SPD1148 Technical Reference Manual

Introduction

This reference manual targets application developers.

It provides complete information on how to use the SPD1148 microcontroller memories and peripherals. For device mechanical and electrical characteristics please refer to the SPD1148 datasheet document.

Contents

List of Figures

List of Tables

Revision history

Revision	Date	Author	Status	Changes
1	2019-10-14	-	Outdated	1. Initial release.
2	2019-12-16	-	Outdated	1. Update Section 26.3. 2. Update Table 24-47 and Table 24-48.
3	2020-02-24	-	Outdated	1. Add Table 1-2. 2. Update Figure 24-1, modify the name of some signals. 3. Update Table 24-15, modify bit-field description. 4. Update Table 24-46, modify bit-field description.
4	2020-06-13	-	Outdated	1. Update Figure 1-1. 2. Update Section 1.4. 3. Update Figure 10-1. 4. Update Figure 10-2. 5. Update Figure 10-14. 6. Update Figure 10-16. 7. Update Table 14-8. 8. Update Table 14-10. 9. Update Table 14-12. 10. Add Figure 25-1. 11. Update Figure 26-1. 12. Update Section 26.2. 13. Add Figure 26-3. 14. Update Section 26.3.
5	2021-05-29	-	Outdated	1. Update Table 10-89 2. Update Table 10-162, Table 10-164 and Table 10-166 3. Update Figure 12-2. 4. Modify content in Section 12.7. 5. Modify content in Section 12.10.2. 6. Update Section 17.3 for UART features. 7. Add Table 20-2. 8. Add Figure 2-2 9. Update Section3.8.11, modified maximum frequency of SIO. 10. Add Section 3.8.1. 11. Update Section 3.8.2 and Section 3.8.4. 12. Update Section 3.9. 13. Update Table 4-1. 14. Update Table 10-11, modify description of GLDPRD bitfield. 15. Update Figure 10-11. 16. Add Table 10-5. 17. Update Figure 10-17 and Figure 10-19.

Revision	Date	Author	Status	Changes
				18. Update Table 10-19. 19. Update Section 19.4.3, add Note for SSP RXONLY mode. 20. Add Section 25.2.5. 21. Update Section 25.4 22. Add Section 26.2.5. 23. Update Section 26.2.2.
6	2021-11-15	-	Outdated	1. Update Table 12-33 and Table 12-153. 2. Update Figure 3-4 and Figure 3-5. 3. Update Section 3.6. 4. Update Table 3-10 and Table 3-11. 5. Update Table 10-7 , modify the reset value of TZSEL. 6. Update Table 15-1. 7. Update Section 17.5.2.2, add Note for receive interrupt. 8. Update Table 17-27 and Table 17-28, modify the UARTFOR.BYTECNT bit-field width. 9. Update Section 24.5. 10. Update Table 24-47 and Table 24-48, add the information of register bit-field PDRVCTL.CPCONEN. 11. Update Table 24-15. 12. Update Section 26.3. 13. Update Table 24-50, modify the description of PDRVOCCTL.DGWIN/ BLANKWIN bit-field. 14. Update Section 26.2.2. 15. Update Figure 4-2. 16. Add Figure 4-5. 17. Update Table 4-3 and Table 4-5.
7	2022-01-15	-	Outdated	1. Update Figure 4-6 and Figure 4-7 2. Update Section 7.2 3. Update Table 8-3 and Table 8-4 4. Update Figure 10-4 5. Update Note of Section 10.6 6. Update Section 13.3. 7. Update Section 16.2. 8. Add Chapter 23. 9. Add Figure 4-1, Figure 4-3 and Figure 4-4. 10. Add Section 13.3. 11. Add Section 14.9 12. Add Section 15.5 and Section 16.4
8	2022-10-13	-	Outdated	1. Update Section 12.11. 2. Update Table 10-13. 3. Update Trip-Zone sub-module logic in Section 10.6. 4. Update Example description in Section 16.2.

Revision	Date	Author	Status	Changes
				5. Update Notes of Table 12-9.
9	2023-08-28	-	Outdated	1. Update Section 17.5.3, added instructions for configuring automatic baud rate.
C/0	2024-05-06	Jiali Zhou	Released	1. Update Figure 10-14. 2. Updated the description of UART compatibility. 3. Modify document style.
C/1	2024-12-16	Jiali Zhou	Released	1. Update the description of VBAT voltage sampling in Section 26.4. 2. Update the description of PWM TZ module. 3. Modify the description of CPPROTEN in Table 24-48. 4. Modify value in Table 19-15. 5. Modify Figure 16-2. 6. Modify Figure 14-3, Figure 14-4. 7. Modify Figure 10-2, Figure 10-3. 8. Modify Section 17.4.2.4 and UARTIIR register. 9. Add CHPENGR register. 10. Update Figure 12-1. 11. Update Figure 4-1. 12. Add note for Section 20.5.1. 13. Modify PWMCLK to ADCCLK in COMP.

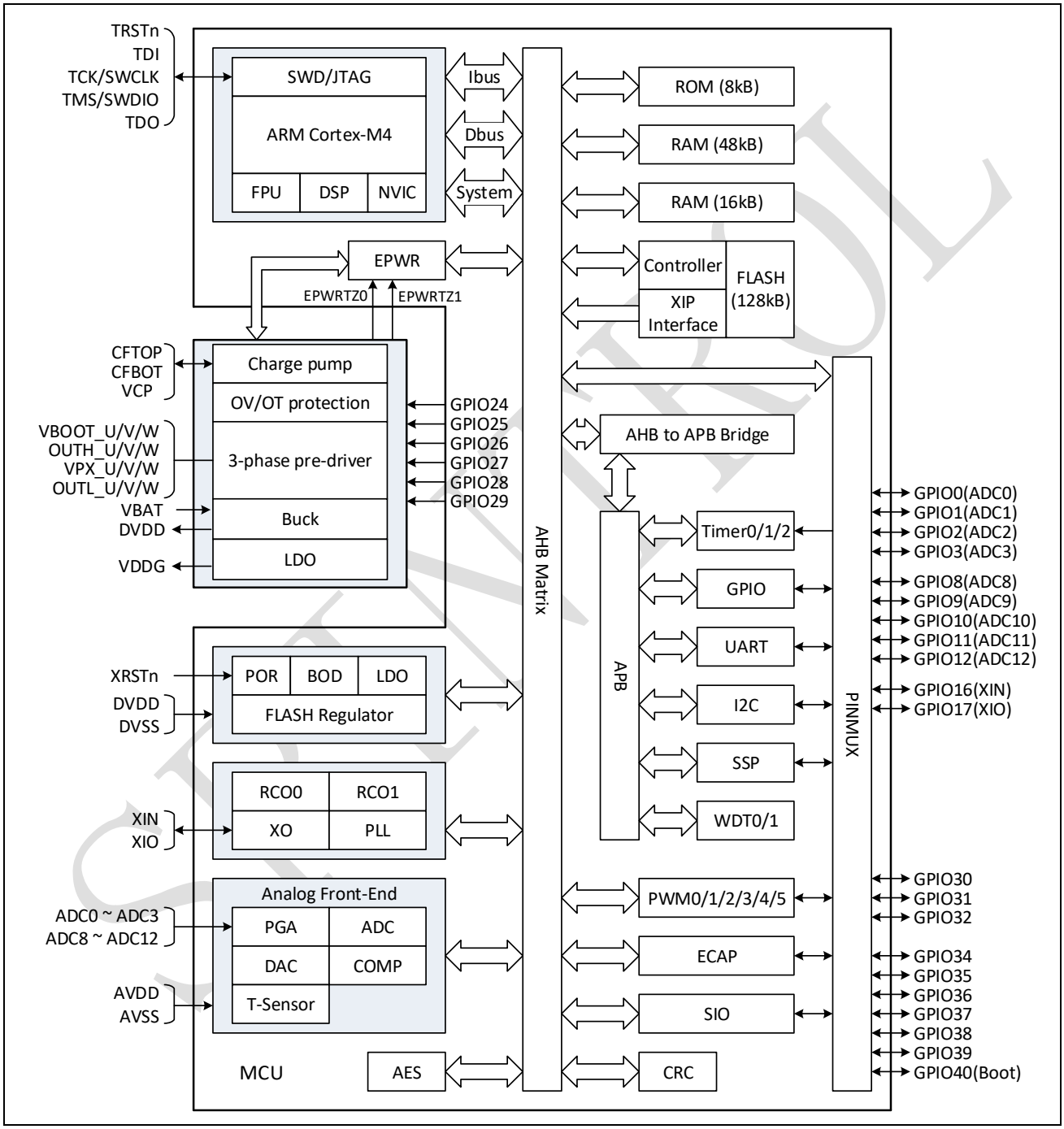
Terms or abbreviations

Terms or abbreviations	Description
MCU	Microcontroller Unit
SWD	Serial Wire Debug
AHB	Advanced High Performance Bus
XIP	Execution In Place
PLL	Phase Locked Loop
BOD	Brownout Detector
PFD	Phase Frequency Detector
NVIC	Nested Vectored Interrupt Controller
UART	Universal Asynchronous Receiver-Transmitter
ADC	Analog-to-Digital Converter
DAC	Digital-to-Analog Converter
PGA	Programmable-Gain Amplifier
CRC	Cyclic Redundancy Check
AES	Advanced Encryption Standard

1 Memory and bus architecture

1.1 System architecture

Figure 1-1: Block diagram of SPD1148

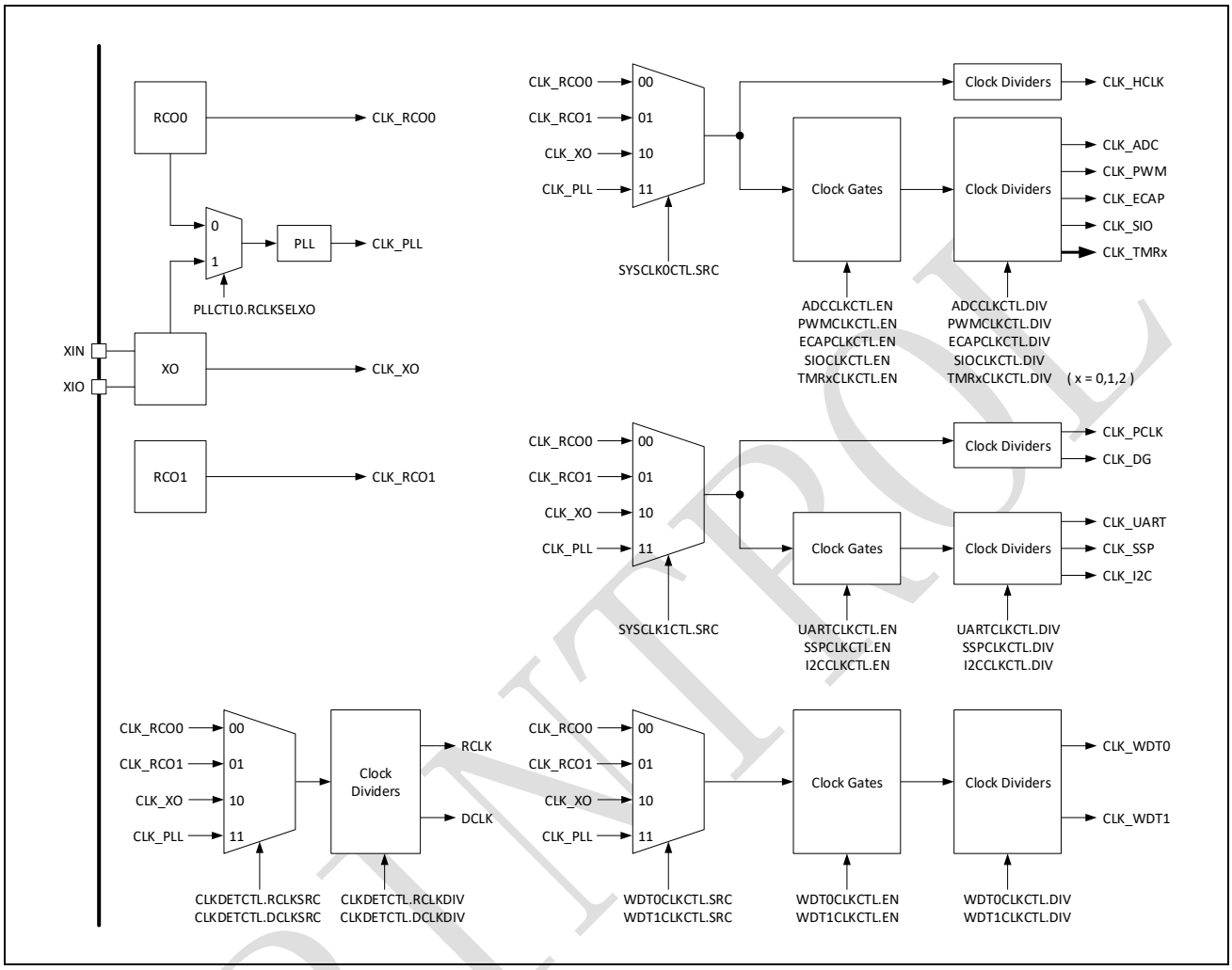


In SPD1148, the main system consists of:

- Three masters:
  - Cortex-M4 I-bus, D-bus and system bus
- 15 slaves:
  - ROM
  - Two SRAMs
  - Flash controller and XIP interface
  - Power and Clock registers
  - AHB peripherals: AES, CRC, AFE, PWM, ECAP, WDT0/1 and SIO
  - AHB to APB bridge, which connect all APB peripherals
- These are interconnected using a low latency AHB bus architecture as shown in Figure 1-1. The AHB-to-APB bridge connects all the APB peripherals and provides full synchronous connections between the AHB and the APB bus. The APB bus is limited to 50MHz.

1.2 Clock tree

Figure 1-2: SPD1148 clock tree



As shown in Figure 1-2, there are four root clocks in SPD1148:

- RCO0: Trimmed at 32MHz
- RCO1: Free running ring oscillator with nominal frequency of 2.2MHz
- XO: External crystal with on-chip active resonator (2-pin configuration) or external clock input (1-pin configuration via XIN or XIO)
- PLL: Provide flexible on-chip clock up to 200MHz. The reference clock can be selected as RCO0 or XO clock

The succeeding smooth MUX provides glitch-free switching. Dedicated clocks are generated for each function building block with independent clock gate and divider.

1.3 Memory map

In SPD1148 device, ROM, SRAMs, registers and I/O ports are organized within the same linear 4GB address space. The bytes are coded in memory in Little Endian format.

All the memory areas that are not allocated to on-chip memories and peripherals are considered “Not Used”. Access to these address results in bus fault. The memory map of SPD1148 is shown in Figure 1-3.

Figure 1-3: SPD1148 memory map

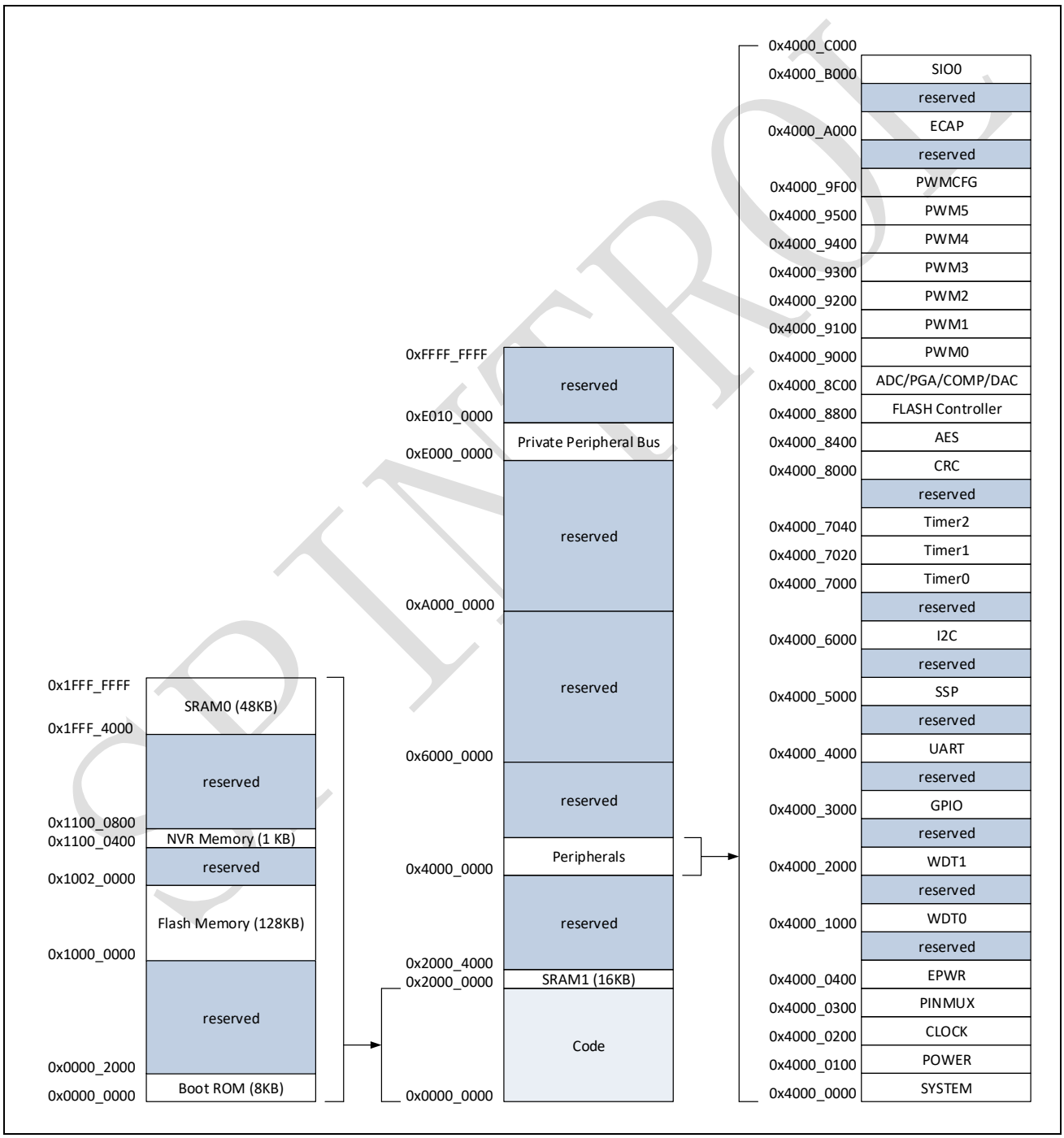


Table 1-1 gives the base address of the peripherals available in SPD1148 device.

Table 1-1: Peripheral Module Base Address

Peripheral Module	Base Address	Register Map
SYSTEM	0x4000 0000	SYSTEM Register Map
POWER	0x4000 0100	POWER Register Map
CLOCK	0x4000 0200	CLOCK Register Map
GPIO	0x4000 3000	GPIO Register Map
PINMUX	0x4100 0300	PINMUX Register Map
AES	0x4000 8400	AES Register Map
CRC	0x4000 8000	CRC Register Map
TIMER0	0x4000 7000	TIMERx Register Map
TIMER1	0x4000 7020	
TIMER2	0x4000 7040	
WDT0	0x4000 1000	WDTx Register Map
WDT1	0x4000 2000	
PWM0	0x4000 9000	PWMx Register Map
PWM1	0x4000 9100	
PWM2	0x4000 9200	
PWM3	0x4000 9300	
PWM4	0x4000 9400	
PWM5	0x4000 9500	
PWMCFG	0x4000 9600	PWMCFG Register Map
ECAP	0x4000 A000	ECAP Register Map
ADC	0x4000 8C00	ADC Register Map
PGA	0x4000 8C00	PGA Register Map
COMP	0x4000 8C00	COMP Register Map
UART	0x4000 4000	UART Register Map
I2C	0x4000 6000	I2C Register Map
SSP	0x4000 5000	SSP Register Map
FLASH	0x4000 8800	FLASH Register Map

1.3.1 Embedded Flash Memory

The SPD1148 is integrated with an embedded flash memory up to 128kB for storing program code and data. An XIP (Execution-in-Place) controller is dedicated to reading of flash memory as shown in Figure 1-1, while another AHB controller interface is used to erase and program the Flash memory. When the SPD1148 starts up, any code in the Flash memory can be directly executed, or loaded into SRAM memory before running from the SRAM.

For the data safety concern, optional ECC feature is designed for both the embedded Flash and the ROM, which is capable to correct 1-bit error and detect 2-bit errors. The error bit correction feature can be enabled or disabled via MEMECCEN register.

The Flash memory is organized into 256 programmable sectors of 128 words each. The Flash memory can be programmed 32 bits (word) at a time. Data can be erased via 512-byte sector erase, 4kB block erase or entire chip erase.

1.3.2 Embedded SRAM

The SPD1148 has implemented two SRAM memories for running code and data. The one SRAM is up to 48KB and the other is 16KB. The SRAMs can be accessed as bytes, half-words (16 bits) or full words (32 bits) with 0 wait states. The start address is 0x1FFF4000 for the 48 KB SRAM, and 0x20000000 for the 16 KB SRAM.

Additionally, the SPD1148 also implements a memory parity check feature for SRAMs to increase the data safety. When enabled, an interrupt would be generated upon the parity error to notify the Cortex-M4 processor. The feature is controlled by MEMIF, MEMIC and MEMIE register in SYSTEM module. Please see MEMIF to MEMIE for register details.

1.4 Boot configuration

In SPD1148, two different boot modes can be selected through BOOT pin and TRSTn pin:

- Boot from the Flash memory (BOOT pin = High, TRSTn = don’t care): after a reset, the boot loader jumps to the Flash memory and runs from the address 0x1000 0000
- ISP mode (BOOT pin = Low, TRSTn = Low): the boot loader reprograms the Flash memory by using UART. During the process, the GPIO34 is configured as UART_TXD and the GPIO35 is configured as UART_RXD.

When BOOT pin is Low, the TRSTn pin must keep Low, or the chip will enter engineering TEST mode.

Table 1-2: Boot modes

Boot mode select pins		Boot Mode
BOOT	TRSTn	
0	0	ISP mode (GPIO34 configured as UART_TXD; GPIO35 configured as UART_RXD)
0	1	Engineering TEST mode. In this mode, the chip would not work normal, so **do not select this mode whenever chip was powered**.
1	x	Normal start mode, boot from Flash memory Note that when TRSTn is high，the debug interface is in active.

2 Power

2.1 Single power domain

The SPD1148 implementation requires a single VBAT power supply. The High Voltage Module produces DVDD/AVDD (3.3V) via Buck DC-DC from VBAT, and pre-driver voltage VDDG (typically 10V) via LDO, also from VBAT. The following content will focus on details of low-voltage MCU operation and registers. Details of High Voltage Module operation and registers are listed in Chapters 24, 25, 26.

The nominal 1.2V digital power is obtained the MCU on-chip regulator fed by DVDD and is output to the pin VCAP12.

During power-up transient, the on-chip regulator guarantees that the current to VCAP12 is below about 30mA. If VCAP12 current exceeds this number during power-up (due to on-board parasitic short, for example), VCAP12 will not come out and the chip will not power up. This safety feature protects the chip during power-up in case of power pin connection problems.

After VCAP12 is charged up, the chip exits reset condition, and the on-chip regulator limits the maximum current delivered to VCAP12 between 500mA and 1A. If the current from VCAP12 exceeds this limit value due to some catastrophic event, the current limit will reset back to about 30mA level and VCAP12 voltage will collapse. This safety feature prevents burn-out of the chip.

Figure 2-1: Power supply overview

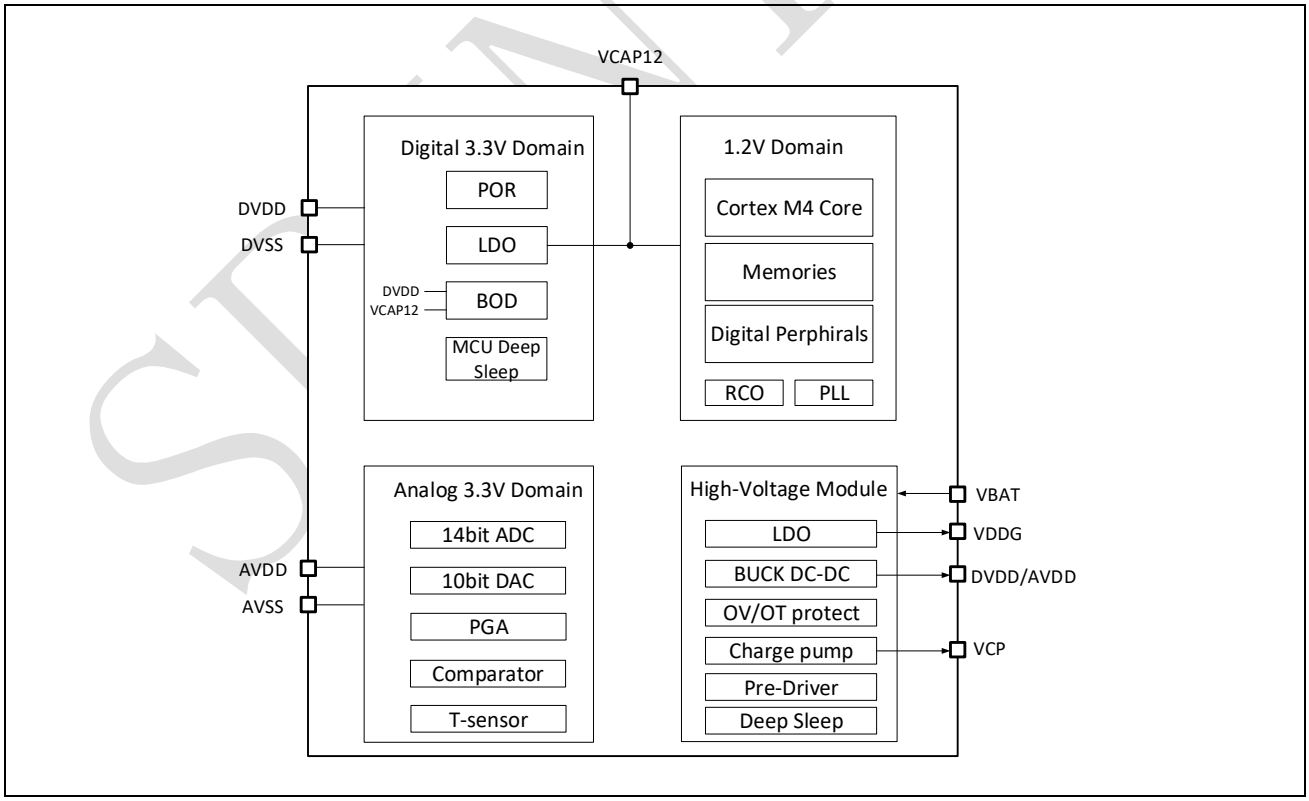
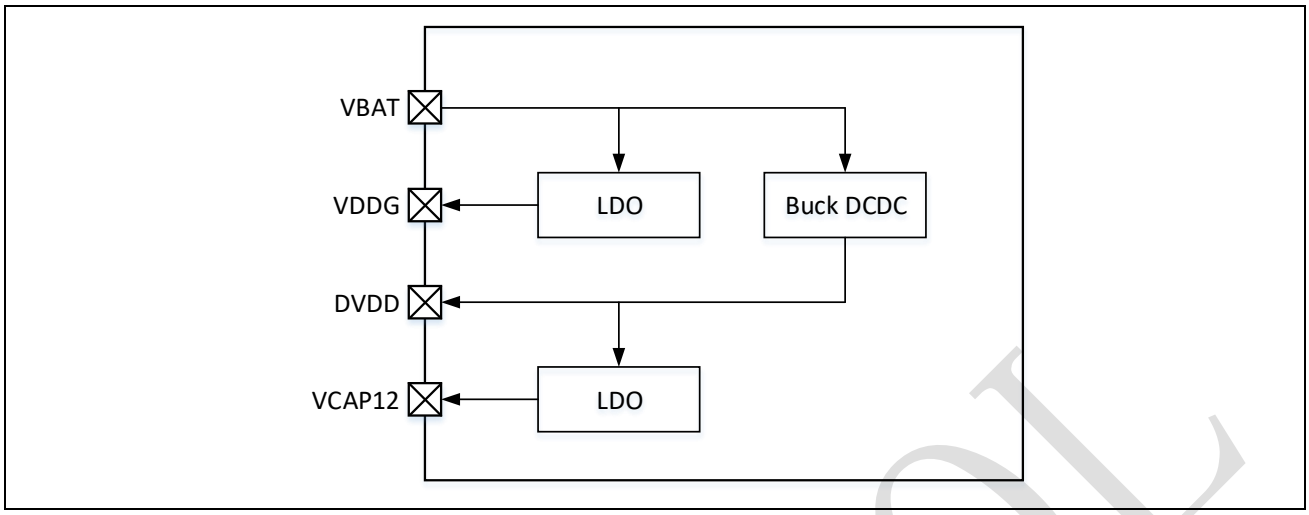


Figure 2-2: Power generation unit



2.2 Analog power

The SPD1148 analog power AVDD is at the same level as the main power DVDD described in Section 2.1. Local bypass capacitors between analog power pin AVDD and analog ground pin AVSS should be placed on board as close to those pins as possible.

2.3 Over-Voltage/Under-Voltage detectors

The Over-Voltage/Under-Voltage detectors are implemented via Brown-out detectors (BOD). The BOD can be used to monitor the various power levels by comparing them to the pre-set thresholds. The BOD for high-voltage powers VBAT and VDDG are described in Section 25.4. This section describes low-voltage BOD for DVDD and VCAP12.

The VDD33H/VDD12H and VDD33L/VDD12L0/VDD12L1 flags are available in the BODIF register, to indicate if the 3.3V/1.2V domain is higher or lower than the pre-set thresholds. The VDD33H/VDD12H and VDD33L/VDD12L0/VDD12L1 events can generate an interrupt if enabled.

The BOD33CTL and BOD12CTL registers determine how to enable 3.3V/1.2V domain brown-out detector and set the too-high and too-low thresholds. The BODIE register enables the 3.3V/1.2V domain too-high and too-low interrupts.

There are two 1.2V domain monitors. This allows additional level of flexibility. For example, the higher-set level can be set to provide a system alert while a lower-set level can be set to initiate a backup subroutine or reset.

Figure 2 3 and Figure 2 4 shows the 3.3V and 1.2V domain brown-out detector structure.

Figure 2-3: 3.3V domain brown-out detector diagram

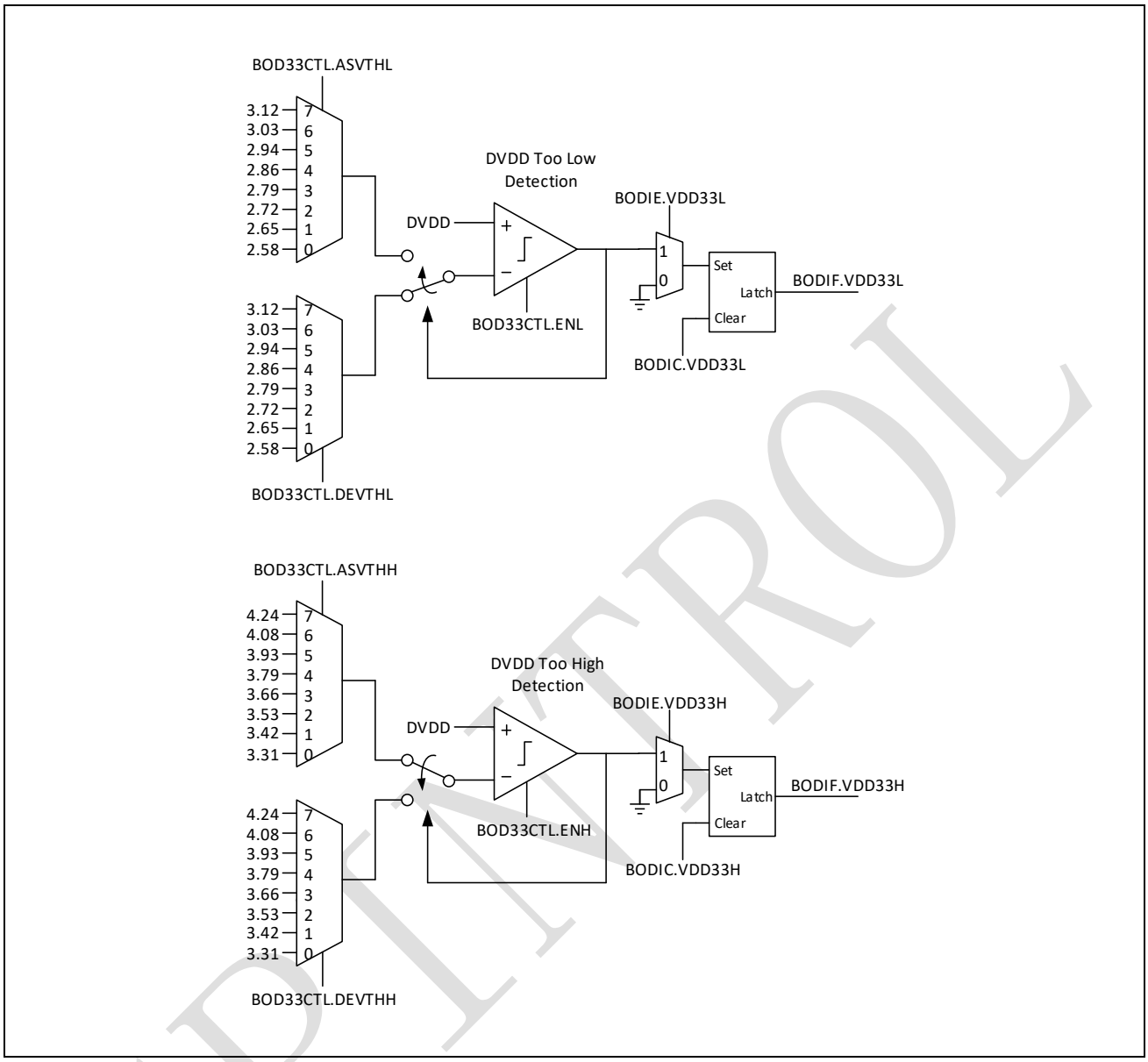
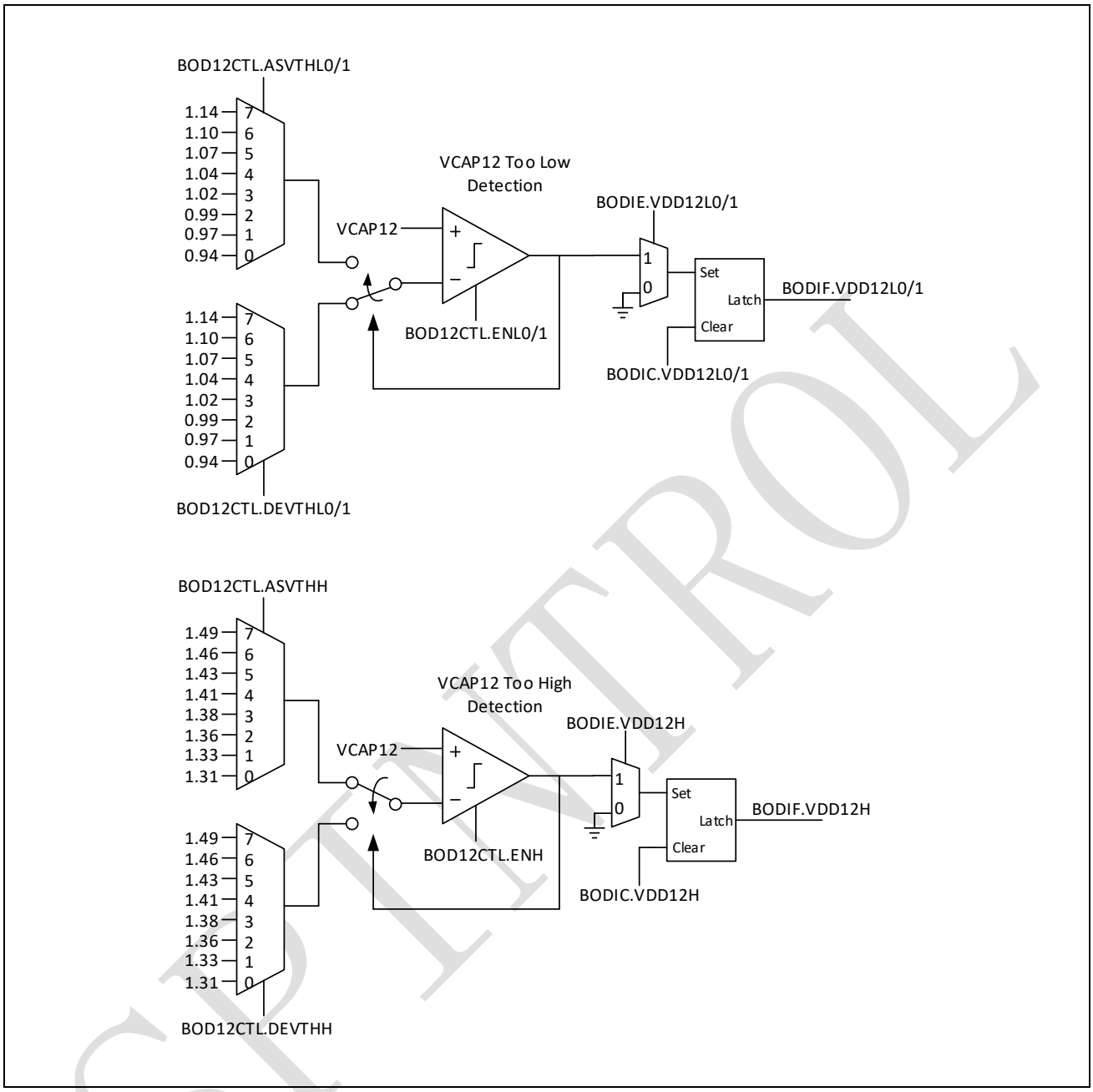


Figure 2-4: 1.2V domain brown-out detector diagram

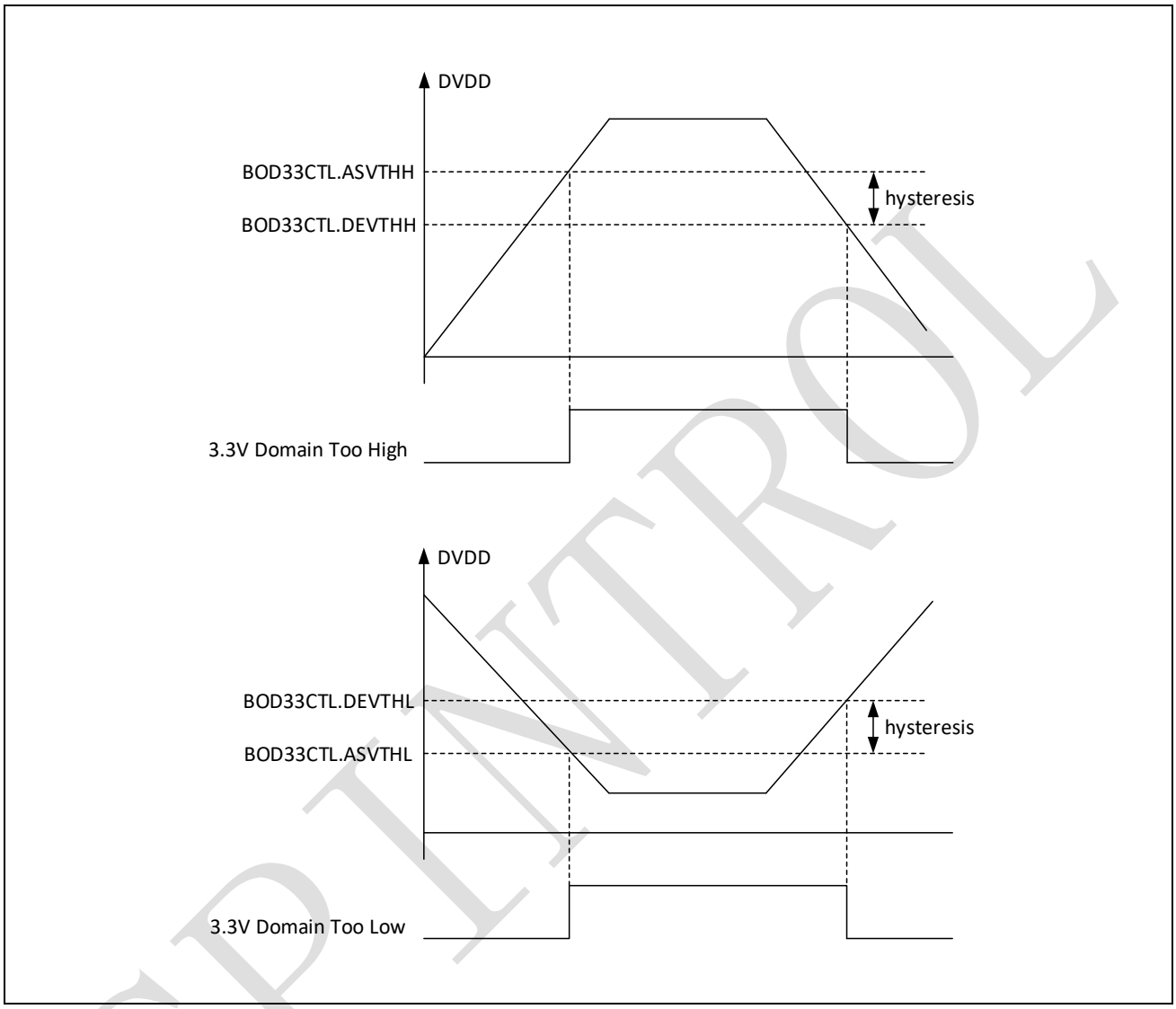


Note: For proper operation of each BOD:

14. In Too High Detection, the 0→1 transition code BOD33CTL.ASVTHH/ BOD12CTL.ASVTHH must be larger or equal to 1→0 transition code BOD33CTL.DEVTHH/ BOD12CTL.DEVTHH.
15. In Too Low Detection, the 0→1 transition code BOD33CTL.ASVTHL/BOD12CTL.ASVTHL0/ BOD12CTL.ASVTHL1 must be smaller or equal to 1→0 transition code BOD33CTL.DEVTHL/BOD12CTL.DEVTHL0/BOD12CTL.DEVTHL1.

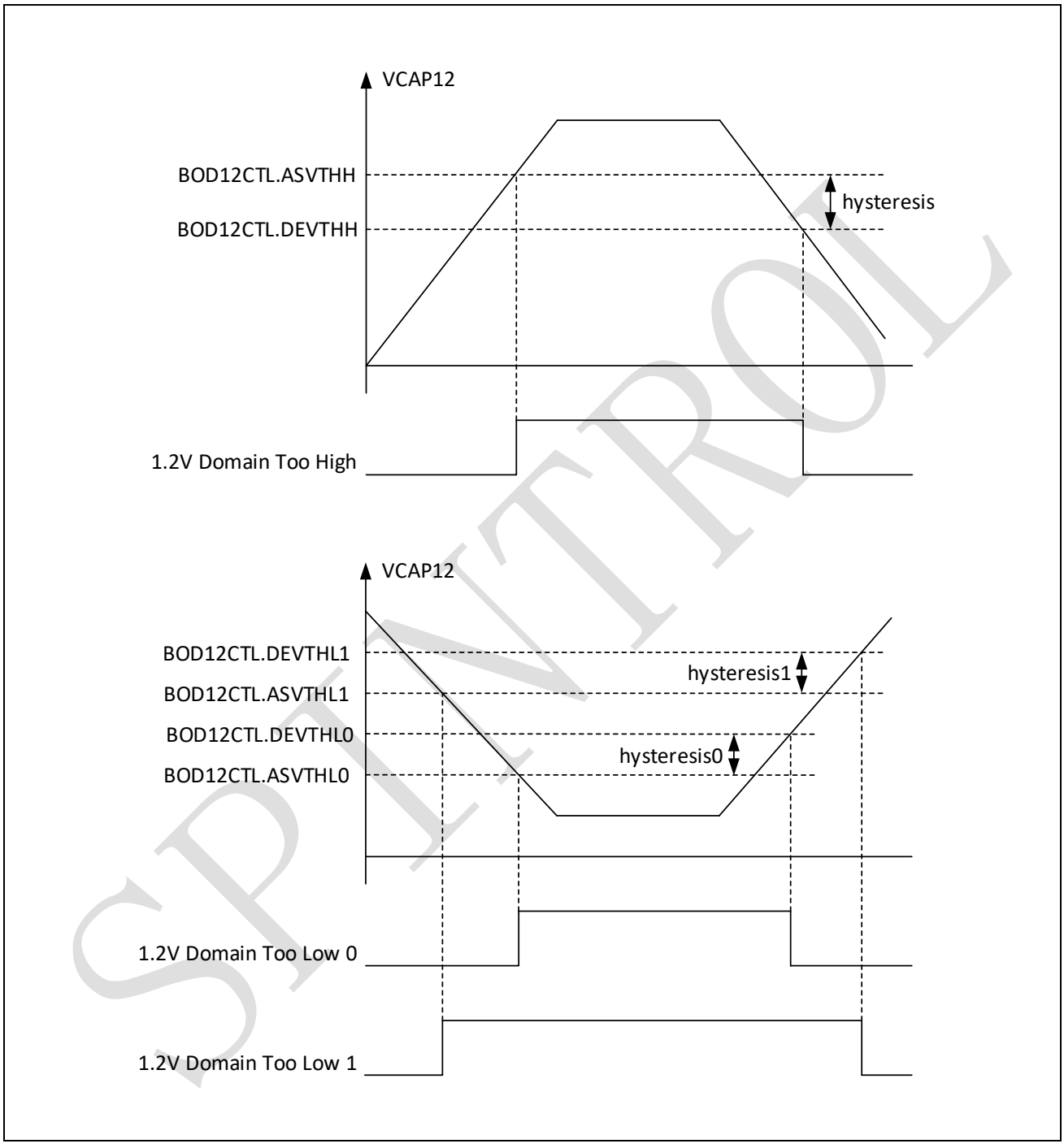
The 3.3V domain too-high or too-low interrupt can be generated when DVDD rises above the TOO_HIGH threshold or when DVDD drops below the TOO_LOW threshold.

Figure 2-5: DVDD thresholds



The 1.2V too-high or too-low interrupt can be generated when VCAP12 rises above the TOO_HIGH threshold or when VCAP12 drops below the TOO_LOW threshold.

Figure 2-6: VCAP12 thresholds



2.4 MCU Deep-sleep mode

MCU deep-sleep mode is implemented for the internal 1.2V LDO in SPD1148. By writing 0x51ee9 to DPSLPKEY register, the 1.2V LDO can be disabled and the system enters into deep-sleep mode for power saving. The whole system can be woken up by pulling down XRSTn pin. Note that a more powerful chip deep sleep mode in VBAT domain shuts down DVDD domain. It is related to high-voltage operation and it is described in Section 25.3.

2.5 Registers

2.5.1 Power register map

Table 2-1: POWER Module Base Address

Peripheral Module	Base Address
POWER	0x4000 0100

Table 2-2: POWER Register Map

Register	Offset	Description	Reset Value
PWRSTS	0x0	Power Status Register	0x00000001
LDOCTL*	0x4	LDO Control Register	0x00000007
PORCTL*	0xC	Low Power POR Control Register	0x00000000
BODIF	0x10	BOD Interrupt Flag Register	0x00000000
BODIC	0x14	BOD Interrupt Clear Register	0x00000000
BODIE*	0x18	BOD Interrupt Enable Register	0x00000000
BODCTL*	0x1C	BOD Control Register	0x00000007
BOD33CTL*	0x20	3.3V BOD Control Register	0x00006E10
BOD12CTL*	0x24	1.2V BOD Control Register	0x006E1010
DPSLPKEY*	0x28	Deep Sleep Enable Key Register	0x00000000
PWRREGKEY	0x2C	Power Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the PWRREGKEY=0x1ACCE551.

2.5.2 Power registers

Table 2-3: Power Status Register (PWRSTS) Layout

PWRSTS (Power Status Register) Offset: 0x0 Default: 0x00000001 Access: POWER -> PWRSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		RESERVED_5	RESERVED_4	RESERVED_3	RESERVED_2	RESERVED_1	FLASHREGRDY

Table 2-4: Power Status Register (PWRSTS) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	RESERVED_5	RO	0x0	Reserved.
4	RESERVED_4	RO	0x0	Reserved.
3	RESERVED_3	RO	0x0	Reserved.
2	RESERVED_2	RO	0x0	Reserved.
1	RESERVED_1	RO	0x0	Reserved.
0	FLASHREGRDY	RO	0x1	Flash regulator ready indicator 0: Not ready 1: Ready

Table 2-5: LDO Control Register (LDOCTL) Layout

LDOCTL (LDO Control Register) Offset: 0x4 Default: 0x00000007 Access: POWER -> LDOCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				VREFTRIM			

Table 2-6: LDO Control Register (LDOCTL) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3:0	VREFTRIM	RW	0x7	Reference voltage trim for LDO 0000: 1.06V 0001: 1.08V 0010: 1.10V 0011: 1.12V 0100: 1.14V 0101: 1.16V 0110: 1.18V 0111: 1.20V 1000: 1.22V 1001: 1.24V 1010: 1.26V 1011: 1.28V 1100: 1.30V 1101: 1.32V 1110: 1.34V 1111: 1.36V

Table 2-7: Low Power POR Control Register (PORCTL) Layout

PORCTL (Low Power POR Control Register) Offset: 0xC Default: 0x00000000
Access: POWER -> PORCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					RESERVED_2	XRSTFILT	

Table 2-8: Low Power POR Control Register (PORCTL) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	RESERVED_2	W1C	0x0	Reserved.
1:0	XRSTFILT	RW	0x0	External reset input deglitch filtering window 00: 500us 01: 1ms 10: 2ms 11: 4ms

Table 2-9: BOD Interrupt Flag Register (BODIF) Layout

BODIF (BOD Interrupt Flag Register) Offset: 0x10 Default: 0x00000000 Access: POWER -> BODIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		BODINT	VDD33H	VDD33L	VDD12H	VDD12L1	VDD12L0

Table 2-10: BOD Interrupt Flag Register (BODIF) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	BODINT	RO	0x0	Latched BOD interrupt flag 0: BOD interrupt is deasserted 1: BOD interrupt is asserted
4	VDD33H	RO	0x0	Latched VDD33H assertion 0: VDD33H is never asserted 1: VDD33H has been asserted
3	VDD33L	RO	0x0	Latched VDD33L assertion 0: VDD33L is never asserted 1: VDD33L has been asserted
2	VDD12H	RO	0x0	Latched VDD12H assertion 0: VDD12H is never asserted 1: VDD12H has been asserted
1	VDD12L1	RO	0x0	Latched VDD12L1 assertion 0: VDD12L1 is never asserted 1: VDD12L1 has been asserted
0	VDD12L0	RO	0x0	Latched VDD12L0 assertion 0: VDD12L0 is never asserted 1: VDD12L0 has been asserted

Table 2-11: BOD Interrupt Clear Register (BODIC) Layout

BODIC (BOD Interrupt Clear Register) Offset: 0x14 Default: 0x00000000 Access: POWER -> BODIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		BODINT	VDD33H	VDD33L	VDD12H	VDD12L1	VDD12L0

Table 2-12: BOD Interrupt Clear Register (BODIC) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	BODINT	W1C	0x0	BOD interrupt clear 0: Write a 0 has no effect and always read back 0 1: Write a 1 clear the BODIF.BODINT. This bit is self-cleared to 0.
4	VDD33H	W1C	0x0	VDD33H interrupt flag clear 0: Write a 0 has no effect and always read back 0 1: Write a 1 clear the BODIF.VDD33H. This bit is self-cleared to 0.
3	VDD33L	W1C	0x0	VDD33L interrupt flag clear 0: Write a 0 has no effect and always read back 0 1: Write a 1 clear the BODIF.VDD33L. This bit is self-cleared to 0.
2	VDD12H	W1C	0x0	VDD12H interrupt flag clear 0: Write a 0 has no effect and always read back 0 1: Write a 1 clear the BODIF.VDD12H. This bit is self-cleared to 0.
1	VDD12L1	W1C	0x0	VDD12L1 interrupt flag clear 0: Write a 0 has no effect and always read back 0 1: Write a 1 clear the BODIF.VDD12L1. This bit is self-cleared to 0.
0	VDD12L0	W1C	0x0	VDD12L0 interrupt flag clear 0: Write a 0 has no effect and always read back 0

Bits	Field Name	Type	Reset	Description
				1: Write a 1 clear the BODIF.VDD12L0. This bit is self-cleared to 0.

Table 2-13: BOD Interrupt Enable Register (BODIE) Layout

BODIE (BOD Interrupt Enable Register) Offset: 0x18 Default: 0x00000000 Access: POWER -> BODIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_5							
23	22	21	20	19	18	17	16
RESERVED_31_5							
15	14	13	12	11	10	9	8
RESERVED_31_5							
7	6	5	4	3	2	1	0
RESERVED_31_5			VDD33H	VDD33L	VDD12H	VDD12L1	VDD12L0

Table 2-14: BOD Interrupt Enable Register (BODIE) Description

Bits	Field Name	Type	Reset	Description
31:5	RESERVED_31_5	RO	0x0	Reserved.
4	VDD33H	RW	0x0	VDD33H interrupt enable 0: VDD33H event does not generate interrupt 1: VDD33H event generates interrupt
3	VDD33L	RW	0x0	VDD33L interrupt enable 0: VDD33L event does not generate interrupt 1: VDD33L event generates interrupt
2	VDD12H	RW	0x0	VDD12H interrupt enable 0: VDD12H event does not generate interrupt 1: VDD12H event generates interrupt
1	VDD12L1	RW	0x0	VDD12L1 interrupt enable 0: VDD12L1 event does not generate interrupt 1: VDD12L1 event generates interrupt
0	VDD12L0	RW	0x0	VDD12L0 interrupt enable 0: VDD12L0 event does not generate interrupt 1: VDD12L0 event generates interrupt

Table 2-15: BOD Control Register (BODCTL) Layout

BODCTL (BOD Control Register) Offset: 0x1C Default: 0x00000007 Access: POWER -> BODCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				VREFTRIM			

Table 2-16: BOD Control Register (BODCTL) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3:0	VREFTRIM	RW	0x7	BOD reference voltage trim 0000: 1.062V 0001: 1.082V 0010: 1.102V 0011: 1.122V 0100: 1.143V 0101: 1.163V 0110: 1.183V 0111: 1.203V 1000: 1.223V 1001: 1.243V 1010: 1.263V 1011: 1.283V 1100: 1.303V 1101: 1.323V 1110: 1.343V 1111: 1.363V

Table 2-17: 3.3V BOD Control Register (BOD33CTL) Layout

BOD33CTL (3.3V BOD Control Register) Offset: 0x20 Default: 0x00006E10 Access: POWER -> BOD33CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_15							
23	22	21	20	19	18	17	16
RESERVED_31_15							
15	14	13	12	11	10	9	8
ESERVED_31_1	DEVTHH			ASVTHH			ENH
7	6	5	4	3	2	1	0
RESERVED_7	DEVTHL			ASVTHL			ENL

Table 2-18: 3.3V BOD Control Register (BOD33CTL) Description

Bits	Field Name	Type	Reset	Description
31:15	RESERVED_31_15	RO	0x0	Reserved.
14:12	DEVTHH	RW	0x6	Select the threshold voltage to deassert VDD33H when 3.3V supply is going low 000: 3.31V 001: 3.42V 010: 3.53V 011: 3.66V 100: 3.79V 101: 3.93V 110: 4.08V 111: 4.24V
11:9	ASVTHH	RW	0x7	Select the threshold voltage to assert VDD33H when 3.3V supply is going high 000: 3.31V 001: 3.42V 010: 3.53V 011: 3.66V 100: 3.79V 101: 3.93V 110: 4.08V 111: 4.24V
8	ENH	RW	0x0	VDD33H BOD enable 0: Disable 1: Enable
7	RESERVED_7	RO	0x0	Reserved.
6:4	DEVTHL	RW	0x1	Select the threshold voltage to deassert VDD33L when 3.3V supply is going high 000: 2.58V 001: 2.65V 010: 2.72V 011: 2.79V 100: 2.86V 101: 2.94V 110: 3.03V 111: 3.12V

Bits	Field Name	Type	Reset	Description
3:1	ASVTHL	RW	0x0	Select the threshold voltage to assert VDD33L when 3.3V supply is going low 000: 2.58V 001: 2.65V 010: 2.72V 011: 2.79V 100: 2.86V 101: 2.94V 110: 3.03V 111: 3.12V
0	ENL	RW	0x0	VDD33L BOD enable 0: Disable 1: Enable

Table 2-19: 1.2V BOD Control Register (BOD12CTL) Layout

BOD12CTL (1.2V BOD Control Register) Offset: 0x24 Default: 0x006E1010 Access: POWER -> BOD12CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_23							
23	22	21	20	19	18	17	16
RESERVED_31_23	DEVTHH			ASVTHH			ENH
15	14	13	12	11	10	9	8
RESERVED_15	DEVTHL1			ASVTHL1			ENL1
7	6	5	4	3	2	1	0
RESERVED_7	DEVTHL0			ASVTHL0			ENL0

Table 2-20: 1.2V BOD Control Register (BOD12CTL) Description

Bits	Field Name	Type	Reset	Description
31:23	RESERVED_31_23	RO	0x0	Reserved.
22:20	DEVTHH	RW	0x6	Select the threshold voltage to deassert VDD12H when 1.2V supply is going low 000: 1.31V 001: 1.33V 010: 1.36V 011: 1.38V 100: 1.41V 101: 1.43V 110: 1.46V 111: 1.49V
19:17	ASVTHH	RW	0x7	Select the threshold voltage to assert VDD12H when 1.2V supply is going high 000: 1.31V 001: 1.33V 010: 1.36V 011: 1.38V 100: 1.41V 101: 1.43V 110: 1.46V 111: 1.49V
16	ENH	RW	0x0	VDD12H BOD enable 0: Disable 1: Enable
15	RESERVED_15	RO	0x0	Reserved.
14:12	DEVTHL1	RW	0x1	Select the threshold voltage to deassert VDD12L1 when 1.2V supply is going high 000: 0.94V 001: 0.97V 010: 0.99V 011: 1.02V 100: 1.04V 101: 1.07V 110: 1.10V 111: 1.14V

Bits	Field Name	Type	Reset	Description
11:9	ASVTHL1	RW	0x0	Select the threshold voltage to assert VDD12L1 when 1.2V supply is going low 000: 0.94V 001: 0.97V 010: 0.99V 011: 1.02V 100: 1.04V 101: 1.07V 110: 1.10V 111: 1.14V
8	ENL1	RW	0x0	VDD12L1 BOD enable 0: Disable 1: Enable
7	RESERVED_7	RO	0x0	Reserved.
6:4	DEVTHL0	RW	0x1	Select the threshold voltage to deassert VDD12L0 when 1.2V supply is going high 000: 0.94V 001: 0.97V 010: 0.99V 011: 1.02V 100: 1.04V 101: 1.07V 110: 1.10V 111: 1.14V
3:1	ASVTHL0	RW	0x0	Select the threshold voltage to assert VDD12L0 when 1.2V supply is going low 000: 0.94V 001: 0.97V 010: 0.99V 011: 1.02V 100: 1.04V 101: 1.07V 110: 1.10V 111: 1.14V
0	ENL0	RW	0x0	VDD12L0 BOD enable 0: Disable 1: Enable

Table 2-21: Deep Sleep Enable Key Register (DPSLPKEY) Layout

DPSLPKEY (Deep Sleep Enable Key Register) Offset: 0x28 Default: 0x00000000 Access: POWER -> DPSLPKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 2-22: Deep Sleep Enable Key Register (DPSLPKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x0	Write 0x51ee9 to force the system go into deep sleep

Table 2-23: Power Register Write-Allow Key Register (PWRREGKEY) Layout

PWRREGKEY (Power Register Write-Allow Key Register) Offset: 0x2C Default: 0x1ACCE551 Access: POWER -> PWRREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 2-24: Power Register Write-Allow Key Register (PWRREGKEY) Description

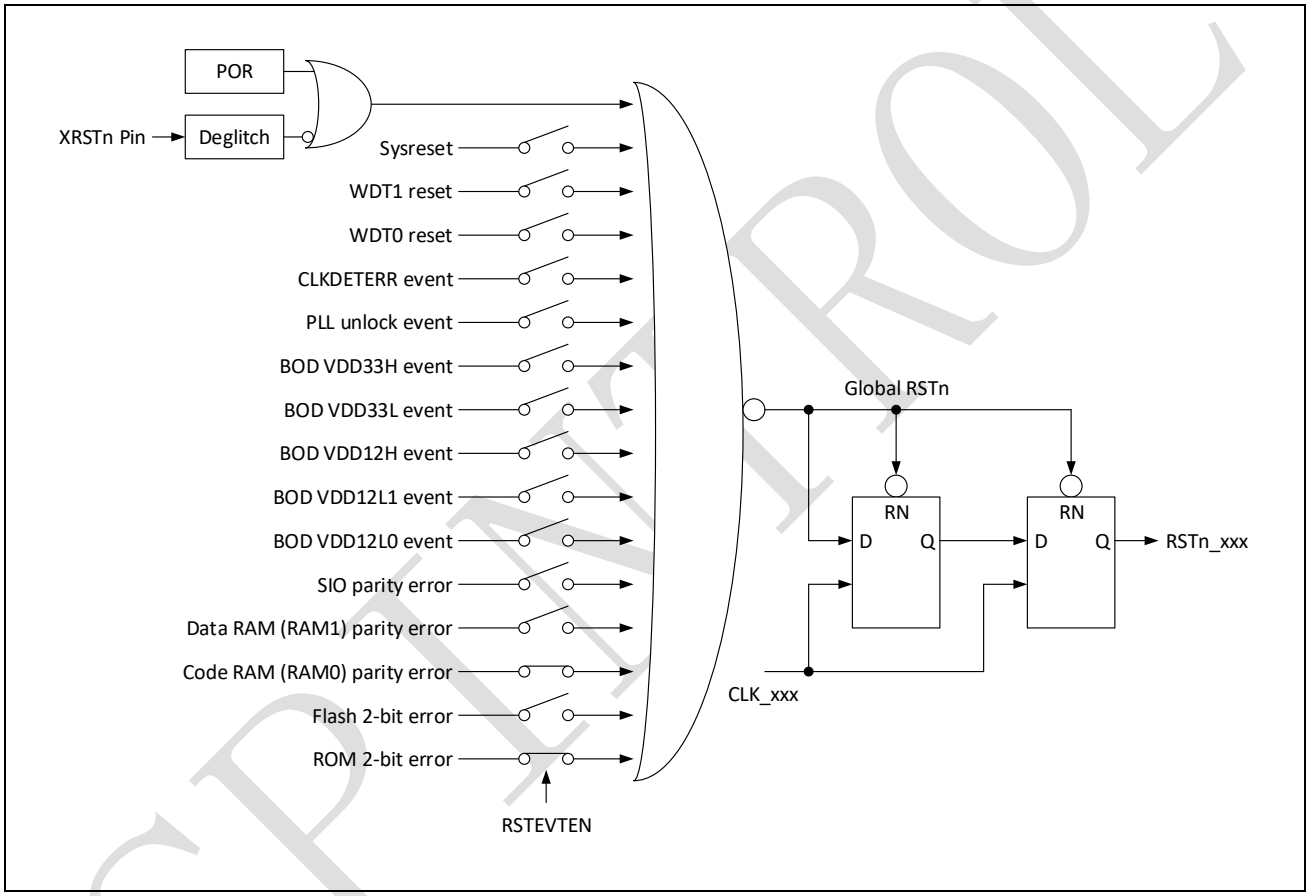
Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected power registers

3 Clocks and reset

3.1 Reset architecture

The global active low reset signal of the SPD1148 is a logic NOR of the reset signals from the power-on-reset (POR) block, the XRSTn pin and the functional logic as shown in Figure 3-1. The functional reset source can be the memory errors, the brown-out detector events, the clock error events, the watchdogs, and the Cortex-M4 SYSRESET request.

Figure 3-1: Active-low POR reset signal diagram



As a diagnosis feature, RSTEVTSTS register records the functional reset events, which can be cleared by corresponding bits in RSTEVTCLR register. Besides, the functional reset sources are programmable via RSTEVTEN register. Details are shown in Table 3-29 to Table 3-34.

3.2 Pin reset

The XRSTn pin of the SPD1148 provides super-set global reset for all function blocks, core and analog blocks. It is pull-up internally to make the chip in normal operation mode. The original XRSTn signal goes through a glitch filter with programmable window size as shown in Table 3-1. Both rising and falling edge glitches are filtered. To assert a valid reset, the XRSTn pin should be kept low for time longer than the window size. By default, the window size is 500us.

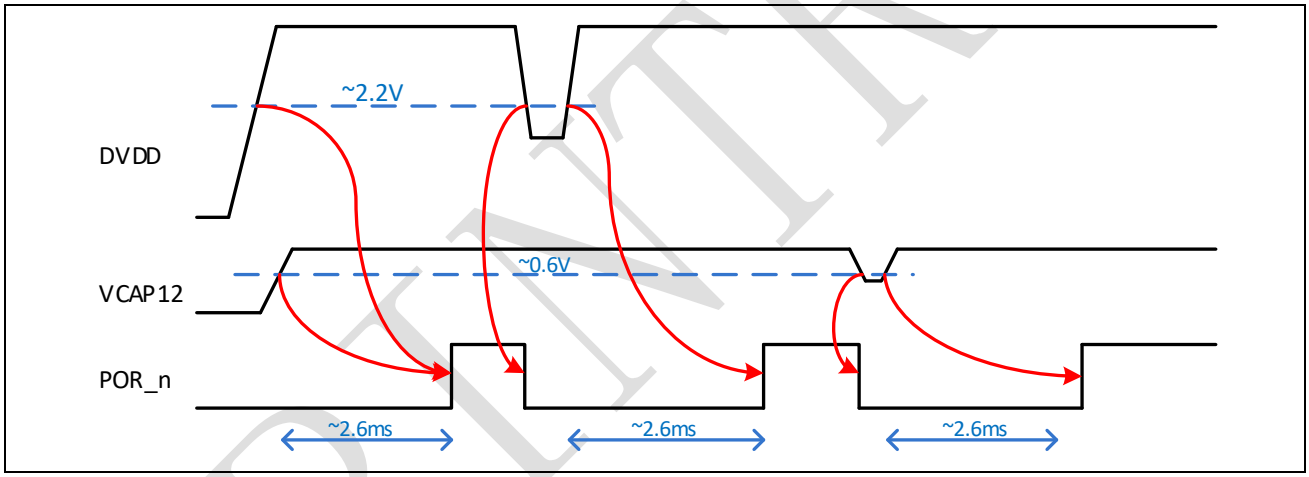
Table 3-1: Pin reset glitch filter window size options

PORCTL.XRSTFILT	Typical Pin Rest Glitch Filter Delay
0	500us
1	1ms
2	2ms
3	4ms

3.3 POR reset

The built-in POR circuits in the SPD1148 guarantee all power up reset sequence requirements and make chip easy to use. It will be asserted for about 2.6ms after power up. The global active-low reset POR_n will go low whenever power levels of DVDD and VCAP12 are below approximately 2.2V and 0.6V respectively as shown in Figure 3-2.

Figure 3-2: Active-low POR reset signal timing diagram

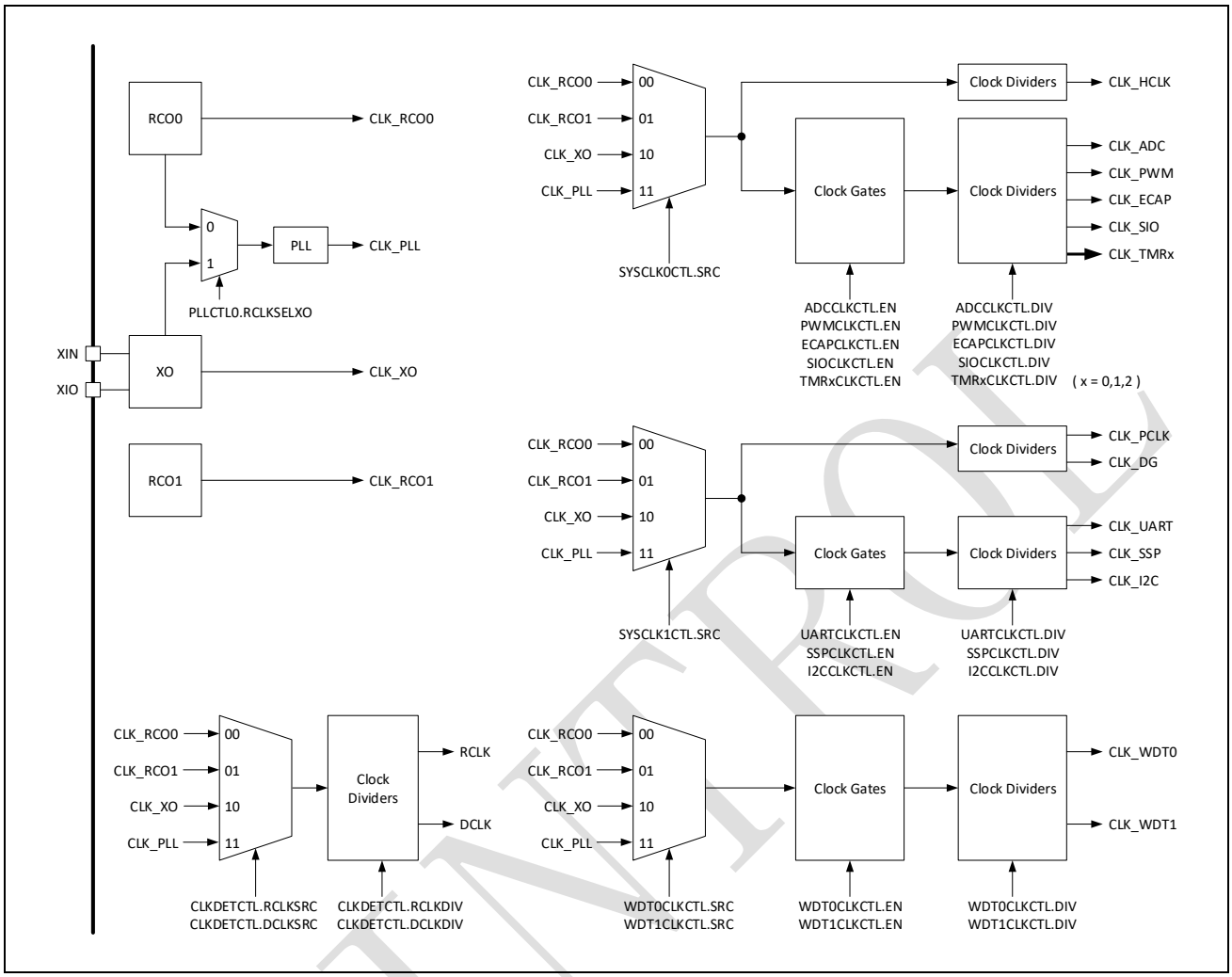


3.4 Clock architecture

The SPD1148 can be clocked by either of the two internal zero-pin RC oscillators, an external oscillator, or by a crystal attached to the on-chip oscillator circuit. A PLL is provided to support high frequency clock from 25MHz to 200MHz while the input frequency for the PLL can be from 4MHz to 56MHz. Combined with programmable dividers and gating blocks, a flexible clock tree is implemented in SPD1148 as shown in Figure 3-3.

Note: The clocks can be brought out to GPIO11/GPIO40 for observation.
Refer to IO_Channel_Defination for details.

Figure 3-3: Clock overview



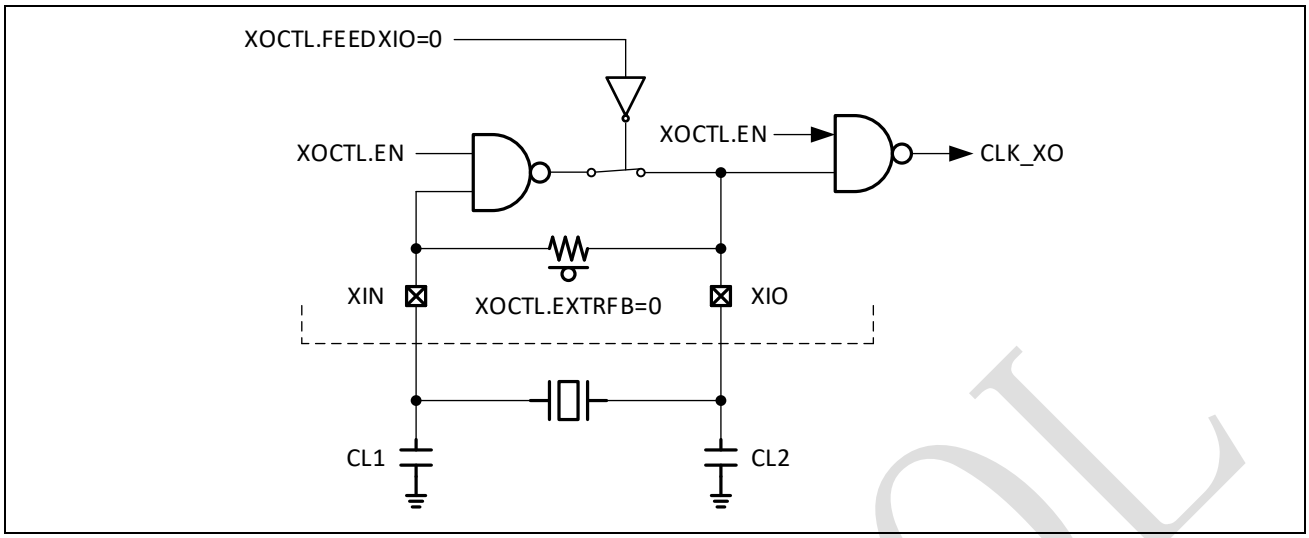
3.5 High frequency RC oscillator

There are two zero-pin RC oscillators implemented on SPD1148. One has been trimmed for minimum frequency variation over temperature and centered at 32MHz. Optional low frequency mode can be enabled to make it oscillates at about 270kHz for power saving. The other is a free-running ring oscillator with a typical frequency of 2.2MHz, which is mainly used for clock safety.

3.6 On-chip crystal oscillator and external clock

Both on-chip and external modes can be used for the crystal oscillator. For the on-chip mode, a crystal is attached as shown in Figure 3-4.

Figure 3-4: On-chip crystal oscillator



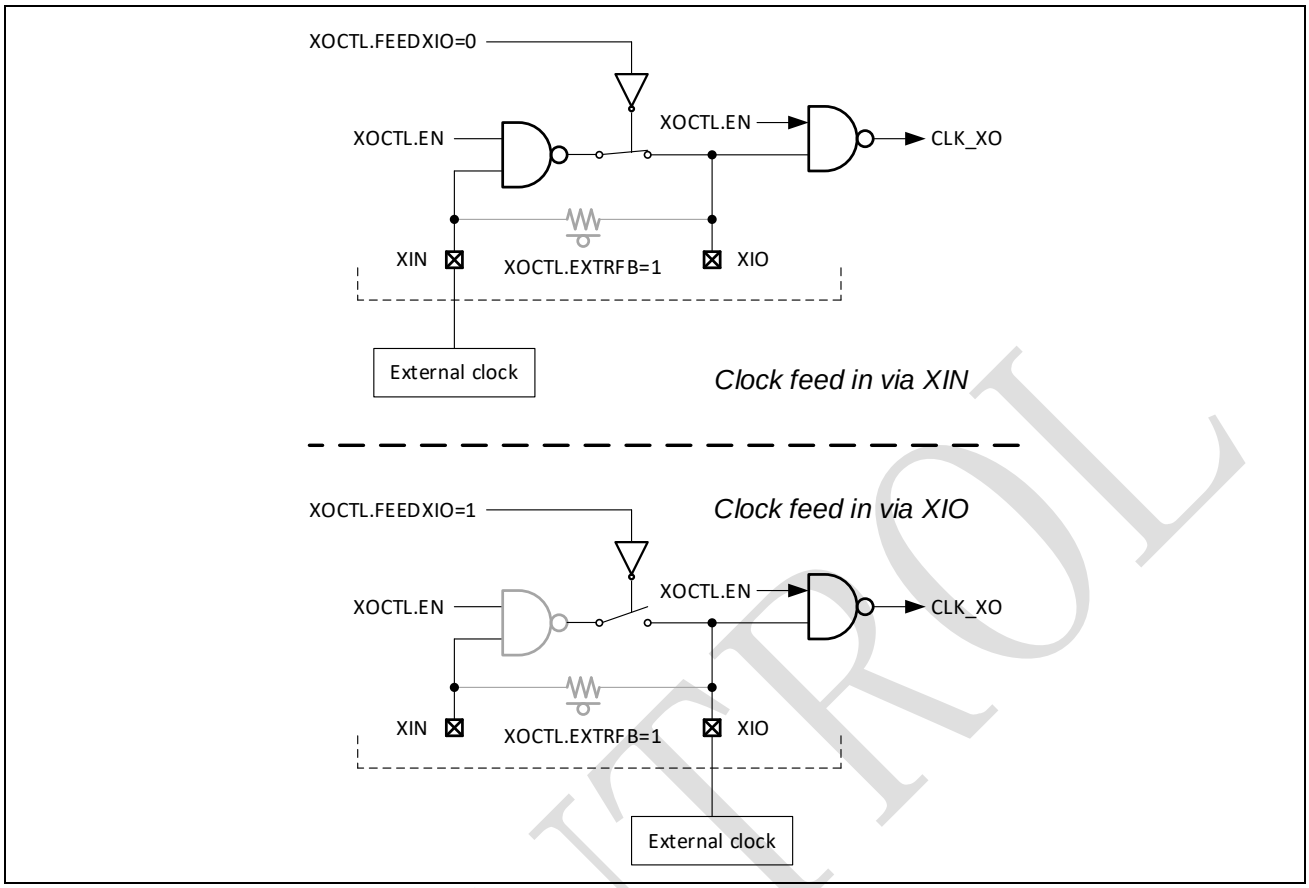
To bias the inverter in the high gain region, a feedback resistor Rf is needed which is normally 1MΩ. SPD1148 provides a built-in feedback resistor by default to save the off-chip components. Please refer to the XO characteristics section of the chip datasheet to select an appropriate crystal and the loading capacitor(CL_eff). The loading capacitor CL_eff is defined as the series combination of CL1 and CL2 as shoiwn in Figure 3-4.

For the external clock mode, no extra software configuration is needed compared with the on-chip mode. Depending on XOCTL.FEEDXIO setup, the clock can be fed in via XIN pin (XIO pin is left open) or XIO pin (XIN pin can be used as normal GPIO) as shown in Figure 3-5.

Since the XIN and XIO are pin shared with GPIO16/GPIO17, the pinmux should be configured correctly before setting XOCTL.EN. For safety concern, XOCTL.EN cannot be written to 1 if GPIO16/GPIO17 pinmux is not a valid configuration for either on-chip or external clock mode. And when XOCTL.EN is 1 so that the XO clock is running, the required GPIO16/GPIO17 pinmux configuration cannot be changed.

Note: If the CLK_XO is used as reference for PLL, the external clock frequency should be within from 4MHz to 56MHz.

Figure 3-5: External clock

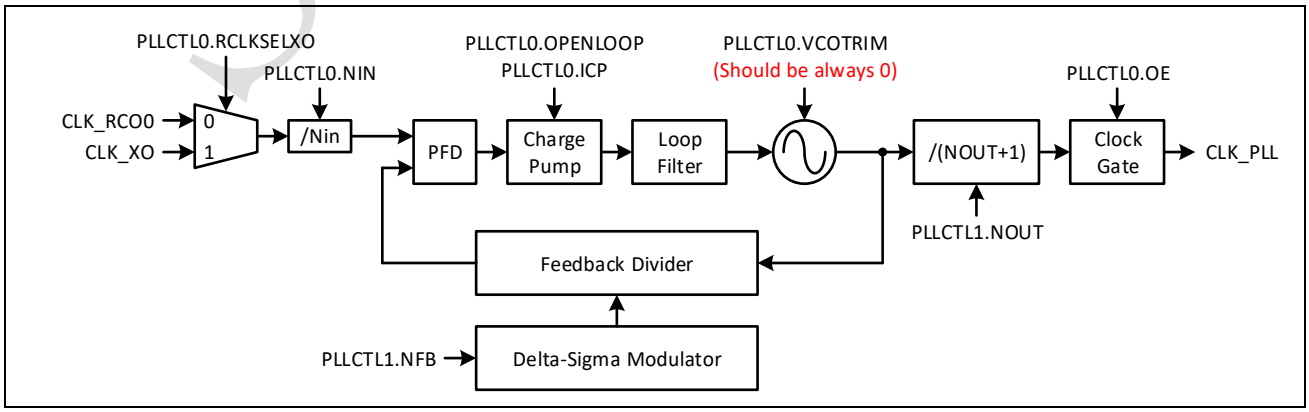


3.7 PLL clock

The architecture of the phase-locked loop in SPD1148 is shown in Figure 3-6. It can take the clock from RCO0 or crystal oscillator as input reference. While the loop is locked, the VCO frequency will be from 400MHz to 600MHz. It is then divided down by programmable divider to provide typically 25MHz to 200MHz clock. The output frequency is given by:

$$f_{PLL} = \frac{f_{REF}}{NIN} \times \frac{NFB}{65536} \times \frac{1}{NOUT + 1}$$

Figure 3-6: PLL clock overview



3.7.1 Reference input and divider

The PLL is designed to take input clock frequency (Fin) from 4MHz to 56MHz, while the reference clock at PFD input (Fpfd) is from 4MHz to 8MHz. Therefore, the input dividing ratio (Fin/Fpfd) should be configured according to Table 3-2.

Table 3-2: Input dividing ratio configuration

Input Frequency (MHz)	Input Dividing Ratio
4~8	1
8~16	2
16~24	3
24~32	4
32~40	5
40~48	6
48~56	7

3.7.2 Output divider

The tuning range of the VCO is designed to be 400MHz to 600MHz. Combined with the output divider, the output clock frequency at CLK_PLL is shown in Table 3-3.

Table 3-3: Output dividing ratio and clock frequency

Desired Output Frequency			Dividing Ratio (NOUT+1)
400MHz	~	600MHz	1
200MHz	~	300MHz	2
150MHz	~	200MHz	3
100MHz	~	150MHz	4
75MHz	~	100MHz	6
50MHz	~	75MHz	8
37.5MHz	~	50MHz	12
25MHz	~	37.5MHz	16

3.7.3 Feedback divider

Based on the criterion on NIN & NOUT derived from Table 3-2 and Table 3-3, the NFB bit in PLLCTL1 register can be calculated as:

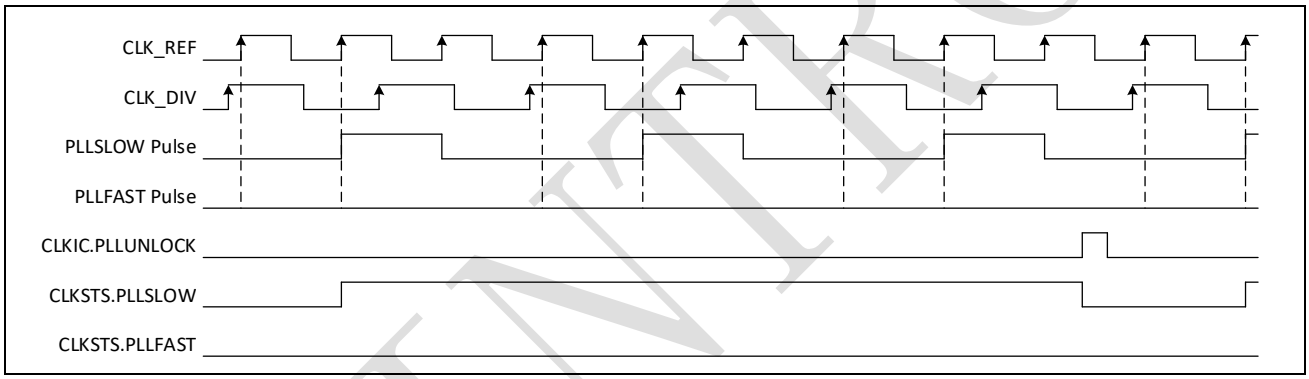
$$NFB = \frac{f_{PLL} \times (NOUT + 1)}{f_{REF} \,/\, NIN} \times 65536$$

3.7.4 PLL unlock detection

A loop unlock detector is included in the phase-frequency detector (PFD) of SPD1148 to provide the diagnosis information for PLL locking status. For the two clocks at the PFD input, if there are two or more rising edges of one clock appears between the two adjacent rising edges of the other clock, internal PLLFAST/PLLSLOW pulse will be generated. These pulses will update the corresponding bit in CLKSTS register. The status bit can be cleared via write a 1 to CLKIC.PLLUNLOCK. Figure 3-7 illustrates a timing diagram for the scenario that reference clock is faster than the divider clock and CLKSTS.PLLSLOW bit is updated.

During the initial locking or a significant change on the feedback dividing ratio of a locked PLL, the frequency of the reference clock and divider clock are different and there will be inevitable PLLFAST or PLLSLOW events latched into CLKSTS register. To avoid false alarm, it should wait some time (about 10us) for the loop to be settled and then assert a CLKIC.PLLUNLOCK to clear the latched status and enable interrupt upon PLL unlock event via CLKIE.PLLUNLOCK if wanted. Neither CLKSTS.PLLFAST nor CLKSTS.PLLSLOW will be set afterwards if the PLL is locked.

Figure 3-7: PLL unlock detection timing diagram example (FREF > FDIV)



3.8 Clock multiplexers and dividers

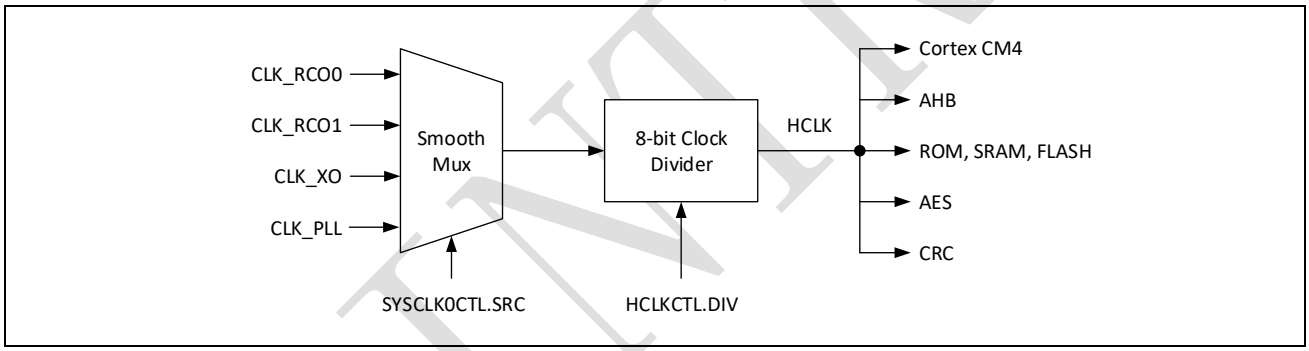
3.8.1 Clock smooth Mux

The smooth clock multiplexers are designed to support glitch-free switch on the fly. To avoid switching to an idle clock and makes the CPU stuck, the multiplexer keeps selecting current clock regardless of the control until the target clock is ready. If the currect selecting clock source is RCO0 and RCO0 has failed, the multiplexer will select RCO1 as current clock source and avoid CPU stucked; Meanwhile, a clock detection error event will be generated if clock detection function enabled. It further triggers a CPU interrupt or a PWM trip-zone event if as enabled by corresponding bit in CLKIE and CLKTZE registers.

3.8.2 HCLK

HCLK is the key clock of the SPD1148 that drives the CPU, AHB, memory, AES and CRC blocks. It is divided down from the always-on clock selected via a smooth multiplexer as shown in Figure 3-8. The maximum allowed frequency of HCLK is 200MHz.

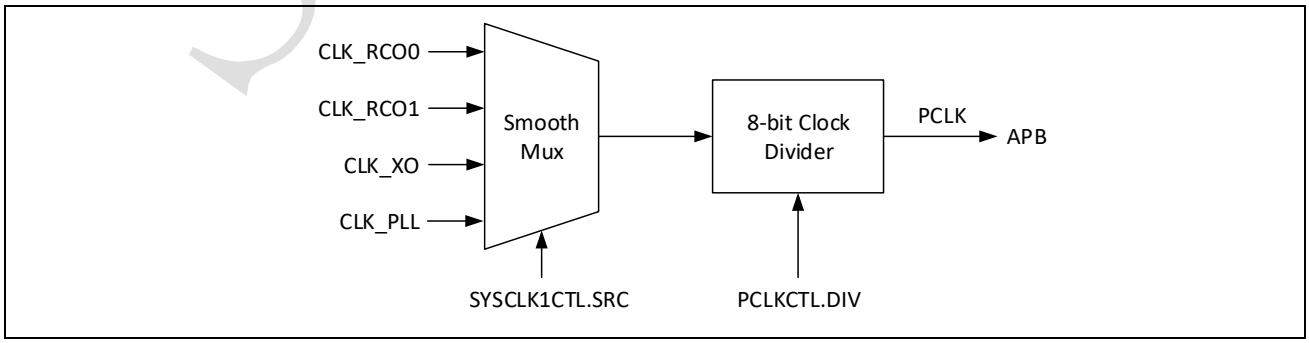
Figure 3-8: HCLK control diagram



3.8.3 APB clock

The APB is driven by PCLK, which is divided down from the always-on clock as shown in Figure 3-9. The maximum allowed frequency of PCLK is 50MHz.

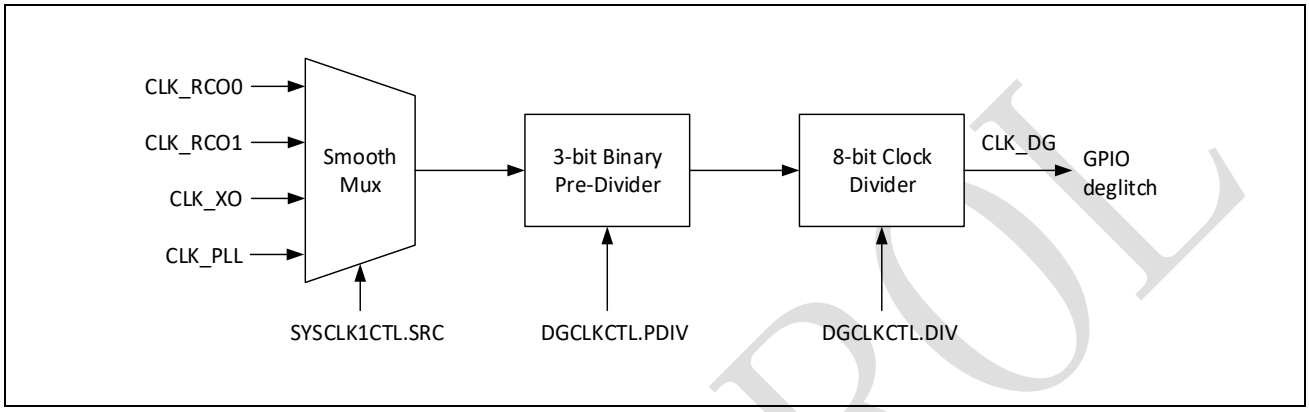
Figure 3-9: PCLK control diagram



3.8.4 GPIO deglitch clock

The GPIO deglitch clock is divided down from the always-on clock by a binary pre-divider and a common 8-bit divider as shown in Figure 3-10. It is used by the GPIOs for input re-timing to filter out the glitches and improve signal integrity. There are 3 clock cycles input delay due to the deglitch filter. The maximum allowed frequency of CLK_DG is 50MHz.

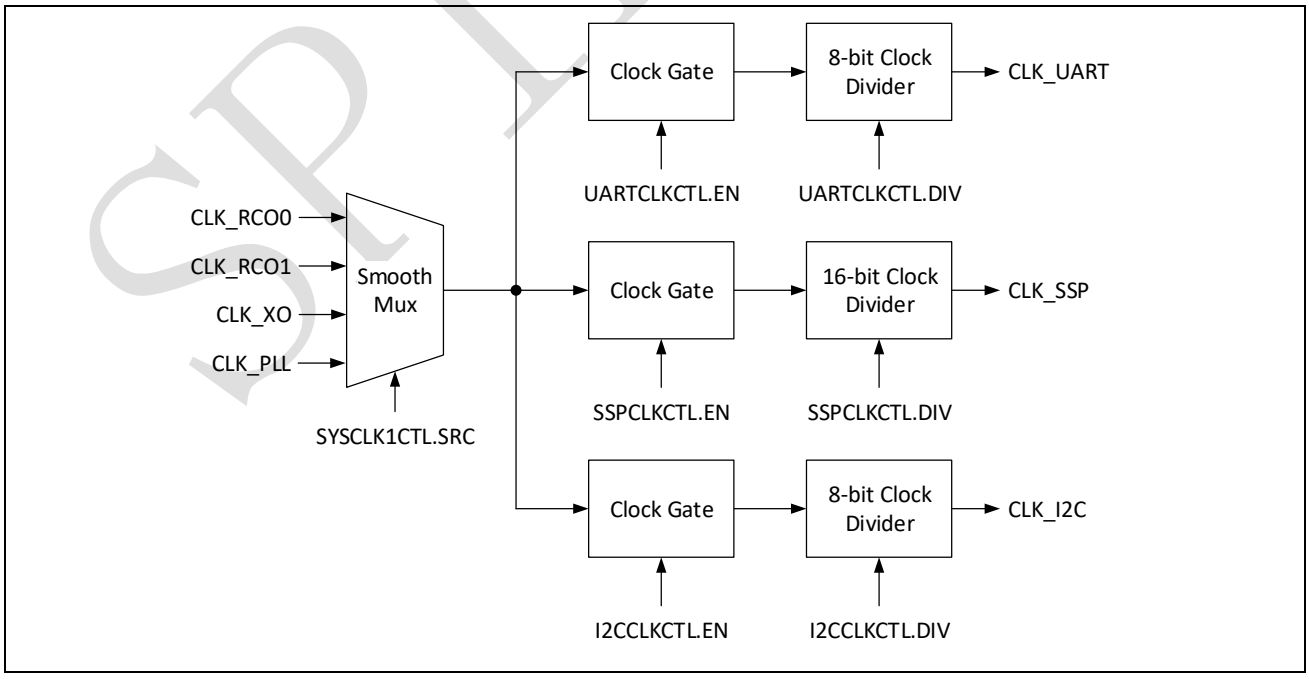
Figure 3-10: GPIO deglitch clock diagram



3.8.5 Communication peripheral clock

The SPD1148 provides SSP, UART and I2C interfaces for serial communication. The corresponding clocks are divided down from the selected system clock and can be gated for power saving as shown in Figure 3-11. The maximum allowed frequency for CLK_UART is 200MHz while the maximum allowed frequency for CLK_SSP and CLK_I2C is 50MHz.

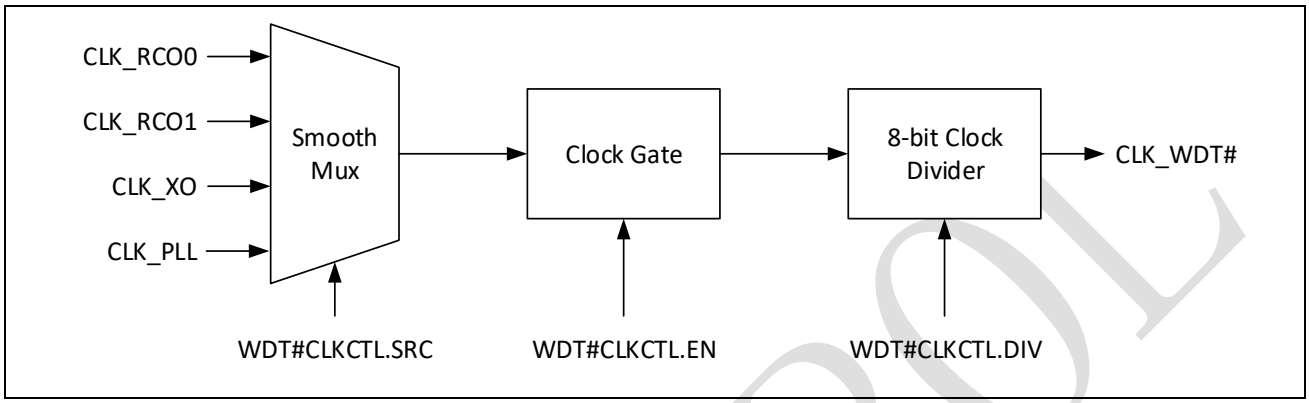
Figure 3-11: Communication peripheral clock diagram



3.8.6 Watchdog timer clock

SPD1148 provides two watchdog timers while each has a dedicated clock that is selected from the four root clocks, gated and divided down as shown in Figure 3-12. Both are enabled by default. The maximum allowed frequency for watchdog timer clock is 200MHz.

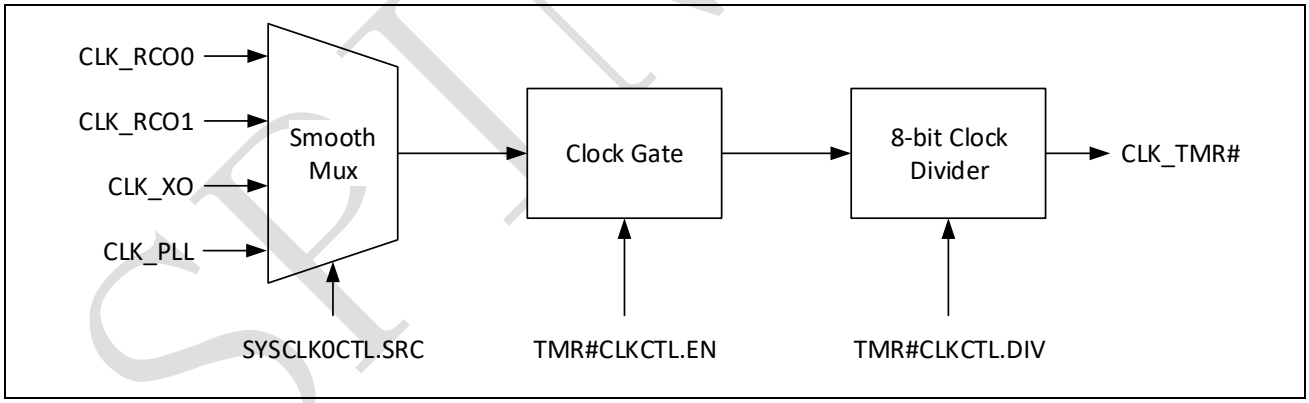
Figure 3-12: Watchdog timer clock diagram



3.8.7 General-purpose timer clock

SPD1148 provides three general timers while each has a dedicated clock that is selected from the four root clocks, gated and divided down as shown in Figure 3-13. The maximum allowed frequency for general timer clock is 200MHz.

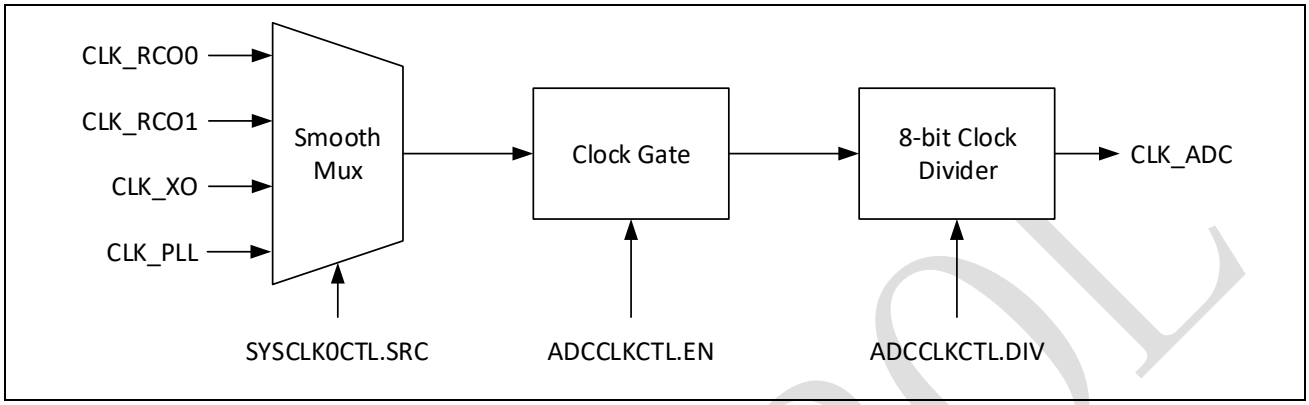
Figure 3-13: General-purpose timer clock diagram



3.8.8 ADC clock

ADC clock is selected from the four root clocks, gated and divided down as shown in Figure 3-14. The maximum allowed frequency is 200MHz.

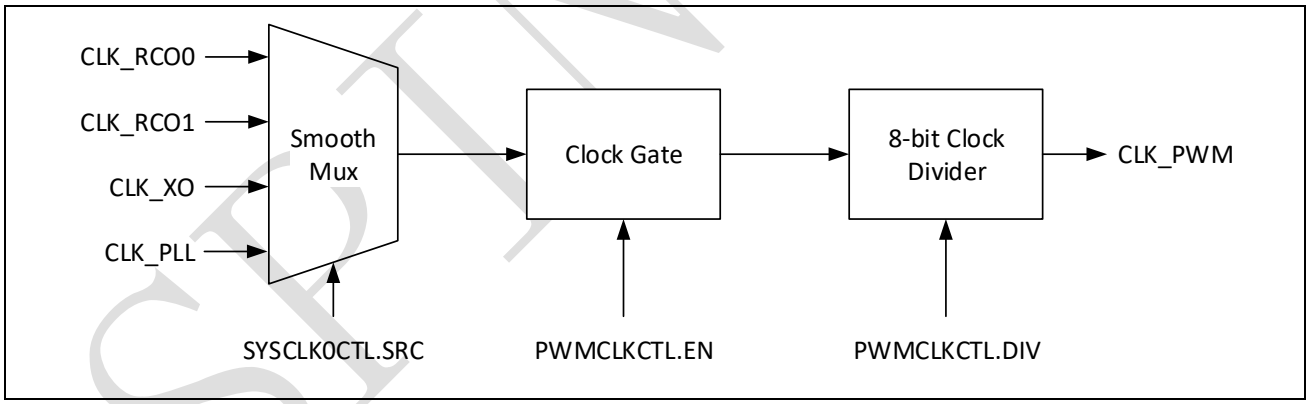
Figure 3-14: ADC clock diagram



3.8.9 PWM clock

The SPD1148 has 6 PWM modules (numbered as from 0 to 5). They share the same clock selected from the four root clocks, gated and divided down as shown in Figure 3-15. The maximum allowed frequency is 200MHz.

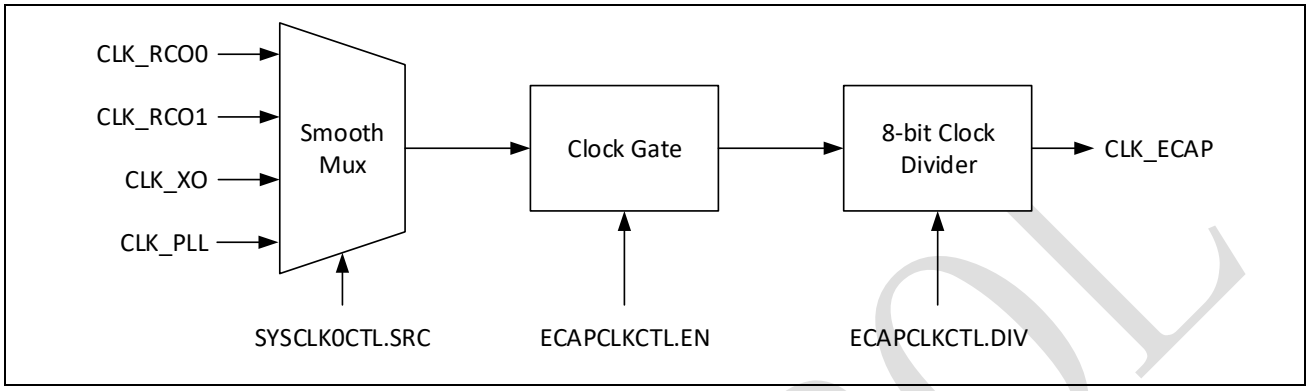
Figure 3-15: PWM clock diagram



3.8.10 ECAP clock

ECAP clock is selected from the four root clocks, gated and divided down as shown in Figure 3-16. The maximum allowed frequency is 200MHz.

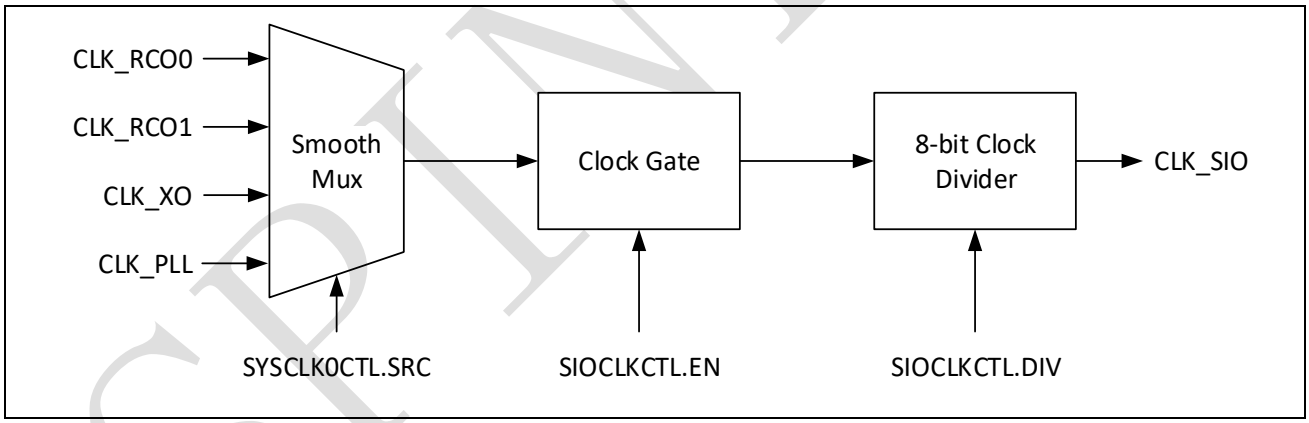
Figure 3-16: ECAP clock diagram



3.8.11 SIO clock

SIO clock is selected from the four root clocks, gated and divided down as shown in Figure 3-17. The maximum allowed frequency is 100MHz.

Figure 3-17: SIO clock diagram



3.9 Clock safety

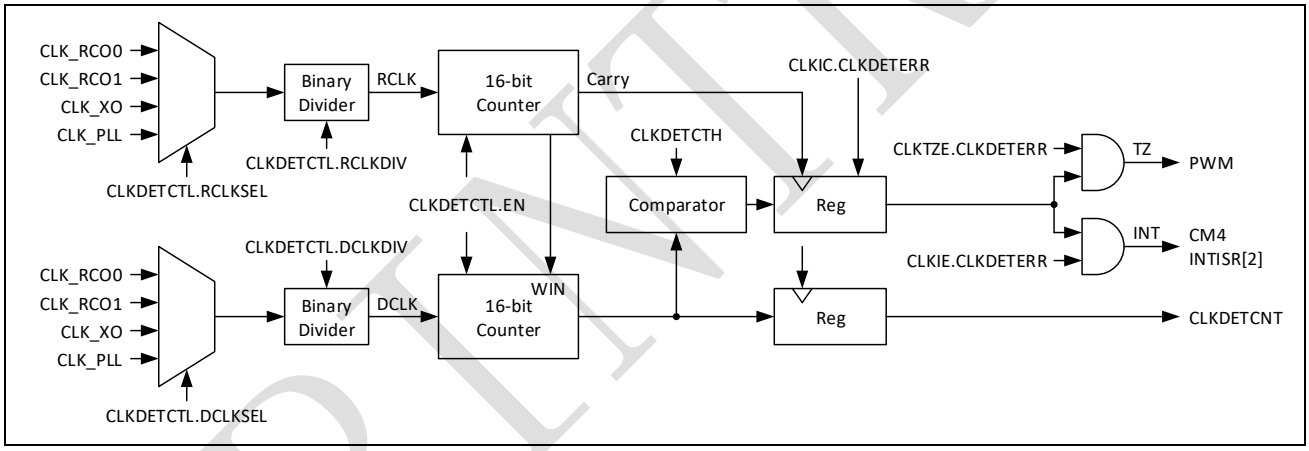
The SPD1148 provides clock cross-detection circuits for clock safety concern. As shown in Figure 3-18, reference clock and detected clock are selected independently. Each of them will go through a binary divider to get the RCLK and DCLK. The RCLK is used to generate the time window, during which period the DCLK is counting. When the counting window is finished, the final value of DCLK counter is latched into CLKDETCNT register. The counted value is also compared with the high and low threshold given in CLKDETCTH register and the error status is latched into CLKIF.CLKDETERR if the value is out of the boundary. It further triggers a CPU interrupt or a PWM trip-zone event if as enabled by corresponding bit in CLKIE and CLKTZE registers. The DCLK can be brought out to GPIO11 by setting GPIO11.MUXSEL=4, or to GPIO40 by setting GPIO40.MUXSEL=3. Refer to IO_Channel_Defination for the details.

CLKDETCNT value is calculated as follows:

$$\text{CLKDETCNT} = \frac{f_{DCLK}}{f_{RCLK}} \times 65536$$

where f_{DCLK} and f_{RCLK} are the frequency of DCLK and RCLK separately, as shown in Figure 3-18.

Figure 3-18: Clock cross-detection



3.10 Registers

3.10.1 Clock register map

Table 3-4: CLOCK Module Base Address

Peripheral Module	Base Address
CLOCK	0x4000 0200

Table 3-5: CLOCK Register Map

Register	Offset	Description	Reset Value
CLKSTS	0x0	Clock Status Register	0x00000003

Register	Offset	Description	Reset Value
RCO0CTL*	0x8	RCO0 Control Register	0x00000579
XOCTL*	0x10	Crystal Oscillator Control Register	0x00000FF2
PLLCTL0*	0x14	PLL Control Register 0	0x00000C40
PLLCTL1*	0x18	PLL Control Register 1	0x0C320000
CLKDETCTL*	0x1C	Clock Detection Control Register	0x00000000
CLKDETCTH*	0x20	Clock Detection Counter Threshold Register	0x00000000
CLKDETCNT	0x24	Clock Detection Counter Register	0x00000000
CLKIF	0x28	Clock Interrupt Flag Register	0x00000000
CLKIC	0x2C	Clock Interrupt Clear Register	0x00000000
CLKIE*	0x30	Clock Interrupt Enable Register	0x00000000
CLKTZE*	0x34	Clock Trip-zone Event Enable Register	0x00000000
SYSCLK0CTL*	0x38	SYSCLK0 Control Register	0x00000000
HCLKCTL*	0x3C	HCLK Control Register	0x00000000
ADCCLKCTL*	0x40	ADC Clock Control Register	0x00000000
PWMCLKCTL*	0x44	PWM Clock Control Register	0x00000000
ECAPCLKCTL*	0x48	ECAP Clock Control Register	0x00000000
TMR0CLKCTL*	0x4C	Timer 0 Clock Control Register	0x00000000
TMR1CLKCTL*	0x50	Timer 1 Clock Control Register	0x00000000
TMR2CLKCTL*	0x54	Timer 2 Clock Control Register	0x00000000
SIOCLKCTL*	0x58	SIO Clock Control Register	0x00000000
SYSCLK1CTL*	0x5C	SYSCLK1 Control Register	0x00000000
PCLKCTL*	0x60	PCLK Control Register	0x00000000
DGCLKCTL*	0x64	Deglitch Clock Control Register	0x00000200
UARTCLKCTL*	0x68	UART Clock Control Register	0x00000000
SSPCLKCTL*	0x6C	SSP Clock Control Register	0x00000000
I2CCLKCTL*	0x70	I2C Clock Control Register	0x00000000
WDT0CLKCTL*	0x74	WDT0 Clock Control Register	0x0000010F
WDT1CLKCTL*	0x78	WDT1 Clock Control Register	0x00000300
CLKREGKEY	0x7C	Clock Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the CLKREGKEY=0x1ACCE551.

3.10.2 Clock registers

Table 3-6: Clock Status Register (CLKSTS) Layout

CLKSTS (Clock Status Register) Offset: 0x0 Default: 0x00000003 Access: CLOCK -> CLKSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							CLKDETERR
7	6	5	4	3	2	1	0
PLLSLOW	PLLFAST	VCOFREQVLD	VCOFREQ	PLLRDY	XORDY	RCO1RDY	RCO0RDY

Table 3-7: Clock Status Register (CLKSTS) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	CLKDETERR	RO	0x0	Indicate error detected by CLKDET 0: 1:
7	PLLSLOW	RO	0x0	Indicate PLL is unlocked and slower than expected. Possible reason can be: 1. Fin/NIN*NFB/65536 is higher than 600MHz (upper boundary of VCO free-running frequency) 2. VCO fails to oscillate 3. Malfunction of feedback divider so that there is no output or the equivalent dividing ratio is higher than expected 0: 1:
6	PLLFAST	RO	0x0	Indicate PLL is unlocked and faster than expected. Possible reason can be: 1. Fin/NIN*NFB/65536 is lower than 400MHz (lower boundary of VCO free-running frequency) 2. The reference clock input is somehow missing 3. Malfunction of feedback divider so that the equivalent dividing ratio is lower than expected. 0: 1:

Bits	Field Name	Type	Reset	Description
5	VCOFREQVLD	RO	0x0	VCOFREQ flag valid indicator 0: VCOFREQ is invalid 1: VCOFREQ is valid
4	VCOFREQ	RO	0x0	VCO frequency flag for trimming 0: VCO in PLL is too slow 1: VCO in PLL is too fast
3	PLLRDY	RO	0x0	PLL analog clock ready indicator 0: Not ready 1: Ready
2	XORDY	RO	0x0	XO clock ready indicator 0: Not ready 1: Ready
1	RCO1RDY	RO	0x1	RCO1 clock ready indicator 0: Not ready 1: Ready
0	RCO0RDY	RO	0x1	RCO0 clock ready indicator 0: Not ready 1: Ready

Table 3-8: RCO0 Control Register (RCO0CTL) Layout

RCO0CTL (RCO0 Control Register) Offset: 0x8 Default: 0x00000579
Access: CLOCK -> RCO0CTL.all

31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					FREQTRIM		
7	6	5	4	3	2	1	0
FREQTRIM						LFMODE	EN

Table 3-9: RCO0 Control Register (RCO0CTL) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10:2	FREQTRIM	RW	0x15E	RCO0 frequency trimming Frequency increases for larger code
1	LFMODE	RW	0x0	RCO0 low frequency mode 0: Normal 32MHz mode (Trimmed) 1: Low frequency mode (Not trimmed)

Bits	Field Name	Type	Reset	Description
0	EN	RW	0x1	RCO0 enable 0: Disable 1: Enable

Table 3-10: Crystal Oscillator Control Register (XOCTL) Layout

XOCTL (Crystal Oscillator Control Register) Offset: 0x10 Default: 0x00000FF2 Access: CLOCK -> XOCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_14							
23	22	21	20	19	18	17	16
RESERVED_31_14							
15	14	13	12	11	10	9	8
RESERVED_31_14		FEEDXIO	EXTRFB	PRECNT			
7	6	5	4	3	2	1	0
PRECNT				RESERVED_3_2		FASTEN	EN

Table 3-11: Crystal Oscillator Control Register (XOCTL) Description

Bits	Field Name	Type	Reset	Description
31:14	RESERVED_31_14	RO	0x0	Reserved.
13	FEEDXIO	RW	0x0	External clock feed into XIO pin 0: External clock feed into XIN pin 1: External clock feed into XIO pin
12	EXTRFB	RW	0x0	Use external feedback resistor 0: Use internal feedback resistor 1: Use external feedback resistor
11:4	PRECNT	RW	0xFF	Pre-counted target value before XO clock is regarded as ready: Target = 1024 * PRECNT
3:2	RESERVED_3_2	RW	0x0	Reserved.
1	FASTEN	RW	0x1	Enable noise injection for fast startup 0: Disable 1: Enable
0	EN	RW	0x0	Crystal oscillator clock enable 0: Disable 1: Enable

Table 3-12: PLL Control Register 0 (PLLCTL0) Layout

PLLCTL0 (PLL Control Register 0) Offset: 0x14 Default: 0x00000C40 Access: CLOCK -> PLLCTL0.all							
31	30	29	28	27	26	25	24
RESERVED_31_14							
23	22	21	20	19	18	17	16
RESERVED_31_14							
15	14	13	12	11	10	9	8
RESERVED_31_14		ICP					OPENLOOP
7	6	5	4	3	2	1	0
RCLKSELXO	VCOTRIM			FCALWIN	FCALEN	OE	EN

Table 3-13: PLL Control Register 0 (PLLCTL0) Description

Bits	Field Name	Type	Reset	Description
31:14	RESERVED_31_14	RO	0x0	Reserved.
13:9	ICP	RW	0x6	PLL charge pump current setting Must be set to 0.96*NFB/(14-VCOTRIM), where NFB is the PLL feedback factor in real number
8	OPENLOOP	RW	0x0	PLL open-loop mode (to trim vco) 0: Close loop 1: Open loop
7	RCLKSELXO	RW	0x0	PLL reference clock select 0: CLK_RCO0 (RC oscillator 0) 1: CLK_XO (Crystal oscillator)
6:4	VCOTRIM	RW	0x4	VCO frequency trimming VCO free-run frequency increases with larger code
3	FCALWIN	RW	0x0	PLL frequency calibration time window For each calibration cycle, keep it as 1 until VCOFREQVLD=1 and release it as 0 0: Window is disabled 1: Window is enabled
2	FCALEN	RW	0x0	PLL frequency calibration enable 0: Disable 1: Enable
1	OE	RW	0x0	PLL digital clock output enable 0: Disable 1: Enable
0	EN	RW	0x0	PLL enable 0: Disable 1: Enable

Table 3-14: PLL Control Register 1 (PLLCTL1) Layout

PLLCTL1 (PLL Control Register 1) Offset: 0x18 Default: 0x0C320000
Access: CLOCK -> PLLCTL1.all

31	30	29	28	27	26	25	24
RESERVED_31	NOUT				NIN		
23	22	21	20	19	18	17	16
NFB							
15	14	13	12	11	10	9	8
NFB							
7	6	5	4	3	2	1	0
NFB							

Table 3-15: PLL Control Register 1 (PLLCTL1) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:27	NOUT	RW	0x1	VCO to digital clock dividing ratio (Fvco/Fout)
26:24	NIN	RW	0x4	PLL input clock dividing ratio (Fin/Fpfd) 000: Invalid 001: Set to 1 if Fin(MHz) is in [4, 8]: Fin/Fpfd=1 010: Set to 2 if Fin(MHz) is in (8, 16]: Fin/Fpfd=2 011: Set to 3 if Fin(MHz) is in (16, 24]: Fin/Fpfd=3 100: Set to 4 if Fin(MHz) is in (24, 32]: Fin/Fpfd=4 101: Set to 5 if Fin(MHz) is in (32, 40]: Fin/Fpfd=5 110: Set to 6 if Fin(MHz) is in (40, 48]: Fin/Fpfd=6 111: Set to 1 if Fin(MHz) is in (48, 56]: Fin/Fpfd=7
23:0	NFB	RW	0x320000	Feedback dividing ratio (Fvco/Fpfd)

Table 3-16: Clock Detection Control Register (CLKDETCTL) Layout

CLKDETCTL (Clock Detection Control Register) Offset: 0x1C Default: 0x00000000
Access: CLOCK -> CLKDETCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					EN	DCLKDIV	
7	6	5	4	3	2	1	0
DCLKDIV	DCLKSEL		RCLKDIV			RCLKSEL	

Table 3-17: Clock Detection Control Register (CLKDETCTL) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	EN	RW	0x0	Clock detection enable When disabled, the interrupts and the clock error events will be cleared 0: Disable clock detection 1: Enable clock detection

Bits	Field Name	Type	Reset	Description
9:7	DCLKDIV	RW	0x0	Detected clock dividing ratio 000: Divide by 1 001: Divide by 2 010: Divide by 4 011: Divide by 8 100: Divide by 16 101: Divide by 32 110: Divide by 64 111: Divide by 128
6:5	DCLKSEL	RW	0x0	Detected clock select 00: CLK_RCO0 01: CLK_RCO1 10: CLK_XO 11: CLK_PLL
4:2	RCLKDIV	RW	0x0	Reference clock dividing ratio 000: Divide by 1 001: Divide by 2 010: Divide by 4 011: Divide by 8 100: Divide by 16 101: Divide by 32 110: Divide by 64 111: Divide by 128
1:0	RCLKSEL	RW	0x0	Reference clock select 00: CLK_RCO0 01: CLK_RCO1 10: CLK_XO 11: CLK_PLL

Table 3-18: Clock Detection Counter Threshold Register (CLKDETCTH) Layout

CLKDETCTH (Clock Detection Counter Threshold Register) Offset: 0x20 Default: 0x00000000 Access: CLOCK -> CLKDETCTH.all							
31	30	29	28	27	26	25	24
HI							
23	22	21	20	19	18	17	16
HI							
15	14	13	12	11	10	9	8
LO							
7	6	5	4	3	2	1	0
LO							

Table 3-19: Clock Detection Counter Threshold Register (CLKDETCTH) Description

Bits	Field Name	Type	Reset	Description
31:16	HI	RW	0x0	Upper counter threshold for clock detect If the final counter value > HI, a trip-zone event will be generated for PWM
15:0	LO	RW	0x0	Lower counter threshold for clock detect If the final counter value < LO, a trip-zone event will be generated for PWM

Table 3-20: Clock Detection Counter Register (CLKDETCNT) Layout

CLKDETCNT (Clock Detection Counter Register) Offset: 0x24 Default: 0x00000000 Access: CLOCK -> CLKDETCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 3-21: Clock Detection Counter Register (CLKDETCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Final counter value of the clock detect

Table 3-22: Clock Interrupt Flag Register (CLKIF) Layout

CLKIF (Clock Interrupt Flag Register) Offset: 0x28 Default: 0x00000000 Access: CLOCK -> CLKIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					CLKINT	CLKDETERR	PLLUNLOCK

Table 3-23: Clock Interrupt Flag Register (CLKIF) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	CLKINT	RO	0x0	Latched clock interrupt 0: No clock interrupt occurred 1: Clock interrupt occurred
1	CLKDETERR	RO	0x0	Latched CLKDET error flag, namely the CLKDETCNT is out of the range defined by CLKDETCTH 0: CLKDETCNT is within the expected range 1: CLKDETCNT is out of the expected range
0	PLLUNLOCK	RO	0x0	Latched PLL unlock flag Detailed reason can be referred to CLKSTS[PLLFAST] and CLKSTS[PLLSLOW] 0: PLL is locked 1: PLL is unlocked

Table 3-24: Clock Interrupt Clear Register (CLKIC) Layout

CLKIC (Clock Interrupt Clear Register) Offset: 0x2C Default: 0x00000000
Access: CLOCK -> CLKIC.all

31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					CLKINT	CLKDETERR	PLLUNLOCK

Table 3-25: Clock Interrupt Clear Register (CLKIC) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	CLKINT	W1C	0x0	Clock interrupt clear A new clock interrupt will be immediately issued if CLKIF is not zero after the clear 0: Write a 0 has no effect and always read back 0 1: Write a 1 clear the CLKIF.CLKINT. This bit is self-cleared to 0.
1	CLKDETERR	W1C	0x0	CLKDET error interrupt flag clear 0: Write a 0 has no effect and always read back 0

Bits	Field Name	Type	Reset	Description
				1: Write a 1 clear the CLKIF.CLKDETERR. This bit is self-cleared to 0.
0	PLLUNLOCK	W1C	0x0	PLL unlock interrupt flag clear 0: Write a 0 has no effect and always read back 0 1: Write a 1 clear the CLKIF.PLLUNLOCK, CLKSTS.PLLFAST, CLKSTS.PLLSLOW. This bit is self-cleared to 0.

Table 3-26: Clock Interrupt Enable Register (CLKIE) Layout

CLKIE (Clock Interrupt Enable Register) Offset: 0x30 Default: 0x00000000 Access: CLOCK -> CLKIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						CLKDETERR	PLLUNLOCK

Table 3-27: Clock Interrupt Enable Register (CLKIE) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	CLKDETERR	RW	0x0	CLKDET error interrupt enable 0: CLKDET error will not trigger clock interrupt 1: CLKDET error will trigger clock interrupt
0	PLLUNLOCK	RW	0x0	PLL unlock interrupt enable 0: PLL unlock event will not trigger clock interrupt 1: PLL unlock event will trigger clock interrupt

Table 3-28: Clock Trip-zone Event Enable Register (CLKTZE) Layout

CLKTZE (Clock Trip-zone Event Enable Register) Offset: 0x34 Default: 0x00000000 Access: CLOCK -> CLKTZE.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						CLKDETERR	PLLUNLOCK

Table 3-29: Clock Trip-zone Event Enable Register (CLKTZE) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	CLKDETERR	RW	0x0	CLKDET error trip-zone event enable 0: CLKDET error will not trigger PWM trip-zone 1: CLKDET error will trigger PWM trip-zone
0	PLLUNLOCK	RW	0x0	PLL unlock trip-zone event enable 0: PLL unlock will not trigger PWM trip-zone 1: PLL unlock will trigger PWM trip-zone

Table 3-30: SYSCLK0 Control Register (SYSCLK0CTL) Layout

SYSCLK0CTL (SYSCLK0 Control Register) Offset: 0x38 Default: 0x00000000 Access: CLOCK -> SYSCLK0CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					SRC		RESERVED_8_0
7	6	5	4	3	2	1	0
RESERVED_8_0							

Table 3-31: SYSCLK0 Control Register (SYSCLK0CTL) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10:9	SRC	RW	0x0	Clock source select 00: RCO0 01: RCO1 10: XO 11: PLL
8:0	RESERVED_8_0	RO	0x0	Reserved.

Table 3-32: HCLK Control Register (HCLKCTL) Layout

HCLKCTL (HCLK Control Register) Offset: 0x3C Default: 0x00000000 Access: CLOCK -> HCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
DIV							

Table 3-33: HCLK Control Register (HCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:0	DIV	RW	0x0	Clock dividing ratio from system clock

Table 3-34: ADC Clock Control Register (ADCCLKCTL) Layout

ADCCLKCTL (ADC Clock Control Register) Offset: 0x40 Default: 0x00000000 Access: CLOCK -> ADCCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-35: ADC Clock Control Register (ADCCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-36: PWM Clock Control Register (PWMCLKCTL) Layout

PWMCLKCTL (PWM Clock Control Register) Offset: 0x44 Default: 0x00000000 Access: CLOCK -> PWMCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-37: PWM Clock Control Register (PWMCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-38: ECAP Clock Control Register (ECAPCLKCTL) Layout

ECAPCLKCTL (ECAP Clock Control Register) Offset: 0x48 Default: 0x00000000 Access: CLOCK -> ECAPCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-39: ECAP Clock Control Register (ECAPCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-40: Timer 0 Clock Control Register (TMR0CLKCTL) Layout

TMR0CLKCTL (Timer 0 Clock Control Register) Offset: 0x4C Default: 0x00000000 Access: CLOCK -> TMR0CLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-41: Timer 0 Clock Control Register (TMR0CLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-42: Timer 1 Clock Control Register (TMR1CLKCTL) Layout

TMR1CLKCTL (Timer 1 Clock Control Register) Offset: 0x50 Default: 0x00000000 Access: CLOCK -> TMR1CLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-43: Timer 1 Clock Control Register (TMR1CLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-44: Timer 2 Clock Control Register (TMR2CLKCTL) Layout

TMR2CLKCTL (Timer 2 Clock Control Register) Offset: 0x54 Default: 0x00000000 Access: CLOCK -> TMR2CLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-45: Timer 2 Clock Control Register (TMR2CLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-46: SIO Clock Control Register (SIOCLKCTL) Layout

SIOCLKCTL (SIO Clock Control Register) Offset: 0x58 Default: 0x00000000 Access: CLOCK -> SIOCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-47: SIO Clock Control Register (SIOCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-48: SYSCLK1 Control Register (SYSCLK1CTL) Layout

SYSCLK1CTL (SYSCLK1 Control Register) Offset: 0x5C Default: 0x00000000 Access: CLOCK -> SYSCLK1CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					SRC		RESERVED_8_(
7	6	5	4	3	2	1	0
RESERVED_8_0							

Table 3-49: SYSCLK1 Control Register (SYSCLK1CTL) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10:9	SRC	RW	0x0	Clock source select 00: RCO0 01: RCO1 10: XO 11: PLL
8:0	RESERVED_8_0	RO	0x0	Reserved.

Table 3-50: PCLK Control Register (PCLKCTL) Layout

PCLKCTL (PCLK Control Register) Offset: 0x60 Default: 0x00000000 Access: CLOCK -> PCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
DIV							

Table 3-51: PCLK Control Register (PCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:0	DIV	RW	0x0	Clock dividing ratio from system clock

Table 3-52: Deglitch Clock Control Register (DGCLKCTL) Layout

DGCLKCTL (Deglitch Clock Control Register) Offset: 0x64 Default: 0x00000200 Access: CLOCK -> DGCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					PDIV		
7	6	5	4	3	2	1	0
DIV							

Table 3-53: Deglitch Clock Control Register (DGCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10:8	PDIV	RW	0x2	Clock binary pre-dividing ratio from system clock 000: Divide by 1 001: Divide by 2 010: Divide by 4 011: Divide by 8 100: Divide by 16 101: Divide by 32 110: Divide by 64 111: Divide by 128
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-54: UART Clock Control Register (UARTCLKCTL) Layout

UARTCLKCTL (UART Clock Control Register) Offset: 0x68 Default: 0x00000000 Access: CLOCK -> UARTCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-55: UART Clock Control Register (UARTCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-56: SSP Clock Control Register (SSPCLKCTL) Layout

SSPCLKCTL (SSP Clock Control Register) Offset: 0x6C Default: 0x00000000 Access: CLOCK -> SSPCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_17							
23	22	21	20	19	18	17	16
RESERVED_31_17							EN
15	14	13	12	11	10	9	8
DIV							
7	6	5	4	3	2	1	0
DIV							

Table 3-57: SSP Clock Control Register (SSPCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:17	RESERVED_31_17	RO	0x0	Reserved.
16	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
15:0	DIV	RW	0x0	Clock dividing ratio

Table 3-58: I2C Clock Control Register (I2CCLKCTL) Layout

I2CCLKCTL (I2C Clock Control Register) Offset: 0x70 Default: 0x00000000 Access: CLOCK -> I2CCLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							EN
7	6	5	4	3	2	1	0
DIV							

Table 3-59: I2C Clock Control Register (I2CCLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	EN	RW	0x0	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-60: WDT0 Clock Control Register (WDT0CLKCTL) Layout

WDT0CLKCTL (WDT0 Clock Control Register) Offset: 0x74 Default: 0x0000010F Access: CLOCK -> WDT0CLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					SRC		EN
7	6	5	4	3	2	1	0
DIV							

Table 3-61: WDT0 Clock Control Register (WDT0CLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10:9	SRC	RW	0x0	Clock source select 00: RCO0 01: RCO1 10: XO 11: PLL
8	EN	RW	0x1	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0xF	Clock dividing ratio

Table 3-62: WDT1 Clock Control Register (WDT1CLKCTL) Layout

WDT1CLKCTL (WDT1 Clock Control Register) Offset: 0x78 Default: 0x00000300 Access: CLOCK -> WDT1CLKCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					SRC		EN
7	6	5	4	3	2	1	0
DIV							

Table 3-63: WDT1 Clock Control Register (WDT1CLKCTL) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10:9	SRC	RW	0x1	Clock source select 00: RCO0 01: RCO1 10: XO 11: PLL
8	EN	RW	0x1	Clock output enable 0: Disable clock output 1: Enable clock output
7:0	DIV	RW	0x0	Clock dividing ratio

Table 3-64: Clock Register Write-Allow Key Register (CLKREGKEY) Layout

CLKREGKEY (Clock Register Write-Allow Key Register) Offset: 0x7C Default: 0x1ACCE551 Access: CLOCK -> CLKREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 3-65: Clock Register Write-Allow Key Register (CLKREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected clock registers

4 General-purpose I/O (GPIO)

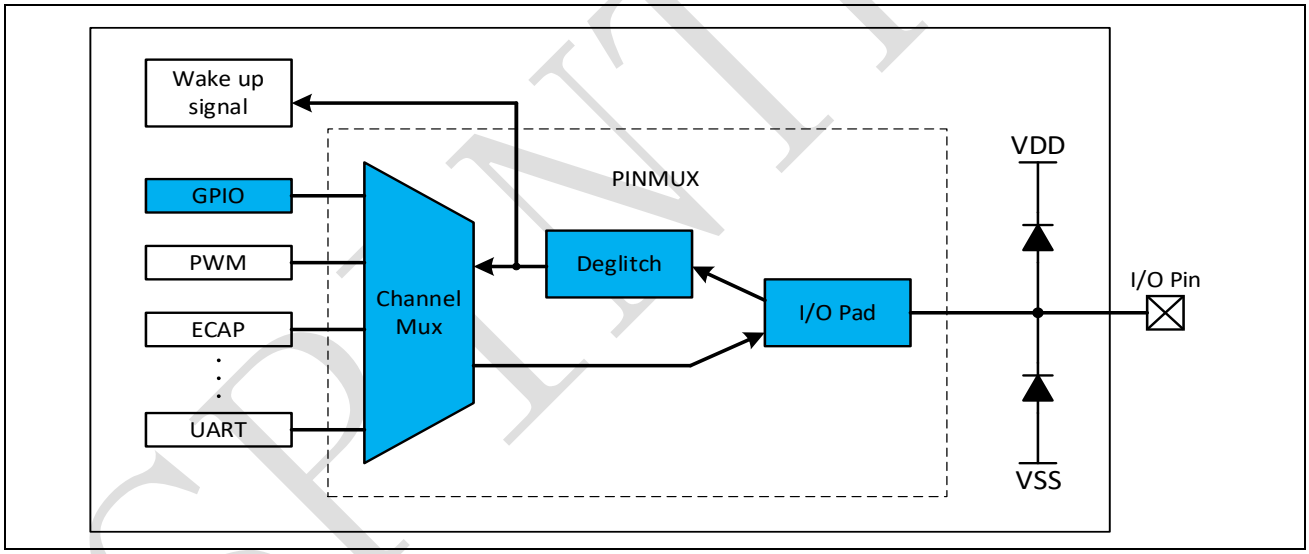
4.1 Overview

The SPD1148 provides 40 I/O pins, while the BOOT pin can be used as GPIO40 with several limitations. These I/O pins are multiplexed so that they can be configured as either general-purpose I/Os or any one of the alternative functions. Each I/O pin can be programmed and used independently.

This chapter describes the detailed I/O control, including pin multiplexing and GPIO-related functions. Figure 4-1 is the structure diagram of the I/O control, which consists of two parts:

- PINMUX module
  - I/O Pad: Pull-up/pull-down control, input/output control, Schmitt input control and output driving strength control
  - Deglitch: Input deglitch control
  - MUX: Pin multiplexing
- GPIO module
  - GPIO function IP control

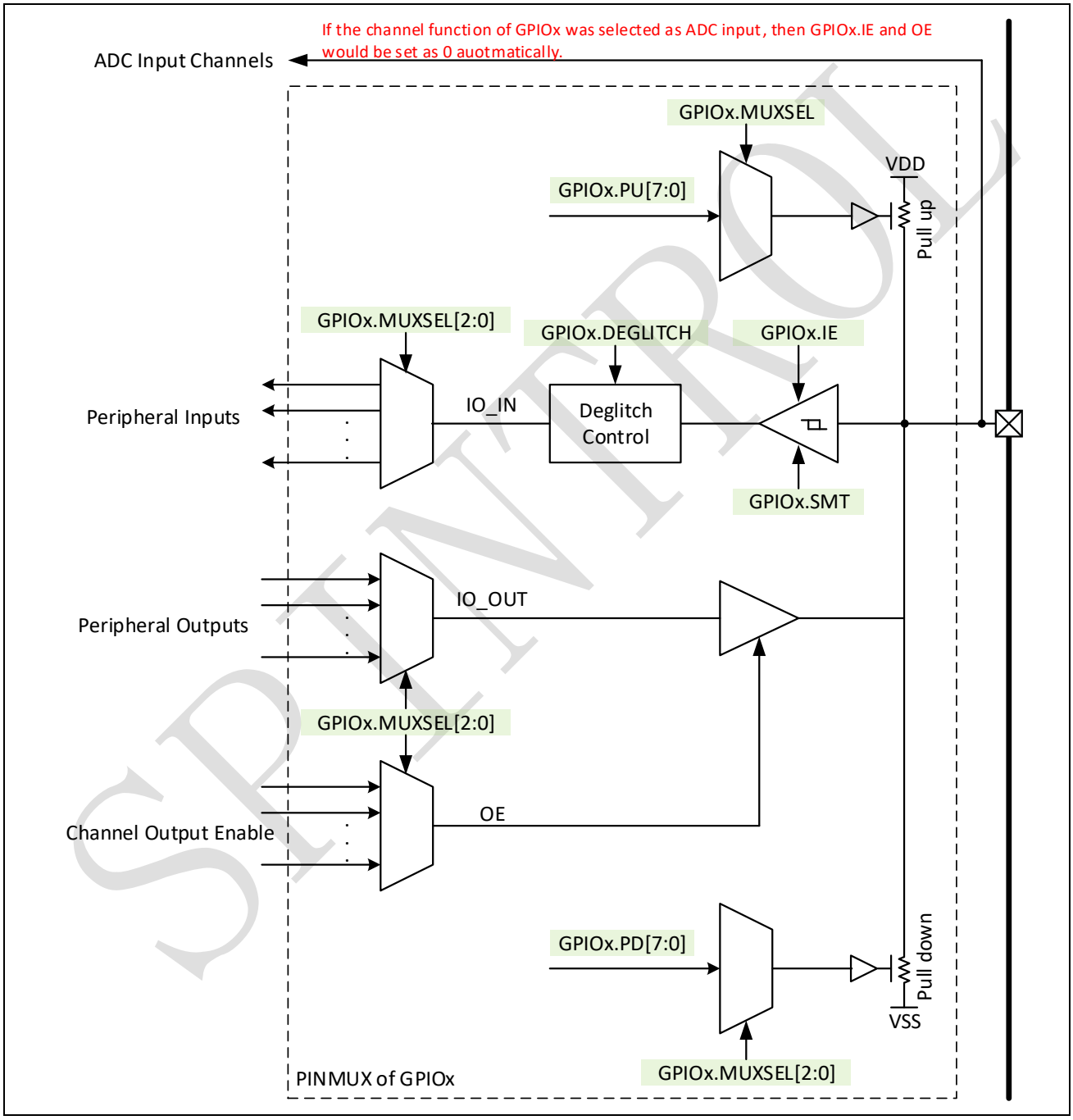
Figure 4-1: Structure diagram of I/O control



4.2 Pin multiplexing (PINMUX)

Figure 4-2 is the structure diagram of the PINMUX. Multiplexers are used for each I/O pin to support overall 8 function channels. The function channel can be either input or output. If the I/O pin is used as input, the output buffer will be disabled and set into high-impedance state. The input signal can also pass through a Schmitt trigger or a re-sampling block to suppress the possible glitches.

Figure 4-2: PINMUX structure diagram

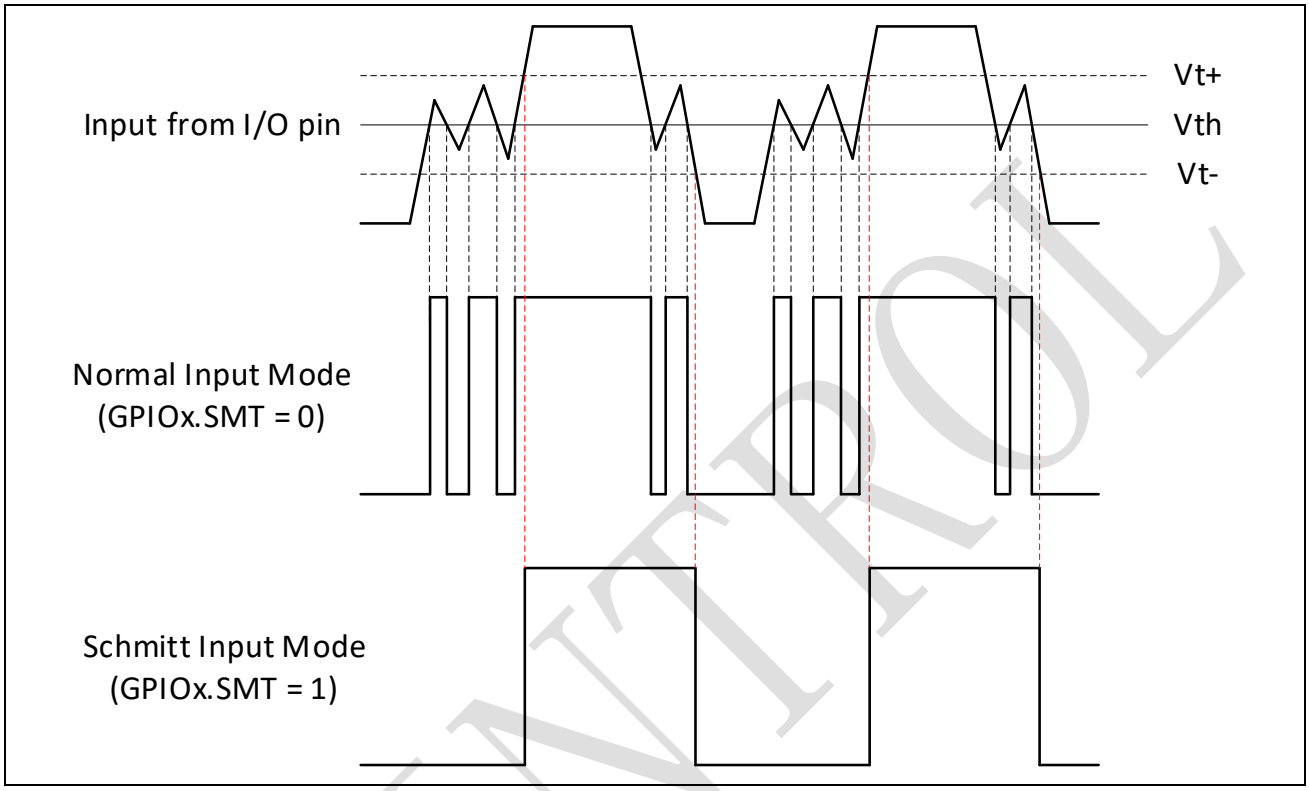


Note: If the channel function of GPIOx selected as ADC input, then the GPIOx.IE and OE would be set as 0 automatically

4.2.1 Schmitt control

The Schimtt input mode can be enabled by setting register GPIOx.SMT. As shown in Figure 4-3, Schmitt input mode can prevent false triggering due to small noise fluctuations (noise) on the input.

Figure 4-3: Schmitt input mode

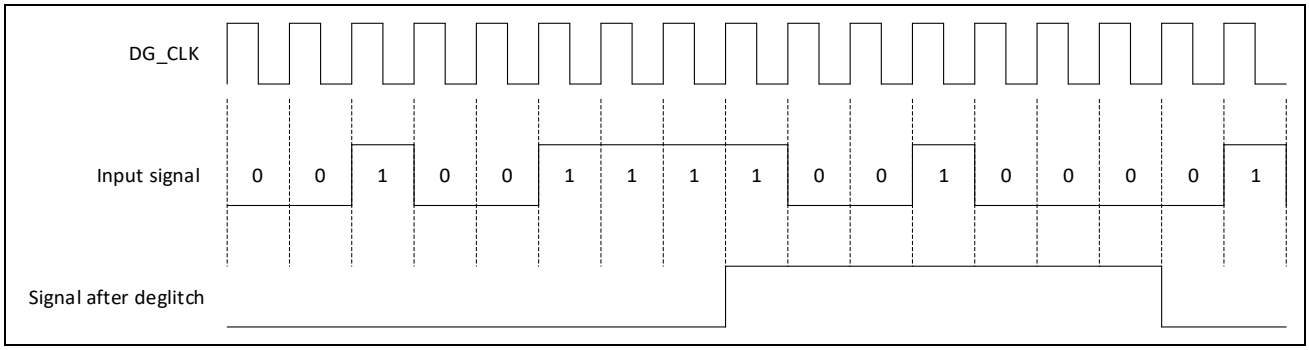


[1] Vth – Input toggling threshold for normal input mode;
[2] Vt+ - Input upper toggling threshold for Schmitt input mode; Vt- - Input lower toggling threshold for Schmitt input mode;
[3] Typically, Vth = 1.5V, Vt+ = 1.67V, Vt- = 1.38V.

4.2.2 Deglitch control

The input deglitching can be enabled by setting register GPIOx.DEGLITCH. Figure 4-4 shows the principle of input deglitching. When the input signal of the Deglitch module remains a value unchanged for 3 consecutive DG_CLK cycles, the output of the Deglitch module will change to this value, otherwise the output of the Deglitch module remains the original value.

Figure 4-4: Input deglitching



4.2.3 I/O alternate functions

The channel function for all the I/O pins are summarized in Table 4-1.

Table 4-1: I/O function channel definition

Name	Channel (Default channel is highlight in red)							
	0	1	2	3	4	5	6	7
GPIO0	GPIO0	**ADC0**					COMP0H	ECAPO
GPIO1	GPIO1	**ADC1**					COMP0L	ECAPO
GPIO2	GPIO2	**ADC2**					COMP1H	ECAPO
GPIO3	GPIO3	**ADC3**					COMP1L	ECAPO
~~GPIO4~~	~~GPIO4~~	~~**ADC4**~~					~~COMP2H~~	~~ECAPO~~
~~GPIO5~~	~~GPIO5~~	~~**ADC5**~~					~~COMP2L~~	~~ECAPO~~
~~GPIO6~~	~~GPIO6~~	~~**ADC6**~~						~~ECAPO~~
~~GPIO7~~	~~GPIO7~~	~~**ADC7**~~						~~ECAPO~~
GPIO8	GPIO8	**ADC8**	SPI_SCLK	PWMSOC			COMP3H	ECAPO
GPIO9	GPIO9	**ADC9**	SPI_SFRM				COMP3L	ECAPO
GPIO10	GPIO10	**ADC10**	SPI_MOSI	SPI_MISO			COMP4H	ECAPO
GPIO11	GPIO11	**ADC11**	SPI_MISO	SPI_MOSI	DCLK		COMP4L	ECAPO
GPIO12	GPIO12	**ADC12**	I2C_SCL					ECAPO
~~GPIO13~~	~~GPIO13~~	~~**ADC13**~~	~~I2C_SDA~~					~~ECAPO~~
~~GPIO14~~	~~GPIO14~~	~~**ADC14**~~	~~UART_TXD~~	~~UART_RXD~~				~~ECAPO~~
~~GPIO15~~	~~GPIO15~~	~~**ADC15**~~	~~UART_RXD~~	~~UART_TXD~~				~~ECAPO~~
GPIO16	**GPIO16**	XIN	UART_TXD	UART_RXD	PWM2A	PWM5A	SIO0_12	ECAPO
GPIO17	**GPIO17**	XIO	UART_RXD	UART_TXD	PWM2B	PWM5B	SIO0_13	ECAPO
~~GPIO18~~	~~**GPIO18**~~	~~PWM3A~~	~~PWM3A~~	~~COMP3H~~	~~PWM0A~~	~~PWM0A~~	~~SIO0_14~~	~~ECAPO~~
~~GPIO19~~	~~**GPIO19**~~	~~PWM4A~~	~~PWM3B~~	~~COMP3L~~	~~PWM1A~~	~~PWM0B~~	~~SIO0_15~~	~~ECAPO~~
~~GPIO20~~	~~**GPIO20**~~			~~COMP4H~~	~~PWM2A~~	~~PWM1A~~	~~SIO0_16~~	~~ECAPO~~
~~GPIO21~~	~~**GPIO21**~~			~~COMP4L~~	~~PWM0B~~	~~PWM1B~~	~~SIO0_17~~	~~ECAPO~~
~~GPIO22~~	~~**GPIO22**~~				~~PWM1B~~	~~PWM2A~~	~~SIO0_0~~	~~ECAPO~~
~~GPIO23~~	~~**GPIO23**~~				~~PWM2B~~	~~PWM2B~~	~~SIO0_1~~	~~ECAPO~~
GPIO24	**GPIO24**			COMP0H	PWM3A	PWM3A	SIO0_2	ECAPO
GPIO25	**GPIO25**			COMP0L	PWM4A	PWM3B	SIO0_3	ECAPO
GPIO26	**GPIO26**			COMP1H	PWM5A	PWM4A	SIO0_4	ECAPO
GPIO27	**GPIO27**			COMP1L	PWM3B	PWM4B	SIO0_5	ECAPO
GPIO28	**GPIO28**			COMP2H	PWM4B	PWM5A	SIO0_6	ECAPO
GPIO29	**GPIO29**			COMP2L	PWM5B	PWM5B	SIO0_7	ECAPO

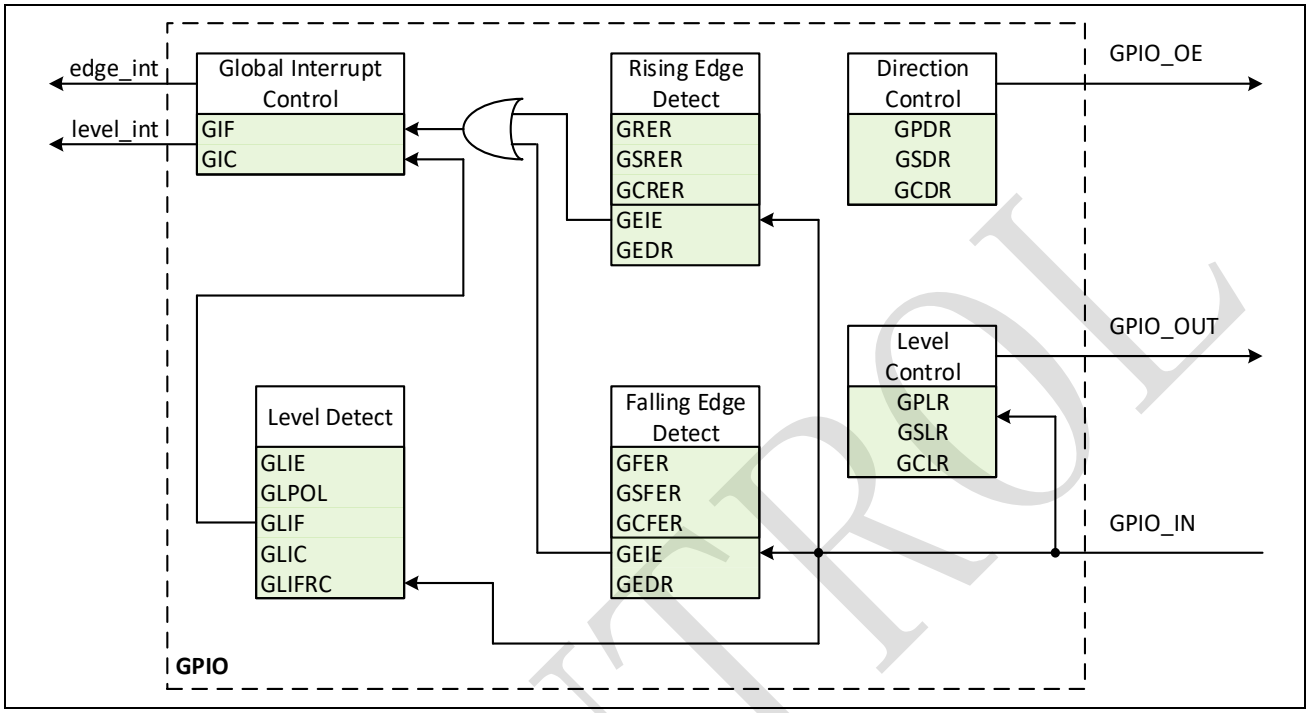
Name	Channel (Default channel is highlight in red)							
	0	1	2	3	4	5	6	7
GPIO30	**GPIO30**	SPI_SCLK	I2C_SCL	COMP3H	PWM3A	PWM0A	SIO0_8	ECAPO
GPIO31	**GPIO31**	SPI_SFRM	I2C_SDA	COMP3L	PWM3B	PWM0B	SIO0_9	ECAPO
GPIO32	**GPIO32**	SPI_MOSI	SPI_MISO	COMP4H	PWM4A	EPWRTZ0O	SIO0_10	ECAPO
~~GPIO33~~	~~**GPIO33**~~	~~SPI_MISO~~	~~SPI_MOSI~~	~~COMP4L~~	~~PWM4B~~	~~EPWRTZ1O~~	~~SIO0_11~~	~~ECAPO~~
GPIO34	**GPIO34**	UART_TXD	UART_RXD	I2C_SDA	SPI_MOSI	SPI_MISO	SIO0_12	ECAPO
GPIO35	**GPIO35**	UART_RXD	UART_TXD	I2C_SCL	SPI_MISO	SPI_MOSI	SIO0_13	ECAPO
GPIO36	**GPIO36**	TDO	UART_RXD	SPI_SCLK	PWM5A	PWM1A	SIO0_14	I2C_SDA
GPIO37	**GPIO37**	TDI	UART_TXD	SPI_SFRM	PWM5B	PWM1B	SIO0_15	I2C_SCL
GPIO38	**GPIO38**	TMS/SWD	I2C_SDA	SPI_MOSI	SPI_MISO	PWM2A	SIO0_16	ECAPO
GPIO39	**GPIO39**	TCK/SWCK	I2C_SCL	SPI_MISO	SPI_MOSI	PWM2B	SIO0_17	ECAPO
GPIO40	**GPIO40**	SPI_SCLK	UART_TXD	DCLK			SIO0_0	ECAPO

Note:
1. In SPD1148, GPIO33 is internally tied to the DVDD. This pin should never be set as output.
2. In SPD1148, GPIO24 ~ GPIO29 are internally connected to Pre-Driver module and should be configured as PWM output function.
3. In SPD1148, the GPIOs with strikeout are not bonded out to the external pin.

4.3 GPIO function IP (GPIO)

Figure 4-5 is the structure diagram of the GPIO.

Figure 4-5: GPIO structure diagram



The GPIO ports are mapped to two groups: GPIO_PORT0 and GPIO_PORT1. Each group is 32 bits width and support bitwise control, but only low 9 bits are used in GPIO_PORT1.

All the GPIO pins are inputs by default. The direction of the GPIO pins is configured by programming the GPIO Pin Direction Register (GPDR0 / GPDR1) or GPIO Set Direction Register (GSDR0 / GSDR1) or GPIO Clear Direction Register (GCDR0 / GCDR1). GSDR and GCDR are bitwise registers, and for each bit in these registers, writing 0 has no effect.

When configured as output pin, programming GPIO Pin Level Register (GPLR0 / GPLR1) or GPIO Set Level Register (GSLR0 / GSLR1) or GPIO Clear Level Register (GCLR0 / GCLR1) determines the GPIO output port status. GSLR and GCLR are bitwise registers, and for each bit in these registers, writing 0 has no effect.

Note:
1. No matter what has been written to GPLR register or GPDR register, readback from GPLR will always reflect the GPIO ports staus, i.e. the status of I/O pads.
2. GPIO33 is internally tied to the DVDD. Therefore, GPLR1[1] is always 1.
3. GPIO33 cannot be set in output direction, i.e. GPDR1[1] and GSDR1[1] should be always 0.

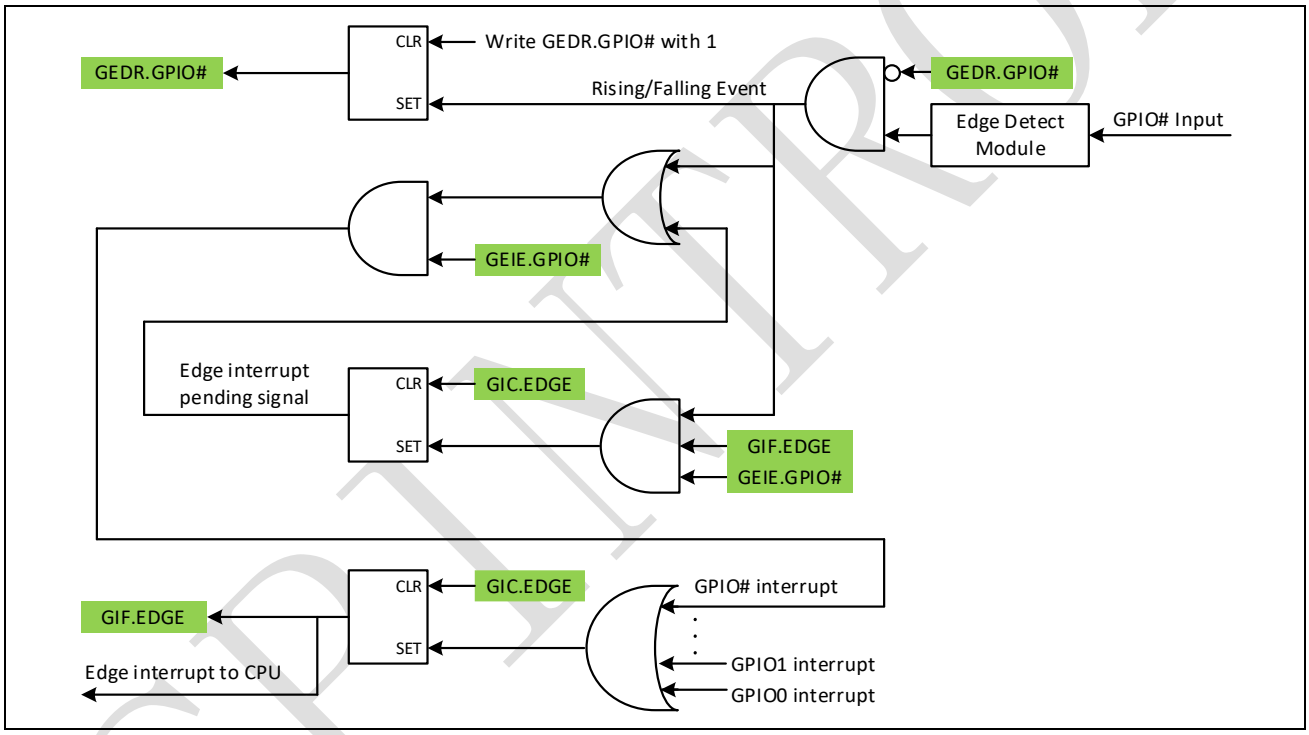
4.3.1 GPIO edge-triggered interrupt

Edge detect module will run when GPIO Rising-edge Detect Enable Register (GRER0 / GRER1) or GPIO Falling-edge Detect Enable Register (GFER0 / GFER1) is set. GSRER, GCRER, GSFER and GCFER are bitwise registers, and for each bit in these registers, writing 0 has no effect.

GPIO Edge Detect Register (GEDR0 / GEDR1) will show the edge detect results. Writing GEDR bit with 1 will clear the associated bit edge detect flag and writing 0 has no effect. The edge event is detected by APB clock synchronize GPIO inputs.

The GPIO edge detect interrupt to CPU will be triggered and the edge detect interrupt flag in GPIO Interrupt Flag Register (GIF.EDGE) will be set if the associated GPIO Edge Interrupt Enable Register (GEIE0 / GEIE1) bit and GEDR bit are all HIGH, as shown in Figure 4-6. Writing 1 to GIC.EDGE will clear the GPIO edge-triggered interrupt flag.

Figure 4-6: GPIO edge-triggered interrupt



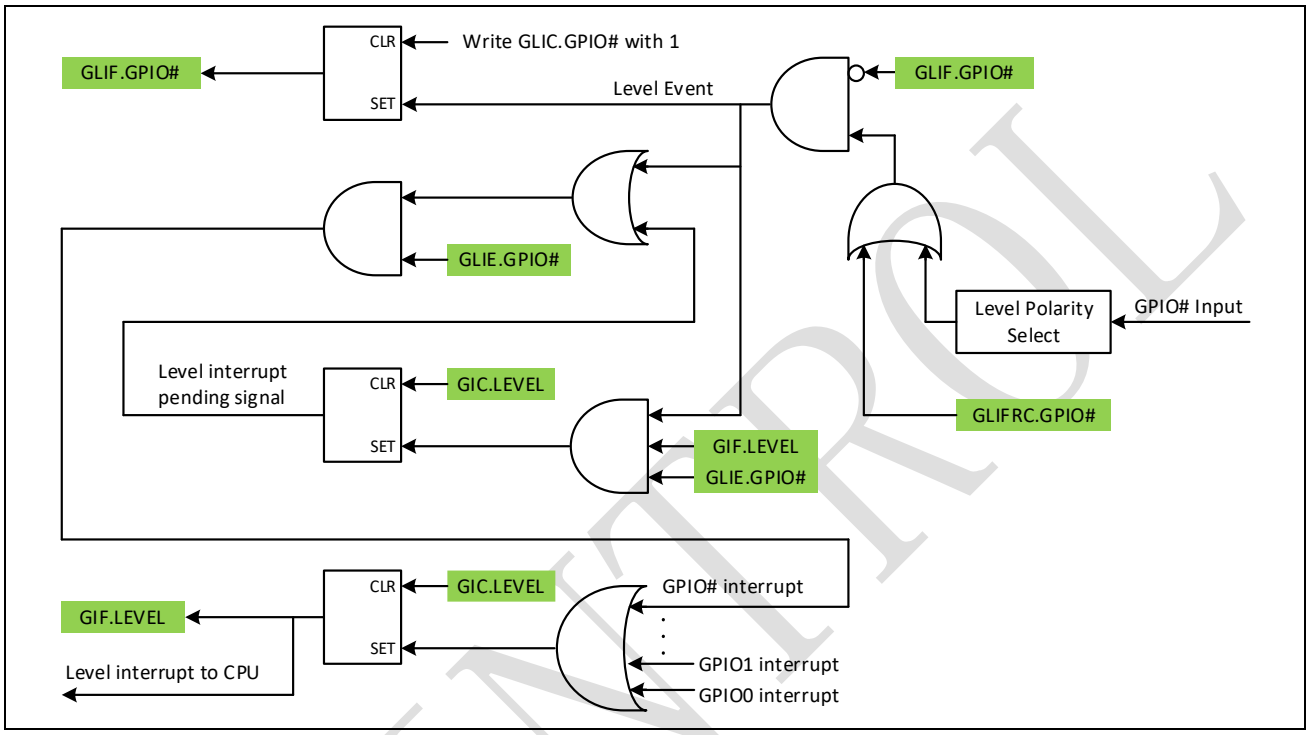
If GIF.EDGE is HIGH, another GPIO pin edge detect event has occurred and its associated bit in GEDR is LOW at present, this will cause edge detect interrupt pending issue. The pending GPIO edge detect interrupt will be released after GIF.EDGE is cleared, and program will enter ISR again for the pending interrupt request. However, if the GPIO pin has been edge-triggered and its associated bit in GEDR is HIGH at present, this pin's event will not trigger interrupt or interrupt pending issue to CPU. Therefore, the program should clear both GEDR and GIF.EDGE before exiting ISR.

4.3.2 GPIO level-triggered interrupt

GPIO level detect module will run when GPIO Level Interrupt Enable Register (GLIE0 / GLIE1) is enabled. Before setting GLIE, GPIO level detected polarity should be determined by programming GPIO Level Interrupt Polarity register (GLIPOL0 / GLIPOL1). Then GPIO level events will trigger level detect interrupt to CPU. The events are recorded by GPIO Level Interrupt Flag Register (GLIF0 / GLIF1)

and level detect interrupt flag in GPIO Interrupt Flag Register (GIF.LEVEL). Program can also write GPIO Level Interrupt Force Register (GLIFRC0 / GLIFRC1) to force level event. As shown in Figure 4-7, Writing 1 to GLIC0 or GLIC1 will clear GPIO level-triggered interrupt flag in GLIF0 or GLIF1 and writing 0 has no effect; Writing 1 to GIC.LEVEL will clear GIF.LEVEL. GPIO level-triggered interrupt pending action is similar with edge-triggered interrupt pending issue.

Figure 4-7: GPIO level-triggered interrupt



4.4 Registers

4.4.1 PINMUX register map

Table 4-2: PINMUX Module Base Address

Peripheral Module	Base Address
PINMUX	0x4000 0300

Table 4-3: PINMUX Register Map

Register	Offset	Description	Reset Value
GPIO0*	0x0	GPIO0 Pin Control Register	0x01000019
GPIO1*	0x4	GPIO1 Pin Control Register	0x01000019
GPIO2*	0x8	GPIO2 Pin Control Register	0x01000019
GPIO3*	0xC	GPIO3 Pin Control Register	0x01000019
GPIO4*	0x10	GPIO4 Pin Control Register	0x01000019
GPIO5*	0x14	GPIO5 Pin Control Register	0x01000019
GPIO6*	0x18	GPIO6 Pin Control Register	0x01000019
GPIO7*	0x1C	GPIO7 Pin Control Register	0x01000019
GPIO8*	0x20	GPIO8 Pin Control Register	0x01000019
GPIO9*	0x24	GPIO9 Pin Control Register	0x01000019
GPIO10*	0x28	GPIO10 Pin Control Register	0x01000019
GPIO11*	0x2C	GPIO11 Pin Control Register	0x01000019
GPIO12*	0x30	GPIO12 Pin Control Register	0x05000419
GPIO13*	0x34	GPIO13 Pin Control Register	0x05000419
GPIO14*	0x38	GPIO14 Pin Control Register	0x09000819
GPIO15*	0x3C	GPIO15 Pin Control Register	0x05000419
GPIO16*	0x40	GPIO16 Pin Control Register	0x08000818
GPIO17*	0x44	GPIO17 Pin Control Register	0x04000418
GPIO18*	0x48	GPIO18 Pin Control Register	0x00000018
GPIO19*	0x4C	GPIO19 Pin Control Register	0x00000018
GPIO20*	0x50	GPIO20 Pin Control Register	0x00000018
GPIO21*	0x54	GPIO21 Pin Control Register	0x00000018
GPIO22*	0x58	GPIO22 Pin Control Register	0x00000018
GPIO23*	0x5C	GPIO23 Pin Control Register	0x00000018

Register	Offset	Description	Reset Value
GPIO24*	0x60	GPIO24 Pin Control Register	0x00000018
GPIO25*	0x64	GPIO25 Pin Control Register	0x00000018
GPIO26*	0x68	GPIO26 Pin Control Register	0x00000018
GPIO27*	0x6C	GPIO27 Pin Control Register	0x00000018
GPIO28*	0x70	GPIO28 Pin Control Register	0x00000018
GPIO29*	0x74	GPIO29 Pin Control Register	0x00000018
GPIO30*	0x78	GPIO30 Pin Control Register	0x04000418
GPIO31*	0x7C	GPIO31 Pin Control Register	0x04000418
GPIO32*	0x80	GPIO32 Pin Control Register	0x00000018
GPIO33*	0x84	GPIO33 Pin Control Register	0x00000018
GPIO34*	0x88	GPIO34 Pin Control Register	0x0D000C18
GPIO35*	0x8C	GPIO35 Pin Control Register	0x0B000A18
GPIO36*	0x90	GPIO36 Pin Control Register	0x04000418
GPIO37*	0x94	GPIO37 Pin Control Register	0x02000018
GPIO38*	0x98	GPIO38 Pin Control Register	0x07000418
GPIO39*	0x9C	GPIO39 Pin Control Register	0x07000418
GPIO40*	0xA0	GPIO40 Pin Control Register	0x01000018
PINMUXREGKEY	0xA4	PINMUX Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the PINMUXREGKEY=0x1ACCE551.

4.4.2 PINMUX registers

Table 4-4: GPIO# Pin Control Register (GPIO#) Layout (# = 0 ~40)

GPIO# (GPIO# Pin Control Register) Offset: # * 4 Access: PINMUX -> GPIO#.all							
31	30	29	28	27	26	25	24
PU							
23	22	21	20	19	18	17	16
PD							
15	14	13	12	11	10	9	8
CHDFLTVAL							
7	6	5	4	3	2	1	0
STRENGTH		DEGLITCH	SMT	IE	MUXSEL		

Table 4-5: GPIO# Pin Control Register (GPIO#) Description (# = 0 ~40)

Bits	Field Name	Type	Reset	Description
31:24	PU	RW	-	Pull up enable
23:16	PD	RW	-	Pull down enable
15:8	CHDFLTVAL	RW	-	Default value for each demuxed input channel
7:6	STRENGTH	RW	-	Output driving strength 00: 5mA 01: 10mA 10: 15mA 11: 20mA
5	DEGLITCH	RW	-	Input deglitch 0: Disable input deglitch 1: Enable input deglitch
4	SMT	RW	-	Switch normal/schmitt input 0: Normal input 1: Schmitt input
3	IE	RW	-	Input enable 0: Input is always 0 1: Enable input
2:0	MUXSEL	RW	-	Channel select The channel definition for each GPIO is shown in Table 4-1

Table 4-6: PINMUX Register Write-Allow Key Register (PINMUXREGKEY) Layout

PINMUXREGKEY (PINMUX Register Write-Allow Key Register) Offset: 0xA4 Default: 0x1ACCE551 Access: PINMUX -> PINMUXREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 4-7: PINMUX Register Write-Allow Key Register (PINMUXREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected pinmux registers

4.4.3 GPIO register map

Table 4-8: GPIO Module Base Address

Peripheral Module	Base Address
GPIO	0x4000 3000

Table 4-9: GPIO Register Map

Register	Offset	Description	Reset Value
GPLR0*	0x0	GPIO Pin Level Register 0	0x00000000
GPLR1*	0x4	GPIO Pin Level Register 1	0x000001CC
GPDR0*	0xC	GPIO Pin Direction Register 0	0x00000000
GPDR1*	0x10	GPIO Pin Direction Register 1	0x00000000
GSLR0*	0x18	GPIO Pin Output Set Register 0	0x00000000
GSLR1*	0x1C	GPIO Pin Output Set Register 1	0x00000000
GCLR0*	0x24	GPIO Pin Output Clear Register 0	0x00000000
GCLR1*	0x28	GPIO Pin Output Clear Register 1	0x00000000
GRER0*	0x30	GPIO Rising Edge Detect Enable Register 0	0x00000000
GRER1*	0x34	GPIO Rising Edge Detect Enable Register 1	0x00000000
GFER0*	0x3C	GPIO Falling Edge Detect Enable Register 0	0x00000000
GFER1*	0x40	GPIO Falling Edge Detect Enable Register 1	0x00000000
GEDR0	0x48	GPIO Edge Detect Status Register 0	0x00000000
GEDR1	0x4C	GPIO Edge Detect Status Register 1	0x00000000
GSDR0*	0x54	GPIO Pin Bitwise Set Direction Register 0	0x00000000
GSDR1*	0x58	GPIO Pin Bitwise Set Direction Register 1	0x00000000
GCDR0*	0x60	GPIO Pin Bitwise Clear Direction Register 0	0x00000000
GCDR1*	0x64	GPIO Pin Bitwise Clear Direction Register 1	0x00000000
GSRER0*	0x6C	GPIO Bitwise Set Rising Edge Detect Enable Register 0	0x00000000
GSRER1*	0x70	GPIO Bitwise Set Rising Edge Detect Enable Register 1	0x00000000

Register	Offset	Description	Reset Value
GCRER0*	0x78	GPIO Bitwise Clear Rising Edge Detect Enable Register 0	0x00000000
GCRER1*	0x7C	GPIO Bitwise Clear Rising Edge Detect Enable Register 1	0x00000000
GSFER0*	0x84	GPIO Bitwise Set Falling Edge Detect Enable Register 0	0x00000000
GSFER1*	0x88	GPIO Bitwise Set Falling Edge Detect Enable Register 1	0x00000000
GCFER0*	0x90	GPIO Bitwise Clear Falling Edge Detect Enable Register 0	0x00000000
GCFER1*	0x94	GPIO Bitwise Clear Falling Edge Detect Enable Register 1	0x00000000
GEIE0*	0x9C	GPIO Edge-Triggered Interrupt Enable Register 0	0x00000000
GEIE1*	0xA0	GPIO Edge-Triggered Interrupt Enable Register 1	0x00000000
GLIF0*	0xA8	GPIO Level-Triggered Interrupt Flag Register 0	0x00000000
GLIF1*	0xAC	GPIO Level-Triggered Interrupt Flag Register 1	0x00000000
GLIE0*	0xB4	GPIO Level-Triggered Interrupt Enable Register 0	0x00000000
GLIE1*	0xB8	GPIO Level-Triggered Interrupt Enable Register 1	0x00000000
GLIC0	0xC0	GPIO Level-Triggered Interrupt Clear Register 0	0x00000000
GLIC1	0xC4	GPIO Level-Triggered Interrupt Clear Register 1	0x00000000
GLIFRC0*	0xCC	GPIO Level-Triggered Interrupt Force Register 0	0x00000000
GLIFRC1*	0xD0	GPIO Level-Triggered Interrupt Force Register 1	0x00000000
GLIPOL0*	0xD8	GPIO Level-Triggered Interrupt Polarity Register 0	0x00000000
GLIPOL1*	0xDC	GPIO Level-Triggered Interrupt Polarity Register 1	0x00000000
GIF	0xE4	GPIO Interrupt Flag Register	0x00000000
GIC	0xE8	GPIO Interrupt Flag Clear Register	0x00000000
GPIOREGKEY	0xEC	GPIO Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the GPIOREGKEY=0x1ACCE551.

4.4.4 GPIO registers

Table 4-10: GPIO Pin Level Register 0 (GPLR0) Layout

GPLR0 (GPIO Pin Level Register 0) Offset: 0x0 Default: 0x00000000 Access: GPIO -> GPLR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-11: GPIO Pin Level Register 0 (GPLR0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	GPIOn pin level readback (n = 0~31) 0: Pin level is low 1: Pin level is high

Table 4-12: GPIO Pin Level Register 1 (GPLR1) Layout

GPLR1 (GPIO Pin Level Register 1) Offset: 0x4 Default: 0x000001CC Access: GPIO -> GPLR1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-13: GPIO Pin Level Register 1 (GPLR1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	RW	0x1CC	GPIOn pin level readback (n = 32~40) 0: Pin level is low 1: Pin level is high

Table 4-14: GPIO Pin Direction Register 0 (GPDR0) Layout

GPDR0 (GPIO Pin Direction Register 0) Offset: 0xC Default: 0x00000000 Access: GPIO -> GPDR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-15: GPIO Pin Direction Register 0 (GPDR0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Set GPIOn direction (n = 0~31) 0: Set GPIOn as input 1: Set GPIOn as output

Table 4-16: GPIO Pin Direction Register 1 (GPDR1) Layout

GPDR1 (GPIO Pin Direction Register 1) Offset: 0x10 Default: 0x00000000 Access: GPIO -> GPDR1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-17: GPIO Pin Direction Register 1 (GPDR1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	RW	0x0	Set GPIOn direction (n = 32~40) 0: Set GPIOn as input 1: Set GPIOn as output

Table 4-18: GPIO Pin Output Set Register 0 (GSLR0) Layout

GSLR0 (GPIO Pin Output Set Register 0) Offset: 0x18 Default: 0x00000000 Access: GPIO -> GSLR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-19: GPIO Pin Output Set Register 0 (GSLR0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	W1S	0x0	Set GPIOn output high (n = 0~31) 0: Unaffected 1: Set GPIOn output to high

Table 4-20: GPIO Pin Output Set Register 1 (GSLR1) Layout

GSLR1 (GPIO Pin Output Set Register 1) Offset: 0x1C Default: 0x00000000 Access: GPIO -> GSLR1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-21: GPIO Pin Output Set Register 1 (GSLR1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	W1S	0x0	Set GPIOn output value (n = 32~40) 0: Unaffected 1: Set GPIOn output to high

Table 4-22: GPIO Pin Output Clear Register 0 (GCLR0) Layout

GCLR0 (GPIO Pin Output Clear Register 0) Offset: 0x24 Default: 0x00000000 Access: GPIO -> GCLR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-23: GPIO Pin Output Clear Register 0 (GCLR0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	W1S	0x0	Set GPIOn output low (n = 0~31) 0: Unaffected 1: Set GPIOn output to low

Table 4-24: GPIO Pin Output Clear Register 1 (GCLR1) Layout

GCLR1 (GPIO Pin Output Clear Register 1) Offset: 0x28 Default: 0x00000000 Access: GPIO -> GCLR1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-25: GPIO Pin Output Clear Register 1 (GCLR1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	W1S	0x0	Set GPIOn output low (n = 32~40) 0: Unaffected 1: Set GPIOn output to low

Table 4-26: GPIO Rising Edge Detect Enable Register 0 (GRER0) Layout

GRER0 (GPIO Rising Edge Detect Enable Register 0) Offset: 0x30 Default: 0x00000000 Access: GPIO -> GRER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-27: GPIO Rising Edge Detect Enable Register 0 (GRER0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Enable GPIOn rising edge detect (n = 0~31) 0: Disable rising edge detection on GPIOn 1: Enable rising edge detection on GPIOn

Table 4-28: GPIO Rising Edge Detect Enable Register 1 (GRER1) Layout

GRER1 (GPIO Rising Edge Detect Enable Register 1) Offset: 0x34 Default: 0x00000000 Access: GPIO -> GRER1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-29: GPIO Rising Edge Detect Enable Register 1 (GRER1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	RW	0x0	Enable GPIOn rising edge detect (n = 32~40) 0: Disable rising edge detection on GPIOn 1: Enable rising edge detection on GPIOn

Table 4-30: GPIO Falling Edge Detect Enable Register 0 (GFER0) Layout

GFER0 (GPIO Falling Edge Detect Enable Register 0) Offset: 0x3C Default: 0x00000000 Access: GPIO -> GFER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-31: GPIO Falling Edge Detect Enable Register 0 (GFER0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Enable GPIOn falling edge detect (n = 0~31) 0: Disable falling edge detection on GPIOn 1: Enable falling edge detection on GPIOn

Table 4-32: GPIO Falling Edge Detect Enable Register 1 (GFER1) Layout

GFER1 (GPIO Falling Edge Detect Enable Register 1) Offset: 0x40 Default: 0x00000000 Access: GPIO -> GFER1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-33: GPIO Falling Edge Detect Enable Register 1 (GFER1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	RW	0x0	Enable GPIOn falling edge detect (n = 32~40) 0: Disable falling edge detection on GPIOn 1: Enable falling edge detection on GPIOn

Table 4-34: GPIO Edge Detect Status Register 0 (GEDR0) Layout

GEDR0 (GPIO Edge Detect Status Register 0) Offset: 0x48 Default: 0x00000000 Access: GPIO -> GEDR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-35: GPIO Edge Detect Status Register 0 (GEDR0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	W1C	0x0	GPIOn edge detect status (n = 0~31) 0: Read 0 indicates no detected rising or falling edge Write a 0 has no effect 1: Read 1 indicates edge was detected Write 1 will clear the latched status

Table 4-36: GPIO Edge Detect Status Register 1 (GEDR1) Layout

GEDR1 (GPIO Edge Detect Status Register 1) Offset: 0x4C Default: 0x00000000 Access: GPIO -> GEDR1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-37: GPIO Edge Detect Status Register 1 (GEDR1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	W1C	0x0	GPIOn edge detect status (n = 32~40) 0: Read 0 indicates no detected rising or falling edge Write a 0 has no effect 1: Read 1 indicates edge was detected Write 1 will clear the latched status

Table 4-38: GPIO Pin Bitwise Set Direction Register 0 (GSDR0) Layout

GSDR0 (GPIO Pin Bitwise Set Direction Register 0) Offset: 0x54 Default: 0x00000000 Access: GPIO -> GSDR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-39: GPIO Pin Bitwise Set Direction Register 0 (GSDR0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	Set GPDR register bit (n = 0~31) 0: Unaffected 1: Set GPDR bit and set GPIOn as output

Table 4-40: GPIO Pin Bitwise Set Direction Register 1 (GSDR1) Layout

GSDR1 (GPIO Pin Bitwise Set Direction Register 1) Offset: 0x58 Default: 0x00000000 Access: GPIO -> GSDR1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-41: GPIO Pin Bitwise Set Direction Register 1 (GSDR1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	WO	0x0	Set GPDR register bit (n = 32~40) 0: Unaffected 1: Set GPDR bit and set GPIOn as output

Table 4-42: GPIO Pin Bitwise Clear Direction Register 0 (GCDR0) Layout

GCDR0 (GPIO Pin Bitwise Clear Direction Register 0) Offset: 0x60 Default: 0x00000000 Access: GPIO -> GCDR0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-43: GPIO Pin Bitwise Clear Direction Register 0 (GCDR0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	Clear GPDR register bit (n = 0~31) 0: Unaffected 1: Clear GPDR bit and set GPIOn as input

Table 4-44: GPIO Pin Bitwise Clear Direction Register 1 (GCDR1) Layout

GCDR1 (GPIO Pin Bitwise Clear Direction Register 1) Offset: 0x64 Default: 0x00000000 Access: GPIO -> GCDR1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-45: GPIO Pin Bitwise Clear Direction Register 1 (GCDR1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	WO	0x0	Clear GPDR register bit (n = 32~40) 0: Unaffected 1: Clear GPDR bit and set GPIOn as input

Table 4-46: GPIO Bitwise Set Rising Edge Detect Enable Register 0 (GSRER0) Layout

GSRER0 (GPIO Bitwise Set Rising Edge Detect Enable Register 0) Offset: 0x6C Default: 0x00000000 Access: GPIO -> GSRER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-47: GPIO Bitwise Set Rising Edge Detect Enable Register 0 (GSRER0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	Set GRER register bit (n = 0~31) 0: Unaffected 1: Set GRER bit and enable rising edge detect for GPIOn

Table 4-48: GPIO Bitwise Set Rising Edge Detect Enable Register 1 (GSRER1) Layout

GSRER1 (GPIO Bitwise Set Rising Edge Detect Enable Register 1) Offset: 0x70 Default: 0x00000000 Access: GPIO -> GSRER1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-49: GPIO Bitwise Set Rising Edge Detect Enable Register 1 (GSRER1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	WO	0x0	Set GRER register bit (n = 32~40) 0: Unaffected 1: Set GRER bit and enable rising edge detect for GPIOn

Table 4-50: GPIO Bitwise Clear Rising Edge Detect Enable Register 0 (GCRER0) Layout

GCRER0 (GPIO Bitwise Clear Rising Edge Detect Enable Register 0) Offset: 0x78 Default: 0x00000000 Access: GPIO -> GCRER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-51: GPIO Bitwise Clear Rising Edge Detect Enable Register 0 (GCRER0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	Clear GRER register bit (n = 0~31) 0: Unaffected 1: Clear GRER bit and disable rising edge detect for GPIOn

Table 4-52: GPIO Bitwise Clear Rising Edge Detect Enable Register 1 (GCRER1) Layout

GCRER1 (GPIO Bitwise Clear Rising Edge Detect Enable Register 1) Offset: 0x7C Default: 0x00000000 Access: GPIO -> GCRER1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-53: GPIO Bitwise Clear Rising Edge Detect Enable Register 1 (GCRER1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	WO	0x0	Clear GRER register bit (n = 32~40) 0: Unaffected 1: Clear GRER bit and disable rising edge detect for GPIOn

Table 4-54: GPIO Bitwise Set Falling Edge Detect Enable Register 0 (GSFER0) Layout

GSFER0 (GPIO Bitwise Set Falling Edge Detect Enable Register 0) Offset: 0x84 Default: 0x00000000 Access: GPIO -> GSFER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-55: GPIO Bitwise Set Falling Edge Detect Enable Register 0 (GSFER0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	Set GFER register bit (n = 0~31) 0: Unaffected 1: Set GFER bit and enable falling edge detect for GPIOn

Table 4-56: GPIO Bitwise Set Falling Edge Detect Enable Register 1 (GSFER1) Layout

GSFER1 (GPIO Bitwise Set Falling Edge Detect Enable Register 1) Offset: 0x88 Default: 0x00000000 Access: GPIO -> GSFER1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-57: GPIO Bitwise Set Falling Edge Detect Enable Register 1 (GSFER1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	WO	0x0	Set GFER register bit (n = 32~40) 0: Unaffected 1: Set GFER bit and enable falling edge detect for GPIOn

Table 4-58: GPIO Bitwise Clear Falling Edge Detect Enable Register 0 (GCFER0) Layout

GCFER0 (GPIO Bitwise Clear Falling Edge Detect Enable Register 0) Offset: 0x90 Default: 0x00000000 Access: GPIO -> GCFER0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-59: GPIO Bitwise Clear Falling Edge Detect Enable Register 0 (GCFER0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	Clear GFER register bit (n = 0~31) 0: Unaffected 1: Clear GFER bit and disable falling edge detect for GPIOn

Table 4-60: GPIO Bitwise Clear Falling Edge Detect Enable Register 1 (GCFER1) Layout

GCFER1 (GPIO Bitwise Clear Falling Edge Detect Enable Register 1) Offset: 0x94 Default: 0x00000000 Access: GPIO -> GCFER1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-61: GPIO Bitwise Clear Falling Edge Detect Enable Register 1 (GCFER1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	WO	0x0	Clear GFER register bit (n = 32~40) 0: Unaffected 1: Clear GFER bit and disable falling edge detect for GPIOn

Table 4-62: GPIO Edge-Triggered Interrupt Enable Register 0 (GEIE0) Layout

GEIE0 (GPIO Edge-Triggered Interrupt Enable Register 0) Offset: 0x9C Default: 0x00000000 Access: GPIO -> GEIE0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-63: GPIO Edge-Triggered Interrupt Enable Register 0 (GEIE0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	GPIOn edge-triggered interrupt enable (n = 0~31) 0: Disable GPIOn edge-triggered interrupt 1: Enable GPIOn edge-triggered interrupt

Table 4-64: GPIO Edge-Triggered Interrupt Enable Register 1 (GEIE1) Layout

GEIE1 (GPIO Edge-Triggered Interrupt Enable Register 1) Offset: 0xA0 Default: 0x00000000 Access: GPIO -> GEIE1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-65: GPIO Edge-Triggered Interrupt Enable Register 1 (GEIE1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	RW	0x0	GPIOn edge-triggered interrupt enable (n = 32~40) 0: Disable GPIOn edge-triggered interrupt 1: Enable GPIOn edge-triggered interrupt

Table 4-66: GPIO Level-Triggered Interrupt Flag Register 0 (GLIF0) Layout

GLIF0 (GPIO Level-Triggered Interrupt Flag Register 0) Offset: 0xA8 Default: 0x00000000 Access: GPIO -> GLIF0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-67: GPIO Level-Triggered Interrupt Flag Register 0 (GLIF0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	GPIOn level-triggered interrupt flag (n = 0~31) 0: GPIOn level-triggered interrupt not occur 1: GPIOn level-triggered interrupt occur

Table 4-68: GPIO Level-Triggered Interrupt Flag Register 1 (GLIF1) Layout

GLIF1 (GPIO Level-Triggered Interrupt Flag Register 1) Offset: 0xAC Default: 0x00000000 Access: GPIO -> GLIF1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-69: GPIO Level-Triggered Interrupt Flag Register 1 (GLIF1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	RO	0x0	GPIOn level-triggered interrupt flag (n = 32~40) 0: GPIOn level-triggered interrupt not occur 1: GPIOn level-triggered interrupt occur

Table 4-70: GPIO Level-Triggered Interrupt Enable Register 0 (GLIE0) Layout

GLIE0 (GPIO Level-Triggered Interrupt Enable Register 0) Offset: 0xB4 Default: 0x00000000 Access: GPIO -> GLIE0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-71: GPIO Level-Triggered Interrupt Enable Register 0 (GLIE0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	GPIOn level-triggered interrupt enable (n = 0~31) 0: Disable GPIOn level-triggered interrupt 1: Enable GPIOn level-triggered interrupt

Table 4-72: GPIO Level-Triggered Interrupt Enable Register 1 (GLIE1) Layout

GLIE1 (GPIO Level-Triggered Interrupt Enable Register 1) Offset: 0xB8 Default: 0x00000000 Access: GPIO -> GLIE1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-73: GPIO Level-Triggered Interrupt Enable Register 1 (GLIE1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	RW	0x0	GPIOn level-triggered interrupt enable (n = 32~40) 0: Disable GPIOn level-triggered interrupt 1: Enable GPIOn level-triggered interrupt

Table 4-74: GPIO Level-Triggered Interrupt Clear Register 0 (GLIC0) Layout

GLIC0 (GPIO Level-Triggered Interrupt Clear Register 0) Offset: 0xC0 Default: 0x00000000 Access: GPIO -> GLIC0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-75: GPIO Level-Triggered Interrupt Clear Register 0 (GLIC0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	W1C	0x0	GPIOn level-triggered interrupt flag clear (n = 0~31) Write 1 will clear the level-triggered interrupt flag and write a 0 has no effect

Table 4-76: GPIO Level-Triggered Interrupt Clear Register 1 (GLIC1) Layout

GLIC1 (GPIO Level-Triggered Interrupt Clear Register 1) Offset: 0xC4 Default: 0x00000000 Access: GPIO -> GLIC1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-77: GPIO Level-Triggered Interrupt Clear Register 1 (GLIC1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	W1C	0x0	GPIOn level-triggered interrupt flag clear (n = 32~40) Write 1 will clear the level-triggered interrupt flag and write a 0 has no effect

Table 4-78: GPIO Level-Triggered Interrupt Force Register 0 (GLIFRC0) Layout

GLIFRC0 (GPIO Level-Triggered Interrupt Force Register 0) Offset: 0xCC Default: 0x00000000 Access: GPIO -> GLIFRC0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-79: GPIO Level-Triggered Interrupt Force Register 0 (GLIFRC0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	W1S	0x0	GPIOn level-triggered interrupt force (n = 0~31) Write 1 will force to generate the GPIOn level-triggered interrupt and write a 0 has no effect

Table 4-80: GPIO Level-Triggered Interrupt Force Register 1 (GLIFRC1) Layout

GLIFRC1 (GPIO Level-Triggered Interrupt Force Register 1) Offset: 0xD0 Default: 0x00000000 Access: GPIO -> GLIFRC1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-81: GPIO Level-Triggered Interrupt Force Register 1 (GLIFRC1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	W1S	0x0	GPIOn level-triggered interrupt force (n = 32~40) Write 1 will force to generate the GPIOn level-triggered interrupt and write a 0 has no effect

Table 4-82: GPIO Level-Triggered Interrupt Polarity Register 0 (GLIPOL0) Layout

GLIPOL0 (GPIO Level-Triggered Interrupt Polarity Register 0) Offset: 0xD8 Default: 0x00000000 Access: GPIO -> GLIPOL0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 4-83: GPIO Level-Triggered Interrupt Polarity Register 0 (GLIPOL0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	GPIOn level-triggered interrupt polarity (n = 0~31) 0: Trigger interrupt when GPIOn is low 1: Trigger interrupt when GPIOn is high

Table 4-84: GPIO Level-Triggered Interrupt Polarity Register 1 (GLIPOL1) Layout

GLIPOL1 (GPIO Level-Triggered Interrupt Polarity Register 1) Offset: 0xDC Default: 0x00000000 Access: GPIO -> GLIPOL1.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							VAL
7	6	5	4	3	2	1	0
VAL							

Table 4-85: GPIO Level-Triggered Interrupt Polarity Register 1 (GLIPOL1) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:0	VAL	RW	0x0	GPIOn level-triggered interrupt polarity (n = 32~40) 0: Trigger interrupt when GPIOn is low 1: Trigger interrupt when GPIOn is high

Table 4-86: GPIO Interrupt Flag Register (GIF) Layout

GIF (GPIO Interrupt Flag Register) Offset: 0xE4 Default: 0x00000000 Access: GPIO -> GIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						LEVEL	EDGE

Table 4-87: GPIO Interrupt Flag Register (GIF) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	LEVEL	RO	0x0	GPIO level-triggered interrupt flag 0: Not Occur 1: Occur
0	EDGE	RO	0x0	GPIO edge-triggered interrupt flag 0: Not Occur 1: Occur

Table 4-88: GPIO Interrupt Flag Clear Register (GIC) Layout

GIC (GPIO Interrupt Flag Clear Register) Offset: 0xE8 Default: 0x00000000 Access: GPIO -> GIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						LEVEL	EDGE

Table 4-89: GPIO Interrupt Flag Clear Register (GIC) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	LEVEL	W1C	0x0	GPIO level-triggered interrupt flag clear 0: Write 0 no effect 1: Clear the interrupt flag
0	EDGE	W1C	0x0	GPIO edge-triggered interrupt flag clear 0: Write 0 no effect 1: Clear the interrupt flag

Table 4-90: GPIO Register Write-Allow Key Register (GPIOREGKEY) Layout

GPIOREGKEY (GPIO Register Write-Allow Key Register) Offset: 0xEC Default: 0x1ACCE551 Access: GPIO -> GPIOREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 4-91: GPIO Register Write-Allow Key Register (GPIOREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected GPIO registers

5 Interrupts and events

5.1 Nested vectored interrupt controller (NVIC)

The Nested Vectored Interrupt Controller (NVIC) is an integral part of ARM Cortex-M4 processor unit. Due to a tight coupling with the CPU, it provides the lowest interrupt latency and efficient processing of late arriving interrupts.

5.1.1 Features

The NVIC supports the following features:

- Up to 51 vectored interrupts (not including the sixteen Cortex-M4 interrupts)
- 16 programmable interrupt priority levels for each interrupt (4 bits of interrupt priority are used). A higher level corresponds to a lower priority, so level 0 is the highest interrupt priority
- Controls system exceptions and peripheral interrupts
- Support for interrupt tail-chaining and late-arrival
- Support for interrupt lazy-stacking
- Support for NMI

5.1.2 Interrupt and exception vectors

When an exception event takes place and is being handled by the Cortex-M4, the processor will need to locate the starting address of the exception handler. Therefore, a vector table mechanism is used. The vector table is an array of word data inside the system memory, each representing the starting address of one exception. The vector table location is controlled by a programmable register in the NVIC called the VTOR (address 0xE000ED08). After reset, the VTOR is reset to 0; therefore, the vector table is located at address 0x0 after reset. The vector address is arranged according to the exception number times four. Table 5-1 is the vector table for SPD1148.

Table 5-1: Vector table for SPD1148

Vector#	IRQ#	Acronym	Priority	Description	Address Offset
-	-	MSP	-	Initial main stack pointer	0x0000_0000
1	-	Reset	-3, the highest	Reset vector	0x0000_0004
2	-14	NMI	-2	Non mask-able interrupt. The watchdog timer 0 is linked to the NMI vector.	0x0000_0008
3	-13	Hard Fault	-1	Hard fault	0x0000_000C
4	-12	MemManage	Configurable	Memory management	0x0000_00010
5	-11	Bus Fault	Configurable	Bus fault	0x0000_0014
6	-10	Usage Fault	Configurable	Usage fault	0x0000_0018

Vector#	IRQ#	Acronym	Priority	Description	Address Offset
7-10	-	-	-	Reserved	0x0000_001C – 0x0000_002B
11	-5	SVCall	Configurable	Supervisor call	0x0000_002C
12	-4	Debug Monitor	Configurable	Debug monitor	0x0000_0030
13	-	-	-	Reserved	0x0000_0034
14	-2	PendSV	Configurable	Pendable service call	0x0000_0038
15	-1	SysTick	Configurable	SysTick timer interrupt	0x0000_003C
16	0	MEM	Configurable	Memory error interrupt	0x0000_0040
17	1	POWER	Configurable	Power error interrupt (BOD)	0x0000_0044
18	2	CLOCK	Configurable	Clock error interrupt	0x0000_0048
19	3	WDT1	Configurable	Watchdog timer 1 interrupt	0x0000_004C
20	4	GPIO_LEVEL	Configurable	GPIO level-trigger interrupt	0x0000_0050
21	5	GPIO_EDGE	Configurable	GPIO edge-trigger interrupt	0x0000_0054
22	6	SIO0A	Configurable	SIO0 interrupt A	0x0000_0058
23	7	SIO0B	Configurable	SIO0 interrupt B	0x0000_005C
24	8	-	-	Reserved	
25	9	-	-	Reserved	
26	10	-	-	Reserved	
27	11	-	-	Reserved	
28	12	UART	Configurable	UART interrupt	0x0000_0070
29	13	SSP	Configurable	SSP interrupt	0x0000_0074
30	14	I2C	Configurable	I2C interrupt	0x0000_0078
31	15	ADC0	Configurable	ADC SOC0 interrupt	0x0000_007C
32	16	ADC1	Configurable	ADC SOC1 interrupt	0x0000_0080
33	17	ADC2	Configurable	ADC SOC2 interrupt	0x0000_0084
34	18	ADC3	Configurable	ADC SOC3 interrupt	0x0000_0088
35	19	ADC4	Configurable	ADC SOC4 interrupt	0x0000_008C
36	20	ADC5	Configurable	ADC SOC5 interrupt	0x0000_0090
37	21	ADC6	Configurable	ADC SOC6 interrupt	0x0000_0094
38	22	ADC7	Configurable	ADC SOC7 interrupt	0x0000_0098
39	23	ADC8	Configurable	ADC SOC8 interrupt	0x0000_009C
40	24	ADC9	Configurable	ADC SOC9 interrupt	0x0000_00A0
41	25	ADC10	Configurable	ADC SOC10 interrupt	0x0000_00A4
42	26	ADC11	Configurable	ADC SOC11 interrupt	0x0000_00A8

Vector#	IRQ#	Acronym	Priority	Description	Address Offset
43	27	ADC12	Configurable	ADC SOC12 interrupt	0x0000_00AC
44	28	ADC13	Configurable	ADC SOC13 interrupt	0x0000_00B0
45	29	ADC14	Configurable	ADC SOC14 interrupt	0x0000_00B4
46	30	ADC15	Configurable	ADC SOC15 interrupt	0x0000_00B8
47	31	ADCPPU0	Configurable	ADC post-process unit 0 interrupt	0x0000_00BC
48	32	ADCPPU1	Configurable	ADC post-process unit 1 interrupt	0x0000_00C0
49	33	ADCPPU2	Configurable	ADC post-process unit 2 interrupt	0x0000_00C4
50	34	ADCPPU3	Configurable	ADC post-process unit 3 interrupt	0x0000_00C8
51	35	ADCPPU4	Configurable	ADC post-process unit 4 interrupt	0x0000_00CC
52	36	ADCPPU5	Configurable	ADC post-process unit 5 interrupt	0x0000_00D0
53	-	-	-	Reserved	0x0000_00D4
54	-	-	-	Reserved	0x0000_00D8
55	39	PWM0	Configurable	PWM0 interrupt	0x0000_00DC
56	40	PWM1	Configurable	PWM1 interrupt	0x0000_00E0
57	41	PWM2	Configurable	PWM2 interrupt	0x0000_00E4
58	42	PWM3	Configurable	PWM3 interrupt	0x0000_00E8
59	43	PWM4	Configurable	PWM4 interrupt	0x0000_00EC
60	44	PWM5	Configurable	PWM5 interrupt	0x0000_00F0
61	-	-	-	Reserved	0x0000_00F4
62	-	-	-	Reserved	0x0000_00F8
63	47	PWM0TZ	Configurable	PWM0 trip-zone interrupt	0x0000_00FC
64	48	PWM1TZ	Configurable	PWM1 trip-zone interrupt	0x0000_0100
65	49	PWM2TZ	Configurable	PWM2 trip-zone interrupt	0x0000_0104
66	50	PWM3TZ	Configurable	PWM3 trip-zone interrupt	0x0000_0108
67	51	PWM4TZ	Configurable	PWM4 trip-zone interrupt	0x0000_010C
68	52	PWM5TZ	Configurable	PWM5 trip-zone interrupt	0x0000_0110
69	-	-	-	Reserved	0x0000_0114
70	-	-	-	Reserved	0x0000_0118
71	55	ECAP	Configurable	ECAP interrupt	0x0000_011C

Vector#	IRQ#	Acronym	Priority	Description	Address Offset
72	56	TIMER0	Configurable	General purpose timer 0 interrupt	0x0000_0120
73	57	TIMER1	Configurable	General purpose timer 1 interrupt	0x0000_0124
74	58	TIMER2	Configurable	General purpose timer 2 interrupt	0x0000_0128
75	59	CRC	Configurable	CRC interrupt	0x0000_012C
76	60	AES	Configurable	AES interrupt	0x0000_0130

5.1.3 NVIC interrupt configuration

NVIC interrupts can be enabled by writing to the SETENA register or disabled by writing to the CLRENA register. In this way, enabling or disabling an interrupt will not affect other interrupt enable states. The SETENA/CLRENA registers are 32 bits wide; each bit represents one interrupt input.

The NVIC interrupt-pending status can be accessed through the Interrupt Set Pending (SETPEND) and Interrupt Clear Pending (CLRPEND) registers. You can cancel a current pended exception through the CLRPEND register, or generate software interrupts through the SETPEND register.

Each external interrupt has an associated priority-level register, which has a maximum width of 8 bits and a minimum width of 3 bits. Each register can be further divided into preempt priority level and sub-priority level based on priority group settings. The SPD1148 implements 4 bits of priority level, so the chip supports 16 levels of programmable priority.

Please see ARM released documents for more information of NVIC configurations.

5.2 SysTick Timer

The Cortex-M4 processor includes an integrated system timer, called SysTick. The SYSTICK timer is a 24-bit down counter. Once it reaches zero, the counter loads the reload value from the RELOAD register. It does not stop until the enable bit in the SYSTICK Control and Status register is cleared.

When the SYSTICK timer changes from 1 to 0, it will set the COUNTFLAG bit in the SYSTICK

Control and Status register. The COUNTFLAG can be cleared by one of the following:

- Read of the SYSTICK Control and Status register by the processor
- Clear of the SYSTICK counter value by writing any value to the SYSTICK Current Value register

The SYSTICK counter can be used to generate SYSTICK exceptions at regular intervals. This is often necessary for the OS, for task and resources management. To enable SYSTICK exception generation, the TICKINT bit should be set.

Please see ARM released documents for more information of SysTick Timer configurations.

6 AES

6.1 AES overview

The SPD1148 implements an AES (Advanced Encryption Standard) engine to provide encryption and decryption services. The AES engine supports ECB, CBC, CTR, CCM* and MMO mode. The key length can be as many as 256 bits. The main features of the AES engine are:

- Supports as many as six block cipher modes: ECB, CBC, CTR, CCM*, MMO and Bypass
- 128, 192 and 256-bit key size
- Supports partial code
- Data IO by register interface
- Interrupt on completed AES operation (Output FIFO is empty and Input FIFO is full)
- Error indications for each block cipher mode
- Separate 4*32-bit input and output FIFOs
- Special feature for security mode
  - CTR: Supports counter modular from 16 to 128
  - CCM*: Supports 0 to (2^{32-1}) bytes associated string and message string
  - Supports 11-13 bytes nonce and supports L(15 – length of nonce) from 2 to 4

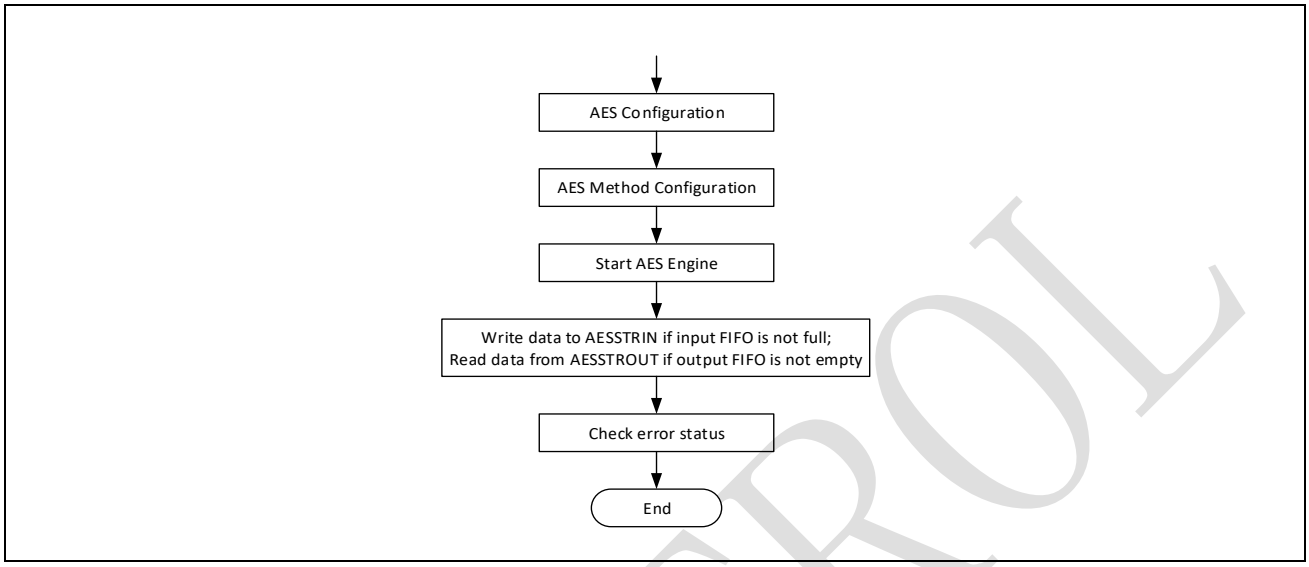
6.2 Functional description

The AES module implements ECB, CBC, CTR, CCM*, MMO and Bypass block cipher modes by efficient hardware.

6.2.1 AES operation flow

See Figure 6-1 for the AES operational flow.

Figure 6-1: AES operation flow diagram



AES operation pseudo code

```
AES_Config_Type aesConfig

aesConfig.mode <- AES_MODE_CBC
aesConfig.encDecSel <- AES_MODE_ENCRYPTION
aesConfig.keySize <- keysize
aesConfig.mStrLen <- length

for i=1 to keysize do
aesConfig.key[i] = key[i]

for i=1 to 4 do
aesConfig.initVect[i] = vector[i]

while j<length or k<length do
if AES input fifo not full do
feed the data plain_text[j]
j++

if AES output fifo not empty do
read the encryption data
k++
```

6.2.2 AES configuration

Ensure correct configuration before starting the AES engine by following these steps:

- Set AES engine to encrypt or decrypt by clearing/setting AESCTL0.DECRYPTEN
- Configure AES block cipher mode by setting AESCTL0.MODE, 0 for ECB mode, 1 for CBC mode, 2 for CTR mode, 5 for CCM* mode, 6 for MMO mode and 7 for BYPASS mode
- Configure AES key size. AES engine supports three types of key size: 128-, 192-, and 256- bit. Configure AES key size parameter by setting AESCTL0.KEYSIZE.
- Fill the key according to key size. AES engine contains eight 32-bit key registers defined as AESKEY0~7. When key size is set to 128-bit, then AESKEY7/6/5/4 is used. When the key size is set to 192-bit, then AESKEY7/6/5/4/3/2 is used. When key size is set to 192-bit, then AESKEY7/6/5/4/3/2 is used. When key size is set to 256-bit, then AESKEY7/6/5/4/3/2/1/0 is used. However, MMO does not support 192- and 256- bit key size, and key size is ignored in Bypass mode.
- For all modes except CCM* mode, set input data size by setting AESMSTRLEN. For CCM* mode, set associate data size by setting AESASTRLEN, set message data size by setting AESMSTRLEN.
- If AES block cipher mode is CTR mode, set CTR mode's counter modular by setting AESCTL0.CTRMOD.
- For CCM* encryption or MMO mode, if MIC/HASH is needed, set AESCTL0.OUTMIC bit to 1 to append MIC/HASH at the end of output stream. If only MIC/HASH is needed, we can block the encrypted data into output FIFO (set AESCTL0.OUTMSG bit to 1), and get MIC/HASH from AESOV3/2/1/0.
- For CCM* encryption mode, set AESCTL0.OUTHDR bit to 1 to output B0 at the beginning of the output stream if it is necessary.
- Fill with initial value according to AES block cipher mode. AES engine contains four 32-bit initial vector registers naming by AESIV0/1/2/3.
- ECB/MMO/BYPASS mode: No initial vectors needed to be configured.
- CTR mode:
  - Set AESIV0 = initial counter
  - Set AESIV1 = Nonce[31:0]
  - Set AESIV2 = Nonce[63:32]
  - Set AESIV3 = Nonce[95:64]
- For CCM* mode
  - Set AESIV0 = Nonce[31:0]
  - Set AESIV1 = Nonce[63:32]
  - Set AESIV2 = Nonce[95:64]
  - Set AESIV3[7:0] = Nonce[103:96]
  - Set AESIV3[15:8] = [15-(Length of Nonce Bytes)]

Note: For Bypass mode, the AES engine ignores the input data; it passes it along unchanged to the output.

6.2.3 Data access method

The AES module contains separate 4*32-bit input and output FIFO. The input and output FIFO can be accessed by CPU.

The CPU writes data into AESSTRIN if the input FIFO is not full; read data from AESSTROUT if output FIFO is not empty. The AES operation finishes as the transfer data size reaches input and output data size.

6.2.4 Starting the AES engine

Clear AES input and output FIFO and reset AES core before starting the AES engine. The AES input and output FIFO can be cleared by setting the AESCTL0.IFIFOCLR bit and AESCTL0.OFIFOCLR bit to 1. The AES core can be reset by setting and then clearing AESCTL1.RST.

6.2.5 Interrupt request

There are three interrupts for the AES engine: input FIFO full interrupt, output FIFO empty interrupt and AES operation done interrupt. Each interrupt can be enabled or cleared by setting AESIE/AESIC registers.

6.2.6 Partial code support

The AES engine can automatically pad the input data when the input data length is not multiple of 128 bits. The AES module supports the following padding scheme for different AES block cipher modes as shown in Table 6-1.

Table 6-1: Padding scheme

Mode	Description
CCM*	Automatically padding 0 for both A string and M string
MMO	Automatically padding “100…00”+2 bytes length information
CBC	Cipher stealing is performed to partial code word, which is proprietary, not according to NIST standard specification
CTR	Partial code word don’t affect the operation
ECB	Check partial case, assert error when partial cases detected

6.2.7 Error status check

Register AESSTS.ERRCODE records the error status for the AES engine when the AES operation has finished. Table 6-2 shows the error code for different AES block cipher modes.

Table 6-2: Error code for different AES block cipher modes

Mode	ERRCODE[2]	ERRCODE[1]	ERRCODE [0]
ECB	N/A	Data is not multiple of 16 bytes	Input data size less than 16 bytes
CBC	N/A	N/A	
CTR	N/A	N/A	
CCM*	MIC mismatch during decryption	N/A	N/A
MMO	N/A	Data is more than 2^{13}-1 bytes	N/A
Bypass	N/A	N/A	N/A

6.2.8 Output vector

The output vector provides some useful information, such as the last cipher block in CBC mode, last counter in CTR mode, MIC value in CCM* mode and HASH value in MMO mode. Register AESOV3/2/1/0 records useful information for different AES block cipher modes. Table 6-3 shows the recorded information in AES output vector for different AES block cipher mode.

Table 6-3: AES output vector

Block Cipher Mode	Output Vector
ECB	N/A
CBC	Last cipher block AESOV0 = cipher[31:0] AESOV1 = cipher[63:32] AESOV2 = cipher[95:64] AESOV3 = cipher[127:96]
CTR	Last counter AESOV0 = counter[31:0] AESOV1 = counter[63:32] AESOV2 = counter[95:64] AESOV3 = counter[127:96]
CCM*	Encryption: MIC value. If MIC is less than 32 byte, always MSB byte is used. Example: 8 byte MIC AESOV2 = MIC[31:0] AESOV3 = MIC[63:32]
MMO	HASH value AESOV0 = HASH[31:0] AESOV1 = HASH[63:32] AESOV2 = HASH[95:64] AESOV3 = HASH[127:96]
Bypass	N/A

6.3 Registers

6.3.1 AES register map

Table 6-4: AES Module Base Address

Peripheral Module	Base Address
AES	0x4000 8400

Table 6-5: AES Register Map

Register	Offset	Description	Reset Value
AESCTL0	0x0	AES Control Register 0	0x00004010
AESSTS	0x8	AES Status Register	0x00000081
AESASTRLEN	0xC	AES Associate String Length Register	0x00000000
AESMSTRLEN	0x10	AES Message String Length Register	0x00000000
AESSTRIN	0x14	AES Input Message Word Register	0x00000000
AESIV0	0x18	AES Initial Vector Register 0	0x00000000
AESIV1	0x1C	AES Initial Vector Register 1	0x00000000
AESIV2	0x20	AES Initial Vector Register 2	0x00000000
AESIV3	0x24	AES Initial Vector Register 3	0x00000000
AESKEY0	0x28	AES Key Register 0	0x00000000
AESKEY1	0x2C	AES Key Register 1	0x00000000
AESKEY2	0x30	AES Key Register 2	0x00000000
AESKEY3	0x34	AES Key Register 3	0x00000000
AESKEY4	0x38	AES Key Register 4	0x00000000
AESKEY5	0x3C	AES Key Register 5	0x00000000
AESKEY6	0x40	AES Key Register 6	0x00000000
AESKEY7	0x44	AES Key Register 7	0x00000000
AESSTROUT	0x48	AES Output Message Word Register	0x00000000
AESOV0	0x4C	AES Output Vector Register 0	0x00000000
AESOV1	0x50	AES Output Vector Register 1	0x00000000
AESOV2	0x54	AES Output Vector Register 2	0x00000000
AESOV3	0x58	AES Output Vector Register 3	0x00000000
AESIF	0x5C	AES Interrupt Status Register	0x00000000
AESIE	0x60	AES Interrupt Enable Register	0x00000000
AESRAWIF	0x64	AES Interrupt Raw Status Register	0x00000000
AESIC	0x68	AES Interrupt Clear Register	0x00000000

6.3.2 AES registers

Table 6-6: AES Control Register 0 (AESCTL0) Layout

AESCTL0 (AES Control Register 0) Offset: 0x0 Default: 0x00004010
Access: AES -> AESCTL0.all

31	30	29	28	27	26	25	24
RESERVED_31_26						CTRMOD	
23	22	21	20	19	18	17	16
CTRMOD					MODE		
15	14	13	12	11	10	9	8
DECRYPTEN	OUTMIC	MICLEN		KEYSIZE		RESERVED_9	RESERVED_8
7	6	5	4	3	2	1	0
RESERVED_7_6		OUTHDR	OUTMSG	OFIFOCLR	IFIFOCLR	RESERVED_1	START

Table 6-7: AES Control Register 0 (AESCTL0) Description

Bits	Field Name	Type	Reset	Description
31:26	RESERVED_31_26	RO	0x0	Reserved.
25:19	CTRMOD	RW	0x0	CTR mode's counter modular modular=2^128:[7'h0-7'hF] modular=2^ctr_mod: others
18:16	MODE	RW	0x0	AES running mode 000: ECB 001: CBC 010: CTR 011: RESERVED 100: RESERVED 101: CCM* 110: MMO 111: BYPASS
15	DECRYPTEN	RW	0x0	Decrypt operation. Ignored in MMO and BYPASS Mode. 0: Encryption 1: Decryption
14	OUTMIC	RW	0x1	Append MIC/HASH at the end of output stream in CCM* mode encryption/MMO mode. 0: Not append MIC/HASH at the end of output stream in CCM* mode encryption or MMO mode 1: Append MIC/HASH at the end of output stream in CCM* mode encryption or MMO mode

Bits	Field Name	Type	Reset	Description
13:12	MICLEN	RW	0x0	Length of MIC field 00: 0 bytes 01: 4 bytes 10: 8 bytes 11: 16 bytes
11:10	KEYSIZE	RW	0x0	Key size parameter 00: 16 bytes 01: 32 bytes 10: 24 bytes 11:
9	RESERVED_9	RW	0x0	Reserved.
8	RESERVED_8	RW	0x0	Reserved.
7:6	RESERVED_7_6	RO	0x0	Reserved.
5	OUTHDR	RW	0x0	Output B0 and l(a) in CCM* mode 0: Don't output B0 and l(a) at the beginning of output stream 1: Output B0 and l(a) at the beginning of output stream
4	OUTMSG	RW	0x1	Output stream to output FIFO 0: Block output stream from output FIFO 1: Forward output stream to output FIFO
3	OFIFOCLR	W1C	0x0	Clear output FIFO 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears the output FIFO This bit is self-cleared to 0
2	IFIFOCLR	W1C	0x0	Clear input FIFO 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears the input FIFO This bit is self-cleared to 0
1	RESERVED_1	RO	0x0	Reserved.
0	START	W1S	0x0	Start AES 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 starts AES operation This bit is self-cleared to 0

Table 6-8: AES Status Register (AESSTS) Layout

AESSTS (AES Status Register) Offset: 0x8 Default: 0x00000081 Access: AES -> AESSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_20							
23	22	21	20	19	18	17	16
RESERVED_31_20				OFIFODEPTH			IFIFODEPTH
15	14	13	12	11	10	9	8
IFIFODEPTH		ERRCODE			RESERVED_10_8		
7	6	5	4	3	2	1	0
OFIFOEMPTY	OFIFORDY	RESERVED_5	IFIFOFULL	RESERVED_3_1			DONE

Table 6-9: AES Status Register (AESSTS) Description

Bits	Field Name	Type	Reset	Description
31:20	RESERVED_31_20	RO	0x0	Reserved.
19:17	OFIFODEPTH	RO	0x0	The output FIFO depth
16:14	IFIFODEPTH	RO	0x0	The input FIFO depth
13:11	ERRCODE	RO	0x0	AES operation error status 000: No operation error 001: Input stream size less than 16 byte in ECB, CBC and CTR mode 010: Data is not multiple of 16 bytes in ECB mode or Data is more than 2^13-1 bytes in MMO mode 011: Data is not multiple of 16 bytes and less than 16 bytes in ECB mode 100: MIC Mismatch during CCM* Decryption 101: Not valid 110: Not valid 111: Not valid
10:8	RESERVED_10_8	RO	0x0	Reserved.
7	OFIFOEMPTY	RO	0x1	Output FIFO empty 0: Output FIFO is not empty 1: Output FIFO is empty
6	OFIFORDY	RO	0x0	Output FIFO is ready to read 0: Output FIFO is not ready to read 1: Output FIFO is ready to read
5	RESERVED_5	RO	0x0	Reserved.
4	IFIFOFULL	RO	0x0	Input FIFO full 0: Input FIFO is not full 1: Input FIFO is full
3:1	RESERVED_3_1	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
0	DONE	RO	0x1	AES operation done 0: AES operation has not done yet 1: AES operation done

Table 6-10: AES Associate String Length Register (AESASTRLEN) Layout

AESASTRLEN (AES Associate String Length Register) Offset: 0xC Default: 0x00000000 Access: AES -> AESASTRLEN.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-11: AES Associate String Length Register (AESASTRLEN) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Size of associate string

Table 6-12: AES Message String Length Register (AESMSTRLEN) Layout

AESMSTRLEN (AES Message String Length Register) Offset: 0x10 Default: 0x00000000 Access: AES -> AESMSTRLEN.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-13: AES Message String Length Register (AESMSTRLEN) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Size of message string

Table 6-14: AES Input Message Word Register (AESSTRIN) Layout

AESSTRIN (AES Input Message Word Register) Offset: 0x14 Default: 0x00000000 Access: AES -> AESSTRIN.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-15: AES Input Message Word Register (AESSTRIN) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	input message word

Table 6-16: AES Initial Vector Register 0 (AESIV0) Layout

AESIV0 (AES Initial Vector Register 0) Offset: 0x18 Default: 0x00000000 Access: AES -> AESIV0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-17: AES Initial Vector Register 0 (AESIV0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 0-3 of initial vector

Table 6-18: AES Initial Vector Register 1 (AESIV1) Layout

AESIV1 (AES Initial Vector Register 1) Offset: 0x1C Default: 0x00000000 Access: AES -> AESIV1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-19: AES Initial Vector Register 1 (AESIV1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 4-7 of initial vector

Table 6-20: AES Initial Vector Register 2 (AESIV2) Layout

AESIV2 (AES Initial Vector Register 2) Offset: 0x20 Default: 0x00000000 Access: AES -> AESIV2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-21: AES Initial Vector Register 2 (AESIV2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 8-11 of initial vector

Table 6-22: AES Initial Vector Register 3 (AESIV3) Layout

AESIV3 (AES Initial Vector Register 3) Offset: 0x24 Default: 0x00000000 Access: AES -> AESIV3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-23: AES Initial Vector Register 3 (AESIV3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 12-15 of initial vector

Table 6-24: AES Key Register 0 (AESKEY0) Layout

AESKEY0 (AES Key Register 0) Offset: 0x28 Default: 0x00000000 Access: AES -> AESKEY0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-25: AES Key Register 0 (AESKEY0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 0-3 of key

Table 6-26: AES Key Register 1 (AESKEY1) Layout

AESKEY1 (AES Key Register 1) Offset: 0x2C Default: 0x00000000 Access: AES -> AESKEY1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-27: AES Key Register 1 (AESKEY1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 4-7 of key

Table 6-28: AES Key Register 2 (AESKEY2) Layout

AESKEY2 (AES Key Register 2) Offset: 0x30 Default: 0x00000000 Access: AES -> AESKEY2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-29: AES Key Register 2 (AESKEY2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 8-11 of key

Table 6-30: AES Key Register 3 (AESKEY3) Layout

AESKEY3 (AES Key Register 3) Offset: 0x34 Default: 0x00000000 Access: AES -> AESKEY3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-31: AES Key Register 3 (AESKEY3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 12-15 of key

Table 6-32: AES Key Register 4 (AESKEY4) Layout

AESKEY4 (AES Key Register 4) Offset: 0x38 Default: 0x00000000 Access: AES -> AESKEY4.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-33: AES Key Register 4 (AESKEY4) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte of 16-19 of key

Table 6-34: AES Key Register 5 (AESKEY5) Layout

AESKEY5 (AES Key Register 5) Offset: 0x3C Default: 0x00000000 Access: AES -> AESKEY5.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-35: AES Key Register 5 (AESKEY5) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 20-23 of key

Table 6-36: AES Key Register 6 (AESKEY6) Layout

AESKEY6 (AES Key Register 6) Offset: 0x40 Default: 0x00000000 Access: AES -> AESKEY6.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-37: AES Key Register 6 (AESKEY6) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 24-27 of key

Table 6-38: AES Key Register 7 (AESKEY7) Layout

AESKEY7 (AES Key Register 7) Offset: 0x44 Default: 0x00000000 Access: AES -> AESKEY7.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-39: AES Key Register 7 (AESKEY7) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Byte 28-31 of key

Table 6-40: AES Output Message Word Register (AESSTROUT) Layout

AESSTROUT (AES Output Message Word Register) Offset: 0x48 Default: 0x00000000 Access: AES -> AESSTROUT.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-41: AES Output Message Word Register (AESSTROUT) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Output message word

Table 6-42: AES Output Vector Register 0 (AESOV0) Layout

AESOV0 (AES Output Vector Register 0) Offset: 0x4C Default: 0x00000000 Access: AES -> AESOV0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-43: AES Output Vector Register 0 (AESOV0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Byte 0-3 of output vector

Table 6-44: AES Output Vector Register 1 (AESOV1) Layout

AESOV1 (AES Output Vector Register 1) Offset: 0x50 Default: 0x00000000 Access: AES -> AESOV1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-45: AES Output Vector Register 1 (AESOV1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Byte 4-7 of output vector

Table 6-46: AES Output Vector Register 2 (AESOV2) Layout

AESOV2 (AES Output Vector Register 2) Offset: 0x54 Default: 0x00000000 Access: AES -> AESOV2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-47: AES Output Vector Register 2 (AESOV2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Byte 8-11 of output vector

Table 6-48: AES Output Vector Register 3 (AESOV3) Layout

AESOV3 (AES Output Vector Register 3) Offset: 0x58 Default: 0x00000000 Access: AES -> AESOV3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 6-49: AES Output Vector Register 3 (AESOV3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Byte 12-15 of output vector

Table 6-50: AES Interrupt Status Register (AESIF) Layout

AESIF (AES Interrupt Status Register) Offset: 0x5C Default: 0x00000000 Access: AES -> AESIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					OFIFOEMPTY	IFIFOFULL	DONE

Table 6-51: AES Interrupt Status Register (AESIF) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	OFIFOEMPTY	RO	0x0	Status of AES output FIFO empty interrupt 0: AES output FIFO empty interrupt not occurred 1: AES output FIFO empty interrupt occurred
1	IFIFOFULL	RO	0x0	Status of AES input FIFO full interrupt 0: AES input FIFO full interrupt not occurred 1: AES input FIFO full interrupt occurred
0	DONE	RO	0x0	Status of AES operation done interrupt 0: AES operation done interrupt not occurred 1: AES operation done interrupt occurred

Table 6-52: AES Interrupt Enable Register (AESIE) Layout

AESIE (AES Interrupt Enable Register) Offset: 0x60 Default: 0x00000000 Access: AES -> AESIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					OFIFOEMPTY	IFIFOFULL	DONE

Table 6-53: AES Interrupt Enable Register (AESIE) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
2	OFIFOEMPTY	RW	0x0	Enable AES output FIFO empty interrupt 0: Disable AES output FIFO empty interrupt 1: Enable AES output FIFO empty interrupt
1	IFIFOFULL	RW	0x0	Enable of AES input FIFO full interrupt 0: Disable AES input FIFO full interrupt 1: Enable AES input FIFO full interrupt
0	DONE	RW	0x0	Enable of AES operation done interrupt 0: Disable AES operation done interrupt 1: Enable AES operation done interrupt

Table 6-54: AES Interrupt Raw Status Register (AESRAWIF) Layout

AESRAWIF (AES Interrupt Raw Status Register) Offset: 0x64 Default: 0x00000000 Access: AES -> AESRAWIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					OFIFOEMPTY	IFIFOFULL	DONE

Table 6-55: AES Interrupt Raw Status Register (AESRAWIF) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	OFIFOEMPTY	RO	0x0	AES output FIFO empty interrupt raw status regardless of mask 0: AES output FIFO empty interrupt not occurred 1: AES output FIFO empty interrupt
1	IFIFOFULL	RO	0x0	AES input FIFO full interrupt raw status regardless of mask 0: AES no input FIFO full interrupt not occurred 1: AES no input FIFO full interrupt occurred
0	DONE	RO	0x0	AES operation done interrupt raw status regardless of mask 0: AES operation done interrupt not occurred 1: AES operation done interrupt occurred

Table 6-56: AES Interrupt Clear Register (AESIC) Layout

AESIC (AES Interrupt Clear Register) Offset: 0x68 Default: 0x00000000 Access: AES -> AESIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					OFIFOEMPTY	IFIFOFULL	DONE

Table 6-57: AES Interrupt Clear Register (AESIC) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	OFIFOEMPTY	W1C	0x0	clearance of AES output FIFO empty interrupt status and raw status 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears both AES output FIFO empty interrupt status and raw status This bit is self-cleared to 0
1	IFIFOFULL	W1C	0x0	Clearance of AES input FIFO full interrupt status and raw status 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears both AES input FIFO full interrupt status and raw status This bit is self-cleared to 0
0	DONE	W1C	0x0	Clearance of AES operation done interrupt status and raw status 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears both AES operation done interrupt status and raw status This bit is self-cleared to 0

7 CRC

7.1 CRC overview

A Cyclic Redundancy Check (CRC) or polynomial code checksum is a hash function designed to detect accidental changes to raw computer data, and is used to verify data transmission or storage generates up to 32-bit CRC code for error detection.

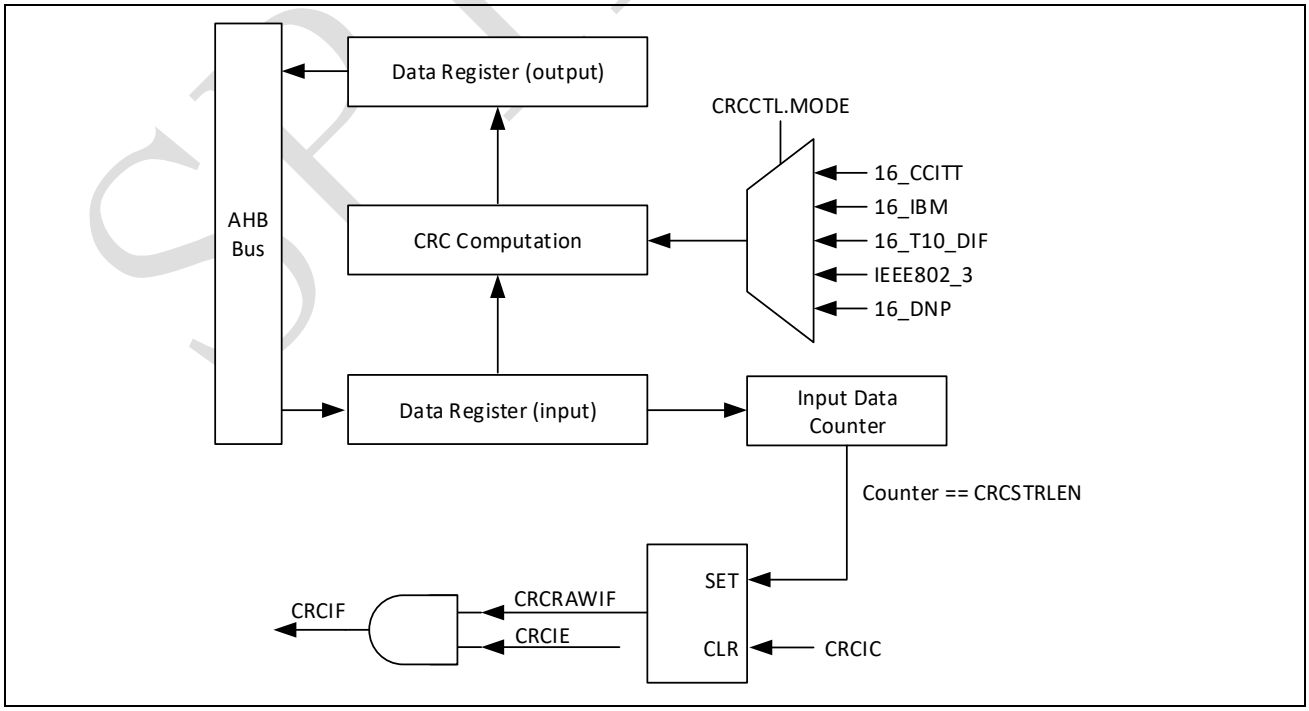
7.2 Main features

A standard AHB slave interface is used to configure the module, receive the bit stream, and output the CRC result. The CRC module supports the following features:

- Supports 32-bit parallel bit stream input, and supports up to 32-bit CRC output
- Supports up to 2^{32}(4294967296) byte length to calculate CRC
- Supports the following CRC standards
  - CRC-16-CCITT, the polynomial is $x^{16}+x^{12}+x^5+1$
  - CRC-16-IBM, the polynomial is $x^{16}+x^{15}+x^2+1$
  - CRC-16-T10-DIF, the polynomial is $x^{16}+x^{15}+x^{11}+x^9+x^8+x^7+x^5+x^4+x^2+1$
  - CRC-32-IEEE 802.3, the polynomial is
  - $X^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
  - CRC-16-DNP, the polynomial is $x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^8+x^6+x^5+x^2+1$

The block diagram is shown in Figure 7-1.

Figure 7-1: CRC block diagram



7.3 CRC operation flow

User can operate the CRC module according to the following steps:

Step 1: Disable CRC (set CRCCTL.EN to 0)

Step 2: Disable the interrupt (set CRCIE.DONE to 0)

Step 3: Clear all the interrupts (set CRCIC.DONE to 1)

Step 4: Configure the stream length (set register CRCSTRLEN)

Step 5: Configure CRC mode (set bit CRCCTL.MODE)

Step 6: Enable the interrupt (set CRCIE.DONE to 1)

Step 7: Enable CRC (set CRCCTL.EN to 1)

Step 8: Write stream in and waiting for interrupt to occur (If interrupt occurred, go to Step 9)

Step 9: Get CRC calculation result (Read register CRCRESULT)

Step 10: CRC operation complete

Note: The CRC input stream registers accepts a word (32 bit) at a time. If the input data is not 4 bytes aligned, pad zeros at the start of the data stream. For example, if the data stream consists of 5 bytes starting from the lower address: 0xA1, 0xA2, 0xA3, 0xA4, 0xA5, the following two words should be written to the stream input register:

- 0xA1000000
- 0xA5A4A3A2

The CRC result bit order is:

- 16 bit CRC: x_0~x_{15} [MSB->LSB]
- 32 bit CRC: x_0~x_{31} [MSB->LSB]

7.4 Registers

7.4.1 CRC register map

Table 7-1: CRC Module Base Address

Peripheral Module	Base Address
CRC	0x4000 8000

Table 7-2: CRC Register Map

Register	Offset	Description	Reset Value
CRCIF	0x0	CRC Interrupt Flag Register	0x00000000
CRCRAWIF	0x4	CRC Raw Interrupt Flag Register	0x00000000
CRCIC	0x8	CRC Interrupt Clear Register	0x00000000
CRCIE	0xC	CRC Interrupt Enable Register	0x00000000
CRCCTL	0x10	CRC Control Register	0x00000000
CRCSTRLEN	0x14	CRC Stream Length Register	0x00000000
CRCSTRIN	0x18	CRC Stream Input Register	0x00000000
CRCRESULT	0x1C	CRC Result Register	0x00000000

7.4.2 CRC registers

Table 7-3: CRC Interrupt Flag Register (CRCIF) Layout

CRCIF (CRC Interrupt Flag Register) Offset: 0x0 Default: 0x00000000 Access: CRC -> CRCIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							DONE

Table 7-4: CRC Interrupt Flag Register (CRCIF) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	DONE	RO	0x0	CRC calculation done interrupt status after mask 0: Interrupt is not occurred 1: Interrupt is occurred

Table 7-5: CRC Raw Interrupt Flag Register (CRCRAWIF) Layout

CRCRAWIF (CRC Raw Interrupt Flag Register) Offset: 0x4 Default: 0x00000000 Access: CRC -> CRCRAWIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							DONE

Table 7-6: CRC Raw Interrupt Flag Register (CRCRAWIF) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	DONE	RO	0x0	CRC calculation done interrupt raw status regardless of mask 0: Interrupt is not occurred 1: Interrupt is occurred

Table 7-7: CRC Interrupt Clear Register (CRCIC) Layout

CRCIC (CRC Interrupt Clear Register) Offset: 0x8 Default: 0x00000000 Access: CRC -> CRCIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							DONE

Table 7-8: CRC Interrupt Clear Register (CRCIC) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	DONE	W1C	0x0	Clearance of CRCIF[0] and CRCRAWIF[0] 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears both CRCIF[0] and CRCRAWIF[0]. This bit is self-cleared to 0

Table 7-9: CRC Interrupt Enable Register (CRCIE) Layout

CRCIE (CRC Interrupt Enable Register) Offset: 0xC Default: 0x00000000 Access: CRC -> CRCIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							DONE

Table 7-10: CRC Interrupt Enable Register (CRCIE) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	DONE	RW	0x0	Enable interrupt triggered by CRCRWIF[0] 0: Disable generation of IRQ and corresponding CRCIF[0] 1: Enable generation of IRQ and corresponding CRCIF[0]

Table 7-11: CRC Control Register (CRCCTL) Layout

CRCCTL (CRC Control Register) Offset: 0x10 Default: 0x00000000 Access: CRC -> CRCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_5							
23	22	21	20	19	18	17	16
RESERVED_31_5							
15	14	13	12	11	10	9	8
RESERVED_31_5							
7	6	5	4	3	2	1	0
RESERVED_31_5			RESERVED_4	MODE			EN

Table 7-12: CRC Control Register (CRCCTL) Description

Bits	Field Name	Type	Reset	Description
31:5	RESERVED_31_5	RO	0x0	Reserved.
4	RESERVED_4	RW	0x0	Reserved.
3:1	MODE	RW	0x0	CRC mode select 000: x^16+x^12+x^5+1 (CRC-16-CCITT, CRC-CCITT) 001: x^16+x^15+x^2+1 (CRC-16, CRC-16-IBM, CRC-16-ANSI) 010: x^16+x^15+x^11+x^9+x^8+x^7+x^5+x^4+x^2+x+1 (CRC-16-T10-DIF) 011: x^32+x^26+x^23+x^22+x^16+x^12+x^11+x^10+x^8+x^7+x^5+x^4+x^2+x+1 (CRC-32-IEEE802.3) 100: x^16+x^13+x^12+x^11+x^10+x^8+x^6+x^5+x^2+1(CRC-16-DNP) 101: Reserved 110: Reserved 111: Reserved
0	EN	RW	0x0	CRC calculate enable 0: Disable CRC calculation 1: Enable CRC calculation, it is automatically cleared when the CRC calculation is finished

Table 7-13: CRC Stream Length Register (CRCSTRLEN) Layout

CRCSTRLEN (CRC Stream Length Register) Offset: 0x14 Default: 0x00000000 Access: CRC -> CRCSTRLEN.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 7-14: CRC Stream Length Register (CRCSTRLEN) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Input stream length minus 1(in unit of byte)

Table 7-15: CRC Stream Input Register (CRCSTRIN) Layout

CRCSTRIN (CRC Stream Input Register) Offset: 0x18 Default: 0x00000000 Access: CRC -> CRCSTRIN.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 7-16: CRC Stream Input Register (CRCSTRIN) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Stream Input data register

Table 7-17: CRC Result Register (CRCRESULT) Layout

CRCRESULT (CRC Result Register) Offset: 0x1C Default: 0x00000000 Access: CRC -> CRCRESULT.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 7-18: CRC Result Register (CRCRESULT) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	CRC calculation result

8 General-purpose timers

8.1 Timer overview

The SPD1148 includes three 32-bit General Purpose Timers (GPT) with following features:

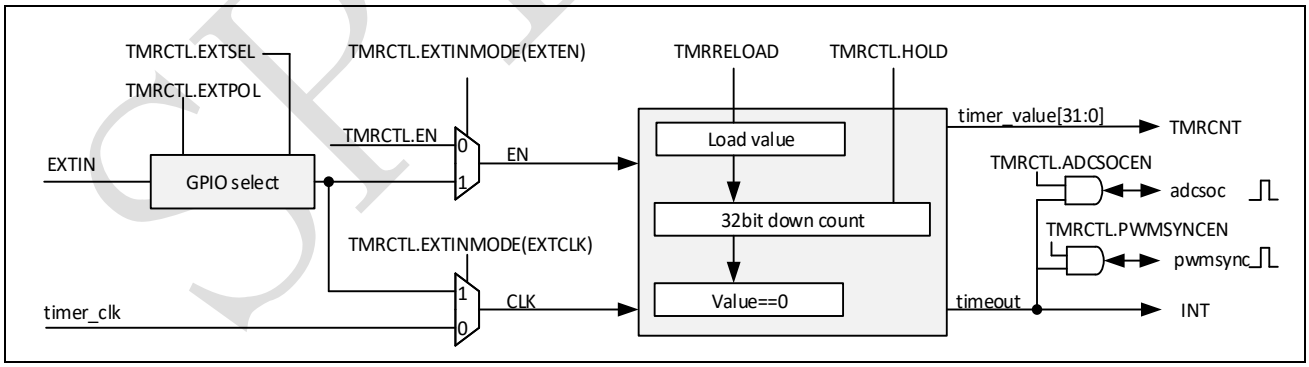
- APB register interface
- Dedicated timer clock, shown in Figure 3-13
- 32-bit down-counter
- Generate interrupt upon counter reaches zero
- Generate ADCSOC event upon counter reaches zero
- Generate PWMSYNC event upon counter reaches zero
- Capture external input as timer enable
- Capture external input as timer clock

8.2 Functional description

Each GPT operates under periodic timer mode. As shown in Figure 8-1, when the timer is enabled, the counter loads the value from TMRRELOAD register and starts to count down until it reaches 0. Then the counter restarts from the reload value and decrease again. Upon the counter equals 0:

- An interrupt will be generated if it is enabled. The interrupt flag held until it is manually cleared.
- An ADCSOC event will be generated if TMRCTL.ADCSOCEN is enabled.
- A PWMSYNC event will be generated if TMRCTL.PWMSYNCEN is enabled.

Figure 8-1: General-purpose timer structure



The low to high transition on external input (EXTIN) can be used as timer enable. It can also be used as external clock. The register TMRCTL.EXTSEL is used to select GPIO pin as the external input (EXTIN). Please see Table 8-3 and Table 8-4 for the details.

The timers are disabled by default after system reset.

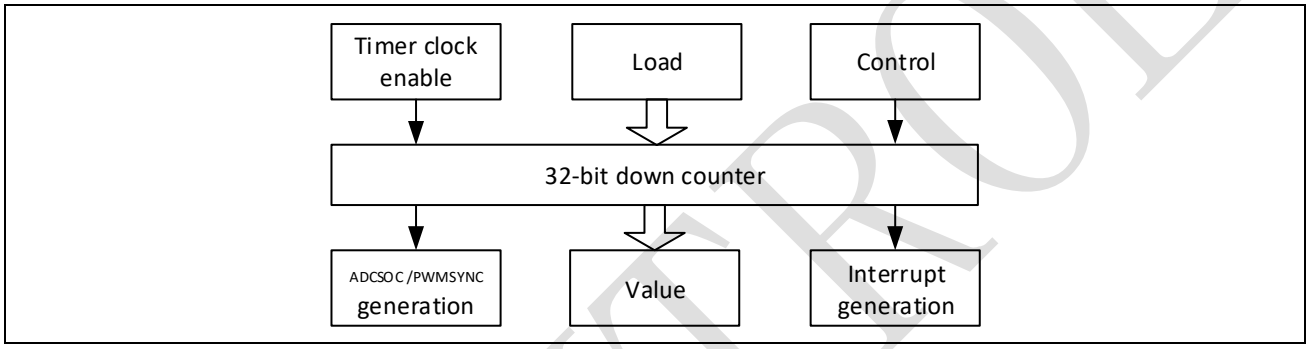
8.3 Initialization sequence

The following sequence of operations should be followed to start the timer.

- Disable the timer and clear timer interrupt
- Configure the timer reload value register (TMRRELOAD)
- Configure the timer control register (TMRCTL)
- Enable the timer to start the timer counter

A block diagram of the timer run mode is shown in Figure 8-2.

Figure 8-2: GPT run flow diagram



8.4 Registers

8.4.1 TIMER register map

Table 8-1: Timer Module Base Address

Peripheral Module	Base Address
TIMER0	0x4000 7000
TIMER1	0x4000 7020
TIMER2	0x4000 7040

Table 8-2: TIMER Register Map

Register	Offset	Description	Reset Value
TMRCTL	0x0	Timer Control Register	0x00000000
TMRCNT	0x4	Timer Counter Value Register	0x00000000
TMRRELOAD	0x8	Timer Reload Value Register	0x00000000
TMRIF	0xC	Timer Interrupt Flag Register	0x00000000
TMRRAWIF	0x10	Timer Raw Interrupt Flag Register	0x00000000
TMRIE	0x14	Timer Interrupt Enable Register	0x00000000
TMRIFRC	0x18	Timer Interrupt Force Register	0x00000000
TMRIC	0x1C	Timer Interrupt Clear Register	0x00000000

8.4.2 TIMER registers

Table 8-3: Timer Control Register (TMRCTL) Layout

TMRCTL (Timer Control Register) Offset: 0x0 Default: 0x00000000
Access: TIMER0 -> TMRCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_13							
23	22	21	20	19	18	17	16
RESERVED_31_13							
15	14	13	12	11	10	9	8
RESERVED_31_13			EXTSEL				
7	6	5	4	3	2	1	0
EXTSEL	EXTPOL	EXTINMODE		PWMSYNCEN	ADCSOCEN	RESERVED_1	EN

Table 8-4: Timer Control Register (TMRCTL) Description

Bits	Field Name	Type	Reset	Description
31:13	RESERVED_31_13	RO	0x0	Reserved.
12:7	EXTSEL	RW	0x0	External input source select (GPIO number)
6	EXTPOL	RW	0x0	External input polarity 0: Active low 1: Active high
5:4	EXTINMODE	RW	0x0	External input mode 00: Disable external input 01: 10: Take external input as timer clock 11: Take external input as timer enable
3	PWMSYNCEN	RW	0x0	PWMSYNC generation enable 0: Do not generate PWMSYNC 1: Generate PWMSYNC whenever TMRCNT counts down to 0
2	ADCSOCEN	RW	0x0	ADCSOC generation enable 0: Do not generate ADCSOC 1: Generate ADCSOC whenever TMRCNT counts down to 0
1	RESERVED_1	RW	0x0	Reserved.
0	EN	RW	0x0	Timer enable 0: Disable Timer 1: Enable Timer

Table 8-5: Timer Counter Value Register (TMRCNT) Layout

TMRCNT (Timer Counter Value Register) Offset: 0x4 Default: 0x00000000 Access: TIMER0 -> TMRCNT.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 8-6: Timer Counter Value Register (TMRCNT) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Current value of timer counter.

Table 8-7: Timer Reload Value Register (TMRRELOAD) Layout

TMRRELOAD (Timer Reload Value Register) Offset: 0x8 Default: 0x00000000 Access: TIMER0 -> TMRRELOAD.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 8-8: Timer Reload Value Register (TMRRELOAD) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Reload value, A Write to this registers sets the current value.

Table 8-9: Timer Interrupt Flag Register (TMRIF) Layout

TMRIF (Timer Interrupt Flag Register) Offset: 0xC Default: 0x00000000 Access: TIMER0 -> TMRIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							INT

Table 8-10: Timer Interrupt Flag Register (TMRIF) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	INT	RO	0x0	Timer interrupt 0: Interrupt not occurred 1: Interrupt occurred and issued to CPU No further interrupt will be issued until the flag is cleared

Table 8-11: Timer Raw Interrupt Flag Register (TMRRAWIF) Layout

TMRRAWIF (Timer Raw Interrupt Flag Register) Offset: 0x10 Default: 0x00000000 Access: TIMER0 -> TMRRAWIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							INT

Table 8-12: Timer Raw Interrupt Flag Register (TMRRAWIF) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	INT	RO	0x0	Timer raw interrupt status. This bit is set whenever TMRCNT counts down to 0 and cleared only by writing a 1 to TMRIC 0: Interrupt not occurred 1: Interrupt occurred

Table 8-13: Timer Interrupt Enable Register (TMRIE) Layout

TMRIE (Timer Interrupt Enable Register) Offset: 0x14 Default: 0x00000000 Access: TIMER0 -> TMRIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							INT

Table 8-14: Timer Interrupt Enable Register (TMRIE) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	INT	RW	0x0	Timer interrupt enable. This bit does not affect TMRRAWIF and PWMSYNC/ADCSOC generation 0: Do not issue interrupt to CPU 1: Issue interrupt whenever TMRRAWIF=1

Table 8-15: Timer Interrupt Force Register (TMRIFRC) Layout

TMRIFRC (Timer Interrupt Force Register) Offset: 0x18 Default: 0x00000000 Access: TIMER0 -> TMRIFRC.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							INT

Table 8-16: Timer Interrupt Force Register (TMRIFRC) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	INT	W1S	0x0	Timer interrupt software force 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 forces the TMRRAWIF flag This bit is self-cleared to 0.

Table 8-17: Timer Interrupt Clear Register (TMRIC) Layout

TMRIC (Timer Interrupt Clear Register) Offset: 0x1C Default: 0x00000000 Access: TIMER0 -> TMRIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							INT

Table 8-18: Timer Interrupt Clear Register (TMRIC) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	INT	W1C	0x0	Timer interrupt clear 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 clears the TMRRAWIF/TMRIF flag This bit is self-cleared to 0.

9 Watchdog timers

9.1 Watchdog overview

The watchdog timer (WDT) regains control in case of system failure (due to a software error) to increase application reliability. The WDT can generate a reset or an interrupt when the counter reaches a given timeout value. The SPD1148 includes two 32-bit WDTs. Registers are controlled via AHB bus.

Each WDT has following features:

- APB register interface
- Dedicated WDT clock, shown in Figure 3-12
- 32-bit down-counter
- Configurable reset or interrupt generation with the given timeout value

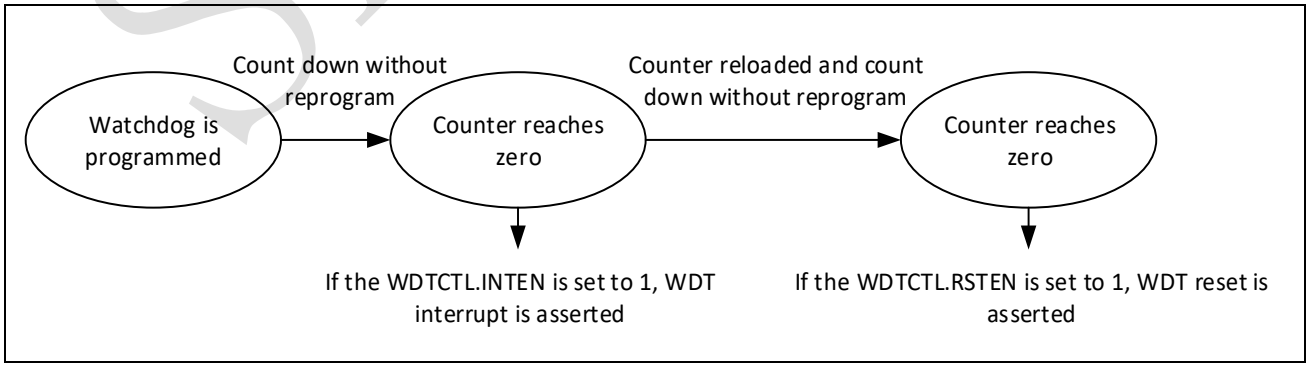
9.2 Functional description

The watchdog counter descends from a preset (timeout) value to zero. The timeout value is obtained by load register WDTLOAD, which should be written after reset and before the WDT is enabled. When the counter reaches zero, either a system reset or an interrupt occurs, depending on the set of the control register WDTCTL.

As a safety feature to prevent rogue software from disabling the watchdog functionality, writing a value of 0x1ACCE551 to WDTREGKEY register enables write access to all other registers, while writing any other value disables the write access. Before configuring the WDT operation, the write access to other registers should be enabled first.

Figure 9-1 shows the flow diagram for the watchdog operation. If the interrupt is enabled, the counter of WDT will be enabled and generate an interrupt when a timeout occurs. Then the counter reloads value from WDTLOAD and continues the count-down sequence. If the interrupt is not cleared by the time a second timeout occurs and the reset is enabled, it generates a system reset request.

Figure 9-1: Watchdog operation flow



When a writing to WDTLOAD occurs, the counter restarts from the new value immediately. When a writing to WDTIC occurs, the interrupt will be cleared and the counter restarts. If the restart occurs

at the same time the watchdog counter reaches zero, the interrupt will not be generated. The WDT can be closed by configuring WDTCTL register when CPU in HALTED mode or LOCKUP mode.

Note: The WDT0 interrupt request signal is connected to the NMI interrupt.

9.3 Initialization sequence

The following sequence of operations must be followed to start the watchdog timer:

- Configure the WDTREGKEY register to enable registers write
- Configure the WDTLOAD register for timeout value
- Configure the WDTCTL.RSTEN bit for reset setup
- Configure interrupt enable bit to start counter and enable interrupt
- Configure the WDTREGKEY Register disable registers write

9.4 Registers

9.4.1 WDT register map

Table 9-1: WDT Module Base Address

Peripheral Module	Base Address
WDT0	0x4000 1000
WDT1	0x4000 2000

Table 9-2: WDT Register Map

Register	Offset	Description	Reset Value
WDTLOAD*	0x0	Watchdog Timer Load Register	0x003D0900
WDTCNT	0x4	Watchdog Timer Current Value Register	0xFFFFFFFF
WDTCTL*	0x8	Watchdog Timer Control Register	0x0000000F
WDTIC*	0xC	Watchdog Timer Clear Interrupt Register	0x00000000
WDTRAWIF	0x10	Watchdog Timer Raw Interrupt Status Register	0x00000000
WDTIF	0x14	Watchdog Timer Interrupt Status Register	0x00000000
WDTREGKEY	0x18	Watchdog Timer Lock Register	0x00000000

Note: 1. Registers marked with * are write-allowed only when the WDTREGKEY=0x1ACCE551.
2. LSB of all register readback will be 1 if WDTREGKEY is not 0x1ACCE551

9.4.2 WDT registers

Table 9-3: Watchdog Timer Load Register (WDTLOAD) Layout

WDTLOAD (Watchdog Timer Load Register) Offset: 0x0 Default: 0x003D0900 Access: WDT0 -> WDTLOAD.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 9-4: Watchdog Timer Load Register (WDTLOAD) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x3D0900	Load value The value from which the counter is to decrease. When this register is written to, the count is immediately restarted from the new value. The minimum valid value is 1.

Table 9-5: Watchdog Timer Current Value Register (WDTCNT) Layout

WDTCNT (Watchdog Timer Current Value Register) Offset: 0x4 Default: 0xFFFFFFFF Access: WDT0 -> WDTCNT.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 9-6: Watchdog Timer Current Value Register (WDTCNT) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0xFFFFFFFF	Current value of the decrementing counter.

Table 9-7: Watchdog Timer Control Register (WDTCTL) Layout

WDTCTL (Watchdog Timer Control Register) Offset: 0x8 Default: 0x0000000F Access: WDT0 -> WDTCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				LOCKUPRUN	HALTEDRUN	RSTEN	INTEN

Table 9-8: Watchdog Timer Control Register (WDTCTL) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	LOCKUPRUN	RW	0x1	Allow watchdog to run during core lockup mode 0: Disable watchdog when core is in lockup mode

Bits	Field Name	Type	Reset	Description
				1: Enable watchdog when core is in lockup mode
2	HALTEDRUN	RW	0x1	Allow watchdog to run during core halted mode 0: Disable watchdog when core is in halted mode 1: Enable watchdog when core is in halted mode
1	RSTEN	RW	0x1	Enable watchdog reset output, Acts as a mask for the reset output. 0: Disable the reset request 1: Enable the reset request
0	INTEN	RW	0x1	Enable the interrupt event Reloads the counter from WDTLOAD when the interrupt upon a rising edge of this bit. 0: Disable the counter and the interrupt 1: Enable the counter and the interrupt

Table 9-9: Watchdog Timer Clear Interrupt Register (WDTIC) Layout

WDTIC (Watchdog Timer Clear Interrupt Register) Offset: 0xC Default: 0x00000000 Access: WDT0 -> WDTIC.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 9-10: Watchdog Timer Clear Interrupt Register (WDTIC) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	Interrupt clear register A write of any value to this register clears the watchdog interrupt, and reloads the counter from WDTLOAD.

Table 9-11: Watchdog Timer Raw Interrupt Status Register (WDTRAWIF) Layout

WDTRAWIF (Watchdog Timer Raw Interrupt Status Register) Offset: 0x10 Default: 0x00000000 Access: WDT0 -> WDTRAWIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							RAWINT

Table 9-12: Watchdog Timer Raw Interrupt Status Register (WDTRAWIF) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	RAWINT	RO	0x0	Raw interrupt status from the counter This register indicates the raw interrupt status from the counter. The value is ANDed with WDTCTL.INTEN to create the masked interrupt, which is passed to the interrupt output pin. 0: Interrupt not occurred 1: Interrupt occurred

Table 9-13: Watchdog Timer Interrupt Status Register (WDTIF) Layout

WDTIF (Watchdog Timer Interrupt Status Register) Offset: 0x14 Default: 0x00000000 Access: WDT0 -> WDTIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							INT

Table 9-14: Watchdog Timer Interrupt Status Register (WDTIF) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
0	INT	RO	0x0	Masked interrupt status from the counter This register indicates the masked interrupt status from the counter. This value is the logical AND of WDTRAWIF.RAWINT and WDTCTL.INTEN, and is the same value that is passed to the interrupt output pin. 0: Interrupt not occurred 1: Interrupt occurred

Table 9-15: Watchdog Timer Lock Register (WDTREGKEY) Layout

WDTREGKEY (Watchdog Timer Lock Register) Offset: 0x18 Default: 0x00000000 Access: WDT0 -> WDTREGKEY.all							
31	30	29	28	27	26	25	24
LCKCTL							
23	22	21	20	19	18	17	16
LCKCTL							
15	14	13	12	11	10	9	8
LCKCTL							
7	6	5	4	3	2	1	0
LCKCTL							LCKSTS

Table 9-16: Watchdog Timer Lock Register (WDTREGKEY) Description

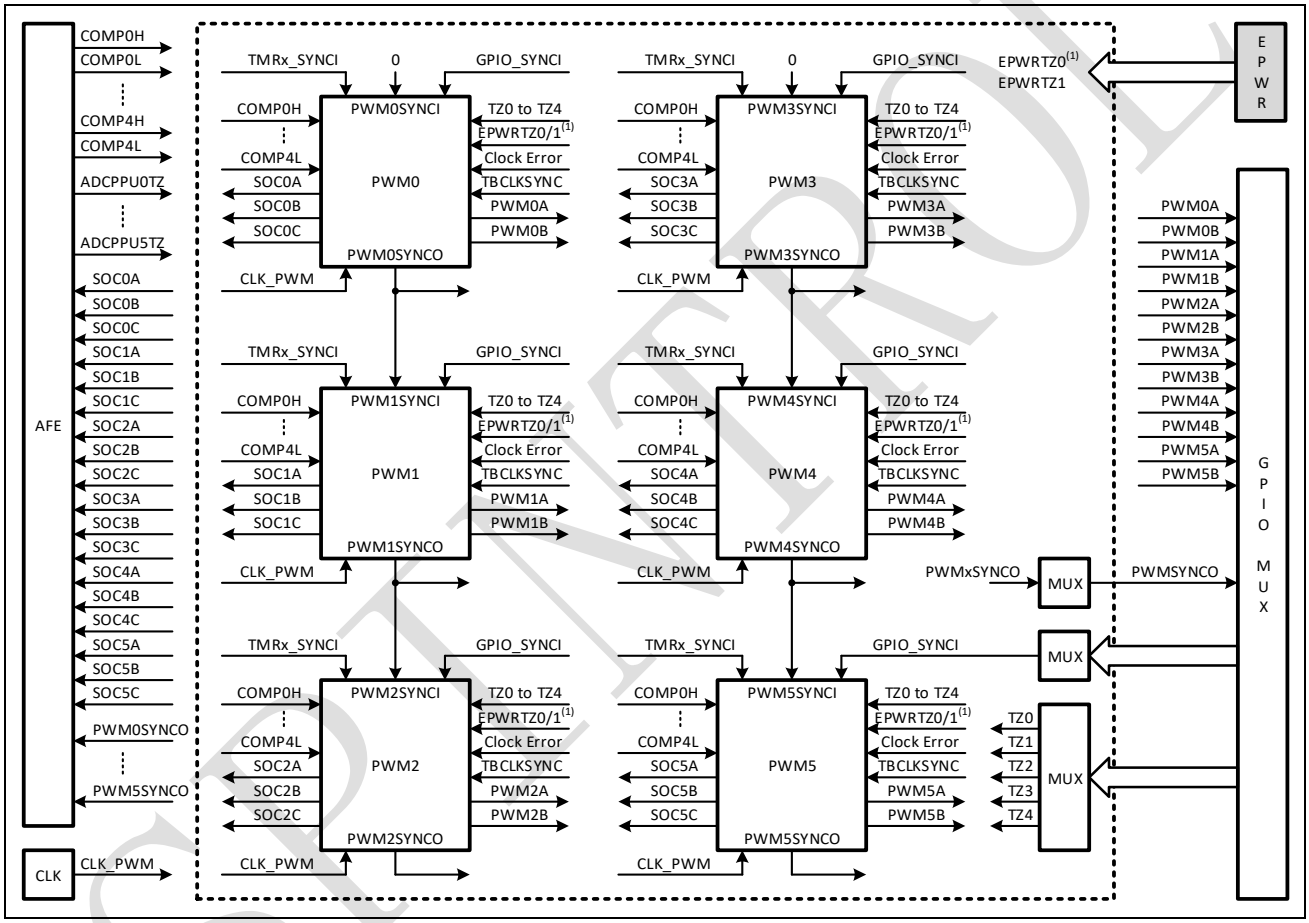
Bits	Field Name	Type	Reset	Description
31:1	LCKCTL	RW	0x0	Enable register writes Enable write access to all other registers by writing 0x1ACCE551. Disable write access by writing any other value.
0	LCKSTS	RO	0x0	Register write lock status 0: Write access to all other registers is enabled (not locked) 1: Write access to all other registers is disabled (locked)

10 PWM

10.1 PWM overview

The PWM plays a key role in power electronic systems such as digital motor control, switch mode power supply control, and uninterruptible power supplies (UPS). The SPD1148 provides 6 PWM modules while each PWM has two outputs: PWMxA and PWMxB. They are chained together via a clock synchronization scheme as shown in Figure 10-1.

Figure 10-1: PWM overview



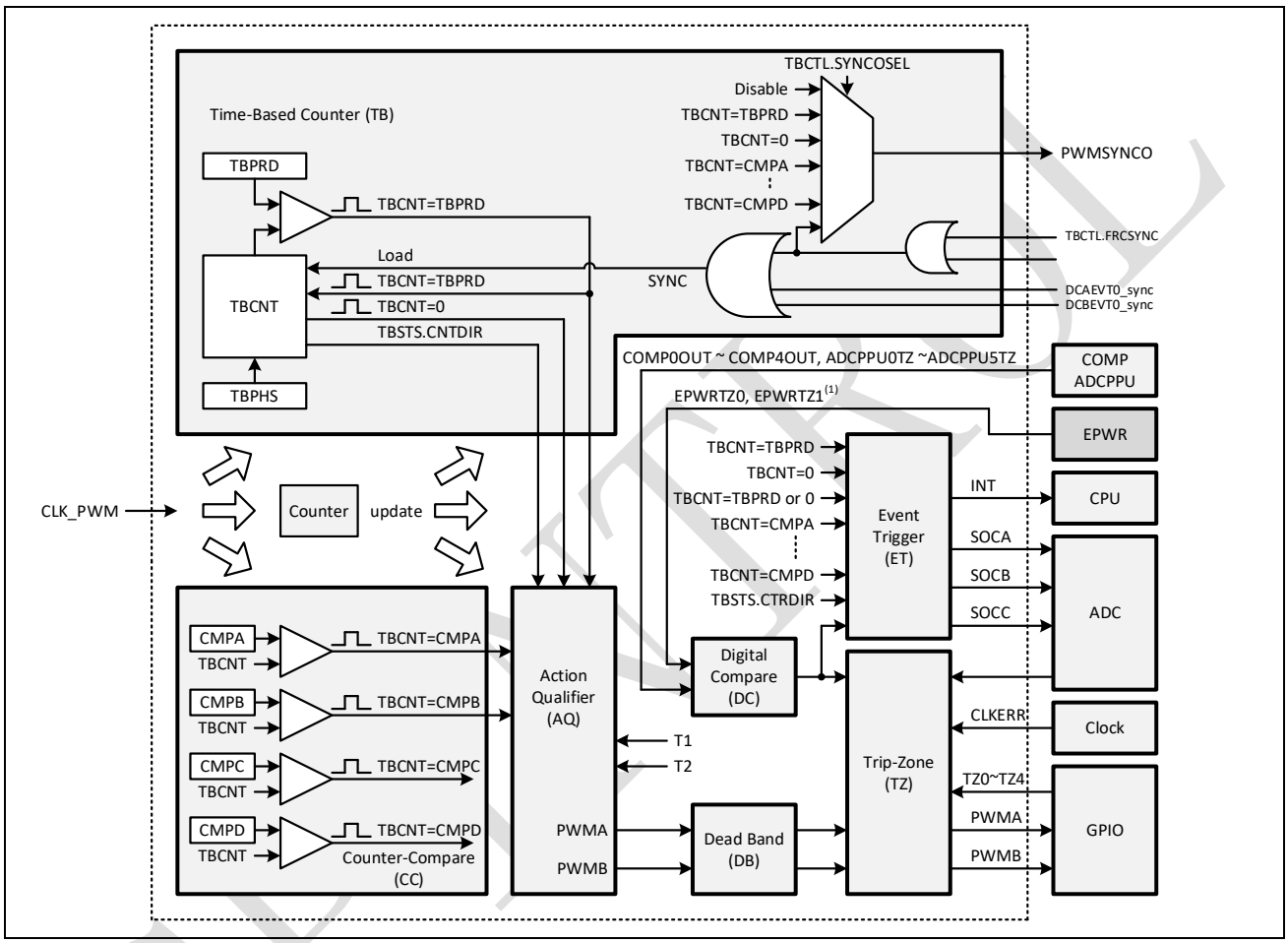
[1] In SPD1148, EPWR module is left unconnected and EPWRTZ0/EPWRTZ1 are always 0; In SPD1148, EPWRTZ0 is always 0.

Each PWM module supports the following features:

- 16-bit time-based counter with period and frequency control.
- Software override control of PWM signals.
- Independent and programmable phase-control support.
- Synchronized phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.

- Either high, low, or high-impedance state logic levels can be forced at the PWM outputs on a trip condition.
- Comparator module outputs and trip zone inputs can generate events, filtered events, or trip conditions.
- All events can trigger both CPU interrupt and ADC start of conversion (SOC).

Figure 10-2: PWM architecture



[1] In SPD1148, EPWR module is left unconnected and EPWRTZ0/EPWRTZ1 are always 0; In SPD1148, EPWRTZ0 is always 0.

Figure 10-2 shows more internal details of a single PWM module. The key signals of a PWM module includes:

- Synchronization IO (PWMSYNCI and PWMSYNCO)

  It is used to synchronize the phase relationship of multiple PWMs. In SPD1148, the PWMSYNCI can be from the GPIO, the timer or the PWMSYNCO of the preceding PWM.

Note: It takes one PWM clock delay from the synchronization input to the actual synchronization action, results in one time-base cycle delay if both TBDIVBIN and TBDIVLIN bits of TBCTL register are 0. Please take this into account when setting the TBPHS register.

The register FRCSYNC in PWMCFG module provides a more flexible and convenient synchronization among the PWMs.

- Trip-zone signals

  Each PWM has 5 trip-zone signals (TZ0~TZ4) configured from the GPIOs to alert the external fault conditions. Clock error event from the CLKDET module or the PLL and the halted/lockup event from the CPU can also be configured as trip-zone events for each PWM.

Note: The signal polarity from the GPIO to the trip-zone logic is programmable. By default, the trip-zone event is triggered when the GPIO is high.

- Comparator results
  The results from the 10 external comparators can generate digital compare events.
- ADC start-of-conversion (PWMxSOCA , PWMxSOCB and PWMxSOCC)
  Each PWM has three ADC start of conversion signals. The event to trigger the start of conversion signal is configured in the Event Trigger sub-module.
- PWM output (PWMxA and PWMxB)

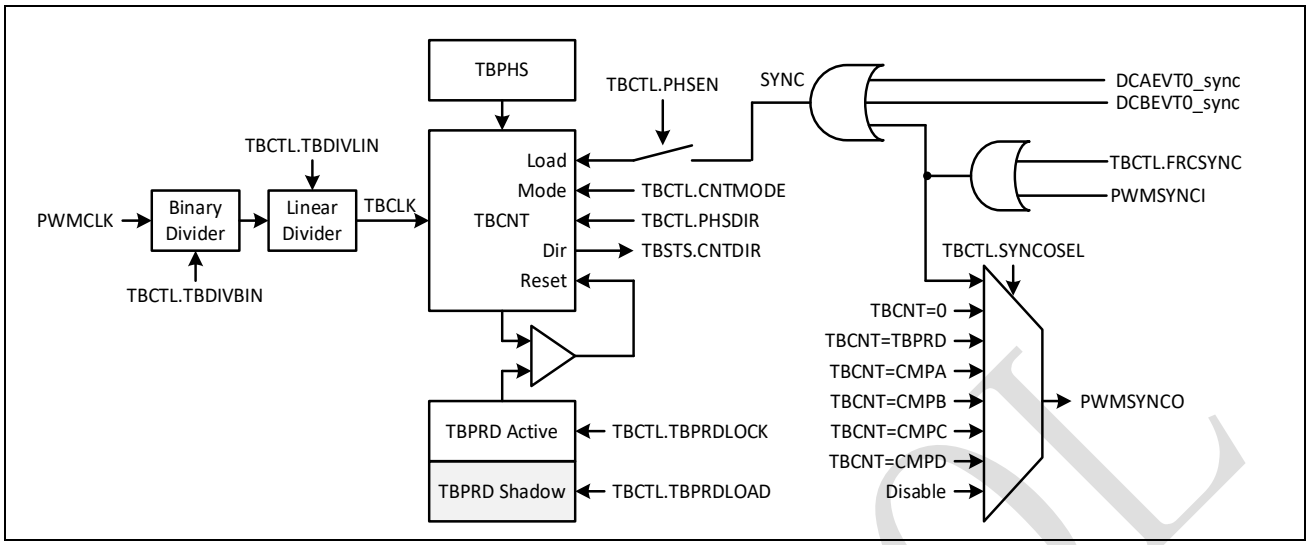
  The PWM outputs can be configured to be multiplexed with the GPIO as in IO_Channel_Defination.

10.2 Time-Base (TB) sub-module

The time-based sub-module handles the timing for the PWM module. The key signals is given in Figure 10-3. It is programmable as below:

- Specify the time-base clock (TBCLK) dividing ratio from the input clock.
- Set the time-base counter (TBCNT) mode:
  - Count-up
  - Count-down
  - Count-up-and-down
- Specify the time-base counter (TBCNT) period.
- Determine the phase relationship with other PWM module.
- Generate the events TBCNT=TBPRD and TBCNT=0.
- Specify the synchronization trigger events input and the action taken on the counter.

Figure 10-3: Time-Base sub-module signals



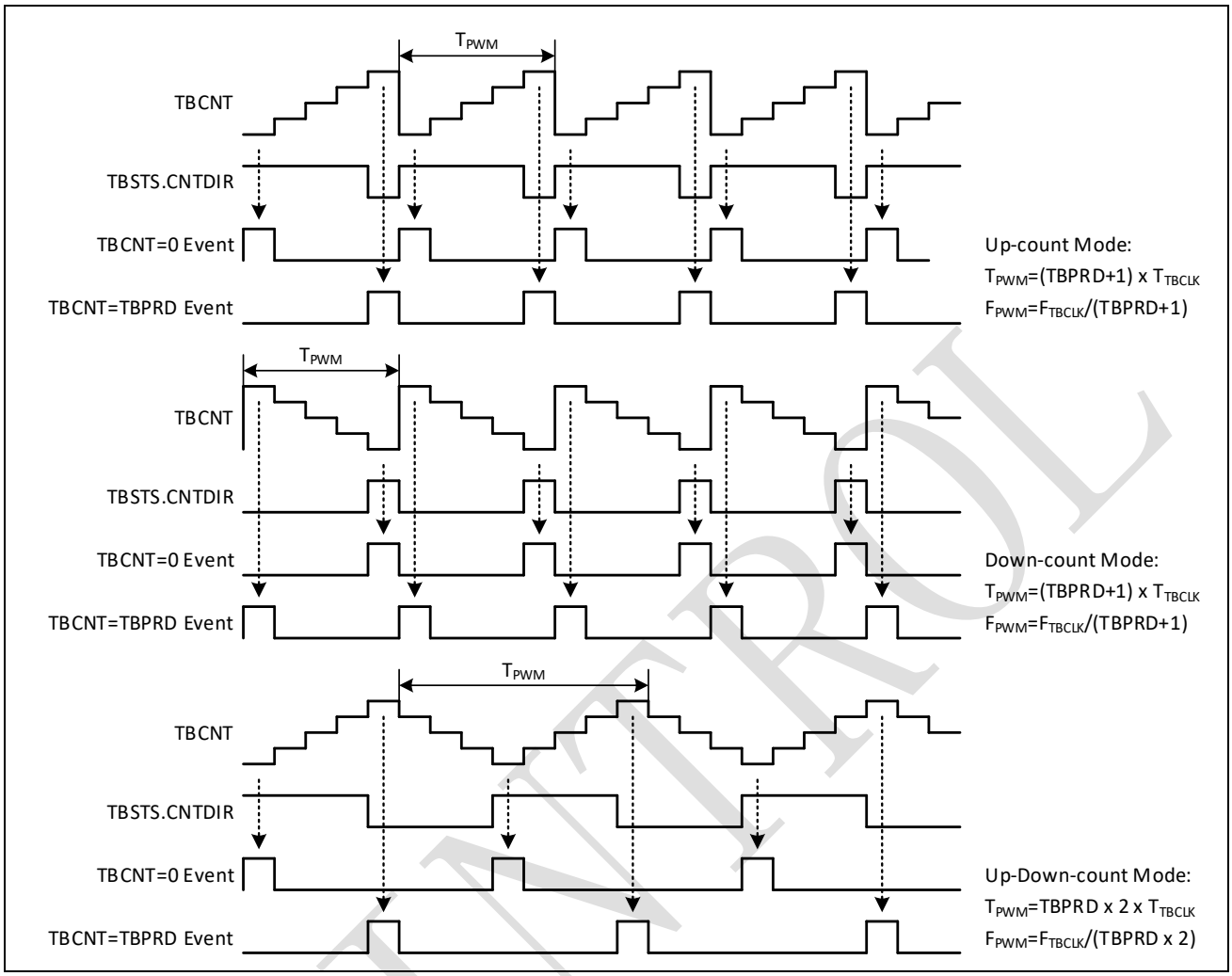
The frequency of the PWM events is controlled by the counter mode (TBCTL.CNTMODE) and the time-base period (TBPRD) register. Figure 10-4 shows the period and frequency relationships for the up-count, down-count, and up-down-count time-base counter modes.

In up-count mode, the time-base counter starts from zero and increases until it reaches TBPRD. Then the TBCNT is reset to zero and starts to increase again.

In down-count mode, the time-base counter starts from TBPRD and decreases until it reaches zero. Then the TBCNT is reset to TBPRD and starts to decrease again.

In up-down-count mode, the time-base counter increases from zero until it reaches TBPRD and then decrease until it reaches zero. Read back from the CNTDIR bit in the TBSTS register is 1 when the counter increases from zero to (TBPRD-1) and 0 when the counter decreases from TBPRD to 1.

Figure 10-4: Time-base counter waveform

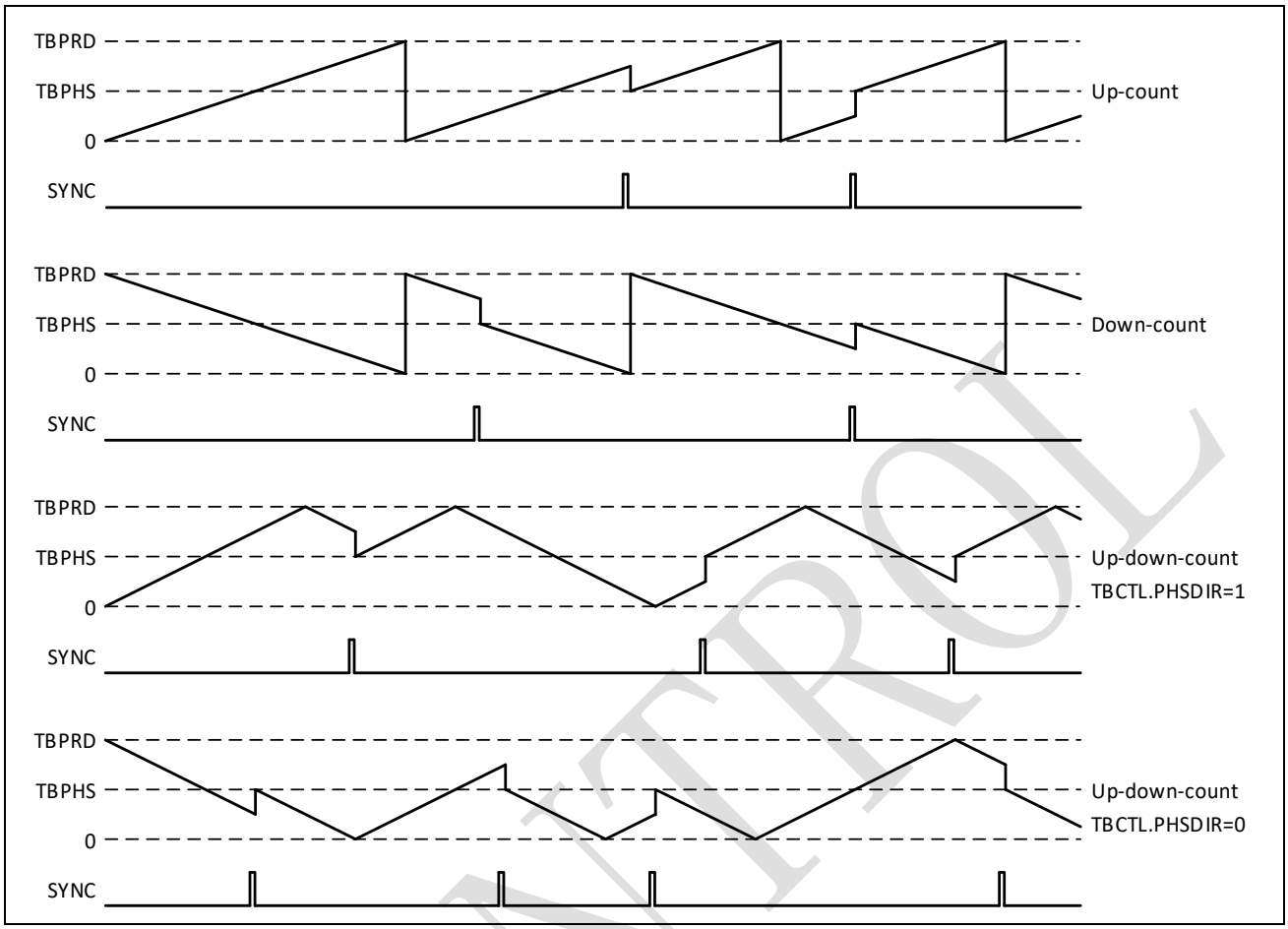


Shadow scheme is provided for TBPRD register in the SPD1148. In this mode, write to the TBPRD register will not immediately change the active value. Instead, the active value of TBPRD is updated from the shadow register when TBCNT=0 or TBCNT=TBPRD as defined by TBCTL.TBPRDLOAD.

While TBCTL.PHSEN=1, phase synchronization is enabled for the TBCNT. In this case, whenever there is a synchronization event, the TBCNT will load the value in time-base phase (TBPHS) register. A timing diagram is given in Figure 10-5. For up-count and down-count mode, the counter keeps the direction after synchronization. For up-down-count mode, the direction is defined by PHSDIR bit in time-base control (TBCTL) register. As shown in Figure 10-3, the synchronization event can be the external input PWMSYNCI signal, the digital compare synchronization A and B, local software forced synchronization, and the global software forced synchronization.

The phase synchronization enables the PWM module to be automatically synchronized to the time base of another PWM module. While TBCTL.PHSEN=0, the PWM will ignore the synchronization input pulse. However, the pulse will go through to PWMSYNCO and used to synchronize other PWM modules.

Figure 10-5: Time-base phase synchronization waveform



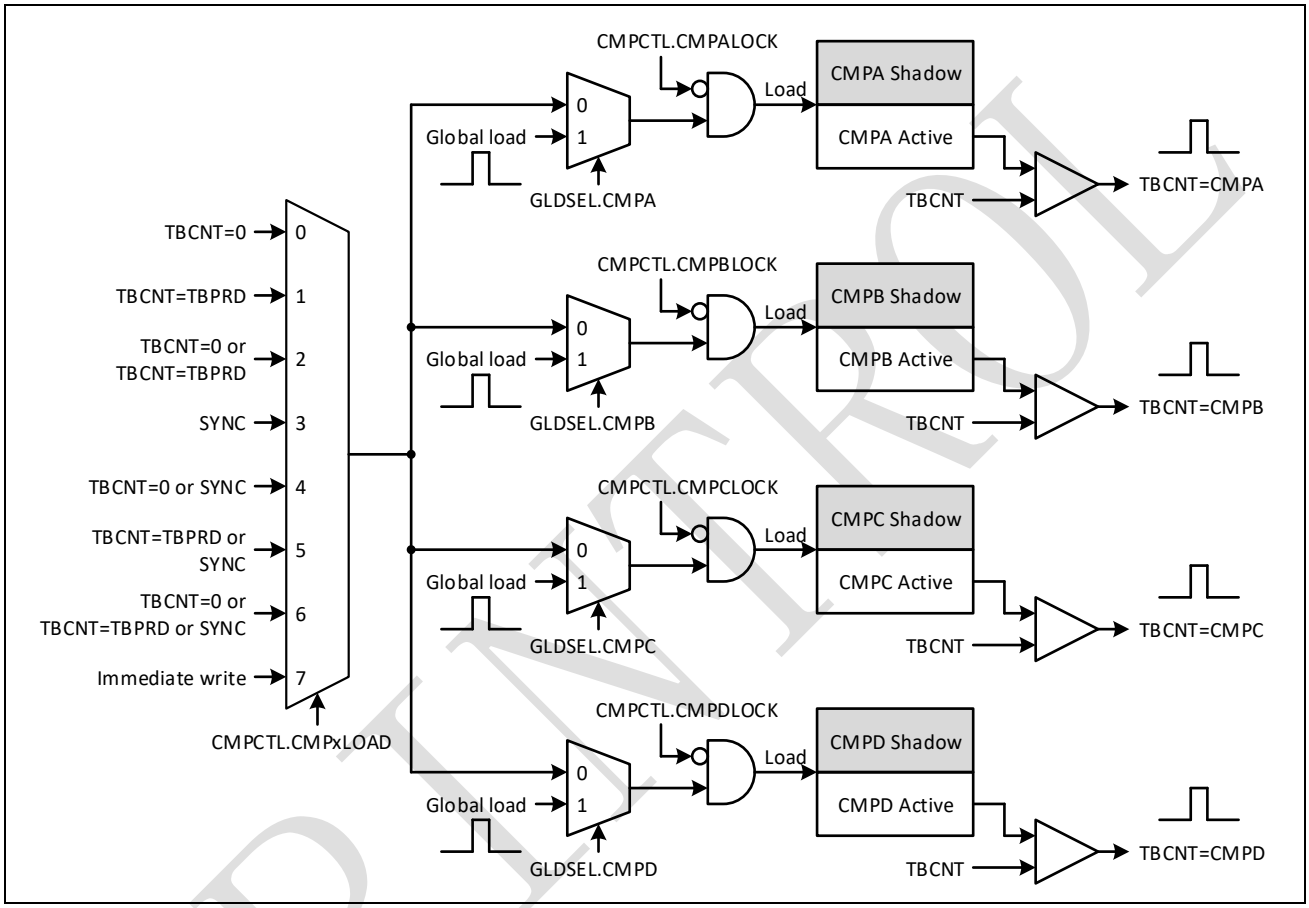
Note: It takes one PWM clock delay from the synchronization input to the actual synchronization action, results in one time-base cycle delay if TBCTL.TBDIVBIN=0 and TBCTL.TBDIVLIN=0. Please take this into account when setting the TBPHS register.

The register FRCSYNC in PWMCFG module provides a more flexible and convenient synchronization among the PWMs.

10.3 Counter Compare (CC) sub-module

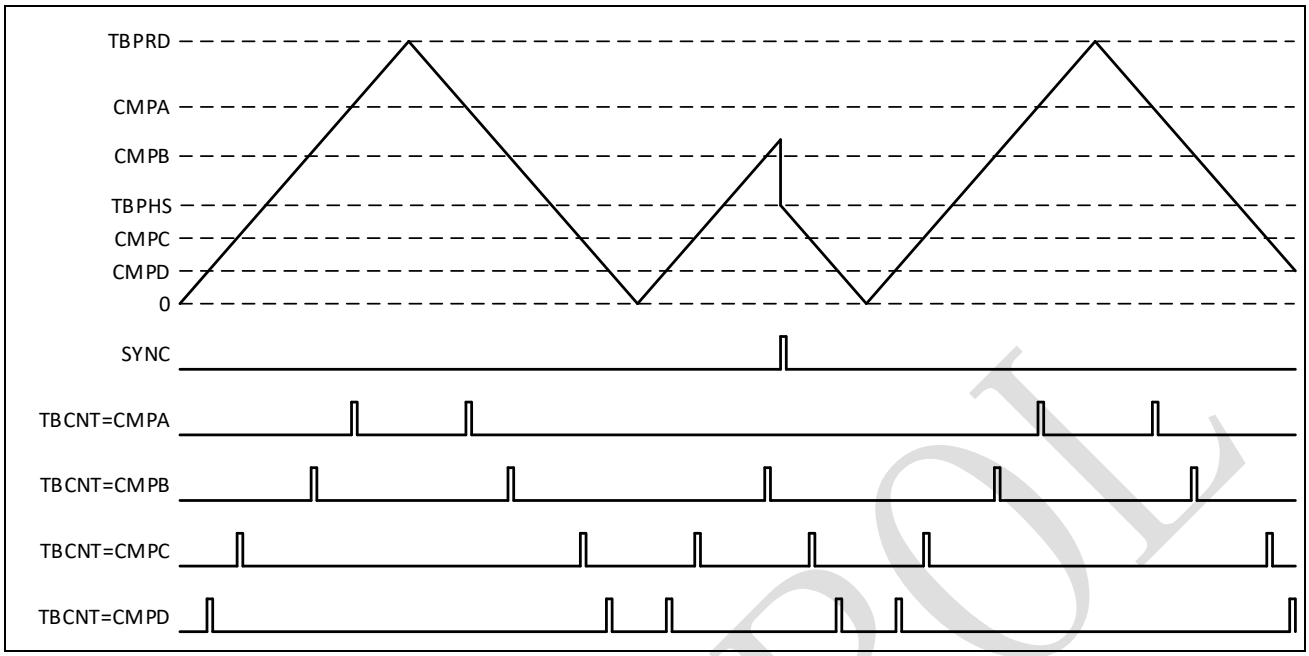
The counter-compare sub-module continuously compare the TBCNT to the counter-compare A (CMPA), counter-compare B (CMPB), counter-compare C (CMPC), and counter-compare D (CMPD) registers. When TBCNT is equal to the counter-compare register, a corresponding event will be generated.

Figure 10-6: Counter-Compare sub-module structure



The basic structure of counter-compare sub-module is shown in Figure 10-6. The counter-compare registers can be configured in shadow mode that the write to the register affects only the shadow register. The active values of the counter-compare register are updated from the shadow register upon the event defined by CMPxLOAD bit in counter compare control (CMPCTL) register. Global shadow load option can be enabled via CMPx bit in GLDSEL register as described in detail in Table 10-14 and Table 10-15. A sample waveform is shown in Figure 10-7.

Figure 10-7: Counter-Compare sub-module waveform



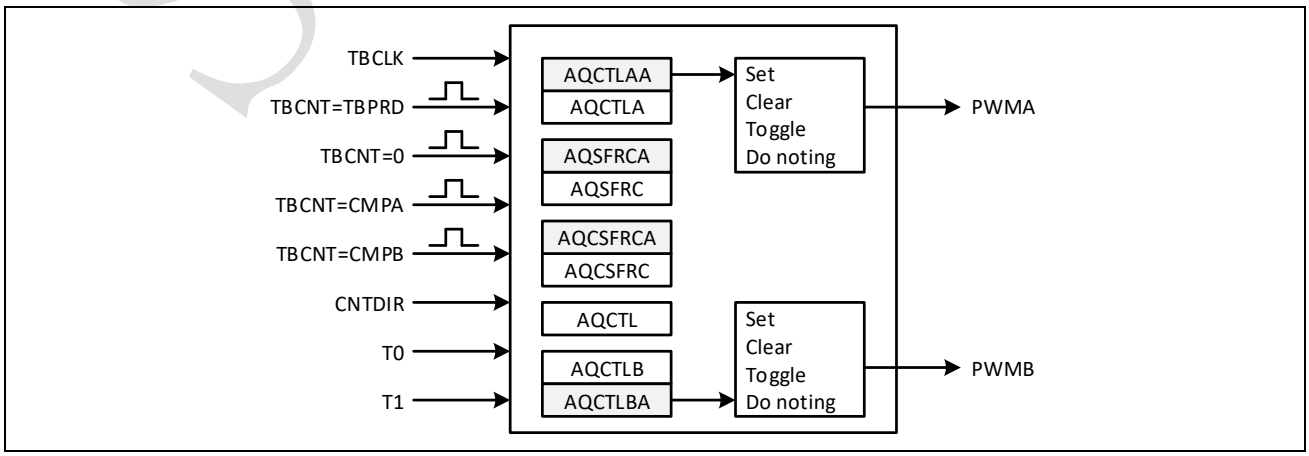
10.4 Action-Qualifier (AQ) sub-module

The action-qualifier sub-module determines which events are converted into various action types and produces the required switched waveforms at PWMA and PWMB outputs.

As shown in Figure 10-8, the events includes:

- TBCNT=TBPRD
- TBCNT=0
- TBCNT=CMPA
- TBCNT=CMPB
- T0/T1 events, which are based on comparator, trip-zone and sync input events
- Software forced event

Figure 10-8: Action-Qualifier sub-module inputs and outputs



The action-qualifier handles these events as well as their priority if they occur concurrently. Based on the events, it takes one of following actions as defined by AQCTLA and AQCTLB registers.

- Set output to a high level
- Clear output to a low level
- Toggle the output
- Do nothing

Shadow mode or the immediate write mode can be selected via AQCTL.AQCTLALOAD, AQCTL.AQCTLBLOAD and AQSFRC.CSFLOAD. Global load is also supported by setting GLDSEL.AQCTLA, GLDSEL.AQCTLB and GLDSEL.AQCSFRC. It is similar to the shadow for CMPx registers as shown in Figure 10-6.

Table 10-1 through Table 10-3 list the priority of action-qualifier in different count modes.

Table 10-1: Action-Qualifier event priority for Up-Count mode

Priority Level	Event
1(Highest)	Software forced event
2	TBCNT=TBPRD (PRD)
3	T0 on up-count (T0U)
4	T1 on up-count (T1U)
5	TBCNT=CMPB on up-count (CBU)
6	TBCNT=CMPA on up-count (CAU)
7 (Lowest)	TBCNT=0 (ZRO)

Table 10-2: Action-Qualifier Event Priority for Down-Count Mode

Priority Level	Event
1(Highest)	Software forced event
2	TBCNT=0 (ZRO)
3	T0 on down-count (T0D)
4	T1 on down-count (T1D)
5	TBCNT=CMPB on down-count (CBD)
6	TBCNT=CMPA on down-count (CAD)
7 (Lowest)	TBCNT=TBPRD (PRD)

Table 10-3: Action-Qualifier Event Priority for Up-Down-Count Mode

Priority Level	Event	
	When increase from 0 to TBPRD-1	When decrease from TBPRD to 1
1(Highest)	Software forced event	Software forced event
2	T0 on up-count (T0U)	T0 on down-count (T0D)

3	T1 on up-count (T1U)	T1 on down-count (T1D)
4	TBCNT=CMPB on up-count (CBU)	TBCNT=CMPB on down-count (CBD)
5	TBCNT=CMPA on up-count (CAU)	TBCNT=CMPA on down-count (CAD)
6 (Lowest)	TBCNT=0 (ZRO)	TBCNT=TBPRD (PRD)

Figure 10-9 and Figure 10-10 demonstrate timing diagrams for the symmetric and asymmetric scenarios of the action qualifier.

Figure 10-9: Up-Down-Count, dual edge symmetric waveform of Action-Qualifier

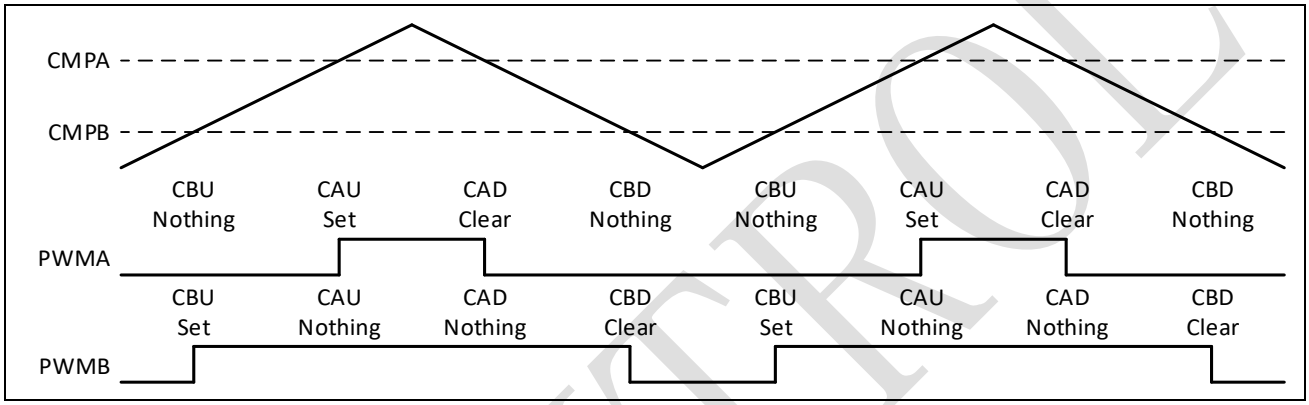
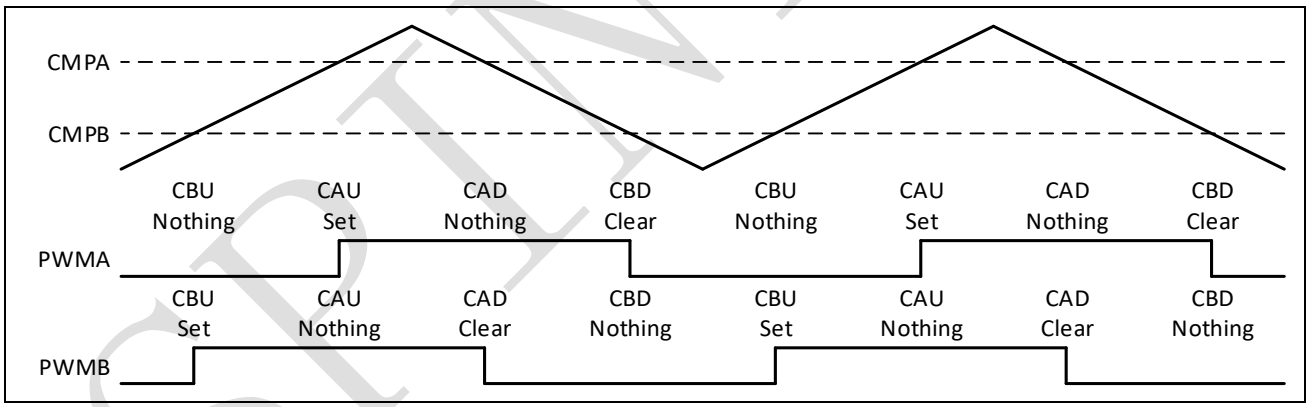


Figure 10-10: Up-Down-Count, dual edge asymmetric waveform of Action-Qualifier



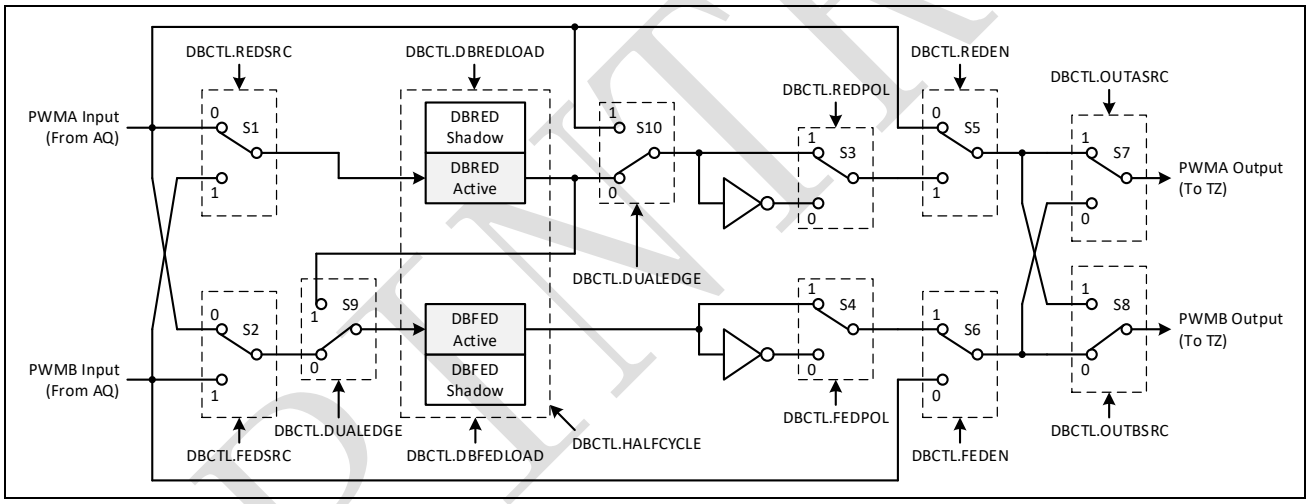
10.5 Dead-Band (DB) sub-module

The dead-band sub-module deals with the signals from the action qualifier to further shaping the waveform with independent and programmable rising-edge and falling-edge delay as well as polarity control.

The key functions of the module includes:

- Generate signal pairs output (PWMA and PWMB) with dead-band relationship from a single PWMA input
- Add programmable delay to rising edge
- Add programmable delay to falling edge
- Support half clock cycle delay
- Adjust the polarity of the output signal
- Bypass the dead-band signal shaping and directly feed through the signals from the action qualifier

Figure 10-11: Dead-Band sub-module structure



Basic structure of the dead-band sub-module is given in Figure 10-11. The REDSRC and FEDSRC bits in dead-band control (DBCTL) register determines the input signal source of the rising edge delay and falling edge delay. All combinations to control the switches from S3 to S10 are supported. Taking DBCTL.REDSRC=0 and DBCTL.FEDSRC=0 for example, when PWMA is configured as the input for both rising-edge and falling-edge delay, the typical operating modes are listed in Table 10-4.

As a special case, dual-edge delay can be achieved by setting the DUALEDGE bit in DBCTL register, when both rising-edge and falling-edge are applied to the same path. This is useful to generate a waveform pair with the same period and duty-cycle but specified phase shift between each other.

Table 10-4: Typical Dead-Band operating modes

Mode	Mode Description	REDPOL S3	FEDPOL S4	REDEN S5	FEDEN S6
1	Both PWMA and PWMB pass through	X	X	0	0
2	Active high complementary	1	0	1	1
3	Active low complementary	0	1	1	1
4	Active high	0	0	1	1
5	Active low	1	1	1	1
6	PWMA output = PWMA input with no delay PWMB output = PWMA input with falling-edge delay	0 or 1	0 or 1	0	1
7	PWMA output = PWMA input with rising-edge delay PWMB output = PWMA input with no delay	0 or 1	0 or 1	1	0

Figure 10-12 illustrates the waveform of typical dead-band sub-module.

Figure 10-12: Typical Dead-Band waveform

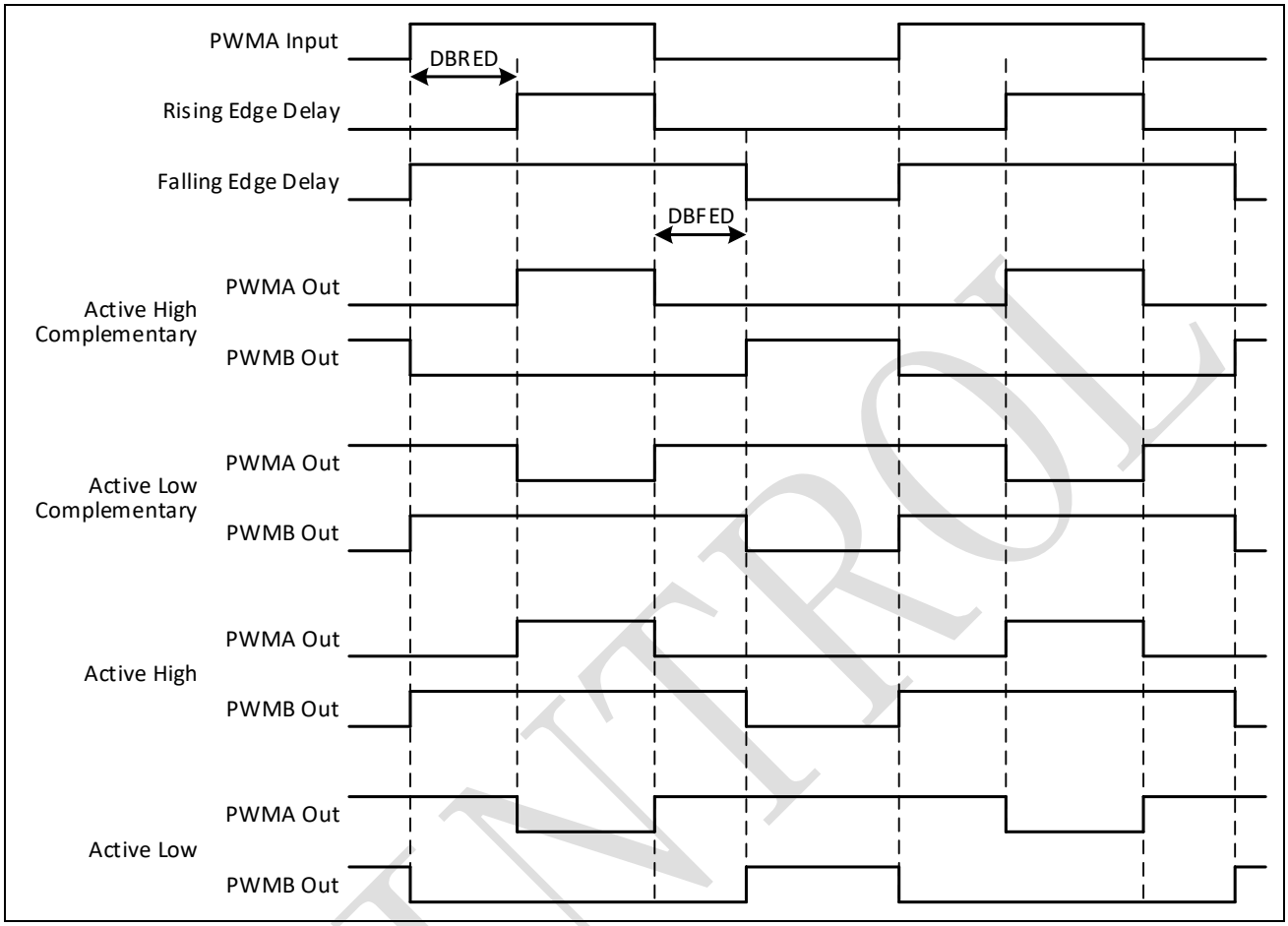
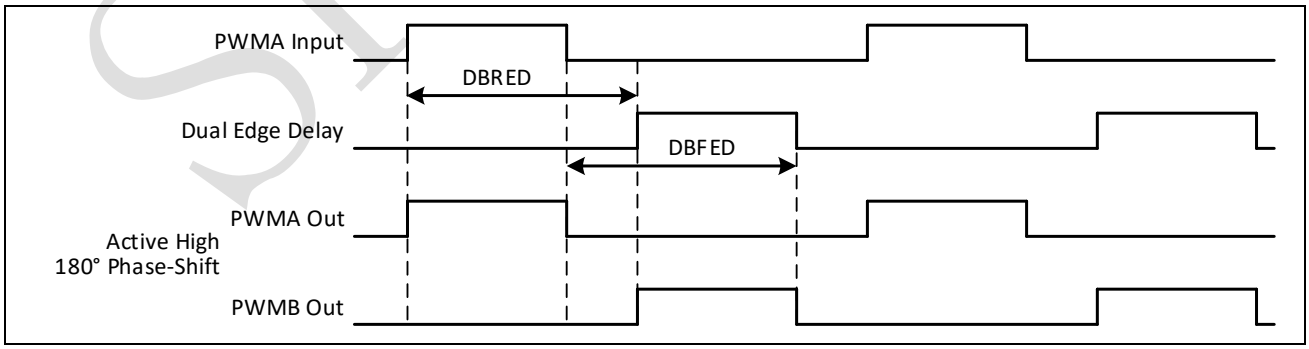


Figure 10-13 illustrates the case to generate waveform with 180° phase shift by enable dual-edge delay and setting both DBRED and DBFED to half the waveform period.

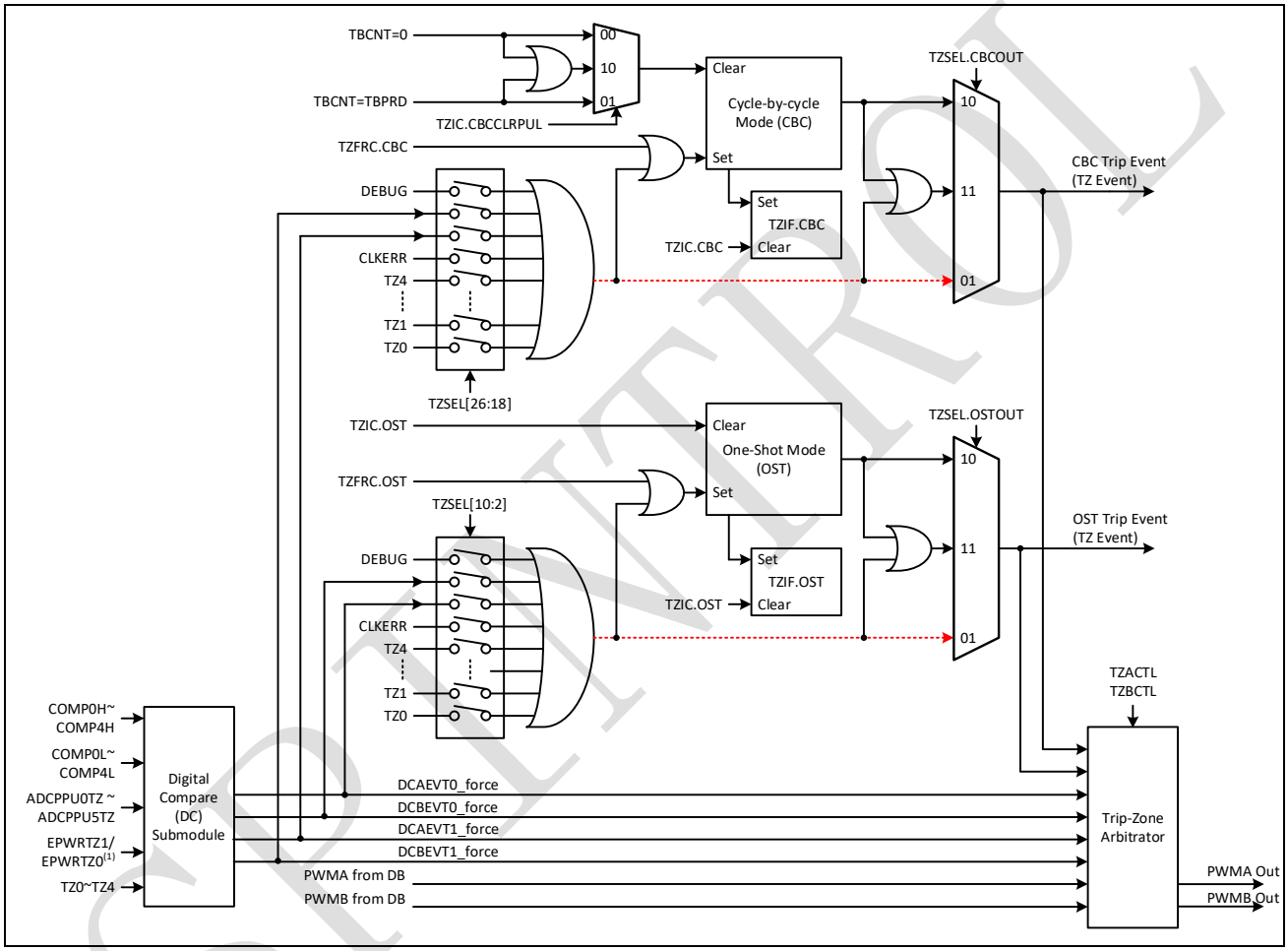
Figure 10-13: Dual-edge delay Dead-Band waveform



10.6 Trip-Zone (TZ) sub-module

The trip-zone sub-module determines the PWM outputs when external fault or trip condition occurs. As shown in Figure 10-14, the input can be from the TZ signal, the clock error event from the PLL unlock detection or the CLKDET module, the debug signal from the core, or the digital compare (DC) output. The five TZ signals (TZ0~TZ4) are sourced from the GPIO multiplexer with programmable signal polarity according to the TZ#SRCCTL registers in PWMCFG module. Software-forced tripping is also supported and interrupt can be generated on any trip-zone input.

Figure 10-14: Trip-Zone sub-module logic



[1] In SPD1148, EPWR module is left unconnected and EPWRTZ0/EPWRTZ1 are always 0; In SPD1148, EPWRTZ0 is always 0.

Upon a fault condition, the PWMA and PMWB outputs can be forced to one of the following:

- High
- Low
- High-impedance
- No action

The event actions on PWMA output is controlled by register TZACTL, and the priority of these event actions is TZU > TZD > DCAEVT0U > DCAEVT0D > DCAEVT1U > DCAEVT1D. It means that when both high priority event and low priority event happen, the output is determined by the configuration of

the high priority event. For example, when both TZU and DCAEVT1U events happen, TZU register field determines the PWM output while DCAEVT1U field is ignored.

Both one-shot trip (OST) and cycle-by-cycle (CBC) trip are supported. The one-shot trip is typically used for major over-current or short circuits condition while the cycle-by-cycle trip is usually used for current limiting operation.

When a one-shot trip event occurs, the action specified in the TZACTL and TZBCTL registers is immediately taken on the PWMA and PWMB outputs. Meanwhile, the one-shot trip event flag (TZIF.OST) is set and a PWM_TZINT interrupt is generated if it is enabled in trip-zone enable interrupt (TZIE) register. The TZIF.OST flag must be manually cleared by writing 1 to the TZIC.OST bit.

When a cycle-by-cycle trip event occurs, the action specified in the TZACTL and TZBCTL registers is immediately taken on the PWMA and PWMB outputs. Meanwhile, the cycle-by-cycle trip event flag (TZIF.CBC) is set and a PWM_TZINT interrupt is generated if it is enabled in the TZIE register. The specified condition is automatically cleared if the trip event is no longer present when the TBCNT reaches zero or TBPRD as specified by TZIC.CBCCLRPUL. However, the TZIF.CBC flag bit remains set until it is manually cleared by writing 1 to the TZIC.CBC bit. If the cycle-by-cycle trip event is still present when the TZIF.CBC flag is cleared, it will be set again immediately.

Note: The asynchronous logic path shown as dashed line in Figure 10-14 is used to observe the original trip-zone event signal when TZSEL.OSTOUT or TZSEL.CBCOUT is set to 1. In real application, the latched path should be enabled to implement the one-shot or cycle-by-cycle trip function as described above. i.e. TZSEL.OSTOUT and TZSEL.CBCOUT should be 2 or 3.

The digital compare DCAEVT0/1 or DCBEVT0/1 event is generated based on the DCAH/DCAL and DCBH/DCBL bits of trip-zone digital compare selection (TZDCSEL) register. When a digital compare event occurs, the action specified in the TZACTL and TZBCTL registers is immediately taken on the PWMA and PWMB outputs. Meanwhile, the DC trip event flag (TZIF.DCAEVT0/1 and TZIF.DCBEVT0/1) is set and a PWM_TZINT interrupt is generated if it is enabled in the TZIE register. The specified condition is automatically cleared when the DC trip event is no longer present. However, the flag bits in the TZIF register remains set until it is manually cleared by writing 1 to the corresponding bits in the TZIC register. If the DC trip event is still present when the TZIF bit is cleared, it will be set again immediately.

Figure 10-15 illustrates the trip-zone interrupt logic.

Figure 10-15: Trip-Zone interrupt generation

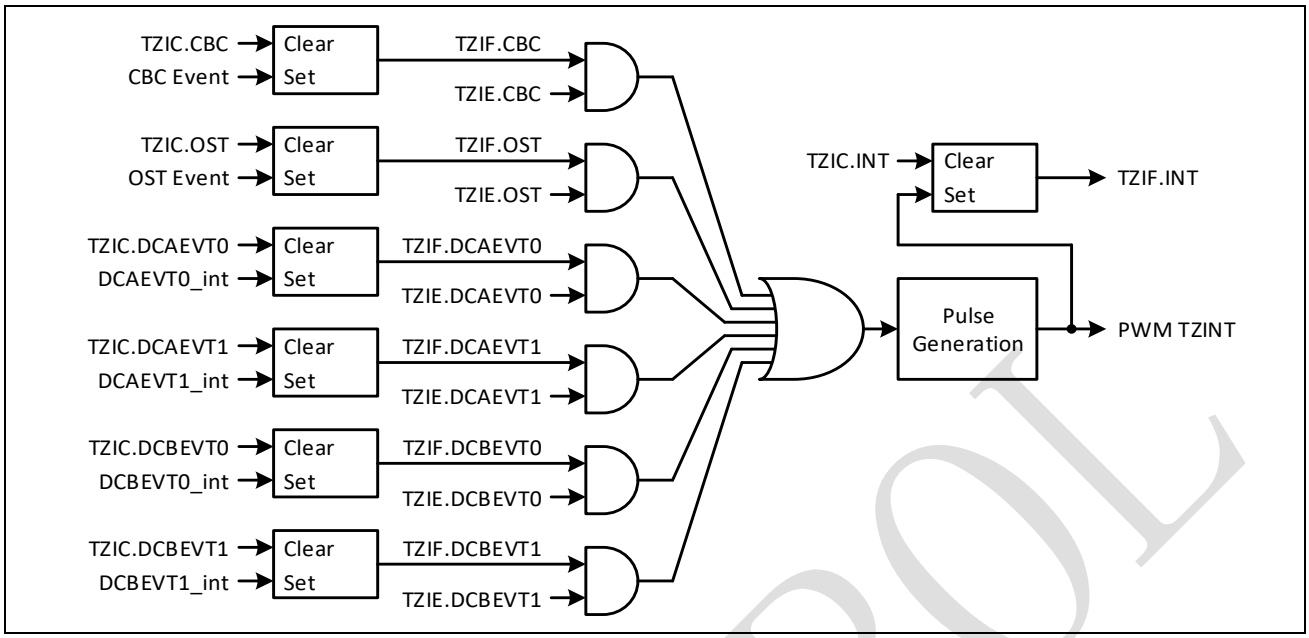


Table 10-5 lists the priority of events that effect the final PWM outputs.

Table 10-5: Priority of events that effect the final PWM outputs

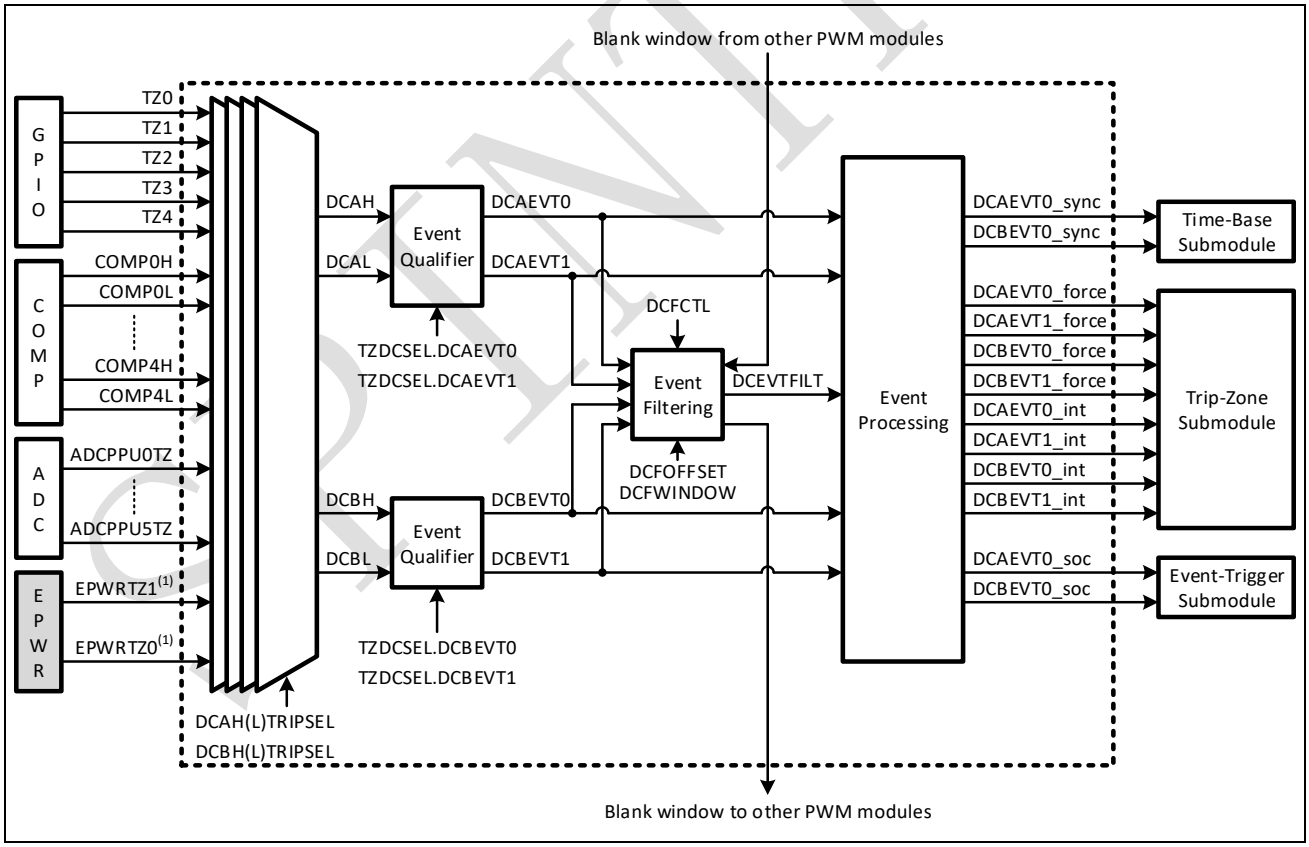
Priority Level	Event
1(Highest)	Trip event
2	Continuous software force via AQCSFRC register
3	Dead-Band control logic
4	One-time software force via AQSFRC register
5 (Lowest)	Action-Qualifier control logic

10.7 Digital Compare (DC) sub-module

The digital compare sub-module takes the analog comparator (COMP) module outputs, the ADC post-processing unit (ADCPPU) module trip-zone outputs, and TZ0, TZ1, TZ2, TZ3 and TZ4 inputs to generate digital compare A high/low (DCAH, DCAL) and digital compare B high/low (DCBH, DCBL) signals. As shown in Figure 10-16, these signals generates raw DCAEVT0/1 and DCBEVT0/1 events, which can be either filtered or directly fed to the event processing logic, so as to generate signals for time-base, trip-zone and event-trigger sub-modules.

The raw DCAEVT0/1 and DCBEVT0/1 events can be filtered to remove noise by inserting a blank window to gate the event for a certain time as shown in Figure 10-17. Upon the TBCNT=TBPRD or TBCNT=0 pulse, the offset counter loads the value from the DCFOFFSET register and starts to count down. When the counter expires, the blank window starts and the duration is certain TBCLK clocks determined by DCFWINDOW register. During the blank window, all events are ignored. Before the window starts and after the window ends, events can generate SOC, SYNC, interrupt and force signals as before. To take into account the cross-talk from/to other blocks in real application, the generated blank windows from PWM modules can be enabled to filter local raw DC events. The polarity of the blank window can be inverted by set BLANKINV bit of DCFCTL register. As shown in the timing diagram in Figure 10-18, the new blank window will immediately start even if the old one does not expire.

Figure 10-16: Digital Compare sub-module structure



[1] In SPD1148, EPWR module is left unconnected and EPWRTZ0/EPWRTZ1 are always 0; In SPD1148, EPWRTZ0 is always 0.

Figure 10-17: Event filtering logic

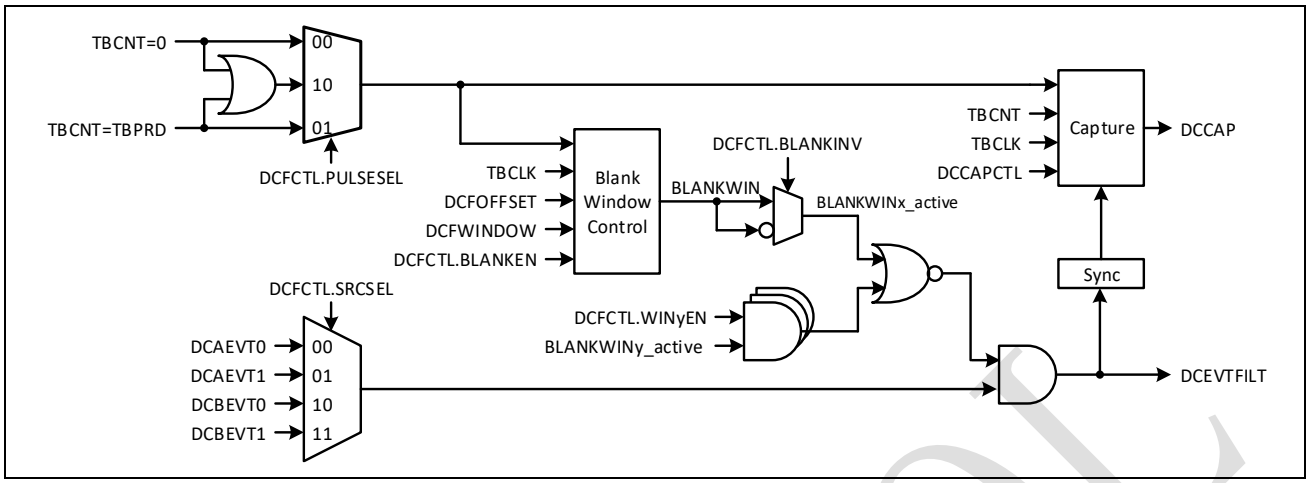
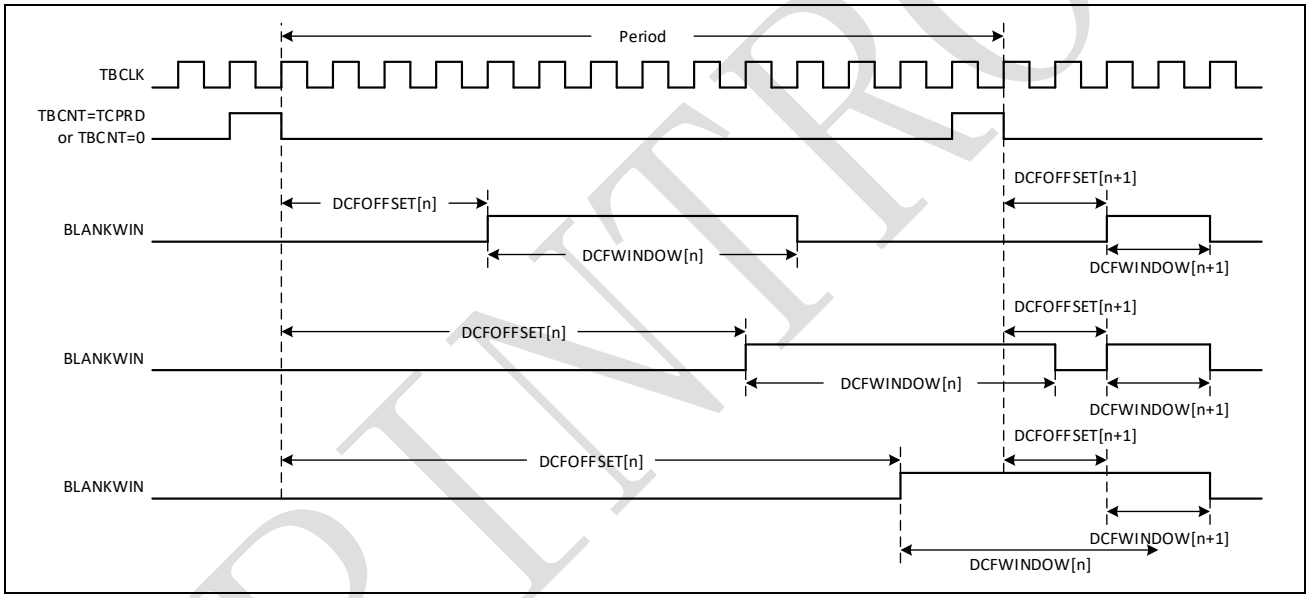


Figure 10-18: Blank window timing diagram



As shown in Figure 10-19, the raw DCAEVT0, DCAEVT1, DCBEVT0, DCBEVT1 and the filtered DCEVTFILT pass through the event processing block and generate the final DCAEVT0, DCAEVT1, DCBEVT0 and DCBEVT1 events according to EVT0SRCSEL and EVT1SRCSEL fields in DCACTL and DCBCTL registers. Based on these events, further signals can be generated as below:

- Force signals for trip-zone sub-module

  DCAEVT0_force, DCAEVT1_force, DCBEVT0_force, DCBEVT1_force

- Synchronous signal for time-base sub-module

  DCAEVT0_sync, DCBEVT0_sync

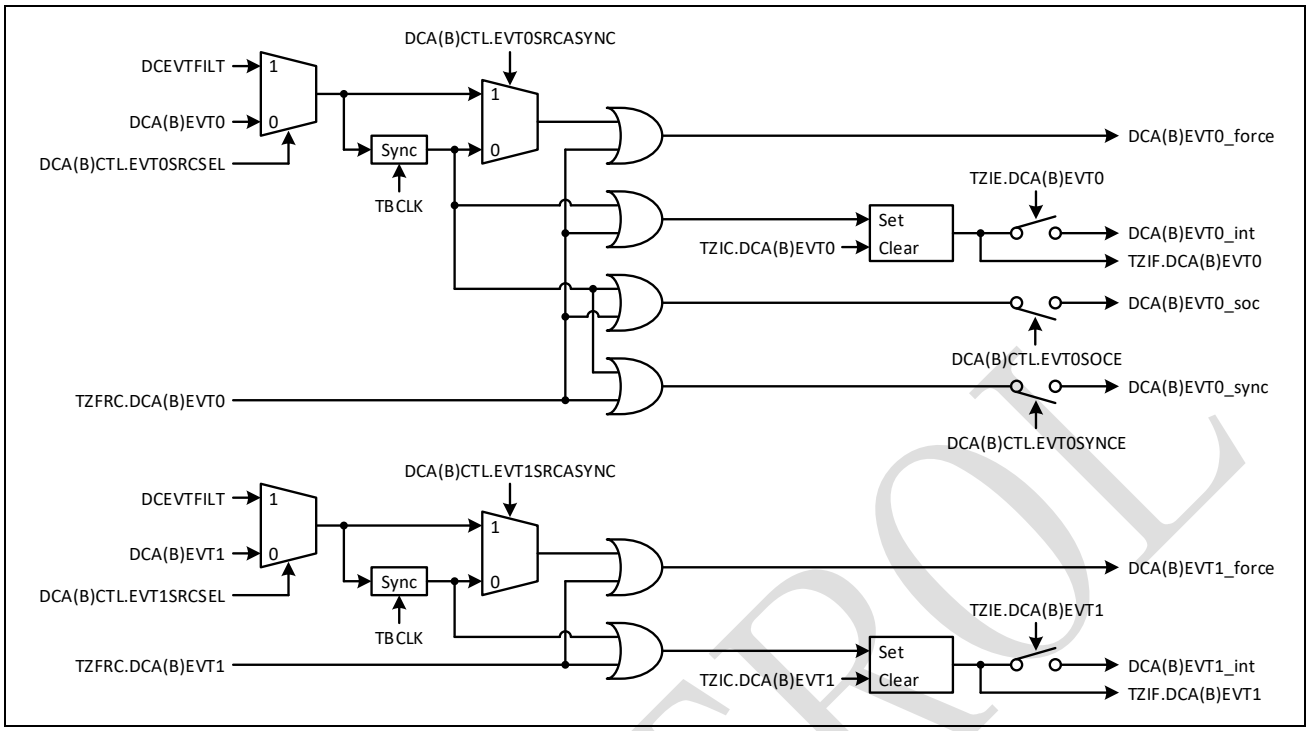
- ADC start of conversion

  DCAEVT0_soc, DCBEVT0_soc

- Interrupt

  DCAEVT0_int, DCAEVT1_int, DCBEVT0_int, DCBEVT1_int

Figure 10-19: Event processing



Below is an example in the scenario that COMP0~COMP2 are used for shut-current monitoring and generate filtered DC trip-zone events to control both PWMxA and PWMxB outputs.

- DCAH is set as the logic OR of COMP0H, COMP0L, COMP1H, COMP1L, COMP2H, and COMP2L via DCAHTRIPSEL register.
- Use DCAH=high to generate the raw DCAEVT0 signal via setting TZDCSEL.DCAEVT0=4.
- Use the raw DCAEVT0 as the DC filter input via setting DCFCTL.SRCSEL=0.
- Use the filtered signal DCEVTFILT as the final DCAEVT0 event via setting DCACTL.EVT0SRCSEL=1.
- Use the filtered signal DCEVTFILT as the final DCBEVT0 event via setting DCBCTL.EVT0SRCSEL=1.
- Configure the PWMxA output action upon final DCAEVT0 trip-zone event via DCAEVT0U and DCAEVT0D fields in TZACTL register.
- Configure the PWMxB output action upon final DCBEVT0 trip-zone event via DCBEVT0U and DCBEVT0D fields in TZBCTL register.

10.8 Event-Trigger (ET) sub-module

The event-trigger sub-module manages the events generated by the time-base sub-module, the counter-compare sub-module, and the digital-compare sub-module to generate an interrupt to the CPU and/or a start-of-conversion (SOC) pulse to the ADC when a selected event occurs.

Figure 10-20: Event-Trigger sub-module structure

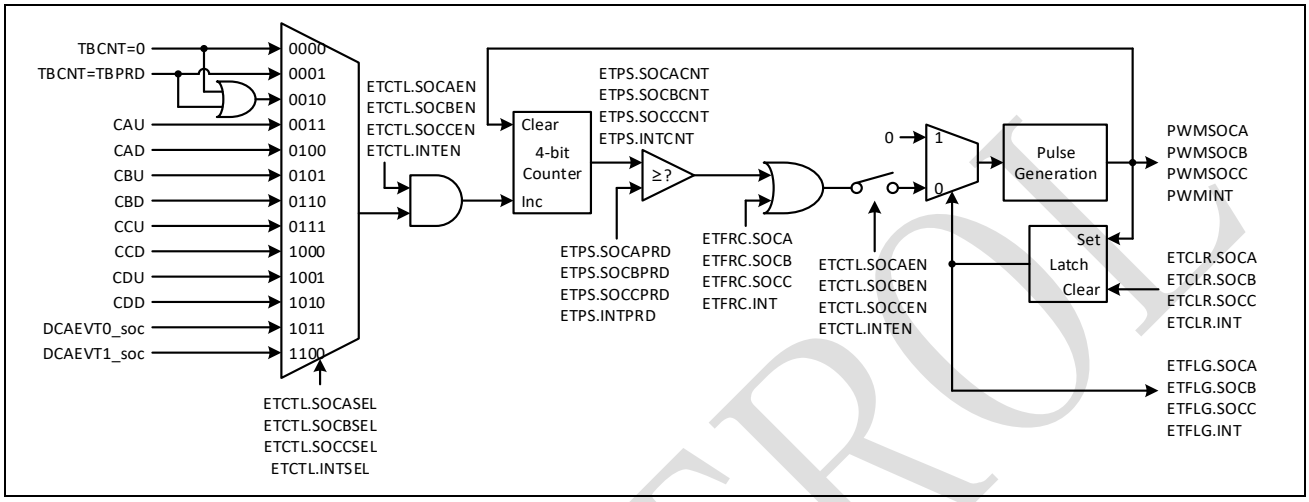


Figure 10-20 illustrates the basic structure of the event-trigger sub-module. Software forced SOCA, SOCB, SOCC and INT pulses are supported in the SPD1148. By using a 4-bit counter, the pulse can be generated in following scenario:

- Never generate
- Generate a pulse on every event
- Generate a pulse on every second event
- ……
- Generate a pulse up to every fifteenth event

10.9 Global reload

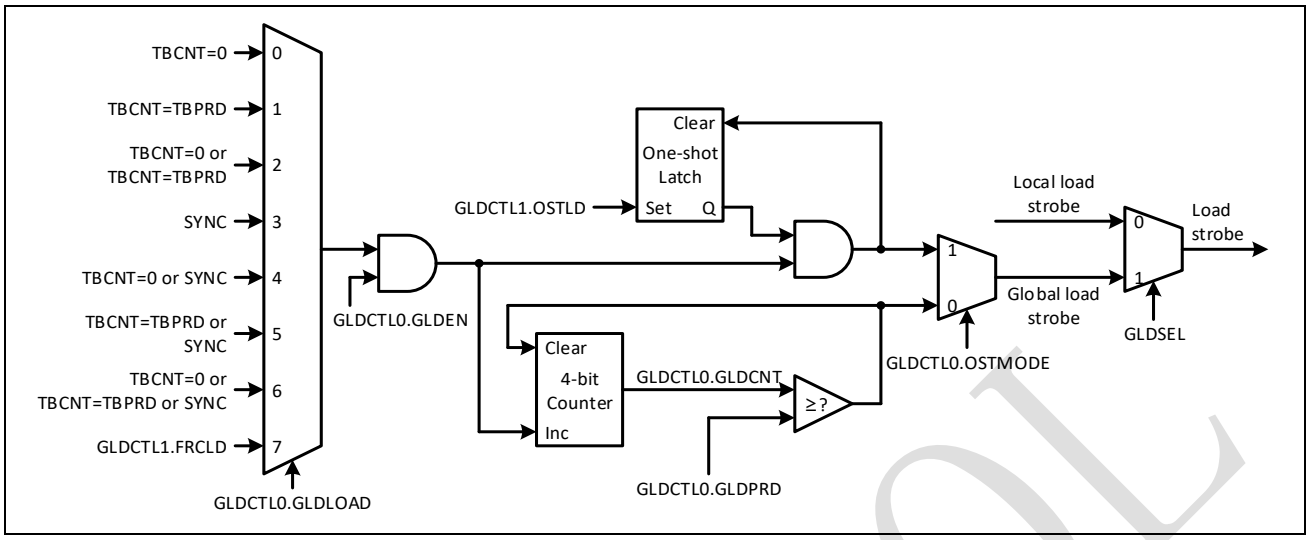
The content transfer from the shadow register to the active register can be done upon the global load strobe as shown in Figure 10-21. This feature is enabled by setting the GLDEN bit of GLDCTL0 register to 1. And the bits in GLDSEL register decide whether the corresponding active register is updated upon the global load strobe or the local load strobe.

When GLDCTL0.OSTMODE=0, the pre-scaler is applied so that the global load strobe is generated every GLDCTL0.GLDPRD occurrences of the global reload events.

When GLDCTL0.OSTMODE=1, one-shot reload mode is enabled so that only the first global reload event will generate the global load strobe after GLDCTL1.OSTLD is setting to 1. Once the global load strobe is generated, the GLDCTL1.OSTLD is cleared to 0 until it is manually set to 1 again. Global reload events during GLDCTL1.OSTLD=0 do not generate any global load strobe.

Software forced global reload can be enabled via writing 1 to GLDCTL1.FRCLD.

Figure 10-21: Global reload structure



10.10 Simultaneous write

In some applications, there is a need of simultaneous writes to the registers of same name in different PWMs. SPD1148 provides a PWM link feature by PWMLINK register, so that write to the register of another PWM module can also update the value of the same register in current PWM module. The registers supporting this feature are TBPRD, CMPA, CMPB, CMPC, CMPD, DBRED, DBFED and GLDCTL1.

10.11 Registers

10.11.1 PWM register map

Table 10-6: PWM Module Base Address

Peripheral Module	Base Address
PWM0	0x4000 9000
PWM1	0x4000 9100
PWM2	0x4000 9200
PWM3	0x4000 9300
PWM4	0x4000 9400
PWM5	0x4000 9500

Table 10-7: PWM Register Map

Register	Offset	Description	Reset Value
SHADOWSTS	0x0	Shadow Status Register	0x00000000
GLDCTL0*	0x4	Global Shadow to Active Load Control Register 0	0x00000000
GLDCTL1	0x8	Global Shadow to Active Load Control Register 1	0x00000000
GLDSEL*	0xC	Global Shadow to Active Load Select Register	0x00000000
PWMLINK*	0x10	PWM Link Control Register	0xFFFFFFFF
TBCTL*	0x14	Time-Base Control Register	0x00000006
TBPRD	0x18	Time-Base Period Register	0x00000000
TBPRDA	0x1C	Time-Base Period Active Register	0x00000000
TBPHS	0x20	Time-Base Phase Register	0x00000000
TBCNT	0x24	Time-Base Counter Register	0x00000000
TBSTS	0x28	Time-Base Status Register	0x00000100
TBSTSCLR	0x2C	Time-Base Status Clear Register	0x00000000
CMPCTL*	0x30	Counter-Compare Control Register	0x00000000
CMPA	0x34	Counter-Compare A Threshold Register	0x00000000
CMPAA	0x38	Counter-Compare A Threshold Active Register	0x00000000
CMPB	0x3C	Counter-Compare B Threshold Register	0x00000000
CMPBA	0x40	Counter-Compare B Threshold Active Register	0x00000000
CMPC	0x44	Counter-Compare C Threshold Register	0x00000000
CMPCA	0x48	Counter-Compare C Threshold Active Register	0x00000000

Register	Offset	Description	Reset Value
CMPD	0x4C	Counter-Compare D Threshold Register	0x00000000
CMPDA	0x50	Counter-Compare D Threshold Active Register	0x00000000
AQCTL*	0x54	Action-Qualifier Control Register	0x00000000
AQCTLA	0x58	Action-Qualifier Output A Control Register	0x00000000
AQCTLAA	0x5C	Action-Qualifier Output A Control Active Register	0x00000000
AQCTLB	0x60	Action-Qualifier Output B Control Register	0x00000000
AQCTLBA	0x64	Action-Qualifier Output B Control Active Register	0x00000000
AQSFRC	0x68	Action-Qualifier Software Force Register	0x00000000
AQCSFRC	0x6C	Action-Qualifier Continuous Software Force Register	0x00000000
AQCSFRCA	0x70	Action-Qualifier Continuous Software Force Active Register	0x00000000
DBCTL*	0x74	Dead-Band Generator Control Register	0x00000070
DBCTLA	0x78	Dead-Band Generator Control Active Register	0x00000070
DBRED	0x7C	Dead-Band Generator Rising Edge Delay Register	0x00000000
DBREDA	0x80	Dead-Band Generator Rising Edge Delay Active Register	0x00000000
DBFED	0x84	Dead-Band Generator Falling Edge Delay Register	0x00000000
DBFEDA	0x88	Dead-Band Generator Falling Edge Delay Active Register	0x00000000
TZSEL*	0x8C	Trip-Zone Event Select Register	0x00030003
TZSTS	0x90	Trip-Zone Status Register	0x00000000
TZSTSCLR	0x94	Trip-Zone Status Clear Register	0x00000000
TZDCSEL*	0x98	Trip-Zone Digital Compare Event Select Register	0x00000000
TZACTL	0x9C	Trip-Zone Output A Control Register	0x00000000
TZBCTL	0xA0	Trip-Zone Ouput B Control Register	0x00000000
TZIF	0xA4	Trip-Zone Flag Register	0x00000000
TZIC	0xA8	Trip-Zone Clear Register	0x00000000
TZIE*	0xAC	Trip-Zone Interrupt Enable Register	0x00000000
TZFRC	0xB0	Trip-Zone Force Register	0x00000000
DCALTRIPSEL*	0xB4	Digital Compare AL Trip Select Register	0x00000000
DCAHTRIPSEL*	0xB8	Digital Compare AH Trip Select Register	0x00000000
DCBLTRIPSEL*	0xBC	Digital Compare BL Trip Select Register	0x00000000

Register	Offset	Description	Reset Value
DCBHTRIPSEL*	0xC0	Digital Compare BH Trip Select Register	0x00000000
DCACTL*	0xC4	Digital Compare A Control Register	0x00000000
DCBCTL*	0xC8	Digital Compare B Control Register	0x00000000
DCFCTL*	0xCC	Digital Compare Filter Register	0x00000010
DCFOFFSET	0xD0	Digital Compare Filter Offset Register	0x00000000
DCFOFFSETCNT	0xD4	Digital Compare Filter Offset Counter Register	0x00000000
DCFWINDOW	0xD8	Digital Compare Filter Window Register	0x00000000
DCFWINDOWCNT	0xDC	Digital Compare Filter Window Counter Register	0x00000000
DCCAPCTL	0xE0	Digital Compare Capture Control Register	0x00000000
DCCAP	0xE4	Digital Compare Counter Capture Register	0x00000000
ETCTL*	0xE8	Event-Trigger Control Register	0x0007B16F
ETPS	0xEC	Event-Trigger Prescale Register	0x00000000
ETFLG	0xF0	Event-Trigger Flag Register	0x00000000
ETCLR	0xF4	Event-Trigger Clear Register	0x00000000
ETFRC	0xF8	Event-Trigger Force Register	0x00000000
PWMREGKEY	0xFC	PWM Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the PWMREGKEY=0x1ACCE551.

10.11.2 PWM registers

Table 10-8: Shadow Status Register (SHADOWSTS) Layout

SHADOWSTS (Shadow Status Register) Offset: 0x0 Default: 0x00000000 Access: PWM0 -> SHADOWSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					DBFED	DBRED	DBCTL
7	6	5	4	3	2	1	0
AQCSFRC	AQCTLB	AQCTLA	CMPD	CMPC	CMPB	CMPA	TBPRD

Table 10-9: Shadow Status Register (SHADOWSTS) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	DBFED	RO	0x0	DBFED shadow status 0: A load from DBFED to DBFEDA clears this bit 1: Write to DBFED set this bit and overwrite current value in DBFED
9	DBRED	RO	0x0	DBRED shadow status 0: A load from DBRED to DBREDA clears this bit 1: Write to DBRED set this bit and overwrite current value in DBRED
8	DBCTL	RO	0x0	DBCTL shadow status 0: A load from DBCTL to DBCTLA clears this bit 1: Write to DBCTL set this bit and overwrite current value in DBCTL
7	AQCSFRC	RO	0x0	AQCSFRC shadow status 0: A load from AQCSFRC to AQCSFRCA clears this bit 1: Write to AQCSFRC set this bit and overwrite current value in AQCSFRC
6	AQCTLB	RO	0x0	AQCTLB shadow status 0: A load from AQCTLB to AQCTLBA clears this bit 1: Write to AQCTLB set this bit and overwrite current value in AQCTLB
5	AQCTLA	RO	0x0	AQCTLA shadow status 0: A load from AQCTLA to AQCTLAA clears this bit 1: Write to AQCTLA set this bit and overwrite current value in AQCTLA

Bits	Field Name	Type	Reset	Description
4	CMPD	RO	0x0	CMPD shadow status 0: A load from CMPD to CMPDA clears this bit 1: Write to CMPD set this bit and overwrite current value in CMPD
3	CMPC	RO	0x0	CMPC shadow status 0: A load from CMPC to CMPCA clears this bit 1: Write to CMPC set this bit and overwrite current value in CMPC
2	CMPB	RO	0x0	CMPB shadow status 0: A load from CMPB to CMPBA clears this bit 1: Write to CMPB set this bit and overwrite current value in CMPB
1	CMPA	RO	0x0	CMPA shadow status 0: A load from CMPA to CMPAA clears this bit 1: Write to CMPA set this bit and overwrite current value in CMPA
0	TBPRD	RO	0x0	TBPRD shadow status 0: A load from TBPRD to TBPRDA clears this bit 1: Write to TBPRD set this bit and overwrite current value in TBPRD

Table 10-10: Global Shadow to Active Load Control Register 0 (GLDCTL0) Layout

GLDCTL0 (Global Shadow to Active Load Control Register 0) Offset: 0x4 Default: 0x00000000
Access: PWM0 -> GLDCTL0.all

31	30	29	28	27	26	25	24
RESERVED_31_13							
23	22	21	20	19	18	17	16
RESERVED_31_13							
15	14	13	12	11	10	9	8
RESERVED_31_13			OSTMODE	GLDCNT			
7	6	5	4	3	2	1	0
GLDPRD				GLDLOAD			GLDEN

Table 10-11: Global Shadow to Active Load Control Register 0 (GLDCTL0) Description

Bits	Field Name	Type	Reset	Description
31:13	RESERVED_31_13	RO	0x0	Reserved.
12	OSTMODE	RW	0x0	One-shot global load mode 0: Global strobe is continuously controlled by GLDPRD and GLDCNT 1: Global strobe is blocked until GLDCTL1[OSTLOAD] is written with 1

Bits	Field Name	Type	Reset	Description
11:8	GLDCNT	RO	0x0	Global shadow to active load strobe counter It indicates how many selected global load events have occurred
7:4	GLDPRD	RW	0x0	Global shadow to active load strobe period No global load pulse will be generated if GLDPRD=0. Otherwise, the global load pulse is generated upon GLDCNT = GLDPRD. It has no effect if OSTMODE=1
3:1	GLDLOAD	RW	0x0	Global shadow to active load mode 000: Load from shadow to active on TBCNT=0 001: Load from shadow to active on TBCNT=TBPRD 010: Load from shadow to active on TBCNT=0 or TBCNT=TBPRD 011: Load from shadow to active on SYNC event 100: Load from shadow to active on SYNC event or TBCNT=0 101: Load from shadow to active on SYNC event or TBCNT=TBPRD 110: Load from shadow to active on SYNC event, TBCNT=0 or TBCNT=TBPRD 111: Load from shadow to active on software by writing 1 to GLDCTL1[FRCLOAD]
0	GLDEN	RW	0x0	Global shadow to active load enable 0: Shadow to active load is defined per register regardless of GLDSEL. i.e. CMPA load is always controlled by CMPCTL[CMPALOAD] 1: Shadow to active load is globally controlled

Table 10-12: Global Shadow to Active Load Control Register 1 (GLDCTL1) Layout

GLDCTL1 (Global Shadow to Active Load Control Register 1) Offset: 0x8 Default: 0x00000000 Access: PWM0 -> GLDCTL1.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						FRCLOAD	OSTLOAD

Table 10-13: Global Shadow to Active Load Control Register 1 (GLDCTL1) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	FRCLOAD	W1S	0x0	Global force shadow to active load 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 generates global load event to the event counter
0	OSTLOAD	W1S	0x0	Start the one-shot load pending 0: Write a 0 has no effect. Read a 0 indicates there is no pending one-shot load 1: Write a 1 initialize a one-shot load pending. Upon event defined by GLDCTL0[GLDLOAD], the global load happens and this bit is self-cleared to 0.

Table 10-14: Global Shadow to Active Load Select Register (GLDSEL) Layout

GLDSEL (Global Shadow to Active Load Select Register) Offset: 0xC Default: 0x00000000
Access: PWM0 -> GLDSEL.all

31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					DBFED	DBRED	DBCTL
7	6	5	4	3	2	1	0
AQCSFRC	AQCTLB	AQCTLA	CMPD	CMPC	CMPB	CMPA	TBPRD

Table 10-15: Global Shadow to Active Load Select Register (GLDSEL) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	DBFED	RW	0x0	DBFED global shadow to active load select 0: Shadow to active load is controlled by DBCTL[DBFEDLOAD] 1: Shadow to active load is controlled by GLDCTL
9	DBRED	RW	0x0	DBRED global shadow to active load select 0: Shadow to active load is controlled by DBCTL[DBREDLOAD] 1: Shadow to active load is controlled by GLDCTL

Bits	Field Name	Type	Reset	Description
8	DBCTL	RW	0x0	DBCTL global shadow to active load select 0: Shadow to active load is controlled by DBCTL[DBCTLLOAD] 1: Shadow to active load is controlled by GLDCTL
7	AQCSFRC	RW	0x0	AQCSFRC global shadow to active load select 0: Shadow to active load is controlled by AQCTL[AQCSFRCLOAD] 1: Shadow to active load is controlled by GLDCTL
6	AQCTLB	RW	0x0	AQCTLB global shadow to active load select 0: Shadow to active load is controlled by AQCTL[AQCTLBLOAD] 1: Shadow to active load is controlled by GLDCTL
5	AQCTLA	RW	0x0	AQCTLA global shadow to active load select 0: Shadow to active load is controlled by AQCTL[AQCTLALOAD] 1: Shadow to active load is controlled by GLDCTL
4	CMPD	RW	0x0	CMPD global shadow to active load select 0: Shadow to active load is controlled by CMPCTL[CMPDLOAD] 1: Shadow to active load is controlled by GLDCTL
3	CMPC	RW	0x0	CMPC global shadow to active load select 0: Shadow to active load is controlled by CMPCTL[CMPCLOAD] 1: Shadow to active load is controlled by GLDCTL
2	CMPB	RW	0x0	CMPB global shadow to active load select 0: Shadow to active load is controlled by CMPCTL[CMPBLOAD] 1: Shadow to active load is controlled by GLDCTL
1	CMPA	RW	0x0	CMPA global shadow to active load select 0: Shadow to active load is controlled by CMPCTL[CMPALOAD] 1: Shadow to active load is controlled by GLDCTL

Bits	Field Name	Type	Reset	Description
0	TBPRD	RW	0x0	TBPRD global shadow to active load select 0: Shadow to active load is controlled by TBCTL[TBPRDLOAD] 1: Shadow to active load is controlled by GLDCTL

Table 10-16: PWM Link Control Register (PWMLINK) Layout

PWMLINK (PWM Link Control Register) Offset: 0x10 Default: 0xFFFFFFFF Access: PWM0 -> PWMLINK.all							
31	30	29	28	27	26	25	24
GLDCTL1				DBFED			
23	22	21	20	19	18	17	16
DBRED				CMPD			
15	14	13	12	11	10	9	8
CMPC				CMPB			
7	6	5	4	3	2	1	0
CMPA				TBPRD			

Table 10-17: PWM Link Control Register (PWMLINK) Description

Bits	Field Name	Type	Reset	Description
31:28	GLDCTL1	RW	0xF	Write to the GLDCTL1 of PWM selected by PWMLINK[GLDCTL1] results in simultaneous write to GLDCTL1 of current PWM module 0000: Select PWM0 0001: Select PWM1 0010: Select PWM2 0011: Select PWM3 0100: Select PWM4 0101: Select PWM5 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111: No PWM selected

Bits	Field Name	Type	Reset	Description
27:24	DBFED	RW	0xF	Write to the DBFED of PWM selected by PWMLINK[DBFED] results in simultaneous write to DBFED of current PWM module 0000: Select PWM0 0001: Select PWM1 0010: Select PWM2 0011: Select PWM3 0100: Select PWM4 0101: Select PWM5 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111: No PWM selected
23:20	DBRED	RW	0xF	Write to the DBRED of PWM selected by PWMLINK[DBRED] results in simultaneous write to DBRED of current PWM module 0000: Select PWM0 0001: Select PWM1 0010: Select PWM2 0011: Select PWM3 0100: Select PWM4 0101: Select PWM5 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111: No PWM selected

Bits	Field Name	Type	Reset	Description
19:16	CMPD	RW	0xF	Write to the CMPD of PWM selected by PWMLINK[CMPD] results in simultaneous write to CMPD of current PWM module 0000: Select PWM0 0001: Select PWM1 0010: Select PWM2 0011: Select PWM3 0100: Select PWM4 0101: Select PWM5 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111: No PWM selected
15:12	CMPC	RW	0xF	Write to the CMPC of PWM selected by PWMLINK[CMPC] results in simultaneous write to CMPC of current PWM module 0000: Select PWM0 0001: Select PWM1 0010: Select PWM2 0011: Select PWM3 0100: Select PWM4 0101: Select PWM5 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111: No PWM selected

Bits	Field Name	Type	Reset	Description
11:8	CMPB	RW	0xF	Write to the CMPB of PWM selected by PWMLINK[CMPB] results in simultaneous write to CMPB of current PWM module 0000: Select PWM0 0001: Select PWM1 0010: Select PWM2 0011: Select PWM3 0100: Select PWM4 0101: Select PWM5 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111: No PWM selected
7:4	CMPA	RW	0xF	Write to the CMPA of PWM selected by PWMLINK[CMPA] results in simultaneous write to CMPA of current PWM module 0000: Select PWM0 0001: Select PWM1 0010: Select PWM2 0011: Select PWM3 0100: Select PWM4 0101: Select PWM5 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111: No PWM selected

Bits	Field Name	Type	Reset	Description
3:0	TBPRD	RW	0xF	Write to the TBPRD of PWM selected by PWMLINK[TBPRD] results in simultaneous write to TBPRD of current PWM module 0000: Select PWM0 0001: Select PWM1 0010: Select PWM2 0011: Select PWM3 0100: Select PWM4 0101: Select PWM5 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111: No PWM selected

Table 10-18: Time-Base Control Register (TBCTL) Layout

TBCTL (Time-Base Control Register) Offset: 0x14 Default: 0x00000006
Access: PWM0 -> TBCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_20							
23	22	21	20	19	18	17	16
RESERVED_31_20				DBGRUN		TBDIVBIN	
15	14	13	12	11	10	9	8
TBDIVBIN	TBDIVLIN			FRCSYNC	SYNCOSEL		
7	6	5	4	3	2	1	0
TBPRDLOCK	TBPRDLOAD		PHSDIR	PHSEN	CNTMODE		CNTRUN

Table 10-19: Time-Base Control Register (TBCTL) Description

Bits	Field Name	Type	Reset	Description
31:20	RESERVED_31_20	RO	0x0	Reserved.
19:18	DBGRUN	RW	0x0	PWM counter behaviour when CPU is halted Note: Debug operation and CPU fault exceptions both can cause CPU halted. 00: Stop after the next time-base counter increment or decrement 01: Stop when TBCNT completes a whole cycle (TBCNT=TBPRD for up-count mode and TBCNT=0 for other modes) 10: Counter keep running when CPU is halted 11: Counter keep running when CPU is halted
17:15	TBDIVBIN	RW	0x0	Binary part of TBCLK dividing ratio from PWM clock input. Overall dividing ratio is 2^TBDIVBIN * (TBDIVLIN+1) 000: Divide by 1 (default on reset) 001: Divide by 2 010: Divide by 4 011: Divide by 8 100: Divide by 16 101: Divide by 32 110: Divide by 64 111: Divide by 128
14:12	TBDIVLIN	RW	0x0	Lineary part of TBCLK dividing ratio from PWM clock input. Overall dividing ratio is 2^TBDIVBIN * (TBDIVLIN+1) 000: Divide by 1 (default on reset) 001: Divide by 2 010: Divide by 3 011: Divide by 4 100: Divide by 5 101: Divide by 6 110: Divide by 7 111: Divide by 8
11	FRCSYNC	W1S	0x0	Software forced synchronization This event is ORed with SYNCI input and is valid only when SYNCOSEL=000 0: Write a 0 has no effect and reads always return a 0 1: Write a 1 forces a one-time synchronization pulse. This bit is self-cleared to 0 and not affected by PWMREGKEY protection

Bits	Field Name	Type	Reset	Description
10:8	SYNCOSEL	RW	0x0	Synchronization output select 000: SYNCI and the software forced sync 001: TBCNT=0 event 010: TBCNT=TBPRD event 011: TBCNT=CMPA event 100: TBCNT=CMPB event 101: TBCNT=CMPC event 110: TBCNT=CMPD event 111: Disable SYNCO
7	TBPRDLOCK	RW	0x0	TBPRD active value lock 0: Active value in TBPRDA is updated according to TBCTL[TBPRDLOAD] 1: Active value in TBPRDA will not change
6:5	TBPRDLOAD	RW	0x0	TBPRD active value load mode It has no effect when TBCTL[TBPRDLOCK]=1 00: Load TBPRD into TBPRDA on TBCNT=0 01: Load TBPRD into TBPRDA on SYNC event 10: Load TBPRD into TBPRDA on SYNC event or TBCNT=0 11: Write TBPRD will immediately change the active value in TBPRDA
4	PHSDIR	RW	0x0	Phase direction It is valid only when TBCNT is in up-down-count mode (CNTMODE=10) 0: Count down after synchronization event 1: Count up after synchronization event
3	PHSEN	RW	0x0	Phase register enable 0: Do not load time-base counter (TBCNT) from time-base phase register (TBPHS) 1: Load TBCNT from TBPHS when synchronization event occurs (SYNCI input, software synchronization or digital compare sync event)
2:1	CNTMODE	RW	0x3	Counter mode 00: Down-count 01: Up-count 10: Up-down-count 11: Stop and freeze (default on reset)
0	CNTRUN	RW	0x0	Counter run/stop control 0: Counter stop 1: Counter free-running

Table 10-20: Time-Base Period Register (TBPRD) Layout

TBPRD (Time-Base Period Register) Offset: 0x18 Default: 0x00000000 Access: PWM0 -> TBPRD.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-21: Time-Base Period Register (TBPRD) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Time-base counter period

Table 10-22: Time-Base Period Active Register (TBPRDA) Layout

TBPRDA (Time-Base Period Active Register) Offset: 0x1C Default: 0x00000000 Access: PWM0 -> TBPRDA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-23: Time-Base Period Active Register (TBPRDA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Time-base counter active period When TBCTL[PRDLOAD]=0x3, write to TBPRD directly affects this register. Otherwise, the value is loaded from TBPRD upon the events defined in TBCTL[PRDLOAD].

Table 10-24: Time-Base Phase Register (TBPHS) Layout

TBPHS (Time-Base Phase Register) Offset: 0x20 Default: 0x00000000 Access: PWM0 -> TBPHS.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-25: Time-Base Phase Register (TBPHS) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Time-base counter phase when synchronization event occurs TBCTL[PHSEN]=0: The synchronization event is ignored and TBCNT keeps its value. TBCTL[PHSEN]=1: Time-base counter (TBCNT) will be loaded with TBPHS when synchronization event occurs.

Table 10-26: Time-Base Counter Register (TBCNT) Layout

TBCNT (Time-Base Counter Register) Offset: 0x24 Default: 0x00000000 Access: PWM0 -> TBCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-27: Time-Base Counter Register (TBCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Current value of the time-base counter

Table 10-28: Time-Base Status Register (TBSTS) Layout

TBSTS (Time-Base Status Register) Offset: 0x28 Default: 0x00000100 Access: PWM0 -> TBSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							CNTDIR
7	6	5	4	3	2	1	0
CNTCMPD	CNTCMPC	CNTCMPB	CNTCMPA	CNTPRD	CNTZRO	CNTMAX	SYNCI

Table 10-29: Time-Base Status Register (TBSTS) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	CNTDIR	RO	0x1	Time-base counter direction status 0: Time-base counter is currently counting down 1: Time-base counter is currently counting up
7	CNTCMPD	RO	0x0	Time-base counter reaching CMPD latched status 0: TBCNT never reaches CMPD 1: TBCNT reached CMPD
6	CNTCMPC	RO	0x0	Time-base counter reaching CMPC latched status 0: TBCNT never reaches CMPC 1: TBCNT reached CMPC
5	CNTCMPB	RO	0x0	Time-base counter reaching CMPB latched status 0: TBCNT never reaches CMPB 1: TBCNT reached CMPB
4	CNTCMPA	RO	0x0	Time-base counter reaching CMPA latched status 0: TBCNTnever reaches CMPA 1: TBCNT reached CMPA
3	CNTPRD	RO	0x0	Time-base counter reaching period latched status 0: TBCNT never reaches TBPRD 1: TBCNT reached TBPRD
2	CNTZRO	RO	0x0	Time-base counter reaching zero latched status 0: TBCNT never reaches 0x0 1: TBCNT reached 0x0

Bits	Field Name	Type	Reset	Description
1	CNTMAX	RO	0x0	Time-base counter reaching max latched status 0: TBCNT never reaches 0xFFFFFFFF 1: TBCNT reached 0xFFFFFFFF
0	SYNCI	RO	0x0	Input synchronization latched status 0: No input synchronization event has occurred 1: An input synchronization event has occurred

Table 10-30: Time-Base Status Clear Register (TBSTSCLR) Layout

TBSTSCLR (Time-Base Status Clear Register) Offset: 0x2C Default: 0x00000000 Access: PWM0 -> TBSTSCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7 CNTCMPD	6 CNTCMPC	5 CNTCMPB	4 CNTCMPA	3 CNTPRD	2 CNTZRO	1 CNTMAX	0 SYNCI

Table 10-31: Time-Base Status Clear Register (TBSTSCLR) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	CNTCMPD	W1C	0x0	Time-base counter reaching CMPD latched status clear 0: Writing a 0 has no effect. Always readback 0. 1: Writing a 1 will clear the latched status. This bit is self-cleared.
6	CNTCMPC	W1C	0x0	Time-base counter reaching CMPC latched status clear 0: Writing a 0 has no effect. Always readback 0. 1: Writing a 1 will clear the latched status. This bit is self-cleared.
5	CNTCMPB	W1C	0x0	Time-base counter reaching CMPB latched status clear 0: Writing a 0 has no effect. Always readback 0. 1: Writing a 1 will clear the latched status. This bit is self-cleared.
4	CNTCMPA	W1C	0x0	Time-base counter reaching CMPA latched status clear 0: Writing a 0 has no effect. Always readback 0. 1: Writing a 1 will clear the latched status. This bit is self-cleared.

Bits	Field Name	Type	Reset	Description
3	CNTPRD	W1C	0x0	Time-base counter reaching period latched status clear 0: Writing a 0 has no effect. Always readback 0. 1: Writing a 1 will clear the latched status. This bit is self-cleared.
2	CNTZRO	W1C	0x0	Time-base counter reaching zero latched status clear 0: Writing a 0 has no effect. Always readback 0. 1: Writing a 1 will clear the latched status. This bit is self-cleared.
1	CNTMAX	W1C	0x0	Time-base counter reaching max latched status clear 0: Writing a 0 has no effect. Always readback 0. 1: Writing a 1 will clear the latched status. This bit is self-cleared.
0	SYNCI	W1C	0x0	Input synchronization latched status clear 0: Writing a 0 has no effect. Always readback 0. 1: Writing a 1 will clear the latched status. This bit is self-cleared.

Table 10-32: Counter-Compare Control Register (CMPCTL) Layout

CMPCTL (Counter-Compare Control Register) Offset: 0x30 Default: 0x00000000 Access: PWM0 -> CMPCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 CMPDLOCK	14	13	12	11 CMPCLOCK	10	9	8
	CMPDLOAD				CMPCLOAD		
7 CMPBLOCK	6	5	4	3 CMPALOCK	2	1	0
	CMPBLOAD				CMPALOAD		

Table 10-33: Counter-Compare Control Register (CMPCTL) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	CMPDLOCK	RW	0x0	CMPD active value lock 0: Active value in CMPDA is updated according to CMPCTL[CMPDLOAD] 1: Active value in CMPDA will not change

Bits	Field Name	Type	Reset	Description
14:12	CMPDLOAD	RW	0x0	CMPD active value load mode It has no effect when CMPCTL[CMPDLOCK]=1 000: Load CMPD into CMPDA on TBCNT=0 001: Load CMPD into CMPDA on TBCNT=TBPRD 010: Load CMPD into CMPDA on TBCNT=0 or TBCNT=TBPRD 011: Load CMPD into CMPDA on SYNC event 100: Load CMPD into CMPDA on SYNC event or TBCNT=0 101: Load CMPD into CMPDA on SYNC event or TBCNT=TBPRD 110: Load CMPD into CMPDA on SYNC event, TBCNT=0 or TBCNT=TBPRD 111: Write CMPD will immediately change the active value in CMPDA
11	CMPCLOCK	RW	0x0	CMPC active value lock 0: Active value in CMPCA is updated according to CMPCTL[CMPCLOAD] 1: Active value in CMPCA will not change
10:8	CMPCLOAD	RW	0x0	CMPC active value load mode It has no effect when CMPCTL[CMPCLOCK]=1 000: Load CMPC into CMPCA on TBCNT=0 001: Load CMPC into CMPCA on TBCNT=TBPRD 010: Load CMPC into CMPCA on TBCNT=0 or TBCNT=TBPRD 011: Load CMPC into CMPCA on SYNC event 100: Load CMPC into CMPCA on SYNC event or TBCNT=0 101: Load CMPC into CMPCA on SYNC event or TBCNT=TBPRD 110: Load CMPC into CMPCA on SYNC event, TBCNT=0 or TBCNT=TBPRD 111: Write CMPC will immediately change the active value in CMPCA
7	CMPBLOCK	RW	0x0	CMPB active value lock 0: Active value in CMPBA is updated according to CMPBTL[CMPBLOAD] 1: Active value in CMPBA will not change

Bits	Field Name	Type	Reset	Description
6:4	CMPBLOAD	RW	0x0	CMPB active value load mode It has no effect when CMPBTL[CMPBLOCK]=1 000: Load CMPB into CMPBA on TBCNT=0 001: Load CMPB into CMPBA on TBCNT=TBPRD 010: Load CMPB into CMPBA on TBCNT=0 or TBCNT=TBPRD 011: Load CMPB into CMPBA on SYNC event 100: Load CMPB into CMPBA on SYNC event or TBCNT=0 101: Load CMPB into CMPBA on SYNC event or TBCNT=TBPRD 110: Load CMPB into CMPBA on SYNC event, TBCNT=0 or TBCNT=TBPRD 111: Write CMPB will immediately change the active value in CMPBA
3	CMPALOCK	RW	0x0	CMPA active value lock 0: Active value in CMPAA is updated according to CMPATL[CMPALOAD] 1: Active value in CMPAA will not change
2:0	CMPALOAD	RW	0x0	CMPA active value load mode It has no effect when CMPATL[CMPALOCK]=1 000: Load CMPA into CMPAA on TBCNT=0 001: Load CMPA into CMPAA on TBCNT=TBPRD 010: Load CMPA into CMPAA on TBCNT=0 or TBCNT=TBPRD 011: Load CMPA into CMPAA on SYNC event 100: Load CMPA into CMPAA on SYNC event or TBCNT=0 101: Load CMPA into CMPAA on SYNC event or TBCNT=TBPRD 110: Load CMPA into CMPAA on SYNC event, TBCNT=0 or TBCNT=TBPRD 111: Write CMPA will immediately change the active value in CMPAA

Table 10-34: Counter-Compare A Threshold Register (CMPA) Layout

CMPA (Counter-Compare A Threshold Register) Offset: 0x34 Default: 0x00000000 Access: PWM0 -> CMPA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-35: Counter-Compare A Threshold Register (CMPA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Comparator A threshold

Table 10-36: Counter-Compare A Threshold Active Register (CMPAA) Layout

CMPAA (Counter-Compare A Threshold Active Register) Offset: 0x38 Default: 0x00000000 Access: PWM0 -> CMPAA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-37: Counter-Compare A Threshold Active Register (CMPAA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Comparator A threshold active value A TBCNT=CMPA event will be generated while TBCNT is equal to the threshold value

Table 10-38: Counter-Compare B Threshold Register (CMPB) Layout

CMPB (Counter-Compare B Threshold Register) Offset: 0x3C Default: 0x00000000 Access: PWM0 -> CMPB.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-39: Counter-Compare B Threshold Register (CMPB) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Comparator B threshold

Table 10-40: Counter-Compare B Threshold Active Register (CMPBA) Layout

CMPBA (Counter-Compare B Threshold Active Register) Offset: 0x40 Default: 0x00000000 Access: PWM0 -> CMPBA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-41: Counter-Compare B Threshold Active Register (CMPBA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Comparator B threshold active value A TBCNT=CMPB event will be generated while TBCNT is equal to the threshold value

Table 10-42: Counter-Compare C Threshold Register (CMPC) Layout

CMPC (Counter-Compare C Threshold Register) Offset: 0x44 Default: 0x00000000 Access: PWM0 -> CMPC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-43: Counter-Compare C Threshold Register (CMPC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Comparator C threshold

Table 10-44: Counter-Compare C Threshold Active Register (CMPCA) Layout

CMPCA (Counter-Compare C Threshold Active Register) Offset: 0x48 Default: 0x00000000 Access: PWM0 -> CMPCA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-45: Counter-Compare C Threshold Active Register (CMPCA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Comparator C threshold active value A TBCNT=CMPC event will be generated while TBCNT is equal to the threshold value

Table 10-46: Counter-Compare D Threshold Register (CMPD) Layout

CMPD (Counter-Compare D Threshold Register) Offset: 0x4C Default: 0x00000000 Access: PWM0 -> CMPD.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-47: Counter-Compare D Threshold Register (CMPD) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Comparator D threshold

Table 10-48: Counter-Compare D Threshold Active Register (CMPDA) Layout

CMPDA (Counter-Compare D Threshold Active Register) Offset: 0x50 Default: 0x00000000 Access: PWM0 -> CMPDA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-49: Counter-Compare D Threshold Active Register (CMPDA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Comparator D threshold active value A TBCNT=CMPD event will be generated while TBCNT is equal to the threshold value

Table 10-50: Action-Qualifier Control Register (AQCTL) Layout

AQCTL (Action-Qualifier Control Register) Offset: 0x54 Default: 0x00000000 Access: PWM0 -> AQCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
T1SEL				T0SEL			
7	6	5	4	3	2	1	0
AQCTLBLOCK	AQCTLBLOAD			AQCTLALOCK	AQCTLALOAD		

Table 10-51: Action-Qualifier Control Register (AQCTL) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:12	T1SEL	RW	0x0	T1 event source select 0000: DCAEVT0 0001: DCAEVT1 0010: DCBEVT0 0011: DCBEVT1 0100: TZ0 0101: TZ1 0110: TZ2 0111: TZ3 1000: TZ4 1001: SYNCI 1010: 1011: 1100: 1101: 1110: 1111:

Bits	Field Name	Type	Reset	Description
11:8	T0SEL	RW	0x0	T0 event source select 0000: DCAEVT0 0001: DCAEVT1 0010: DCBEVT0 0011: DCBEVT1 0100: TZ0 0101: TZ1 0110: TZ2 0111: TZ3 1000: TZ4 1001: SYNCI 1010: 1011: 1100: 1101: 1110: 1111:
7	AQCTLBLOCK	RW	0x0	AQCTLB active value lock 0: Active value in AQCTLBA is updated according to AQCTL[AQCTLBLOAD] 1: Active value in AQCTLBA will not change
6:4	AQCTLBLOAD	RW	0x0	AQCTLB active value load mode It has no effect when AQCTL[AQCTLBLOCK]=1 000: Load AQCTLB into AQCTLBA on TBCNT=0 001: Load AQCTLB into AQCTLBA on TBCNT=TBPRD 010: Load AQCTLB into AQCTLBA on TBCNT=0 or TBCNT=TBPRD 011: Load AQCTLB into AQCTLBA on SYNC event 100: Load AQCTLB into AQCTLBA on SYNC event or TBCNT=0 101: Load AQCTLB into AQCTLBA on SYNC event or TBCNT=TBPRD 110: Load AQCTLB into AQCTLBA on SYNC event, TBCNT=0 or TBCNT=TBPRD 111: Write AQCTLB immediately change the active value in AQCTLBA
3	AQCTLALOCK	RW	0x0	AQCTLA active value lock 0: Active value in AQCTLAA is updated according to AQCTL[AQCTLALOAD] 1: Active value in AQCTLAA will not change

Bits	Field Name	Type	Reset	Description
2:0	AQCTLALOAD	RW	0x0	AQCTLA active value load mode It has no effect when AQCTL[AQCTLALOCK]=1 000: Load AQCTLA into AQCTLAA on TBCNT=0 001: Load AQCTLA into AQCTLAA on TBCNT=TBPRD 010: Load AQCTLA into AQCTLAA on TBCNT=0 or TBCNT=TBPRD 011: Load AQCTLA into AQCTLAA on SYNC event 100: Load AQCTLA into AQCTLAA on SYNC event or TBCNT=0 101: Load AQCTLA into AQCTLAA on SYNC event or TBCNT=TBPRD 110: Load AQCTLA into AQCTLAA on SYNC event, TBCNT=0 or TBCNT=TBPRD 111: Write AQCTLA immediately change the active value in AQCTLAA

Table 10-52: Action-Qualifier Output A Control Register (AQCTLA) Layout

AQCTLA (Action-Qualifier Output A Control Register) Offset: 0x58 Default: 0x00000000
Access: PWM0 -> AQCTLA.all

31	30	29	28	27	26	25	24
RESERVED_31_20							
23	22	21	20	19	18	17	16
RESERVED_31_20				T1D		T1U	
15	14	13	12	11	10	9	8
T0D		T0U		CBD		CBU	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	

Table 10-53: Action-Qualifier Output A Control Register (AQCTLA) Description

Bits	Field Name	Type	Reset	Description
31:20	RESERVED_31_20	RO	0x0	Reserved.
19:18	T1D	RW	0x0	Action on output A when T1 event and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
17:16	T1U	RW	0x0	Action on output A when T1 event and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)

Bits	Field Name	Type	Reset	Description
15:14	T0D	RW	0x0	Action on output A when T0 event and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
13:12	T0U	RW	0x0	Action on output A when T0 event and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
11:10	CBD	RW	0x0	Action on output A when TBCNT=CMPB and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
9:8	CBU	RW	0x0	Action on output A when TBCNT=CMPB and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
7:6	CAD	RW	0x0	Action on output A when TBCNT=CMPA and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
5:4	CAU	RW	0x0	Action on output A when TBCNT=CMPA and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
3:2	PRD	RW	0x0	Action on output A when TBCNT=TBPRD 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)

Bits	Field Name	Type	Reset	Description
1:0	ZRO	RW	0x0	Action on output A when TBCNT=Zero 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)

Table 10-54: Action-Qualifier Output A Control Active Register (AQCTLAA) Layout

AQCTLAA (Action-Qualifier Output A Control Active Register) Offset: 0x5C Default: 0x00000000
Access: PWM0 -> AQCTLAA.all

31	30	29	28	27	26	25	24
RESERVED_31_20							
23	22	21	20	19	18	17	16
RESERVED_31_20				T1D		T1U	
15	14	13	12	11	10	9	8
T0D		T0U		CBD		CBU	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	

Table 10-55: Action-Qualifier Output A Control Active Register (AQCTLAA) Description

Bits	Field Name	Type	Reset	Description
31:20	RESERVED_31_20	RO	0x0	Reserved.
19:18	T1D	RO	0x0	Action on output A when T1 event and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
17:16	T1U	RO	0x0	Action on output A when T1 event and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
15:14	T0D	RO	0x0	Action on output A when T0 event and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)

Bits	Field Name	Type	Reset	Description
13:12	T0U	RO	0x0	Action on output A when T0 event and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
11:10	CBD	RO	0x0	Action on output A when TBCNT=CMPB and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
9:8	CBU	RO	0x0	Action on output A when TBCNT=CMPB and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
7:6	CAD	RO	0x0	Action on output A when TBCNT=CMPA and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
5:4	CAU	RO	0x0	Action on output A when TBCNT=CMPA and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
3:2	PRD	RO	0x0	Action on output A when TBCNT=TBPRD 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)
1:0	ZRO	RO	0x0	Action on output A when TBCNT=Zero 00: Do nothing (action disabled) 01: Clear (force PWM output A to low) 10: Set (force PWM output A to high) 11: Toggle (invert current PWM output A status)

Table 10-56: Action-Qualifier Output B Control Register (AQCTLB) Layout

AQCTLB (Action-Qualifier Output B Control Register) Offset: 0x60 Default: 0x00000000 Access: PWM0 -> AQCTLB.all							
31	30	29	28	27	26	25	24
RESERVED_31_20							
23	22	21	20	19	18	17	16
RESERVED_31_20				T1D		T1U	
15	14	13	12	11	10	9	8
T0D		T0U		CBD		CBU	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	

Table 10-57: Action-Qualifier Output B Control Register (AQCTLB) Description

Bits	Field Name	Type	Reset	Description
31:20	RESERVED_31_20	RO	0x0	Reserved.
19:18	T1D	RW	0x0	Action on output B when T1 event and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
17:16	T1U	RW	0x0	Action on output B when T1 event and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
15:14	T0D	RW	0x0	Action on output B when T0 event and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
13:12	T0U	RW	0x0	Action on output B when T0 event and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)

Bits	Field Name	Type	Reset	Description
11:10	CBD	RW	0x0	Action on output B when TBCNT=CMPA and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
9:8	CBU	RW	0x0	Action on output B when TBCNT=CMPB and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
7:6	CAD	RW	0x0	Action on output B when TBCNT=CMPA and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
5:4	CAU	RW	0x0	Action on output B when TBCNT=CMPA and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
3:2	PRD	RW	0x0	Action on output B when TBCNT=TBPRD 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
1:0	ZRO	RW	0x0	Action on output B when TBCNT=Zero 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)

Table 10-58: Action-Qualifier Output B Control Active Register (AQCTLBA) Layout

AQCTLBA (Action-Qualifier Output B Control Active Register) Offset: 0x64 Default: 0x00000000 Access: PWM0 -> AQCTLBA.all							
31	30	29	28	27	26	25	24
RESERVED_31_20							
23	22	21	20	19	18	17	16
RESERVED_31_20				T1D		T1U	
15	14	13	12	11	10	9	8
T0D		T0U		CBD		CBU	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	

Table 10-59: Action-Qualifier Output B Control Active Register (AQCTLBA) Description

Bits	Field Name	Type	Reset	Description
31:20	RESERVED_31_20	RO	0x0	Reserved.
19:18	T1D	RO	0x0	Action on output B when T1 event and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
17:16	T1U	RO	0x0	Action on output B when T1 event and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
15:14	T0D	RO	0x0	Action on output B when T0 event and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
13:12	T0U	RO	0x0	Action on output B when T0 event and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)

Bits	Field Name	Type	Reset	Description
11:10	CBD	RO	0x0	Action on output B when TBCNT=CMPA and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
9:8	CBU	RO	0x0	Action on output B when TBCNT=CMPB and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
7:6	CAD	RO	0x0	Action on output B when TBCNT=CMPA and TBCNT is counting down 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
5:4	CAU	RO	0x0	Action on output B when TBCNT=CMPA and TBCNT is counting up 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
3:2	PRD	RO	0x0	Action on output B when TBCNT=TBPRD 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)
1:0	ZRO	RO	0x0	Action on output B when TBCNT=Zero 00: Do nothing (action disabled) 01: Clear (force PWM output B to low) 10: Set (force PWM output B to high) 11: Toggle (invert current PWM output B status)

Table 10-60: Action-Qualifier Software Force Register (AQSFRC) Layout

AQSFRC (Action-Qualifier Software Force Register) Offset: 0x68 Default: 0x00000000 Access: PWM0 -> AQSFRC.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
CSFLOAD		OTSFB	ACTSFB		OTSFA	ACTSFA	

Table 10-61: Action-Qualifier Software Force Register (AQSFRC) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:6	CSFLOAD	RW	0x0	Options to load continuous software force register (AQCSFRC) active value from shadow register 00: Load from AQCSFRC into AQCSFRCA on TBCNT=0 01: Load from AQCSFRC into AQCSFRCA on TBCNT=TBPRD 10: Load from AQCSFRC into AQCSFRCA on TBCNT=Zero or TBCNT=TBPRD 11: Write AQCSFRC immediately change the active value in AQCSFRCA
5	OTSFB	W1S	0x0	One-time software forced event on output B This bit is auto cleared once a write to it is complete 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 initialize a one-cycle force pulse. This bit is self-cleared to 0
4:3	ACTSFB	RW	0x0	Action when one-time software force B is invoked 00: Do nothing (action disabled) 01: Clear (force PWM output B to low 10: Set (force PWM output B to high 11: Toggle (invert current PWM AQ output B status)
2	OTSFA	W1S	0x0	One-time software forced event on output A This bit is auto cleared once a write to it is complete 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 initialize a one-cycle force pulse. This bit is self-cleared to 0

Bits	Field Name	Type	Reset	Description
1:0	ACTSFA	RW	0x0	Action when one-time software force A is invoked 00: Do nothing (action disabled) 01: Clear (force PWM output A to low 10: Set (force PWM output A to high 11: Toggle (invert current PWM AQ output A status)

Table 10-62: Action-Qualifier Continuous Software Force Register (AQCSFRC) Layout

AQCSFRC (Action-Qualifier Continuous Software Force Register) Offset: 0x6C Default: 0x00000000
Access: PWM0 -> AQCSFRC.all

31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				CSFB		CSFA	

Table 10-63: Action-Qualifier Continuous Software Force Register (AQCSFRC) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3:2	CSFB	RW	0x0	Continuous software force on output B 00: Forcing disabled 01: Force a continuous low on output B 10: Force a continuous high on output B 11: Invalid option
1:0	CSFA	RW	0x0	Continuous software force on output A 00: Forcing disabled 01: Force a continuous low on output A 10: Force a continuous high on output A 11: Invalid option

Table 10-64: Action-Qualifier Continuous Software Force Active Register (AQCSFRCA) Layout

AQCSFRCA (Action-Qualifier Continuous Software Force Active Register) Offset: 0x70 Default: 0x00000000 Access: PWM0 -> AQCSFRCA.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				CSFB		CSFA	

Table 10-65: Action-Qualifier Continuous Software Force Active Register (AQCSFRCA) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3:2	CSFB	RO	0x0	Continuous software force on output B 00: Forcing disabled 01: Force a continuous low on output B 10: Force a continuous high on output B 11: Invalid option
1:0	CSFA	RO	0x0	Continuous software force on output A 00: Forcing disabled 01: Force a continuous low on output A 10: Force a continuous high on output A 11: Invalid option

Table 10-66: Dead-Band Generator Control Register (DBCTL) Layout

DBCTL (Dead-Band Generator Control Register) Offset: 0x74 Default: 0x00000070 Access: PWM0 -> DBCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_19							
23	22	21	20	19	18	17	16
RESERVED_31_19					DBFEDLOCK	DBFEDLOAD	
15	14	13	12	11	10	9	8
DBREDLOCK	DBREDLOAD		DBCTLLOCK	DBCTLLOAD		HALFCYCLE	DUALEDGE
7	6	5	4	3	2	1	0
OUTBSRC	OUTASRC	FEDPOL	REDPOL	FEDSRC	REDSRC	FEDEN	REDEN

Table 10-67: Dead-Band Generator Control Register (DBCTL) Description

Bits	Field Name	Type	Reset	Description
31:19	RESERVED_31_19	RO	0x0	Reserved.
18	DBFEDLOCK	RW	0x0	DBFED active value lock 0: Active value in DBFEDA is updated according to DBCTL[DBFEDLOAD] 1: Active value in DBFEDA will not change
17:16	DBFEDLOAD	RW	0x0	DBFED active value load mode It has no effect when DBCTL[DBFEDLOCK]=1 00: Load DBFED into DBFEDA on TBCNT=0 01: Load DBFED into DBFEDA on TBCNT=TBPRD 10: Load DBFED into DBFEDA on TBCNT=0 or TBCNT=TBPRD 11: Write DBFED immediately change the active value in DBFEDA
15	DBREDLOCK	RW	0x0	DBRED active value lock 0: Active value in DBREDA is updated according to DBCTL[DBREDLOAD] 1: Active value in DBREDA will not change
14:13	DBREDLOAD	RW	0x0	DBRED active value load mode It has no effect when DBCTL[DBFEDLOCK]=1 00: Load DBRED into DBREDA on TBCNT=0 01: Load DBRED into DBREDA on TBCNT=TBPRD 10: Load DBRED into DBREDA on TBCNT=0 or TBCNT=TBPRD 11: Write DBRED will immediately change the active value in DBREDA
12	DBCTLLOCK	RW	0x0	DBCTL[9:0] active value lock 0: Active value in DBCTLA is updated according to DBCTL[DBCTLLOAD] 1: Active value in DBCTLA will not change
11:10	DBCTLLOAD	RW	0x0	DBCTL[9:0] active value load mode It has no effect when DBCTL[DBCTLLOCK]=1 00: Load DBCTL[9:0] into DBCTLA on TBCNT=0 01: Load DBCTL[9:0] into DBCTLA on TBCNT=TBPRD 10: Load DBCTL[9:0] into DBCTLA on TBCNT=0 or TBCNT=TBPRD 11: Write DBCTL[9:0] will immediately change the active value in DBCTLA

Bits	Field Name	Type	Reset	Description
9	HALFCYCLE	RW	0x0	Half cycle clocking enable 0: Full cycle clocking enabled. The dead-band counter is clocked at TBCLK. 1: Half cycle clocking enabled. The dead-band counter is clocked at 2x TBCLK.
8	DUALEDGE	RW	0x0	Dual-edge delay mode 0: Rising-edge delay and falling-edge delay are two separate paths 1: Rising-edge delay path output serves as input for Falling-edge delay path
7	OUTBSRC	RW	0x0	Output B source 0: Falling-edge delay path is selected as output B 1: Rising-edge delay path is selected as output B
6	OUTASRC	RW	0x1	Output A source 0: Falling-edge delay path is selected as output A 1: Rising-edge delay path is selected as output A
5	FEDPOL	RW	0x1	Falling-edge delay output polarity 0: Falling-edge delay output is inverted to have active-low polarity 1: Falling-edge delay output keeps the original active-high polarity
4	REDPOL	RW	0x1	Rising-edge delay output polarity 0: Rising-edge delay output is inverted to have active-low polarity 1: Rising-edge delay output keeps the original active-high polarity
3	FEDSRC	RW	0x0	Falling-edge delay source 0: Input A is the source for falling-edge delay 1: Input B is the source for falling-edge delay
2	REDSRC	RW	0x0	Rising-edge delay source 0: Input A is the source for rising-edge delay 1: Input B is the source for rising-edge delay
1	FEDEN	RW	0x0	Falling-edge delay enable 0: Falling-edge delay is bypassed and the input from action-qualifier goes to the submodule directly 1: Falling-edge delay is enabled and applied at output B

Bits	Field Name	Type	Reset	Description
0	REDEN	RW	0x0	Rising-edge delay enable 0: Rising-edge delay is bypassed and the input from action-qualifier goes to the submodule directly 1: Rising-edge delay is enabled and applied at output A

Table 10-68: Dead-Band Generator Control Active Register (DBCTLA) Layout

DBCTLA (Dead-Band Generator Control Active Register) Offset: 0x78 Default: 0x00000070
Access: PWM0 -> DBCTLA.all

31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						HALFCYCLE	DUALEDGE
7	6	5	4	3	2	1	0
OUTBSRC	OUTASRC	FEDPOL	REDPOL	FEDSRC	REDSRC	FEDEN	REDEN

Table 10-69: Dead-Band Generator Control Active Register (DBCTLA) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9	HALFCYCLE	RO	0x0	Half cycle clocking enable 0: Full cycle clocking enabled. The dead-band counter is clocked at TBCLK. 1: Half cycle clocking enabled. The dead-band counter is clocked at 2x TBCLK.
8	DUALEDGE	RO	0x0	Dual-edge delay mode 0: Rising-edge delay and falling-edge delay are two separate paths 1: Rising-edge delay path output serves as input for Falling-edge delay path
7	OUTBSRC	RO	0x0	Output B source 0: Falling-edge delay path is selected as output B 1: Rising-edge delay path is selected as output B
6	OUTASRC	RO	0x1	Output A source 0: Falling-edge delay path is selected as output A 1: Rising-edge delay path is selected as output A

Bits	Field Name	Type	Reset	Description
5	FEDPOL	RO	0x1	Falling-edge delay output polarity 0: Falling-edge delay output is inverted to have active-low polarity 1: Falling-edge delay output keeps the original active-high polarity
4	REDPOL	RO	0x1	Rising-edge delay output polarity 0: Rising-edge delay output is inverted to have active-low polarity 1: Rising-edge delay output keeps the original active-high polarity
3	FEDSRC	RO	0x0	Falling-edge delay source 0: Input A is the source for falling-edge delay 1: Input B is the source for falling-edge delay
2	REDSRC	RO	0x0	Rising-edge delay source 0: Input A is the source for rising-edge delay 1: Input B is the source for rising-edge delay
1	FEDEN	RO	0x0	Falling-edge delay enable 0: Falling-edge delay is bypassed and the input from action-qualifier goes to the submodule directly 1: Falling-edge delay is enabled and applied at output B
0	REDEN	RO	0x0	Rising-edge delay enable 0: Rising-edge delay is bypassed and the input from action-qualifier goes to the submodule directly 1: Rising-edge delay is enabled and applied at output A

Table 10-70: Dead-Band Generator Rising Edge Delay Register (DBRED) Layout

DBRED (Dead-Band Generator Rising Edge Delay Register) Offset: 0x7C Default: 0x00000000 Access: PWM0 -> DBRED.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-71: Dead-Band Generator Rising Edge Delay Register (DBRED) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Rising-edge delay

Table 10-72: Dead-Band Generator Rising Edge Delay Active Register (DBREDA) Layout

DBREDA (Dead-Band Generator Rising Edge Delay Active Register) Offset: 0x80 Default: 0x00000000 Access: PWM0 -> DBREDA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-73: Dead-Band Generator Rising Edge Delay Active Register (DBREDA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Rising-edge delay

Table 10-74: Dead-Band Generator Falling Edge Delay Register (DBFED) Layout

DBFED (Dead-Band Generator Falling Edge Delay Register) Offset: 0x84 Default: 0x00000000 Access: PWM0 -> DBFED.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-75: Dead-Band Generator Falling Edge Delay Register (DBFED) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Falling-edge delay

Table 10-76: Dead-Band Generator Falling Edge Delay Active Register (DBFEDA) Layout

DBFEDA (Dead-Band Generator Falling Edge Delay Active Register) Offset: 0x88 Default: 0x00000000 Access: PWM0 -> DBFEDA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-77: Dead-Band Generator Falling Edge Delay Active Register (DBFEDA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Falling-edge delay

Table 10-78: Trip-Zone Event Select Register (TZSEL) Layout

TZSEL (Trip-Zone Event Select Register) Offset: 0x8C Default: 0x00030003 Access: PWM0 -> TZSEL.all							
31	30	29	28	27	26	25	24
RESERVED_31_27					DBGCBC	DCBEVT1	DCAEVT1
23	22	21	20	19	18	17	16
CLKERRCBC	TZ4CBC	TZ3CBC	TZ2CBC	TZ1CBC	TZ0CBC	CBCOUT	
15	14	13	12	11	10	9	8
RESERVED_15_11					DBGOST	DCBEVT0	DCAEVT0
7	6	5	4	3	2	1	0
CLKERROST	TZ4OST	TZ3OST	TZ2OST	TZ1OST	TZ0OST	OSTOUT	

Table 10-79: Trip-Zone Event Select Register (TZSEL) Description

Bits	Field Name	Type	Reset	Description
31:27	RESERVED_31_27	RO	0x0	Reserved.
26	DBGCBC	RW	0x0	JTAG debug as cycle-by-cycle trip source (ORed) 0: Disable 1: Enable
25	DCBEVT1	RW	0x0	Digital compare B event 1 as a cycle-by-cycle trip source (ORed) 0: Disable 1: Enable
24	DCAEVT1	RW	0x0	Digital compare A event 1 as a cycle-by-cycle trip source (ORed) 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
23	CLKERRCBC	RW	0x0	Clock error as a cycle-by-cycle trip source (ORed) 0: Disable 1: Enable
22	TZ4CBC	RW	0x0	TZ4 as a cycle-by-cycle trip source (ORed) 0: Disable 1: Enable
21	TZ3CBC	RW	0x0	TZ3 as a cycle-by-cycle trip source (ORed) 0: Disable 1: Enable
20	TZ2CBC	RW	0x0	TZ2 as a cycle-by-cycle trip source (ORed) 0: Disable 1: Enable
19	TZ1CBC	RW	0x0	TZ1 as a cycle-by-cycle trip source (ORed) 0: Disable 1: Enable
18	TZ0CBC	RW	0x0	TZ0 as a cycle-by-cycle trip source (ORed) 0: Disable 1: Enable
17:16	CBCOUT	RW	0x3	Cycle-by-cycle trip-zone output select 00: Disable cycle-by-cycle trip-zone 01: Select only the asynchronous path. i.e. The original event OR gate output 10: Select only the latched path 11: Select both the asynchronous path and the latched path (ORed)
15:11	RESERVED_15_11	RO	0x0	Reserved.
10	DBGOST	RW	0x0	JTAG debug as a one-shot trip source (ORed) 0: Disable 1: Enable
9	DCBEVT0	RW	0x0	Digital compare B event 0 as a one-shot trip source (ORed) 0: Disable 1: Enable
8	DCAEVT0	RW	0x0	Digital compare A event 0 as a one-shot trip source (ORed) 0: Disable 1: Enable
7	CLKERROST	RW	0x0	Clock error as a one-shot trip source (ORed) 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
6	TZ4OST	RW	0x0	TZ4 as a one-shot trip source (ORed) 0: Disable 1: Enable
5	TZ3OST	RW	0x0	TZ3 as a one-shot trip source (ORed) 0: Disable 1: Enable
4	TZ2OST	RW	0x0	TZ2 as a one-shot trip source (ORed) 0: Disable 1: Enable
3	TZ1OST	RW	0x0	TZ1 as a one-shot trip source (ORed) 0: Disable 1: Enable
2	TZ0OST	RW	0x0	TZ0 as a one-shot trip source (ORed) 0: Disable 1: Enable
1:0	OSTOUT	RW	0x3	One-shot trip-zone output select 00: Disable one-shot trip-zone 01: Select only the asynchronous path. i.e. The original event OR gate output 10: Select only the latched path 11: Select both the asynchronous path and the latched path (ORed)

Table 10-80: Trip-Zone Status Register (TZSTS) Layout

TZSTS (Trip-Zone Status Register) Offset: 0x90 Default: 0x00000000 Access: PWM0 -> TZSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_18							
23	22	21	20	19	18	17	16
RESERVED_31_18						DBGCBC	DCBEVT1
15 DCAEVT1	14 CLKERRCBC	13 TZ4CBC	12 TZ3CBC	11 TZ2CBC	10 TZ1CBC	9 TZ0CBC	8 DBGOST
7 DCBEVT0	6 DCAEVT0	5 CLKERROST	4 TZ4OST	3 TZ3OST	2 TZ2OST	1 TZ1OST	0 TZ0OST

Table 10-81: Trip-Zone Status Register (TZSTS) Description

Bits	Field Name	Type	Reset	Description
31:18	RESERVED_31_18	RO	0x0	Reserved.
17	DBGCBC	RO	0x0	Latched JTAG debug cycle-by-cycle trip event status 0: No event has occurred 1: Event occurred

Bits	Field Name	Type	Reset	Description
16	DCBEVT1	RO	0x0	Latched digital compare B event 1 cycle-by-cycle trip status 0: No event has occurred 1: Event occurred
15	DCAEVT1	RO	0x0	Latched digital compare A event 1 cycle-by-cycle trip status 0: No event has occurred 1: Event occurred
14	CLKERRCBC	RO	0x0	Latched clock error cycle-by-cycle trip event status 0: No event has occurred 1: Event occurred
13	TZ4CBC	RO	0x0	Latched TZ4 cycle-by-cycle trip event status 0: No event has occurred 1: Event occurred
12	TZ3CBC	RO	0x0	Latched TZ3 cycle-by-cycle trip event status 0: No event has occurred 1: Event occurred
11	TZ2CBC	RO	0x0	Latched TZ2 cycle-by-cycle trip event status 0: No event has occurred 1: Event occurred
10	TZ1CBC	RO	0x0	Latched TZ1 cycle-by-cycle trip event status 0: No event has occurred 1: Event occurred
9	TZ0CBC	RO	0x0	Latched TZ0 cycle-by-cycle trip event status 0: No event has occurred 1: Event occurred
8	DBGOST	RO	0x0	Latched JTAG debug one-shot trip event status 0: No event has occurred 1: Event occurred
7	DCBEVT0	RO	0x0	Latched digital compare B event 0 one-shot trip status 0: No event has occurred 1: Event occurred
6	DCAEVT0	RO	0x0	Latched digital compare A event 0 one-shot trip status 0: No event has occurred 1: Event occurred
5	CLKERROST	RO	0x0	Latched clock error one-shot trip event status 0: No event has occurred 1: Event occurred

Bits	Field Name	Type	Reset	Description
4	TZ4OST	RO	0x0	Latched TZ4 one-shot trip event status 0: No event has occurred 1: Event occurred
3	TZ3OST	RO	0x0	Latched TZ3 one-shot trip event status 0: No event has occurred 1: Event occurred
2	TZ2OST	RO	0x0	Latched TZ2 one-shot trip event status 0: No event has occurred 1: Event occurred
1	TZ1OST	RO	0x0	Latched TZ1 one-shot trip event status 0: No event has occurred 1: Event occurred
0	TZ0OST	RO	0x0	Latched TZ0 one-shot trip event status 0: No event has occurred 1: Event occurred

Table 10-82: Trip-Zone Status Clear Register (TZSTSCLR) Layout

TZSTSCLR (Trip-Zone Status Clear Register) Offset: 0x94 Default: 0x00000000 Access: PWM0 -> TZSTSCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_18							
23	22	21	20	19	18	17	16
RESERVED_31_18						DBGCBC	DCBEVT1
15	14	13	12	11	10	9	8
DCAEVT1	CLKERRCBC	TZ4CBC	TZ3CBC	TZ2CBC	TZ1CBC	TZ0CBC	DBGOST
7	6	5	4	3	2	1	0
DCBEVT0	DCAEVT0	CLKERROST	TZ4OST	TZ3OST	TZ2OST	TZ1OST	TZ0OST

Table 10-83: Trip-Zone Status Clear Register (TZSTSCLR) Description

Bits	Field Name	Type	Reset	Description
31:18	RESERVED_31_18	RO	0x0	Reserved.
17	DBGCBC	W1C	0x0	Latched JTAG debug cycle-by-cycle trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
16	DCBEVT1	W1C	0x0	Latched digital compare B event 1 cycle-by-cycle trip status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0

Bits	Field Name	Type	Reset	Description
15	DCAEVT1	W1C	0x0	Latched digital compare A event 1 cycle-by-cycle trip status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
14	CLKERRCBC	W1C	0x0	Latched clock error cycle-by-cycle trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
13	TZ4CBC	W1C	0x0	Latched TZ4 cycle-by-cycle trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
12	TZ3CBC	W1C	0x0	Latched TZ3 cycle-by-cycle trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
11	TZ2CBC	W1C	0x0	Latched TZ2 cycle-by-cycle trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
10	TZ1CBC	W1C	0x0	Latched TZ1 cycle-by-cycle trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
9	TZ0CBC	W1C	0x0	Latched TZ0 cycle-by-cycle trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
8	DBGOST	W1C	0x0	Latched JTAG debug one-shot trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0

Bits	Field Name	Type	Reset	Description
7	DCBEVT0	W1C	0x0	Latched digital output B event 0 one-shot trip status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
6	DCAEVT0	W1C	0x0	Latched digital output A event 0 one-shot trip status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
5	CLKERROST	W1C	0x0	Latched clock error one-shot trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
4	TZ4OST	W1C	0x0	Latched TZ4 one-shot trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
3	TZ3OST	W1C	0x0	Latched TZ3 one-shot trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
2	TZ2OST	W1C	0x0	Latched TZ2 one-shot trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
1	TZ1OST	W1C	0x0	Latched TZ1 one-shot trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
0	TZ0OST	W1C	0x0	Latched TZ0 one-shot trip event status clear 0: Write a 0 has no effect. Always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0

Table 10-84: Trip-Zone Digital Compare Event Select Register (TZDCSEL) Layout

TZDCSEL (Trip-Zone Digital Compare Event Select Register) Offset: 0x98 Default: 0x00000000 Access: PWM0 -> TZDCSEL.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				DCBEVT1			DCBEVT0
7	6	5	4	3	2	1	0
DCBEVT0		DCAEVT1			DCAEVT0		

Table 10-85: Trip-Zone Digital Compare Event Select Register (TZDCSEL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11:9	DCBEVT1	RW	0x0	Digital compare B event 1 option 000: Event disabled 001: DCBL=low, DCBH=don't care 010: DCBL=high, DCBH=don't care 011: DCBL=don't care, DCBH=low 100: DCBL=don't care, DCBH=high 101: DCBL=low, DCBH=high 110: DCBL=high, DCBH=low 111: DCBL=high, DCBH=high
8:6	DCBEVT0	RW	0x0	Digital compare B event 0 option 000: Event disabled 001: DCBL=low, DCBH=don't care 010: DCBL=high, DCBH=don't care 011: DCBL=don't care, DCBH=low 100: DCBL=don't care, DCBH=high 101: DCBL=low, DCBH=high 110: DCBL=high, DCBH=low 111: DCBL=high, DCBH=high
5:3	DCAEVT1	RW	0x0	Digital compare A event 1 option 000: Event disabled 001: DCAL=low, DCAH=don't care 010: DCAL=high, DCAH=don't care 011: DCAL=don't care, DCAH=low 100: DCAL=don't care, DCAH=high 101: DCAL=low, DCAH=high 110: DCAL=high, DCAH=low 111: DCAL=high, DCAH=high

Bits	Field Name	Type	Reset	Description
2:0	DCAEVT0	RW	0x0	Digital compare A event 0 option 000: Event disabled 001: DCAL=low, DCAH=don't care 010: DCAL=high, DCAH=don't care 011: DCAL=don't care, DCAH=low 100: DCAL=don't care, DCAH=high 101: DCAL=low, DCAH=high 110: DCAL=high, DCAH=low 111: DCAL=high, DCAH=high

Table 10-86: Trip-Zone Output A Control Register (TZACTL) Layout

TZACTL (Trip-Zone Output A Control Register) Offset: 0x9C Default: 0x00000000
Access: PWM0 -> TZACTL.all

31	30	29	28	27	26	25	24
RESERVED_31_18							
23	22	21	20	19	18	17	16
RESERVED_31_18						DCAEVT1D	
15	14	13	12	11	10	9	8
DCAEVT1D	DCAEVT1U			DCAEVT0D			DCAEVT0U
7	6	5	4	3	2	1	0
DCAEVT0U		TZAD			TZAU		

Table 10-87: Trip-Zone Output A Control Register (TZACTL) Description

Bits	Field Name	Type	Reset	Description
31:18	RESERVED_31_18	RO	0x0	Reserved.
17:15	DCAEVT1D	RW	0x0	Action on output A when a digital compare A event 1 occurs and TBCNT is counting down 000: Set output A to high-impedance 001: Force output A to low state 010: Force output A to high state 011: Force output A to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output A

Bits	Field Name	Type	Reset	Description
14:12	DCAEVT1U	RW	0x0	Action on output A when a digital compare A event 1 occurs and TBCNT is counting up 000: Set output A to high-impedance 001: Force output A to low state 010: Force output A to high state 011: Force output A to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output A
11:9	DCAEVT0D	RW	0x0	Action on output A when a digital compare A event 0 occurs and TBCNT is counting down 000: Set output A to high-impedance 001: Force output A to low state 010: Force output A to high state 011: Force output A to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output A
8:6	DCAEVT0U	RW	0x0	Action on output A when a digital compare A event 0 occurs and TBCNT is counting up 000: Set output A to high-impedance 001: Force output A to low state 010: Force output A to high state 011: Force output A to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output A

Bits	Field Name	Type	Reset	Description
5:3	TZAD	RW	0x0	Action on output A when a trip event occurs and TBCNT is counting down 000: Set output A to high-impedance 001: Force output A to low state 010: Force output A to high state 011: Force output A to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output A
2:0	TZAU	RW	0x0	Action on output A when a trip event occurs and TBCNT is counting up 000: Set output A to high-impedance 001: Force output A to low state 010: Force output A to high state 011: Force output A to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output A

Table 10-88: Trip-Zone Ouput B Control Register (TZBCTL) Layout

TZBCTL (Trip-Zone Ouput B Control Register) Offset: 0xA0 Default: 0x00000000 Access: PWM0 -> TZBCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_18							
23	22	21	20	19	18	17	16
RESERVED_31_18						DCBEVT1D	
15	14	13	12	11	10	9	8
DCBEVT1D	DCBEVT1U			DCBEVT0D			DCBEVT0U
7	6	5	4	3	2	1	0
DCBEVT0U		TZBD			TZBU		

Table 10-89: Trip-Zone Ouput B Control Register (TZBCTL) Description

Bits	Field Name	Type	Reset	Description
31:18	RESERVED_31_18	RO	0x0	Reserved.
17:15	DCBEVT1D	RW	0x0	Action on output B when a digital compare B event 1 occurs and TBCNT is counting down 000: Set output B to high-impedance 001: Force output B to high state 010: Force output B to low state 011: Force output B to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output B
14:12	DCBEVT1U	RW	0x0	Action on output B when a digital compare B event 1 occurs and TBCNT is counting up 000: Set output B to high-impedance 001: Force output B to low state 010: Force output B to high state 011: Force output B to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output B
11:9	DCBEVT0D	RW	0x0	Action on output B when a digital compare B event 0 occurs and TBCNT is counting down 000: Set output B to high-impedance 001: Force output B to low state 010: Force output B to high state 011: Force output B to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output B

Bits	Field Name	Type	Reset	Description
8:6	DCBEVT0U	RW	0x0	Action on output B when a digital compare B event 0 occurs and TBCNT is counting up 000: Set output B to high-impedance 001: Force output B to low state 010: Force output B to high state 011: Force output B to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output B
5:3	TZBD	RW	0x0	Action on output B when a trip event occurs and TBCNT is counting down 000: Set output B to high-impedance 001: Force output B to low state 010: Force output B to high state 011: Force output B to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output B
2:0	TZBU	RW	0x0	Action on output B when a trip event occurs and TBCNT is counting up 000: Set output B to high-impedance 001: Force output B to low state 010: Force output B to high state 011: Force output B to invert current state (high->low, low->high) 100: 101: 110: 111: Do nothing for output B

Table 10-90: Trip-Zone Flag Register (TZIF) Layout

TZIF (Trip-Zone Flag Register) Offset: 0xA4 Default: 0x00000000 Access: PWM0 -> TZIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	INT	DCBEVT1	DCBEVT0	DCAEVT1	DCAEVT0	CBC	OST

Table 10-91: Trip-Zone Flag Register (TZIF) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	INT	RO	0x0	Latched trip interrupt status flag 0: No interrupt has been generated 1: Trip-zone interrupt was generated. No further trip-zone interrupt will be generated until this flag is cleared by writing 1 to TZIC.INT. If TZIF[5:0]&TZIE is not 0, another interrupt pulse will be generated when this flag is cleared.
5	DCBEVT1	RO	0x0	Latched digital compare B event 1 status flag 0: DCBEVT1 did not occur 1: DCBEVT1 has occurred. It is cleared by writing 1 to TZIC.DCBEVT1.
4	DCBEVT0	RO	0x0	Latched digital compare B event 0 status flag 0: DCBEVT0 did not occur 1: DCBEVT0 has occurred. It is cleared by writing 1 to TZIC.DCBEVT0.
3	DCAEVT1	RO	0x0	Latched digital compare A event 1 status flag 0: DCAEVT1 did not occur 1: DCAEVT1 has occurred. It is cleared by writing 1 to TZIC.DCAEVT1.
2	DCAEVT0	RO	0x0	Latched digital compare A event 0 status flag 0: DCAEVT0 did not occur 1: DCAEVT0 has occurred. It is cleared by writing 1 to TZIC.DCAEVT0.

Bits	Field Name	Type	Reset	Description
1	CBC	RO	0x0	Latched cycle-by-cycle trip event status flag 0: No cycle-by-cycle trip event has occurred 1: A cycle-by-cycle trip event occurred. It will remain set until it is manually cleared by wring 1 to TZCLR.CBC. If the cycle-by-cycle trip event is still present when the CBC bit is cleared, it will be immediately set again.
0	OST	RO	0x0	Latched one-shot trip event status flag 0: No one-shot trip event has occurred 1: A one-shot trip event occurred. It is cleared by writing 1 to TZCLR.OST.

Table 10-92: Trip-Zone Clear Register (TZIC) Layout

TZIC (Trip-Zone Clear Register) Offset: 0xA8 Default: 0x00000000 Access: PWM0 -> TZIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							CBCCLRMODE
7 CBCCLRMODE	6 INT	5 DCBEVT1	4 DCBEVT0	3 DCAEVT1	2 DCAEVT0	1 CBC	0 OST

Table 10-93: Trip-Zone Clear Register (TZIC) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:7	CBCCLRMODE	RW	0x0	Latched CBC trip status clear mode 00: Clear CBC trip on TBCNT=0 01: Clear CBC trip on TBCNT=TBPRD 10: Clear CBC trip on TBCNT=0 or TBCNT=TBPRD 11: Clear disabled. i.e. Never clear the CBC trip
6	INT	W1C	0x0	Clear global interrupt status flag 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears TZIF.INT and TZ interrupt
5	DCBEVT1	W1C	0x0	Clear digital compare B event 1 status flag 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears TZIF.DCBEVT1

Bits	Field Name	Type	Reset	Description
4	DCBEVT0	W1C	0x0	Clear digital compare B event 0 status flag 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears TZIF.DCBEVT0
3	DCAEVT1	W1C	0x0	Clear digital compare A event 1 status flag 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears TZIF.DCAEVT1
2	DCAEVT0	W1C	0x0	Clear digital compare A event 0 status flag 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears TZIF.DCAEVT0
1	CBC	W1C	0x0	Clear cycle-by-cycle trip status flag 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears TZIF.CBC
0	OST	W1C	0x0	Clear one-shot trip status flag 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears TZIF.OST

Table 10-94: Trip-Zone Interrupt Enable Register (TZIE) Layout

TZIE (Trip-Zone Interrupt Enable Register) Offset: 0xAC Default: 0x00000000 Access: PWM0 -> TZIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		DCBEVT1	DCBEVT0	DCAEVT1	DCAEVT0	CBC	OST

Table 10-95: Trip-Zone Interrupt Enable Register (TZIE) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	DCBEVT1	RW	0x0	Digital compare B event 1 interrupt enable 0: Disable 1: Enable
4	DCBEVT0	RW	0x0	Digital compare B event 0 interrupt enable 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
3	DCAEVT1	RW	0x0	Digital compare A event 1 interrupt enable 0: Disable 1: Enable
2	DCAEVT0	RW	0x0	Digital compare A event 0 interrupt enable 0: Disable 1: Enable
1	CBC	RW	0x0	Cycle-by-cycle trip-zone interrupt enable 0: Disable 1: Enable
0	OST	RW	0x0	One-shot trip-zone interrupt enable 0: Disable 1: Enable

Table 10-96: Trip-Zone Force Register (TZFRC) Layout

TZFRC (Trip-Zone Force Register) Offset: 0xB0 Default: 0x00000000 Access: PWM0 -> TZFRC.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		DCBEVT1	DCBEVT0	DCAEVT1	DCAEVT0	CBC	OST

Table 10-97: Trip-Zone Force Register (TZFRC) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	DCBEVT1	W1S	0x0	Software force a digital compare B event 1 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 forces a digital compare B event 1 and sets TZIF.DCBEVT1
4	DCBEVT0	W1S	0x0	Software force a digital compare B event 0 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 forces a digital compare B event 0 and sets TZIF.DCBEVT0
3	DCAEVT1	W1S	0x0	Software force a digital compare A event 1 0: Write a 0 has no effect and always reads back a 0

Bits	Field Name	Type	Reset	Description
				1: Write a 1 forces a digital compare A event 1 and sets TZIF.DCAEVT1
2	DCAEVT0	W1S	0x0	Software force a digital compare A event 0 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 forces a digital compare A event 0 and sets TZIF.DCAEVT0
1	CBC	W1S	0x0	Software force a cycle-by-cycle trip event 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 forces a cycle-by-cycle trip event and sets TZIF.CBC
0	OST	W1S	0x0	Software force a one-shot trip event 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 forces a one-shot trip event and sets TZIF.OST

Table 10-98: Digital Compare AL Trip Select Register (DCALTRIPSEL) Layout

DCALTRIPSEL (Digital Compare AL Trip Select Register) Offset: 0xB4 Default: 0x00000000
Access: PWM0 -> DCALTRIPSEL.all

31	30	29	28	27	26	25	24
RESERVED_31_26						COMP4H	COMP4L
23	22	21	20	19	18	17	16
COMP3H	COMP3L	COMP2H	COMP2L	COMP1H	COMP1L	COMP0H	COMP0L
15	14	13	12	11	10	9	8
RESERVED_15_14		ADCPPU5TZ	ADCPPU4TZ	ADCPPU3TZ	ADCPPU2TZ	ADCPPU1TZ	ADCPPU0TZ
7	6	5	4	3	2	1	0
RESERVED_7	EPWRTZ1	EPWRTZ0	TZ4	TZ3	TZ2	TZ1	TZ0

Table 10-99: Digital Compare AL Trip Select Register (DCALTRIPSEL) Description

Bits	Field Name	Type	Reset	Description
31:26	RESERVED_31_26	RO	0x0	Reserved.
25	COMP4H	RW	0x0	Enable COMP4H as DCAL trip event (ORed) 0: Disable 1: Enable
24	COMP4L	RW	0x0	Enable COMP4L as DCAL trip event (ORed) 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
23	COMP3H	RW	0x0	Enable COMP3H as DCAL trip event (ORed) 0: Disable 1: Enable
22	COMP3L	RW	0x0	Enable COMP3L as DCAL trip event (ORed) 0: Disable 1: Enable
21	COMP2H	RW	0x0	Enable COMP2H as DCAL trip event (ORed) 0: Disable 1: Enable
20	COMP2L	RW	0x0	Enable COMP2L as DCAL trip event (ORed) 0: Disable 1: Enable
19	COMP1H	RW	0x0	Enable COMP1H as DCAL trip event (ORed) 0: Disable 1: Enable
18	COMP1L	RW	0x0	Enable COMP1L as DCAL trip event (ORed) 0: Disable 1: Enable
17	COMP0H	RW	0x0	Enable COMP0H as DCAL trip event (ORed) 0: Disable 1: Enable
16	COMP0L	RW	0x0	Enable COMP0L as DCAL trip event (ORed) 0: Disable 1: Enable
15:14	RESERVED_15_14	RO	0x0	Reserved.
13	ADCPPU5TZ	RW	0x0	Enable ADCPPU5TZ as DCAL trip event (ORed) 0: Disable 1: Enable
12	ADCPPU4TZ	RW	0x0	Enable ADCPPU4TZ as DCAL trip event (ORed) 0: Disable 1: Enable
11	ADCPPU3TZ	RW	0x0	Enable ADCPPU3TZ as DCAL trip event (ORed) 0: Disable 1: Enable
10	ADCPPU2TZ	RW	0x0	Enable ADCPPU2TZ as DCAL trip event (ORed) 0: Disable 1: Enable
9	ADCPPU1TZ	RW	0x0	Enable ADCPPU1TZ as DCAL trip event (ORed) 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
8	ADCPPU0TZ	RW	0x0	Enable ADCPPU0TZ as DCAL trip event (ORed) 0: Disable 1: Enable
7	RESERVED_7	RO	0x0	Reserved.
6	EPWRTZ1	RW	0x0	Enable EPWRTZ1 as DCAL trip event (ORed) 0: Disable 1: Enable
5	EPWRTZ0	RW	0x0	Enable EPWRTZ0 as DCAL trip event (ORed) 0: Disable 1: Enable
4	TZ4	RW	0x0	Enable TZ4 as DCAL trip event (ORed) 0: Disable 1: Enable
3	TZ3	RW	0x0	Enable TZ3 as DCAL trip event (ORed) 0: Disable 1: Enable
2	TZ2	RW	0x0	Enable TZ2 as DCAL trip event (ORed) 0: Disable 1: Enable
1	TZ1	RW	0x0	Enable TZ1 as DCAL trip event (ORed) 0: Disable 1: Enable
0	TZ0	RW	0x0	Enable TZ0 as DCAL trip event (ORed) 0: Disable 1: Enable

Table 10-100: Digital Compare AH Trip Select Register (DCAHTRIPSEL) Layout

DCAHTRIPSEL (Digital Compare AH Trip Select Register) Offset: 0xB8 Default: 0x00000000
Access: PWM0 -> DCAHTRIPSEL.all

31	30	29	28	27	26	25	24
RESERVED_31_26						COMP4H	COMP4L
23	22	21	20	19	18	17	16
COMP3H	COMP3L	COMP2H	COMP2L	COMP1H	COMP1L	COMP0H	COMP0L
15	14	13	12	11	10	9	8
RESERVED_15_14		ADCPPU5TZ	ADCPPU4TZ	ADCPPU3TZ	ADCPPU2TZ	ADCPPU1TZ	ADCPPU0TZ
7	6	5	4	3	2	1	0
RESERVED_7	EPWRTZ1	EPWRTZ0	TZ4	TZ3	TZ2	TZ1	TZ0

Table 10-101: Digital Compare AH Trip Select Register (DCAHTRIPSEL) Description

Bits	Field Name	Type	Reset	Description
31:26	RESERVED_31_26	RO	0x0	Reserved.
25	COMP4H	RW	0x0	Enable COMP4H as DCAH trip event (ORed) 0: Disable 1: Enable
24	COMP4L	RW	0x0	Enable COMP4L as DCAH trip event (ORed) 0: Disable 1: Enable
23	COMP3H	RW	0x0	Enable COMP3H as DCAH trip event (ORed) 0: Disable 1: Enable
22	COMP3L	RW	0x0	Enable COMP3L as DCAH trip event (ORed) 0: Disable 1: Enable
21	COMP2H	RW	0x0	Enable COMP2H as DCAH trip event (ORed) 0: Disable 1: Enable
20	COMP2L	RW	0x0	Enable COMP2L as DCAH trip event (ORed) 0: Disable 1: Enable
19	COMP1H	RW	0x0	Enable COMP1H as DCAH trip event (ORed) 0: Disable 1: Enable
18	COMP1L	RW	0x0	Enable COMP1L as DCAH trip event (ORed) 0: Disable 1: Enable
17	COMP0H	RW	0x0	Enable COMP0H as DCAH trip event (ORed) 0: Disable 1: Enable
16	COMP0L	RW	0x0	Enable COMP0L as DCAH trip event (ORed) 0: Disable 1: Enable
15:14	RESERVED_15_14	RO	0x0	Reserved.
13	ADCPPU5TZ	RW	0x0	Enable ADCPPU5TZ as DCAH trip event (ORed) 0: Disable 1: Enable
12	ADCPPU4TZ	RW	0x0	Enable ADCPPU4TZ as DCAH trip event (ORed) 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
11	ADCPPU3TZ	RW	0x0	Enable ADCPPU3TZ as DCAH trip event (ORed) 0: Disable 1: Enable
10	ADCPPU2TZ	RW	0x0	Enable ADCPPU2TZ as DCAH trip event (ORed) 0: Disable 1: Enable
9	ADCPPU1TZ	RW	0x0	Enable ADCPPU1TZ as DCAH trip event (ORed) 0: Disable 1: Enable
8	ADCPPU0TZ	RW	0x0	Enable ADCPPU0TZ as DCAH trip event (ORed) 0: Disable 1: Enable
7	RESERVED_7	RO	0x0	Reserved.
6	EPWRTZ1	RW	0x0	Enable EPWRTZ1 as DCAH trip event (ORed) 0: Disable 1: Enable
5	EPWRTZ0	RW	0x0	Enable EPWRTZ0 as DCAH trip event (ORed) 0: Disable 1: Enable
4	TZ4	RW	0x0	Enable TZ4 as DCAH trip event (ORed) 0: Disable 1: Enable
3	TZ3	RW	0x0	Enable TZ3 as DCAH trip event (ORed) 0: Disable 1: Enable
2	TZ2	RW	0x0	Enable TZ2 as DCAH trip event (ORed) 0: Disable 1: Enable
1	TZ1	RW	0x0	Enable TZ1 as DCAH trip event (ORed) 0: Disable 1: Enable
0	TZ0	RW	0x0	Enable TZ0 as DCAH trip event (ORed) 0: Disable 1: Enable

Table 10-102: Digital Compare BL Trip Select Register (DCBLTRIPSEL) Layout

DCBLTRIPSEL (Digital Compare BL Trip Select Register) Offset: 0xBC Default: 0x00000000 Access: PWM0 -> DCBLTRIPSEL.all							
31	30	29	28	27	26	25	24
RESERVED_31_26						COMP4H	COMP4L
23	22	21	20	19	18	17	16
COMP3H	COMP3L	COMP2H	COMP2L	COMP1H	COMP1L	COMP0H	COMP0L
15	14	13	12	11	10	9	8
RESERVED_15_14		ADCPPU5TZ	ADCPPU4TZ	ADCPPU3TZ	ADCPPU2TZ	ADCPPU1TZ	ADCPPU0TZ
7	6	5	4	3	2	1	0
RESERVED_7	EPWRTZ1	EPWRTZ0	TZ4	TZ3	TZ2	TZ1	TZ0

Table 10-103: Digital Compare BL Trip Select Register (DCBLTRIPSEL) Description

Bits	Field Name	Type	Reset	Description
31:26	RESERVED_31_26	RO	0x0	Reserved.
25	COMP4H	RW	0x0	Enable COMP4H as DCBL trip event (ORed) 0: Disable 1: Enable
24	COMP4L	RW	0x0	Enable COMP4L as DCBL trip event (ORed) 0: Disable 1: Enable
23	COMP3H	RW	0x0	Enable COMP3H as DCBL trip event (ORed) 0: Disable 1: Enable
22	COMP3L	RW	0x0	Enable COMP3L as DCBL trip event (ORed) 0: Disable 1: Enable
21	COMP2H	RW	0x0	Enable COMP2H as DCBL trip event (ORed) 0: Disable 1: Enable
20	COMP2L	RW	0x0	Enable COMP2L as DCBL trip event (ORed) 0: Disable 1: Enable
19	COMP1H	RW	0x0	Enable COMP1H as DCBL trip event (ORed) 0: Disable 1: Enable
18	COMP1L	RW	0x0	Enable COMP1L as DCBL trip event (ORed) 0: Disable 1: Enable
17	COMP0H	RW	0x0	Enable COMP0H as DCBL trip event (ORed) 0: Disable 1: Enable
16	COMP0L	RW	0x0	Enable COMP0L as DCBL trip event (ORed) 0: Disable 1: Enable
15:14	RESERVED_15_14	RO	0x0	Reserved.
13	ADCPPU5TZ	RW	0x0	Enable ADCPPU5TZ as DCBL trip event (ORed) 0: Disable 1: Enable
12	ADCPPU4TZ	RW	0x0	Enable ADCPPU4TZ as DCBL trip event (ORed) 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
11	ADCPPU3TZ	RW	0x0	Enable ADCPPU3TZ as DCBL trip event (ORed) 0: Disable 1: Enable
10	ADCPPU2TZ	RW	0x0	Enable ADCPPU2TZ as DCBL trip event (ORed) 0: Disable 1: Enable
9	ADCPPU1TZ	RW	0x0	Enable ADCPPU1TZ as DCBL trip event (ORed) 0: Disable 1: Enable
8	ADCPPU0TZ	RW	0x0	Enable ADCPPU0TZ as DCBL trip event (ORed) 0: Disable 1: Enable
7	RESERVED_7	RO	0x0	Reserved.
6	EPWRTZ1	RW	0x0	Enable EPWRTZ1 as DCBL trip event (ORed) 0: Disable 1: Enable
5	EPWRTZ0	RW	0x0	Enable EPWRTZ0 as DCBL trip event (ORed) 0: Disable 1: Enable
4	TZ4	RW	0x0	Enable TZ4 as DCBL trip event (ORed) 0: Disable 1: Enable
3	TZ3	RW	0x0	Enable TZ3 as DCBL trip event (ORed) 0: Disable 1: Enable
2	TZ2	RW	0x0	Enable TZ2 as DCBL trip event (ORed) 0: Disable 1: Enable
1	TZ1	RW	0x0	Enable TZ1 as DCBL trip event (ORed) 0: Disable 1: Enable
0	TZ0	RW	0x0	Enable TZ0 as DCBL trip event (ORed) 0: Disable 1: Enable

Table 10-104: Digital Compare BH Trip Select Register (DCBHTRIPSEL) Layout

DCBHTRIPSEL (Digital Compare BH Trip Select Register) Offset: 0xC0 Default: 0x00000000 Access: PWM0 -> DCBHTRIPSEL.all							
31	30	29	28	27	26	25	24
RESERVED_31_26						COMP4H	COMP4L
23	22	21	20	19	18	17	16
COMP3H	COMP3L	COMP2H	COMP2L	COMP1H	COMP1L	COMP0H	COMP0L
15	14	13	12	11	10	9	8
RESERVED_15_14		ADCPPU5TZ	ADCPPU4TZ	ADCPPU3TZ	ADCPPU2TZ	ADCPPU1TZ	ADCPPU0TZ
7	6	5	4	3	2	1	0
RESERVED_7	EPWRTZ1	EPWRTZ0	TZ4	TZ3	TZ2	TZ1	TZ0

Table 10-105: Digital Compare BH Trip Select Register (DCBHTRIPSEL) Description

Bits	Field Name	Type	Reset	Description
31:26	RESERVED_31_26	RO	0x0	Reserved.
25	COMP4H	RW	0x0	Enable COMP4H as DCBH trip event (ORed) 0: Disable 1: Enable
24	COMP4L	RW	0x0	Enable COMP4L as DCBH trip event (ORed) 0: Disable 1: Enable
23	COMP3H	RW	0x0	Enable COMP3H as DCBH trip event (ORed) 0: Disable 1: Enable
22	COMP3L	RW	0x0	Enable COMP3L as DCBH trip event (ORed) 0: Disable 1: Enable
21	COMP2H	RW	0x0	Enable COMP2H as DCBH trip event (ORed) 0: Disable 1: Enable
20	COMP2L	RW	0x0	Enable COMP2L as DCBH trip event (ORed) 0: Disable 1: Enable
19	COMP1H	RW	0x0	Enable COMP1H as DCBH trip event (ORed) 0: Disable 1: Enable
18	COMP1L	RW	0x0	Enable COMP1L as DCBH trip event (ORed) 0: Disable 1: Enable
17	COMP0H	RW	0x0	Enable COMP0H as DCBH trip event (ORed) 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
16	COMP0L	RW	0x0	Enable COMP0L as DCBH trip event (ORed) 0: Disable 1: Enable
15:14	RESERVED_15_14	RO	0x0	Reserved.
13	ADCPPU5TZ	RW	0x0	Enable ADCPPU5TZ as DCBH trip event (ORed) 0: Disable 1: Enable
12	ADCPPU4TZ	RW	0x0	Enable ADCPPU4TZ as DCBH trip event (ORed) 0: Disable 1: Enable
11	ADCPPU3TZ	RW	0x0	Enable ADCPPU3TZ as DCBH trip event (ORed) 0: Disable 1: Enable
10	ADCPPU2TZ	RW	0x0	Enable ADCPPU2TZ as DCBH trip event (ORed) 0: Disable 1: Enable
9	ADCPPU1TZ	RW	0x0	Enable ADCPPU1TZ as DCBH trip event (ORed) 0: Disable 1: Enable
8	ADCPPU0TZ	RW	0x0	Enable ADCPPU0TZ as DCBH trip event (ORed) 0: Disable 1: Enable
7	RESERVED_7	RO	0x0	Reserved.
6	EPWRTZ1	RW	0x0	Enable EPWRTZ1 as DCBH trip event (ORed) 0: Disable 1: Enable
5	EPWRTZ0	RW	0x0	Enable EPWRTZ0 as DCBH trip event (ORed) 0: Disable 1: Enable
4	TZ4	RW	0x0	Enable TZ4 as DCBH trip event (ORed) 0: Disable 1: Enable
3	TZ3	RW	0x0	Enable TZ3 as DCBH trip event (ORed) 0: Disable 1: Enable
2	TZ2	RW	0x0	Enable TZ2 as DCBH trip event (ORed) 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
1	TZ1	RW	0x0	Enable TZ1 as DCBH trip event (ORed) 0: Disable 1: Enable
0	TZ0	RW	0x0	Enable TZ0 as DCBH trip event (ORed) 0: Disable 1: Enable

Table 10-106: Digital Compare A Control Register (DCACTL) Layout

DCACTL (Digital Compare A Control Register) Offset: 0xC4 Default: 0x00000000
Access: PWM0 -> DCACTL.all

31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		EVT1SRCASYNC	EVT1SRCSEL	EVT0SYNCE	EVT0SOCE	EVT0SRCASYNC	EVT0SRCSEL

Table 10-107: Digital Compare A Control Register (DCACTL) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	EVT1SRCASYNC	RW	0x0	DCAEVT1 source signal is asynchronous/synchronous 0: Synchronous 1: Asynchronous
4	EVT1SRCSEL	RW	0x0	DCAEVT1 source signal select 0: Source signal is DCAEVT1 1: Source signal is DCEVTFILT
3	EVT0SYNCE	RW	0x0	DCAEVT0 SYNC generation 0: Disable 1: Enable
2	EVT0SOCE	RW	0x0	DCAEVT0 SOC generation 0: Disable 1: Enable
1	EVT0SRCASYNC	RW	0x0	DCAEVT0 source signal is asynchronous/synchronous (1/0) 0: Synchronous 1: Asynchronous

Bits	Field Name	Type	Reset	Description
0	EVT0SRCSEL	RW	0x0	DCAEVT0 source signal select 0: Source signal is DCAEVT0 1: Source signal is DCEVTFILT

Table 10-108: Digital Compare B Control Register (DCBCTL) Layout

DCBCTL (Digital Compare B Control Register) Offset: 0xC8 Default: 0x00000000 Access: PWM0 -> DCBCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		EVT1SRCASYNC	EVT1SRCSEL	EVT0SYNCE	EVT0SOCE	EVT0SRCASYNC	EVT0SRCSEL

Table 10-109: Digital Compare B Control Register (DCBCTL) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	EVT1SRCASYNC	RW	0x0	DCBEVT1 source signal is asynchronous/synchronous (1/0) 0: Synchronous 1: Asynchronous
4	EVT1SRCSEL	RW	0x0	DCBEVT1 source signal select 0: Source signal is DCBEVT1 1: Source signal is DCEVTFILT
3	EVT0SYNCE	RW	0x0	DCBEVT0 SYNC generation 0: Disable 1: Enable
2	EVT0SOCE	RW	0x0	DCBEVT0 SOC generation 0: Disable 1: Enable
1	EVT0SRCASYNC	RW	0x0	DCBEVT0 source signal is asynchronous/synchronous 0: Synchronous 1: Asynchronous
0	EVT0SRCSEL	RW	0x0	DCBEVT0 source signal select 0: Source signal is DCBEVT0 1: Source signal is DCEVTFILT

Table 10-110: Digital Compare Filter Register (DCFCTL) Layout

DCFCTL (Digital Compare Filter Register) Offset: 0xCC Default: 0x00000010 Access: PWM0 -> DCFCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				WIN5EN	WIN4EN	WIN3EN	WIN2EN
7	6	5	4	3	2	1	0
WIN1EN	WIN0EN	PULSESEL		BLANKINV	BLANKEN	SRCSEL	

Table 10-111: Digital Compare Filter Register (DCFCTL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11	WIN5EN	RW	0x0	Enable blank window from PWM5 (ORed) 0: Disable 1: Enable
10	WIN4EN	RW	0x0	Enable blank window from PWM4 (ORed) 0: Disable 1: Enable
9	WIN3EN	RW	0x0	Enable blank window from PWM3 (ORed) 0: Disable 1: Enable
8	WIN2EN	RW	0x0	Enable blank window from PWM2 (ORed) 0: Disable 1: Enable
7	WIN1EN	RW	0x0	Enable blank window from PWM1 (ORed) 0: Disable 1: Enable
6	WIN0EN	RW	0x0	Enable blank window from PWM0 (ORed) 0: Disable 1: Enable
5:4	PULSESEL	RW	0x1	Pulse select for blanking and capture alignment for this module 00: Align on TBCNT=0 01: Align on TBCNT=TBPRD 10: Align on TBCNT=0 or TBCNT=TBPRD 11: Reserved
3	BLANKINV	RW	0x0	blanking window inversion 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
2	BLANKEN	RW	0x0	blanking window 0: Disable 1: Enable
1:0	SRCSEL	RW	0x0	DCEVTFILT signal source select 00: Source is DCAEVT0 01: Source is DCAEVT1 10: Source is DCBEVT0 11: Source is DCBEVT1

Table 10-112: Digital Compare Filter Offset Register (DCFOFFSET) Layout

DCFOFFSET (Digital Compare Filter Offset Register) Offset: 0xD0 Default: 0x00000000 Access: PWM0 -> DCFOFFSET.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-113: Digital Compare Filter Offset Register (DCFOFFSET) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Blanking window offset It defines the number of TBCLK cycles from the blanking window reference specified by DCFCTL.PULSESEL to the point when the blanking window is applied.

Table 10-114: Digital Compare Filter Offset Counter Register (DCFOFFSETCNT) Layout

DCFOFFSETCNT (Digital Compare Filter Offset Counter Register) Offset: 0xD4 Default: 0x00000000 Access: PWM0 -> DCFOFFSETCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-115: Digital Compare Filter Offset Counter Register (DCFOFFSETCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Blanking offset counter It is read only and indicates the current value of the offset counter. The counter counts down to zero and then stops until it is re-loaded on the next event defined by DCFCTL.PULSESEL.

Table 10-116: Digital Compare Filter Window Register (DCFWINDOW) Layout

DCFWINDOW (Digital Compare Filter Window Register) Offset: 0xD8 Default: 0x00000000
Access: PWM0 -> DCFWINDOW.all

31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-117: Digital Compare Filter Window Register (DCFWINDOW) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Blanking window width in TBCLK cycles The blanking window begins when the offset counter expires. When this occurs, the window counter is loaded and begins to count down. If the blanking window is currently active and the offset counter expires, the blanking window counter is restarted. The blanking window can cross a PWM period boundary. No blanking window is generated when it is set to zero.

Table 10-118: Digital Compare Filter Window Counter Register (DCFWINDOWCNT) Layout

DCFWINDOWCNT (Digital Compare Filter Window Counter Register) Offset: 0xDC Default: 0x00000000 Access: PWM0 -> DCFWINDOWCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-119: Digital Compare Filter Window Counter Register (DCFWINDOWCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Blanking window counter It is read only and indicates the current value of the window counter. The counter counts down to zero and then stops until it is re-loaded when the offset counter reaches zero again.

Table 10-120: Digital Compare Capture Control Register (DCCAPCTL) Layout

DCCAPCTL (Digital Compare Capture Control Register) Offset: 0xE0 Default: 0x00000000 Access: PWM0 -> DCCAPCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				CAPSTS	CAPCLR	DIRECT	CAPEN

Table 10-121: Digital Compare Capture Control Register (DCCAPCTL) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	CAPSTS	RO	0x0	Latched capture status 0: No DC capture event occurred 1: DC capture event has occurred
2	CAPCLR	W1C	0x0	Latched capture status clear 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears the latched status. This bit is self-cleared to 0
1	DIRECT	RW	0x0	DCCAP direct load mode 0: Shadow mode. The DCCAP active value is copied to shadow register on a TBCNT=TBPRD or TBCNT=zero event as defined by DCFCTL.PULSESEL. Reads of the DCCAP register always returns the shadow register contents. 1: Direct mode. Reads of the DCCAP register always returns the active value.
0	CAPEN	RW	0x0	TBCNT counter capture enable 0: Disable 1: Enable

Table 10-122: Digital Compare Counter Capture Register (DCCAP) Layout

DCCAP (Digital Compare Counter Capture Register) Offset: 0xE4 Default: 0x00000000 Access: PWM0 -> DCCAP.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 10-123: Digital Compare Counter Capture Register (DCCAP) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Captured TBCNT value upon a rising-edge of DCEVTFLT In shadow mode, the captured value is copied to the shadow register on a TBCNT=TBPRD or TBCNT=zero event as defined by DCFCTL.PULSESEL. Reads of the DCCAP register always returns the shadow register contents. In direct mode, reads of the DCCAP register always returns the active value.

Table 10-124: Event-Trigger Control Register (ETCTL) Layout

ETCTL (Event-Trigger Control Register) Offset: 0xE8 Default: 0x0007B16F Access: PWM0 -> ETCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_20							
23	22	21	20	19	18	17	16
RESERVED_31_20				SOCCEN	SOCCSEL		
15	14	13	12	11	10	9	8
SOCCSEL	SOCBEN	SOCBSEL				SOCAEN	SOCASEL
7	6	5	4	3	2	1	0
SOCASEL			INTEN	INTSEL			

Table 10-125: Event-Trigger Control Register (ETCTL) Description

Bits	Field Name	Type	Reset	Description
31:20	RESERVED_31_20	RO	0x0	Reserved.
19	SOCCEN	RW	0x0	ADC start of conversion C (SOCC) pulse enable 0: Disable 1: Enable
18:15	SOCCSEL	RW	0xF	SOCC event select 0000: Enable event TBCNT=0 0001: Enable event TBCNT=TBPRD 0010: Enable event TBCNT=0 or TBCNT=TBPRD 0011: Enable event TBCNT=CMPA when TBCNT is counting up 0100: Enable event TBCNT=CMPA when TBCNT is counting down 0101: Enable event TBCNT=CMPB when TBCNT is counting up 0110: Enable event TBCNT=CMPB when TBCNT is counting down 0111: Enable event TBCNT=CMPC when TBCNT is counting up 1000: Enable event TBCNT=CMPC when TBCNT is counting down 1001: Enable event TBCNT=CMPD when TBCNT is counting up 1010: Enable event TBCNT=CMPD when TBCNT is counting down 1011: Enable DCAEVT0.SOC event 1100: Enable DCBEVT0.SOC event 1101: 1110: 1111:
14	SOCBEN	RW	0x0	ADC start of conversion B (SOCB) pulse enable 0: Disable 1: Enable

13:10	SOCBSEL	RW	0xC	SOCB event select 0000: Select event TBCNT=0 0001: Select event TBCNT=TBPRD 0010: Select event TBCNT=0 or TBCNT=TBPRD 0011: Select event TBCNT=CMPA when TBCNT is counting up 0100: Select event TBCNT=CMPA when TBCNT is counting down 0101: Select event TBCNT=CMPB when TBCNT is counting up 0110: Select event TBCNT=CMPB when TBCNT is counting down 0111: Select event TBCNT=CMPC when TBCNT is counting up 1000: Select event TBCNT=CMPC when TBCNT is counting down 1001: Select event TBCNT=CMPD when TBCNT is counting up 1010: Select event TBCNT=CMPD when TBCNT is counting down 1011: Select DCAEVT0.SOC event 1100: Select DCBEVT0.SOC event 1101: 1110: 1111:
9	SOCAEN	RW	0x0	ADC start of conversion A (SOCA) pulse enable 0: Disable 1: Enable

8:5	SOCASEL	RW	0xB	SOCA event select 0000: Select event TBCNT=0 0001: Select event TBCNT=TBPRD 0010: Select event TBCNT=0 or TBCNT=TBPRD 0011: Select event TBCNT=CMPA when TBCNT is counting up 0100: Select event TBCNT=CMPA when TBCNT is counting down 0101: Select event TBCNT=CMPB when TBCNT is counting up 0110: Select event TBCNT=CMPB when TBCNT is counting down 0111: Select event TBCNT=CMPC when TBCNT is counting up 1000: Select event TBCNT=CMPC when TBCNT is counting down 1001: Select event TBCNT=CMPD when TBCNT is counting up 1010: Select event TBCNT=CMPD when TBCNT is counting down 1011: Select DCAEVT0.SOC event 1100: Select DCBEVT0.SOC event 1101: 1110: 1111:
4	INTEN	RW	0x0	PWM interrupt enable 0: Disable 1: Enable

3:0	INTSEL	RW	0xF	PWM interrupt select 0000: Select event TBCNT=0 0001: Select event TBCNT=TBPRD 0010: Select event TBCNT=0 or TBCNT=TBPRD 0011: Select event TBCNT=CMPA when TBCNT is counting up 0100: Select event TBCNT=CMPA when TBCNT is counting down 0101: Select event TBCNT=CMPB when TBCNT is counting up 0110: Select event TBCNT=CMPB when TBCNT is counting down 0111: Select event TBCNT=CMPC when TBCNT is counting up 1000: Select event TBCNT=CMPC when TBCNT is counting down 1001: Select event TBCNT=CMPD when TBCNT is counting up 1010: Select event TBCNT=CMPD when TBCNT is counting down 1011: Select DCAEVT0.SOC event 1100: Select DCBEVT0.SOC event 1101: 1110: 1111:

Table 10-126: Event-Trigger Prescale Register (ETPS) Layout

ETPS (Event-Trigger Prescale Register) Offset: 0xEC Default: 0x00000000 Access: PWM0 -> ETPS.all							
31	30	29	28	27	26	25	24
SOCCCNT				SOCCPRD			
23	22	21	20	19	18	17	16
SOCBCNT				SOCBPRD			
15	14	13	12	11	10	9	8
SOCACNT				SOCAPRD			
7	6	5	4	3	2	1	0
INTCNT				INTPRD			

Table 10-127: Event-Trigger Prescale Register (ETPS) Description

Bits	Field Name	Type	Reset	Description
31:28	SOCCCNT	RO	0x0	PWM ADC start-of-conversion C event (SOCC) counter It indicates how many selected ETSEL.SOCCSEL events have occurred. It is automatically cleared when an SOCC pulse is generated. It can be initialized via write. 0000: No events have occurred 0001: 1 event has occurred 0010: 2 events have occurred 0011: 3 events have occurred 0100: 4 events have occurred 0101: 5 events have occurred 0110: 6 events have occurred 0111: 7 events have occurred 1000: 8 events have occurred 1001: 9 events have occurred 1010: 10 events have occurred 1011: 11 events have occurred 1100: 12 events have occurred 1101: 13 events have occurred 1110: 14 events have occurred 1111: 15 events have occurred

Bits	Field Name	Type	Reset	Description
27:24	SOCCPRD	RW	0x0	PWM ADC start-of-conversion C event (SOCC) period select SOCC is generated upon SOCCPRD<=SOCCCNT and ETCTL[SOCCEN]=1. No SOCC will be generated if SOCCPRD=0 0000: Disable SOCC event counter. No SOCC pulse will be generated. 0001: Generate SOCC pulse when ETPS.SOCCCNT=0001 (first event) 0010: Generate SOCC pulse when ETPS.SOCCCNT=0010 (second event) 0011: Generate SOCC pulse when ETPS.SOCCCNT=0011 (third event) 0100: Generate SOCC pulse when ETPS.SOCCCNT=0100 (fourth event) 0101: Generate SOCC pulse when ETPS.SOCCCNT=0101 (fifth event) 0110: Generate SOCC pulse when ETPS.SOCCCNT=0110(sixth event) 0111: Generate SOCC pulse when ETPS.SOCCCNT=0111 (seventh event) 1000: Generate SOCC pulse when ETPS.SOCCCNT=1000 (eighth event) 1001: Generate SOCC pulse when ETPS.SOCCCNT=1001 (ninth event) 1010: Generate SOCC pulse when ETPS.SOCCCNT=1010 (tenth event) 1011: Generate SOCC pulse when ETPS.SOCCCNT=1011 (eleventh event) 1100: Generate SOCC pulse when ETPS.SOCCCNT=1100(twelfth event) 1101: Generate SOCC pulse when ETPS.SOCCCNT=1101 (thirteenth event) 1110: Generate SOCC pulse when ETPS.SOCCCNT=1110(fourteenth event) 1111: Generate SOCC pulse when ETPS.SOCCCNT=1111 (fifteenth event)

Bits	Field Name	Type	Reset	Description
23:20	SOCBCNT	RO	0x0	PWM ADC start-of-conversion B event (SOCB) counter It indicates how many selected ETSEL.SOCBSEL events have occurred. It is automatically cleared when an SOCB pulse is generated. It can be initialized via write. 0000: No events have occurred 0001: 1 event has occurred 0010: 2 events have occurred 0011: 3 events have occurred 0100: 4 events have occurred 0101: 5 events have occurred 0110: 6 events have occurred 0111: 7 events have occurred 1000: 8 events have occurred 1001: 9 events have occurred 1010: 10 events have occurred 1011: 11 events have occurred 1100: 12 events have occurred 1101: 13 events have occurred 1110: 14 events have occurred 1111: 15 events have occurred

Bits	Field Name	Type	Reset	Description
19:16	SOCBPRD	RW	0x0	PWM ADC start-of-conversion B event (SOCB) period select SOCB is generated upon SOCBPRD<=SOCBCNT and ETCTL[SOCBEN]=1. No SOCB will be generated if SOCBPRD=0 0000: Disable SOCB event counter. No SOCB pulse will be generated. 0001: Generate SOCB pulse when ETPS.SOCBCNT=0001 (first event) 0010: Generate SOCB pulse when ETPS.SOCBCNT=0010 (second event) 0011: Generate SOCB pulse when ETPS.SOCBCNT=0011 (third event) 0100: Generate SOCB pulse when ETPS.SOCBCNT=0100 (fourth event) 0101: Generate SOCB pulse when ETPS.SOCBCNT=0101 (fifth event) 0110: Generate SOCB pulse when ETPS.SOCBCNT=0110(sixth event) 0111: Generate SOCB pulse when ETPS.SOCBCNT=0111 (seventh event) 1000: Generate SOCB pulse when ETPS.SOCBCNT=1000 (eighth event) 1001: Generate SOCB pulse when ETPS.SOCBCNT=1001 (ninth event) 1010: Generate SOCB pulse when ETPS.SOCBCNT=1010 (tenth event) 1011: Generate SOCB pulse when ETPS.SOCBCNT=1011 (eleventh event) 1100: Generate SOCB pulse when ETPS.SOCBCNT=1100(twelfth event) 1101: Generate SOCB pulse when ETPS.SOCBCNT=1101 (thirteenth event) 1110: Generate SOCB pulse when ETPS.SOCBCNT=1110(fourteenth event) 1111: Generate SOCB pulse when ETPS.SOCBCNT=1111 (fifteenth event)

Bits	Field Name	Type	Reset	Description
15:12	SOCACNT	RO	0x0	PWM ADC start-of-conversion A event (SOCA) counter It indicates how many selected ETSEL.SOCASEL events have occurred. It is automatically cleared when an SOCA pulse is generated. It can be initialized via write. 0000: No events have occurred 0001: 1 event has occurred 0010: 2 events have occurred 0011: 3 events have occurred 0100: 4 events have occurred 0101: 5 events have occurred 0110: 6 events have occurred 0111: 7 events have occurred 1000: 8 events have occurred 1001: 9 events have occurred 1010: 10 events have occurred 1011: 11 events have occurred 1100: 12 events have occurred 1101: 13 events have occurred 1110: 14 events have occurred 1111: 15 events have occurred

Bits	Field Name	Type	Reset	Description
11:8	SOCAPRD	RW	0x0	PWM ADC start-of-conversion A event (SOCA) period select SOCA is generated upon SOCAPRD<=SOCACNT and ETCTL[SOCAEN]=1. No SOCA will be generated if SOCAPRD=0 0000: Disable SOCA event counter. No SOCA pulse will be generated. 0001: Generate SOCA pulse when ETPS.SOCACNT=01 (first event) 0010: Generate SOCA pulse when ETPS.SOCACNT=10 (second event) 0011: Generate SOCA pulse when ETPS.SOCACNT=11 (third event) 0100: Generate SOCA pulse when ETPS.SOCACNT=0100 (fourth event) 0101: Generate SOCA pulse when ETPS.SOCACNT=0101 (fifth event) 0110: Generate SOCA pulse when ETPS.SOCACNT=0110(sixth event) 0111: Generate SOCA pulse when ETPS.SOCACNT=0111 (seventh event) 1000: Generate SOCA pulse when ETPS.SOCACNT=1000 (eighth event) 1001: Generate SOCA pulse when ETPS.SOCACNT=1001 (ninth event) 1010: Generate SOCA pulse when ETPS.SOCACNT=1010 (tenth event) 1011: Generate SOCA pulse when ETPS.SOCACNT=1011 (eleventh event) 1100: Generate SOCA pulse when ETPS.SOCACNT=1100(twelfth event) 1101: Generate SOCA pulse when ETPS.SOCACNT=1101 (thirteenth event) 1110: Generate SOCA pulse when ETPS.SOCACNT=1110(fourteenth event) 1111: Generate SOCA pulse when ETPS.SOCACNT=1111 (fifteenth event)

Bits	Field Name	Type	Reset	Description
7:4	INTCNT	RO	0x0	PWM interrupt counter It indicates how many selected ETSEL.INTSEL events have occurred. It is automatically cleared when an interrupt pulse is generated. It can be initialized via write. 0000: No events have occurred 0001: 1 event has occurred 0010: 2 events have occurred 0011: 3 events have occurred 0100: 4 events have occurred 0101: 5 events have occurred 0110: 6 events have occurred 0111: 7 events have occurred 1000: 8 events have occurred 1001: 9 events have occurred 1010: 10 events have occurred 1011: 11 events have occurred 1100: 12 events have occurred 1101: 13 events have occurred 1110: 14 events have occurred 1111: 15 events have occurred

Bits	Field Name	Type	Reset	Description
3:0	INTPRD	RW	0x0	PWM interrupt period select Interrupt is generated upon INTPRD<=INTCNT and ETCTL[INTEN]=1. No interrupt will be generated if INTPRD=0 0000: Disable the interrupt event counter. No interrupt will be generated and ETFRC.INT is ignored. 0001: Generate an interrupt when ETPS.INTCNT=0001 (first event) 0010: Generate an interrupt when ETPS.INTCNT=0010 (second event) 0011: Generate an interrupt when ETPS.INTCNT=0011 (third event) 0100: Generate an interrupt when ETPS.INTCNT=0100 (fourth event) 0101: Generate an interrupt when ETPS.INTCNT=0101 (fifth event) 0110: Generate an interrupt when ETPS.INTCNT=0110(sixth event) 0111: Generate an interrupt when ETPS.INTCNT=0111 (seventh event) 1000: Generate an interrupt when ETPS.INTCNT=1000 (eighth event) 1001: Generate an interrupt when ETPS.INTCNT=1001 (ninth event) 1010: Generate an interrupt when ETPS.INTCNT=1010 (tenth event) 1011: Generate an interrupt when ETPS.INTCNT=1011 (eleventh event) 1100: Generate an interrupt when ETPS.INTCNT=1100(twelfth event) 1101: Generate an interrupt when ETPS.INTCNT=1101 (thirteenth event) 1110: Generate an interrupt when ETPS.INTCNT=1110(fourteenth event) 1111: Generate an interrupt when ETPS.INTCNT=1111 (fifteenth event)

Table 10-128: Event-Trigger Flag Register (ETFLG) Layout

ETFLG (Event-Trigger Flag Register) Offset: 0xF0 Default: 0x00000000 Access: PWM0 -> ETFLG.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				SOCC	SOCB	SOCA	INT

Table 10-129: Event-Trigger Flag Register (ETFLG) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	SOCC	RO	0x0	Latched PWM ADC start-of-conversion C flag 0: Indicates no event occurred 1: Indicates a SOCC event was generated. The pulse will continue to be generated even if the flag is set.
2	SOCB	RO	0x0	Latched PWM ADC start-of-conversion B flag 0: Indicates no event occurred 1: Indicates a SOCB event was generated. The pulse will continue to be generated even if the flag is set.
1	SOCA	RO	0x0	Latched PWM ADC start-of-conversion A flag 0: Indicates no event occurred 1: Indicates a SOCA event was generated. The pulse will continue to be generated even if the flag is set.
0	INT	RO	0x0	Latched PWM interrupt flag 0: Indicates no event occurred 1: Indicates a PWM interrupt was generated. No further interrupts will be generated until the flag bit is cleared.

Table 10-130: Event-Trigger Clear Register (ETCLR) Layout

ETCLR (Event-Trigger Clear Register) Offset: 0xF4 Default: 0x00000000 Access: PWM0 -> ETCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				SOCC	SOCB	SOCA	INT

Table 10-131: Event-Trigger Clear Register (ETCLR) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	SOCC	W1S	0x0	Latched PWM ADC start-of-conversion C flag clear 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears ETFLG.SOCC
2	SOCB	W1S	0x0	Latched PWM ADC start-of-conversion B flag clear 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears ETFLG.SOCB
1	SOCA	W1S	0x0	Latched PWM ADC start-of-conversion A flag clear 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears ETFLG.SOCA
0	INT	W1S	0x0	Latched PWM interrupt flag clear 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 clears ETFLG.INT

Table 10-132: Event-Trigger Force Register (ETFRC) Layout

ETFRC (Event-Trigger Force Register) Offset: 0xF8 Default: 0x00000000 Access: PWM0 -> ETFRC.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				SOCC	SOCB	SOCA	INT

Table 10-133: Event-Trigger Force Register (ETFRC) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	SOCC	W1S	0x0	Software force a start-of-conversion C pulse This bit is ignored if ETSEL.SOCCEN=0 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 generates an interrupt and sets the ETFLG.SOCC flag
2	SOCB	W1S	0x0	Software force a start-of-conversion B pulse This bit is ignored if ETSEL.SOCBEN=0 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 generates an interrupt and sets the ETFLG.SOCB flag
1	SOCA	W1S	0x0	Software force a start-of-conversion A pulse This bit is ignored if ETSEL.SOCAEN=0 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 generates an interrupt and sets the ETFLG.SOCA flag
0	INT	W1S	0x0	Software force a PWM interrupt This bit is ignored if ETSEL.INTEN=0 or ETPS.INTPRD=00 0: Write a 0 has no effect and always reads back a 0 1: Write a 1 generates an interrupt and sets the ETFLG.INT flag

Table 10-134: PWM Register Write-Allow Key Register (PWMREGKEY) Layout

PWMREGKEY (PWM Register Write-Allow Key Register) Offset: 0xFC Default: 0x1ACCE551 Access: PWM0 -> PWMREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 10-135: PWM Register Write-Allow Key Register (PWMREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected PWM registers

10.11.3 PWMCFG register map

Table 10-136: PWMCFG Module Base Address

Peripheral Module	Base Address
PWMCFG	0x4000 9F00

Table 10-137: PWMCFG Register Map

Register	Offset	Description	Reset Value
TZ0SRCCTL*	0x0	TZ0 Source Control Register	0x0000007F
TZ1SRCCTL*	0x4	TZ1 Source Control Register	0x0000007F
TZ2SRCCTL*	0x8	TZ2 Source Control Register	0x0000007F
TZ3SRCCTL*	0xC	TZ3 Source Control Register	0x0000007F
TZ4SRCCTL*	0x10	TZ4 Source Control Register	0x0000007F
FRCSYNC	0x14	Global PWM Force Synchronization Register	0x00000000
GPIOSYNCCTL*	0x18	GPIO PWMSYNCI Control Register	0x0000007F
GPIOSYNCEN*	0x1C	GPIO Force PWMSYNCI Enable Register	0x00000009
TMR0SYNCEN*	0x20	Timer 0 Force PWMSYNCI Enable Register	0x00000000
TMR1SYNCEN*	0x24	Timer 1 Force PWMSYNCI Enable Register	0x00000000
TMR2SYNCEN*	0x28	Timer 2 Force PWMSYNCI Enable Register	0x00000000
SYNCOCTL*	0x2C	PWMSYNCO Control Register	0x00000007
SOCAOCTL*	0x30	PWMSOCA Output Control Register	0x00000007
SOCBOCTL*	0x34	PWMSOCB Output Control Register	0x00000007
SOCCOCTL*	0x38	PWMSOCC Output Control Register	0x00000007
PWMCFGREGKEY	0x3C	PWMCFG Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the PWMCFGKEY=0x1ACCE551.

10.11.4 PWMCFG registers

Table 10-138: TZ0 Source Control Register (TZ0SRCCTL) Layout

TZ0SRCCTL (TZ0 Source Control Register) Offset: 0x0 Default: 0x0000007F Access: PWMCFG -> TZ0SRCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 10-139: TZ0 Source Control Register (TZ0SRCCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	TZ0 polarity 0: Active when low 1: Active when high
5:0	IOSEL	RW	0x3F	GPIO number for TZ0 event

Table 10-140: TZ1 Source Control Register (TZ1SRCCTL) Layout

TZ1SRCCTL (TZ1 Source Control Register) Offset: 0x4 Default: 0x0000007F Access: PWMCFG -> TZ1SRCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 10-141: TZ1 Source Control Register (TZ1SRCCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	TZ1 polarity 0: Active when low 1: Active when high
5:0	IOSEL	RW	0x3F	GPIO number for TZ1 event

Table 10-142: TZ2 Source Control Register (TZ2SRCCTL) Layout

TZ2SRCCTL (TZ2 Source Control Register) Offset: 0x8 Default: 0x0000007F Access: PWMCFG -> TZ2SRCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 10-143: TZ2 Source Control Register (TZ2SRCCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	TZ2 polarity 0: Active when low 1: Active when high
5:0	IOSEL	RW	0x3F	GPIO number for TZ2 event

Table 10-144: TZ3 Source Control Register (TZ3SRCCTL) Layout

TZ3SRCCTL (TZ3 Source Control Register) Offset: 0xC Default: 0x0000007F Access: PWMCFG -> TZ3SRCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 10-145: TZ3 Source Control Register (TZ3SRCCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	TZ3 polarity 0: Active when low 1: Active when high
5:0	IOSEL	RW	0x3F	GPIO number for TZ3 event

Table 10-146: TZ4 Source Control Register (TZ4SRCCTL) Layout

TZ4SRCCTL (TZ4 Source Control Register) Offset: 0x10 Default: 0x0000007F Access: PWMCFG -> TZ4SRCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 10-147: TZ4 Source Control Register (TZ4SRCCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	TZ4 polarity 0: Active when low 1: Active when high
5:0	IOSEL	RW	0x3F	GPIO number for TZ4 event

Table 10-148: Global PWM Force Synchronization Register (FRCSYNC) Layout

FRCSYNC (Global PWM Force Synchronization Register) Offset: 0x14 Default: 0x00000000 Access: PWMCFG -> FRCSYNC.all							
31	30	29	28	27	26	25	24
PWMCLK	RESERVED_30_6						
23	22	21	20	19	18	17	16
RESERVED_30_6							
15	14	13	12	11	10	9	8
RESERVED_30_6							
7	6	5	4	3	2	1	0
RESERVED_30_6		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	PWM1SYNC	PWM0SYNC

Table 10-149: Global PWM Force Synchronization Register (FRCSYNC) Description

Bits	Field Name	Type	Reset	Description
31	PWMCLK	W1S	0x0	Software forced synchronization on clock for all PWMs 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 will force a SYNC on all PWM clocks This bit is self-cleared
30:6	RESERVED_30_6	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
5	PWM5SYNC	W1S	0x0	Software forced synchronization on PWM5 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 will force a SYNC event on PWM5. This bit is self-cleared
4	PWM4SYNC	W1S	0x0	Software forced synchronization on PWM4 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 will force a SYNC event on PWM4. This bit is self-cleared
3	PWM3SYNC	W1S	0x0	Software forced synchronization on PWM3 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 will force a SYNC event on PWM3. This bit is self-cleared
2	PWM2SYNC	W1S	0x0	Software forced synchronization on PWM2 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 will force a SYNC event on PWM2. This bit is self-cleared
1	PWM1SYNC	W1S	0x0	Software forced synchronization on PWM1 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 will force a SYNC event on PWM1. This bit is self-cleared
0	PWM0SYNC	W1S	0x0	Software forced synchronization on PWM0 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 will force a SYNC event on PWM0. This bit is self-cleared

Table 10-150: GPIO PWMSYNCI Control Register (GPIOSYNCCTL) Layout

GPIOSYNCCTL (GPIO PWMSYNCI Control Register) Offset: 0x18 Default: 0x0000007F Access: PWMCFG -> GPIOSYNCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 10-151: GPIO PWMSYNCI Control Register (GPIOSYNCCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	PWMSYNCI input polarity 0: Active when low 1: Active when high
5:0	IOSEL	RW	0x3F	GPIO number for PWMSYNCI input

Table 10-152: GPIO Force PWMSYNCI Enable Register (GPIOSYNCEN) Layout

GPIOSYNCEN (GPIO Force PWMSYNCI Enable Register) Offset: 0x1C Default: 0x00000009 Access: PWMCFG -> GPIOSYNCEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	PWM1SYNC	PWM0SYNC

Table 10-153: GPIO Force PWMSYNCI Enable Register (GPIOSYNCEN) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	PWM5SYNC	RW	0x0	Enable PWM5 synchronization by GPIO 0: Disable 1: Enable
4	PWM4SYNC	RW	0x0	Enable PWM4 synchronization by GPIO 0: Disable 1: Enable
3	PWM3SYNC	RW	0x1	Enable PWM3 synchronization by GPIO 0: Disable 1: Enable
2	PWM2SYNC	RW	0x0	Enable PWM2 synchronization by GPIO 0: Disable 1: Enable
1	PWM1SYNC	RW	0x0	Enable PWM1 synchronization by GPIO 0: Disable 1: Enable
0	PWM0SYNC	RW	0x1	Enable PWM0 synchronization by GPIO 0: Disable 1: Enable

Table 10-154: Timer 0 Force PWMSYNCI Enable Register (TMR0SYNCEN) Layout

TMR0SYNCEN (Timer 0 Force PWMSYNCI Enable Register) Offset: 0x20 Default: 0x00000000 Access: PWMCFG -> TMR0SYNCEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	PWM1SYNC	PWM0SYNC

Table 10-155: Timer 0 Force PWMSYNCI Enable Register (TMR0SYNCEN) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	PWM5SYNC	RW	0x0	Enable PWM5 synchronization by Timer 0 0: Disable 1: Enable
4	PWM4SYNC	RW	0x0	Enable PWM4 synchronization by Timer 0 0: Disable 1: Enable
3	PWM3SYNC	RW	0x0	Enable PWM3 synchronization by Timer 0 0: Disable 1: Enable
2	PWM2SYNC	RW	0x0	Enable PWM2 synchronization by Timer 0 0: Disable 1: Enable
1	PWM1SYNC	RW	0x0	Enable PWM1 synchronization by Timer 0 0: Disable 1: Enable
0	PWM0SYNC	RW	0x0	Enable PWM0 synchronization by Timer 0 0: Disable 1: Enable

Table 10-156: Timer 1 Force PWMSYNCI Enable Register (TMR1SYNCEN) Layout

TMR1SYNCEN (Timer 1 Force PWMSYNCI Enable Register) Offset: 0x24 Default: 0x00000000 Access: PWMCFG -> TMR1SYNCEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	PWM1SYNC	PWM0SYNC

Table 10-157: Timer 1 Force PWMSYNCI Enable Register (TMR1SYNCEN) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	PWM5SYNC	RW	0x0	Enable PWM5 synchronization by Timer 1 0: Disable 1: Enable
4	PWM4SYNC	RW	0x0	Enable PWM4 synchronization by Timer 1 0: Disable 1: Enable
3	PWM3SYNC	RW	0x0	Enable PWM3 synchronization by Timer 1 0: Disable 1: Enable
2	PWM2SYNC	RW	0x0	Enable PWM2 synchronization by Timer 1 0: Disable 1: Enable
1	PWM1SYNC	RW	0x0	Enable PWM1 synchronization by Timer 1 0: Disable 1: Enable
0	PWM0SYNC	RW	0x0	Enable PWM0 synchronization by Timer 1 0: Disable 1: Enable

Table 10-158: Timer 2 Force PWMSYNCI Enable Register (TMR2SYNCEN) Layout

TMR2SYNCEN (Timer 2 Force PWMSYNCI Enable Register) Offset: 0x28 Default: 0x00000000 Access: PWMCFG -> TMR2SYNCEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		PWM5SYNC	PWM4SYNC	PWM3SYNC	PWM2SYNC	PWM1SYNC	PWM0SYNC

Table 10-159: Timer 2 Force PWMSYNCI Enable Register (TMR2SYNCEN) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	PWM5SYNC	RW	0x0	Enable PWM5 synchronization by Timer 2 0: Disable 1: Enable
4	PWM4SYNC	RW	0x0	Enable PWM4 synchronization by Timer 2 0: Disable 1: Enable
3	PWM3SYNC	RW	0x0	Enable PWM3 synchronization by Timer 2 0: Disable 1: Enable
2	PWM2SYNC	RW	0x0	Enable PWM2 synchronization by Timer 2 0: Disable 1: Enable
1	PWM1SYNC	RW	0x0	Enable PWM1 synchronization by Timer 2 0: Disable 1: Enable
0	PWM0SYNC	RW	0x0	Enable PWM0 synchronization by Timer 2 0: Disable 1: Enable

Table 10-160: PWMSYNCO Control Register (SYNCOCTL) Layout

SYNCOCTL (PWMSYNCO Control Register) Offset: 0x2C Default: 0x00000007 Access: PWMCFG -> SYNCOCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		SRCSEL			POL	DURATION	

Table 10-161: PWMSYNCO Control Register (SYNCOCTL) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5:3	SRCSEL	RW	0x0	PWMSYNCO source select 000: Source select from PWM0 001: Source select from PWM1 010: Source select from PWM2 011: Source select from PWM3 100: Source select from PWM4 101: Source select from PWM5 110: 111:
2	POL	RW	0x1	Polarity of PWMSYNCO to the pin 0: Active when low 1: Active when high
1:0	DURATION	RW	0x3	Pulse duration of PWMSYNCO to the pin 00: Duration of PWMSYNCO pulse to the pin is 4 PWM clocks 01: Duration of PWMSYNCO pulse to the pin is 8 PWM clocks 10: Duration of PWMSYNCO pulse to the pin is 16 PWM clocks 11: Duration of PWMSYNCO pulse to the pin is 32 PWM clocks

Table 10-162: PWMSOCA Output Control Register (SOCAOCTL) Layout

SOCAOCTL (PWMSOCA Output Control Register) Offset: 0x30 Default: 0x00000007 Access: PWMCFG -> SOCAOCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							SOCA5EN
7	6	5	4	3	2	1	0
SOCA4EN	SOCA3EN	SOCA2EN	SOCA1EN	SOCA0EN	POL	DURATION	

Table 10-163: PWMSOCA Output Control Register (SOCAOCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	SOCA5EN	RW	0x0	Enable PWM5SOCA output to the pin 0: Disable 1: Enable
7	SOCA4EN	RW	0x0	Enable PWM4SOCA output to the pin 0: Disable 1: Enable
6	SOCA3EN	RW	0x0	Enable PWM3SOCA output to the pin 0: Disable 1: Enable
5	SOCA2EN	RW	0x0	Enable PWM2SOCA output to the pin 0: Disable 1: Enable
4	SOCA1EN	RW	0x0	Enable PWM1SOCA output to the pin 0: Disable 1: Enable
3	SOCA0EN	RW	0x0	Enable PWM0SOCA output to the pin 0: Disable 1: Enable
2	POL	RW	0x1	Polarity of PWMSOCA output to the pin 0: Active when low 1: Active when high

Bits	Field Name	Type	Reset	Description
1:0	DURATION	RW	0x3	Pulse duration of PWMSOCA output to the pin 00: Duration of PWMSOCA pulse to the pin is 4 PWM clocks 01: Duration of PWMSOCA pulse to the pin is 8 PWM clocks 10: Duration of PWMSOCA pulse to the pin is 16 PWM clocks 11: Duration of PWMSOCA pulse to the pin is 32 PWM clocks

Table 10-164: PWMSOCB Output Control Register (SOCBOCTL) Layout

SOCBOCTL (PWMSOCB Output Control Register) Offset: 0x34 Default: 0x00000007
Access: PWMCFG -> SOCBOCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							SOCB5EN
7	6	5	4	3	2	1	0
SOCB4EN	SOCB3EN	SOCB2EN	SOCB1EN	SOCB0EN	POL	DURATION	

Table 10-165: PWMSOCB Output Control Register (SOCBOCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	SOCB5EN	RW	0x0	Enable PWM5SOCB output to the pin 0: Disable 1: Enable
7	SOCB4EN	RW	0x0	Enable PWM4SOCB output to the pin 0: Disable 1: Enable
6	SOCB3EN	RW	0x0	Enable PWM3SOCB output to the pin 0: Disable 1: Enable
5	SOCB2EN	RW	0x0	Enable PWM2SOCB output to the pin 0: Disable 1: Enable
4	SOCB1EN	RW	0x0	Enable PWM1SOCB output to the pin 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
3	SOCB0EN	RW	0x0	Enable PWM0SOCB output to the pin 0: Disable 1: Enable
2	POL	RW	0x1	Polarity of PWMSOCB output to the pin 0: Active when low 1: Active when high
1:0	DURATION	RW	0x3	Pulse duration of PWMSOCB output to the pin 00: Duration of PWMSOCB pulse to the pin is 4 PWM clocks 01: Duration of PWMSOCB pulse to the pin is 8 PWM clocks 10: Duration of PWMSOCB pulse to the pin is 16 PWM clocks 11: Duration of PWMSOCB pulse to the pin is 32 PWM clocks

Table 10-166: PWMSOCC Output Control Register (SOCCOCTL) Layout

SOCCOCTL (PWMSOCC Output Control Register) Offset: 0x38 Default: 0x00000007 Access: PWMCFG -> SOCCOCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							SOCC5EN
7	6	5	4	3	2	1	0
SOCC4EN	SOCC3EN	SOCC2EN	SOCC1EN	SOCC0EN	POL	DURATION	

Table 10-167: PWMSOCC Output Control Register (SOCCOCTL) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	SOCC5EN	RW	0x0	Enable PWM5SOCC output to the pin 0: Disable 1: Enable
7	SOCC4EN	RW	0x0	Enable PWM4SOCC output to the pin 0: Disable 1: Enable
6	SOCC3EN	RW	0x0	Enable PWM3SOCC output to the pin 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
5	SOCC2EN	RW	0x0	Enable PWM2SOCC output to the pin 0: Disable 1: Enable
4	SOCC1EN	RW	0x0	Enable PWM1SOCC output to the pin 0: Disable 1: Enable
3	SOCC0EN	RW	0x0	Enable PWM0SOCC output to the pin 0: Disable 1: Enable
2	POL	RW	0x1	Polarity of PWMSOCC output to the pin 0: Active when low 1: Active when high
1:0	DURATION	RW	0x3	Pulse duration of PWMSOCC output to the pin 00: Duration of PWMSOCC pulse to the pin is 4 PWM clocks 01: Duration of PWMSOCC pulse to the pin is 8 PWM clocks 10: Duration of PWMSOCC pulse to the pin is 16 PWM clocks 11: Duration of PWMSOCC pulse to the pin is 32 PWM clocks

Table 10-168: PWMCFG Register Write-Allow Key Register (PWMCFGREGKEY) Layout

PWMCFGREGKEY (PWMCFG Register Write-Allow Key Register) Offset: 0x3C Default: 0x1ACCE551 Access: PWMCFG -> PWMCFGREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 10-169: PWMCFG Register Write-Allow Key Register (PWMCFGREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected PWMCFG registers

11 ECAP

11.1 Introduction

The enhanced capture (ECAP) module is used in situations where accurate timing of external events is important. This chapter describes the module and how to use it.

Uses for ECAP include:

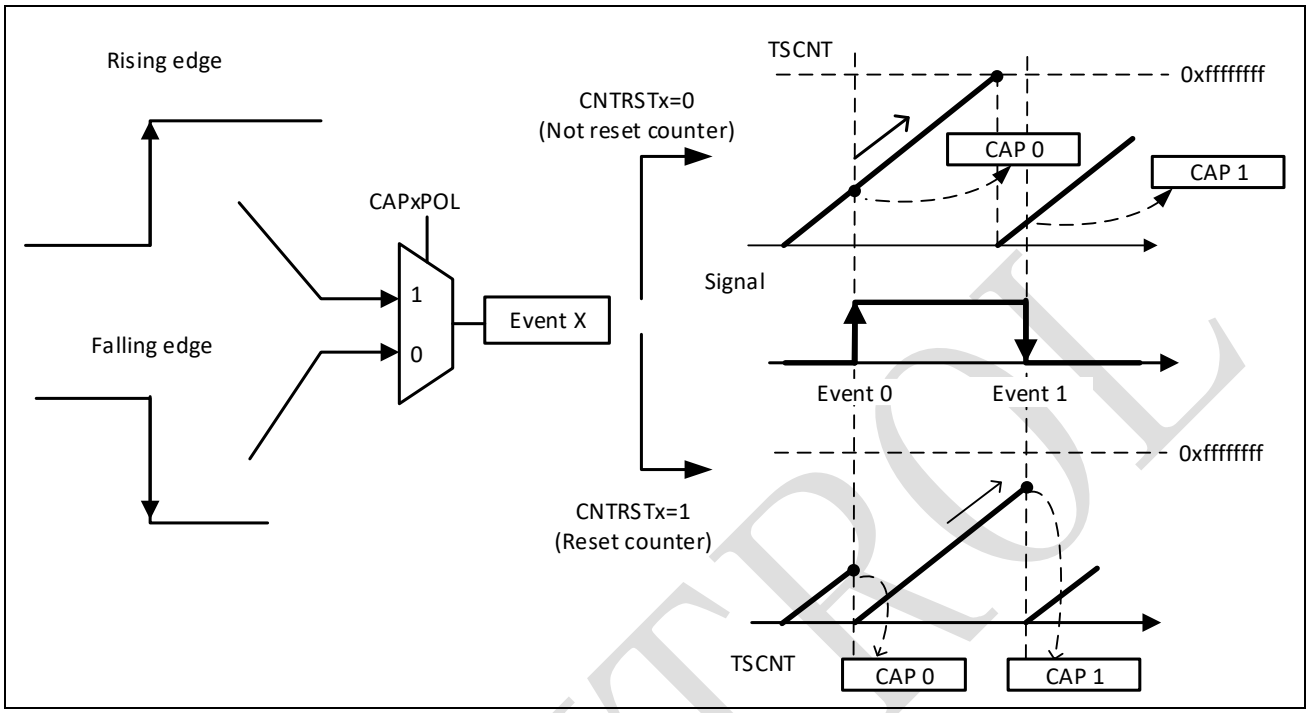
- Speed measurements of rotating machinery (e.g., e-bike motor with Hall sensors)
- Measure period and duty cycle of pulse signals
- Decoding information derived from duty cycle encoded systems

The ECAP module described in this guide includes the following features:

- 32-bit time base with 5 ns time resolution (max 200 MHz system clock)
- 4-event time-stamp registers (each 32 bits)
- Edge polarity selection for up to four sequenced time-stamp capture events
- Interrupt on either of the four events
- Single shot capture of up to four event time-stamps
- Continuous mode capture of time-stamps in a four-deep circular buffer
- Absolute time-stamp capture
- Difference (Delta) mode time-stamp capture
- All above resources dedicated to a single input pin
- The ECAP module can be configured as a PWM output if not used in capture mode

11.2 Description

Figure 11-1: Basic operation of enhanced capture



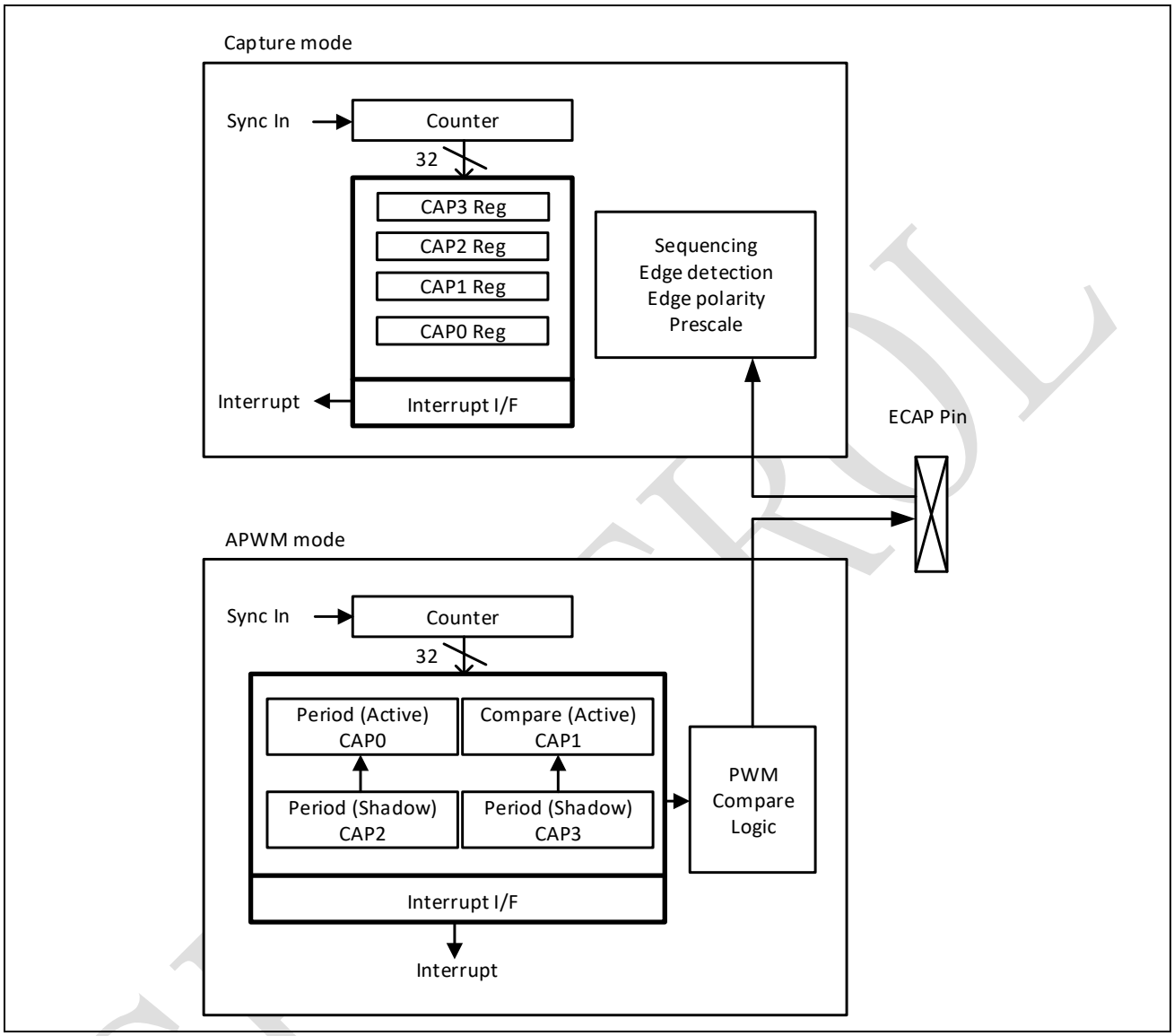
The philosophy of enhanced capture (ECAP) module is regarded “rising” and “falling” edge of signal captured as abstract symbols–called events, which can be named as four number sequence with time. In other words, Event0 to Event3 symbolize consequence in time, and each of them can be programmed as rising or falling edge. This is set by CAPCTL.CAPxPOL field.

The second is to set counter behavior after event happened. This behavior can clear ECAP counter or do nothing about it by setting CAPCTL.CNTRSTx.

With above two basic configuration of events and counter, users can make complicated capture function such as period and duty calculation with other registers illustrated later. Figure 11-1 shows the basic two operation of ECAP.

11.3 Capture and APWM operating mode

Figure 11-2: Capture and APWM modes of operation



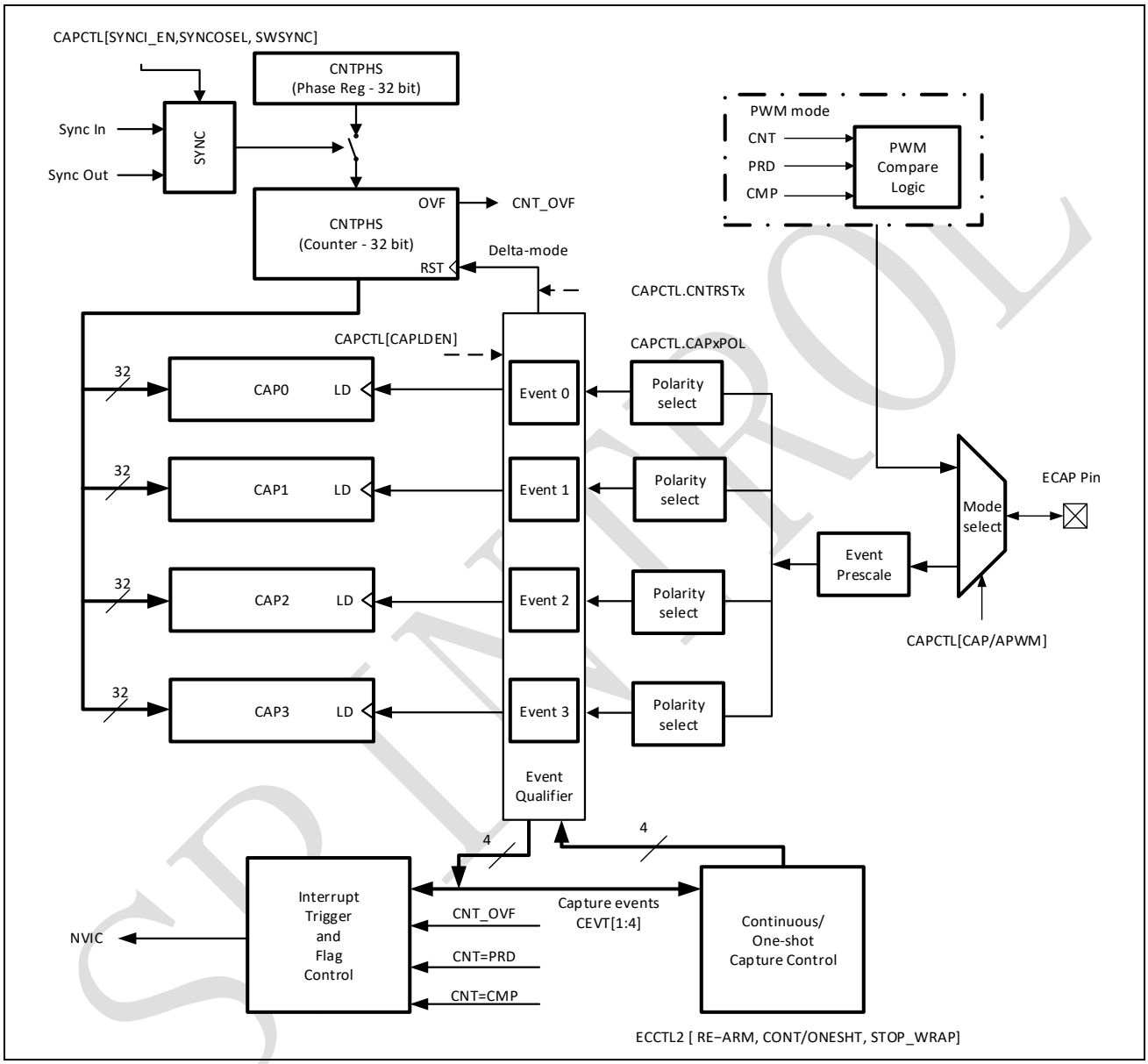
User can take ECAP module resources to implement a single-channel PWM generator (with 32 bit capabilities) when it is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms.

The CAP0 and CAP1 registers become the active period and compare registers, respectively, while CAP2 and CAP3 registers become the period and capture shadow registers, respectively. Figure 11-2 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.

11.4 Capture mode description

Figure 11-3 shows the various components that implement the capture function. The ECAP Pin is sourced from GPIO pin according to CAPSRCCTL register.

Figure 11-3: Capture function diagram



11.4.1 Event prescaler

An input capture signal (continuous pulse series) can be prescaled by N = CAPCTL.EVTDIV+1. This is useful when very high frequency signals are used as inputs. Figure 11-4 shows a functional diagram and Figure 11-5 shows the operation of the prescale function.

Figure 11-4: Event prescale control

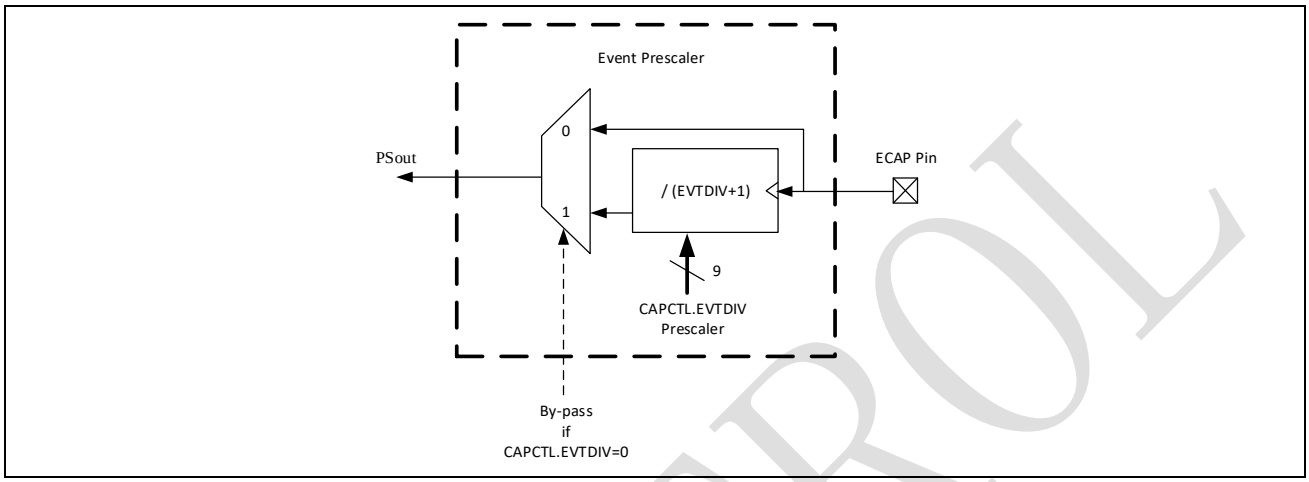
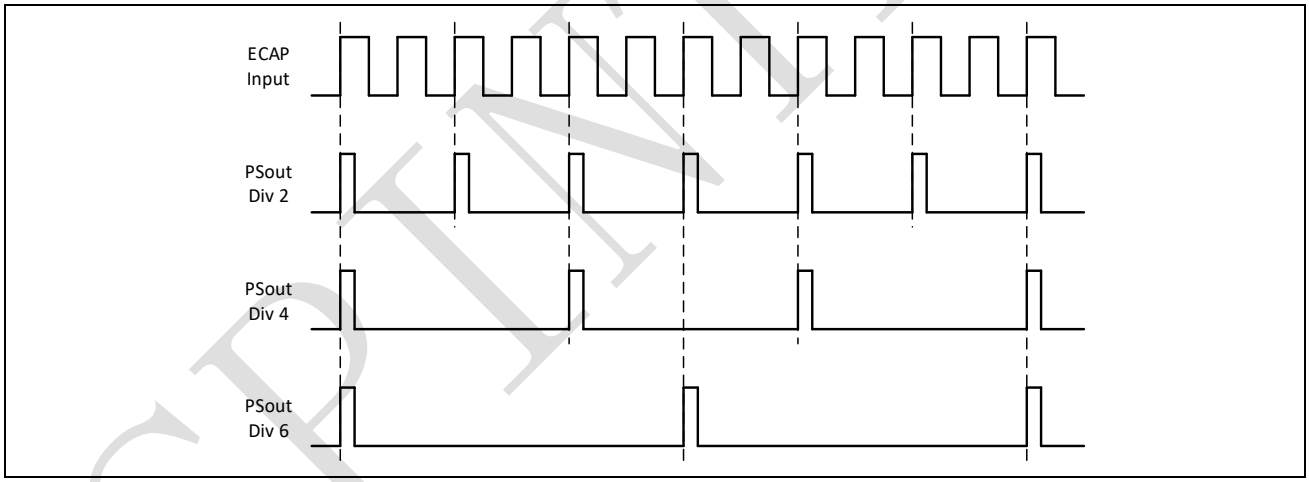


Figure 11-5: Prescale function waveforms



Note: The ECAP event prescaler is implemented with existing clock divider IP, so there is extra four event cycles (regarded as clock cycles to the divider) required to initialize the divider after power-up. For one shot mode, in which the user cares about exactly each incoming event, the first 4 events may not work correctly. We add software work-around to purposely generate 4 dummy events via GPIO toggling to initialize the event prescaler in ECAP module demo code.

11.4.2 Edge polarity select and qualifier

Four independent edge polarity (rising edge/falling edge) selection MUXes are used, one for each capture event.

- Each edge (up to 4) is event qualified by the Modulo-4 sequencer.
- The edge event is gated to its respective CAPx register by the Mod4 counter. The CAPx register is loaded on the falling edge.

11.4.3 Continuous/one-shot control

The Mod4 (2 bit) counter is incremented via edge qualified events (CEVT0-CEVT3). The function of this counter index which event in sequence occurs now. It will cooperate with CAPCTL.STOPWARP to control the length of such capture event sequence.

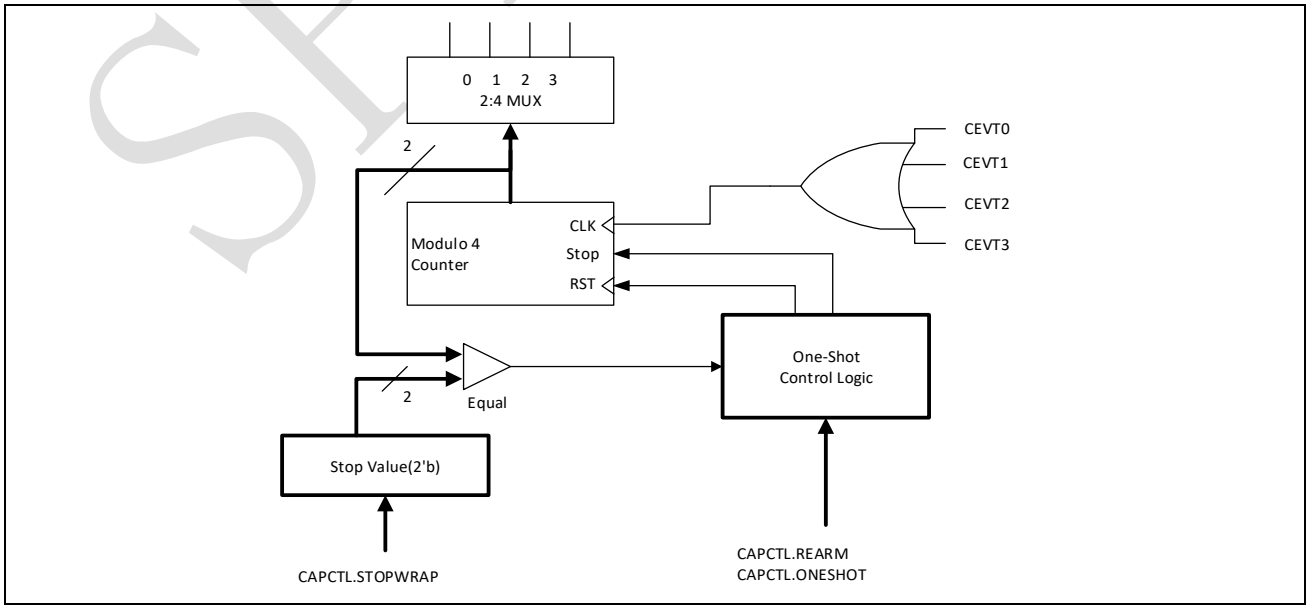
- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.
- A 2-bit stop register (CAPCTL.STOPWARP) is used to compare the Mod4 counter output, and when equal stops the Mod4 counter and inhibits further loads of the CAP0-CAP3 registers. This occurs during one-shot operation.

The continuous/one-shot block controls the start/stop and reset (zero) functions of the Mod4 counter via a mono-shot type of action that can be triggered by the stop-value comparator and re-armed via software control.

Once armed, the ECAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of CAP0-3 registers (i.e., time-stamps).

Re-arming prepares the ECAP module for another capture sequence. Also re-arming clears (to zero) theMod4 counter and permits loading of CAP0-3 registers again, providing the CAPLDEN bit is set. In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to CAP0-3 in a circular buffer sequence.

Figure 11-6: Details of the continuous/one-shot block

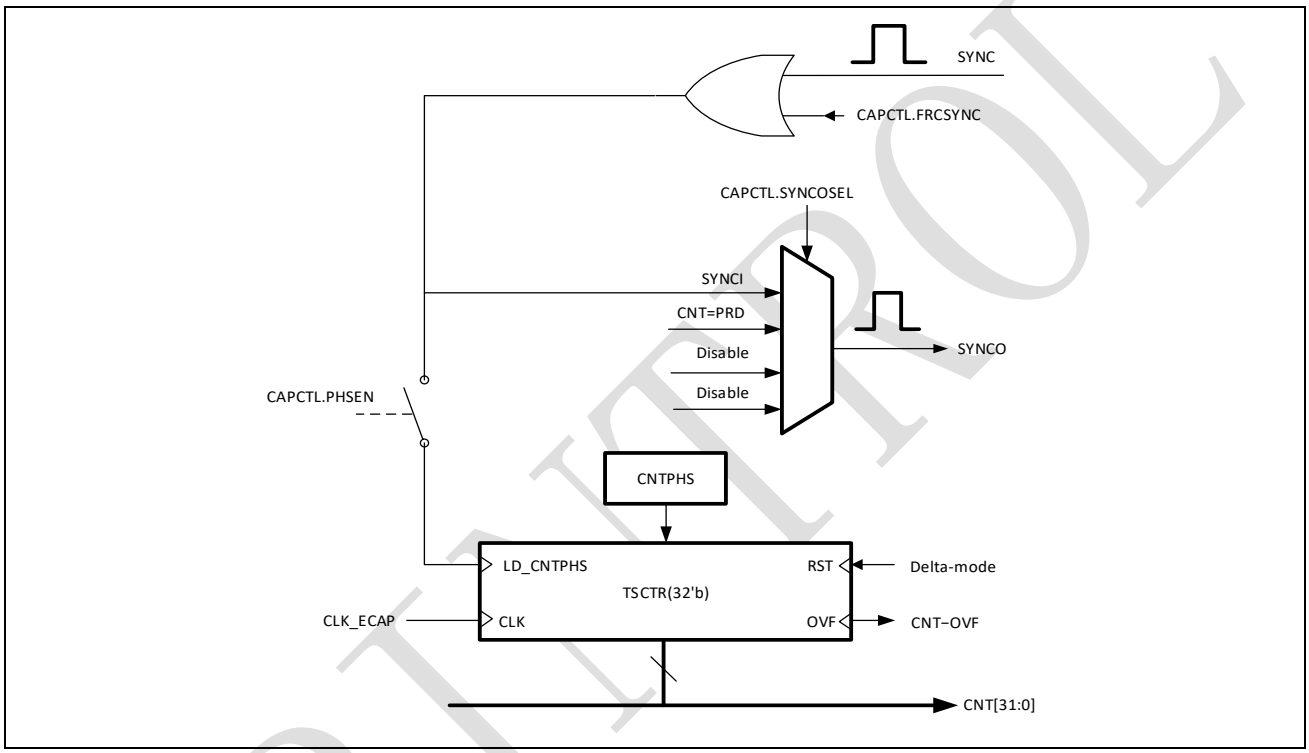


11.4.4 32-bit counter and phase control

This counter provides the time-base for event captures, and is clocked via the CLK_ECAP. A phase register is provided to achieve synchronization with other counters, via a hardware and software forced sync. This is useful in APWM mode when a phase offset between modules is needed.

On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then it is reset to 0 by any of the LD0-LD3 signals.

Figure 11-7: Details of the counter and synchronization block



11.4.5 CAP0-CAP3 registers

These 32-bit registers are fed by the 32-bit counter timer bus TSCNT and are loaded (i.e., capture a time-stamp) when their respective LD inputs are strobed. Loading of the capture registers can be inhibited via control bit CAPLDEN. During one-shot operation, this bit is cleared (loading is inhibited) automatically when a stop condition occurs, i.e. Stop Value = Mod4.

CAP0 and CAP1 registers become the active period and compare registers, respectively, in APWM mode.CAP2 and CAP3 registers become the respective shadow registers (APRD and ACMP) for CAP0 and CAP1 during APWM operation.

11.4.6 Interrupt control

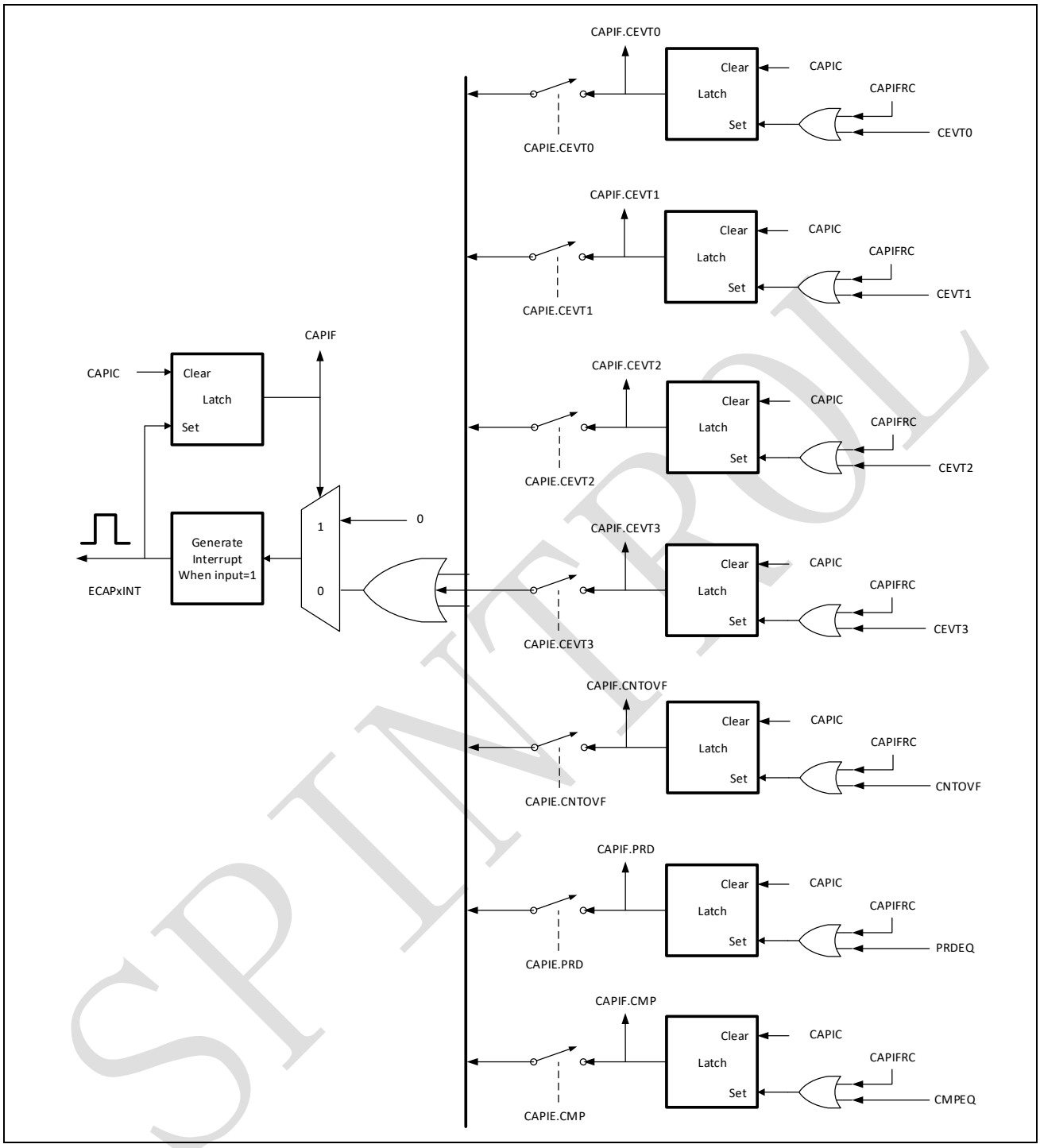
An Interrupt can be generated on capture events (CEVT0-CEVT3, CNTOVF) or APWM events (CNT = PRD, CNT = CMP).A counter overflow event (0xFFFFFFFF->0x00000000) is also provided as an interrupt source (CNTOVF).The capture events are edge and sequencer qualified (i.e., ordered in time) by the polarity select andMod4 gating, respectively.

One of these events can be selected as the interrupt source (from the ECAP module) going to the PIE. Seven interrupt events (CEVT0, CEVT1, CEVT2, CEVT3, CNTOVF, CNT=PRD, CNT=CMP) can be generated. The interrupt enable register (CAPIE) is used to enable/disable individual interrupt event sources.

The interrupt flag register (CAPIF) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the NVIC only if any of the interrupt events happens and the corresponding flag bit as well as the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event via the interrupt clear register (CAPIC) before any other interrupt pulses are generated. Keeping the interrupt event flag bit as 1 prevents interrupt generation by future occurrence of the same event, while keeping the INT flag as 1 prevents interrupt generation by all future events. You can force an interrupt event via the interrupt force register (CAPIFRC).This is useful for test purposes.

Note: The CEVT0, CEVT1, CEVT2, CEVT3 flags are only active in capture mode (CAPCTL.CAPAPWM = 0). The CNT=PRD, CNT=CMP flags are only valid in APWM mode (CAPCTL.CAPAPWM = 1). CNTOVF flag is valid in both modes.

Figure 11-8: Interrupts in ECAP module



11.4.7 Shadow load and lockout control

In capture mode, this logic inhibits (locks out) any shadow loading of CAP0 or CAP1 from APRD and ACMP registers, respectively.

In APWM mode, shadow loading is active and two choices are permitted:

- Immediate – APRD or ACMP are transferred to CAP0 or CAP1 immediately upon writing a new value.
- On period equal, i.e., CNT[31:0] = PRD[31:0]

11.5 APWM mode operation

Main operating highlights of the APWM section:

- The time-stamp counter bus is made available for comparison via 2 digital (32-bit) comparators.
- When CAP0/1 registers are not used in capture mode, their contents can be used as Period and Compare values in APWM mode.
- Double buffering is achieved via shadow registers APRD and ACMP (CAP2/3). The shadow register contents are transferred over to CAP0/1 registers either immediately upon a write, or on a CNT = PRD trigger.
- In APWM mode, writing to CAP0/CAP1 active registers will also write the same value to the corresponding shadow registers CAP2/CAP3. This emulates immediate mode. Writing to the shadow registers CAP2/CAP3 will invoke the shadow mode.
- During initialization, you must write to the active registers for both period and compare. This automatically copies the initial values into the shadow values. For subsequent compare updates, i.e., during run-time, you only need to use the shadow registers.

Figure 11-9: Active and shadow register when APWM mode

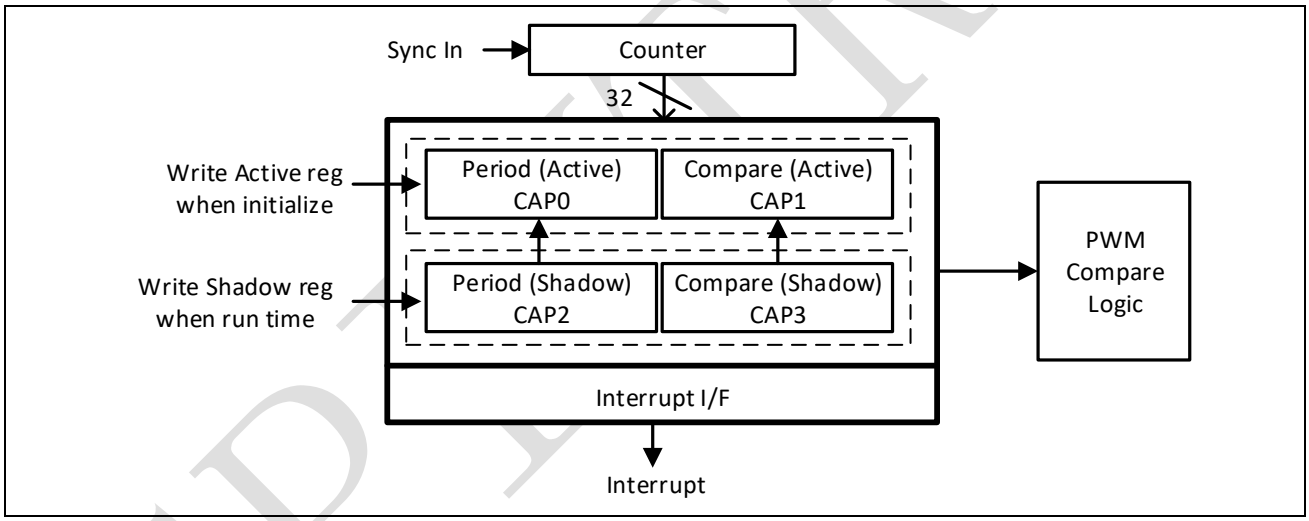
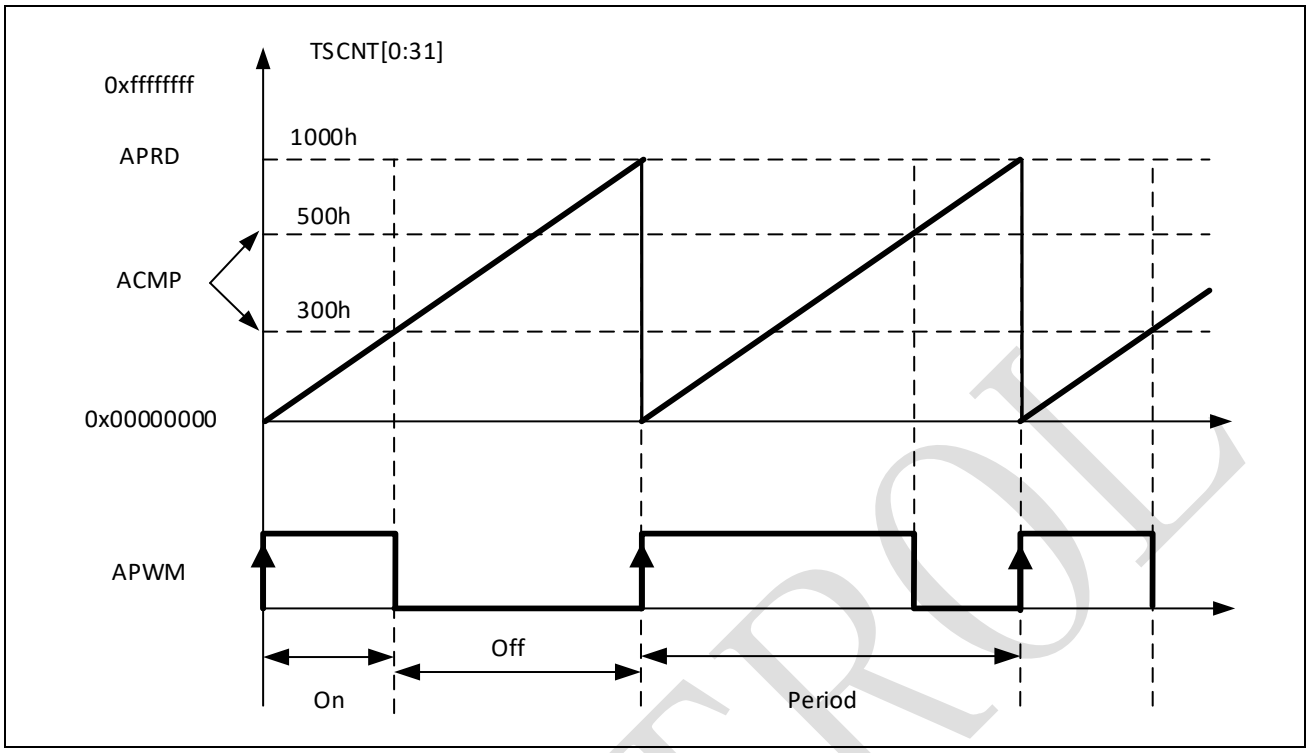


Figure 11-10: PWM waveform details Of APWM mode operation (active High)



11.6 Application of the ECAP module

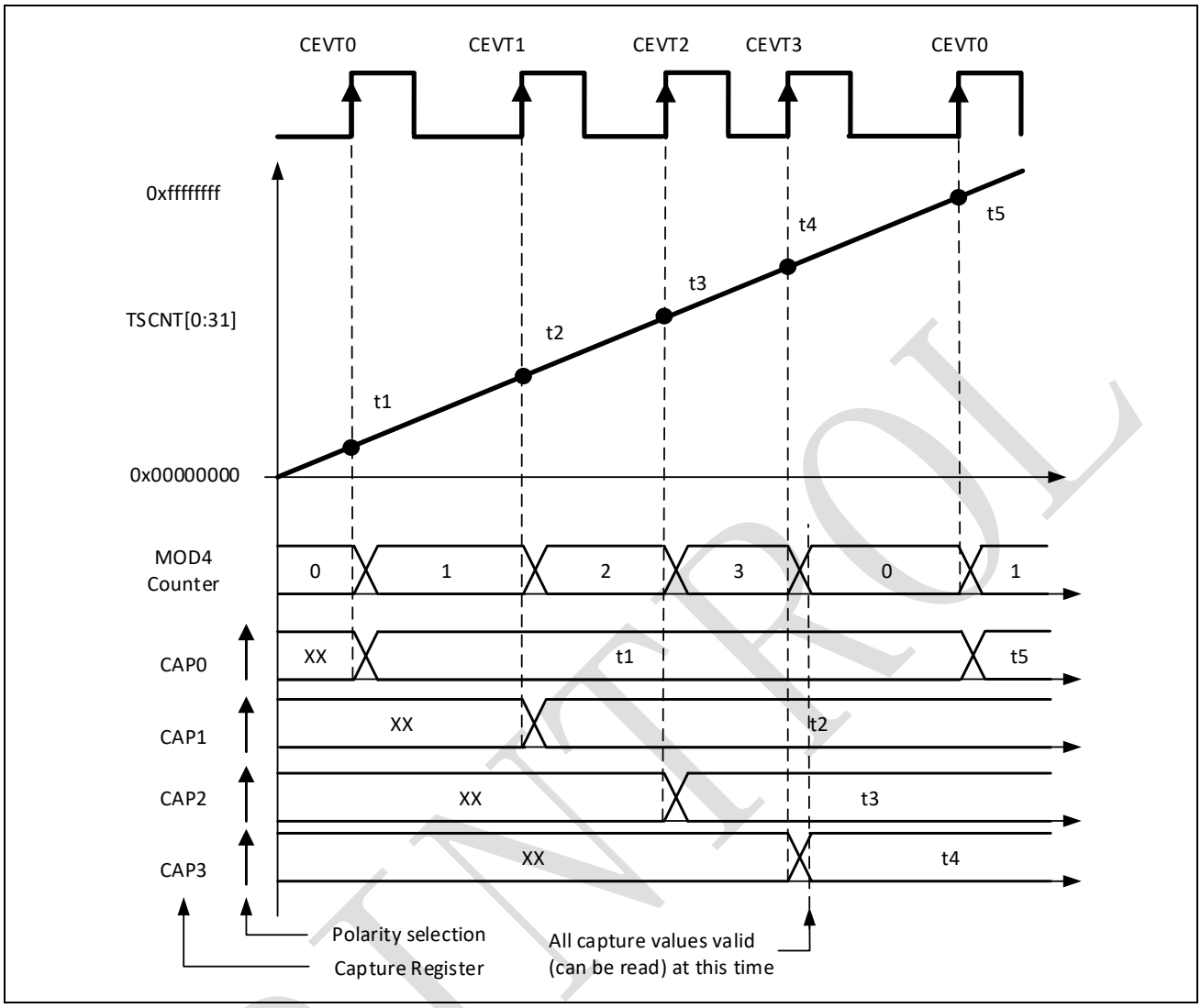
The following sections will provide Applications examples and code snippets to show how to configure and operate the ECAP module. For clarity and ease of use, the examples use the SPD1148 ECAP "C" header files.

Example 1 – Absolute time-stamp operation rising edge trigger

Figure 11-11 shows an example of continuous capture operation (Mod4 counter wraps around). In this figure, TSCNT counts-up without resetting and capture events are qualified on the rising edge only, this gives period (and frequency) information.

On an event, the TSCNT contents (i.e., time-stamp) is first captured, then Mod4 counter is incremented to the next state. When the TSCNT reaches 0xFFFFFFFF (i.e. maximum value), it wraps around to 0x00000000 (not shown in Figure 11-11), if this occurs, the CNTOVF (counter overflow) flag is set, and an interrupt (if enabled) occurs. Captured Time-stamps are valid at the point indicated by the diagram, i.e. after the 4th event, hence event CEVT3 can conveniently be used to trigger an interrupt and the CPU can read data from the CAPx registers.

Figure 11-11: Capture sequence for absolute time-stamp and rising edge detect



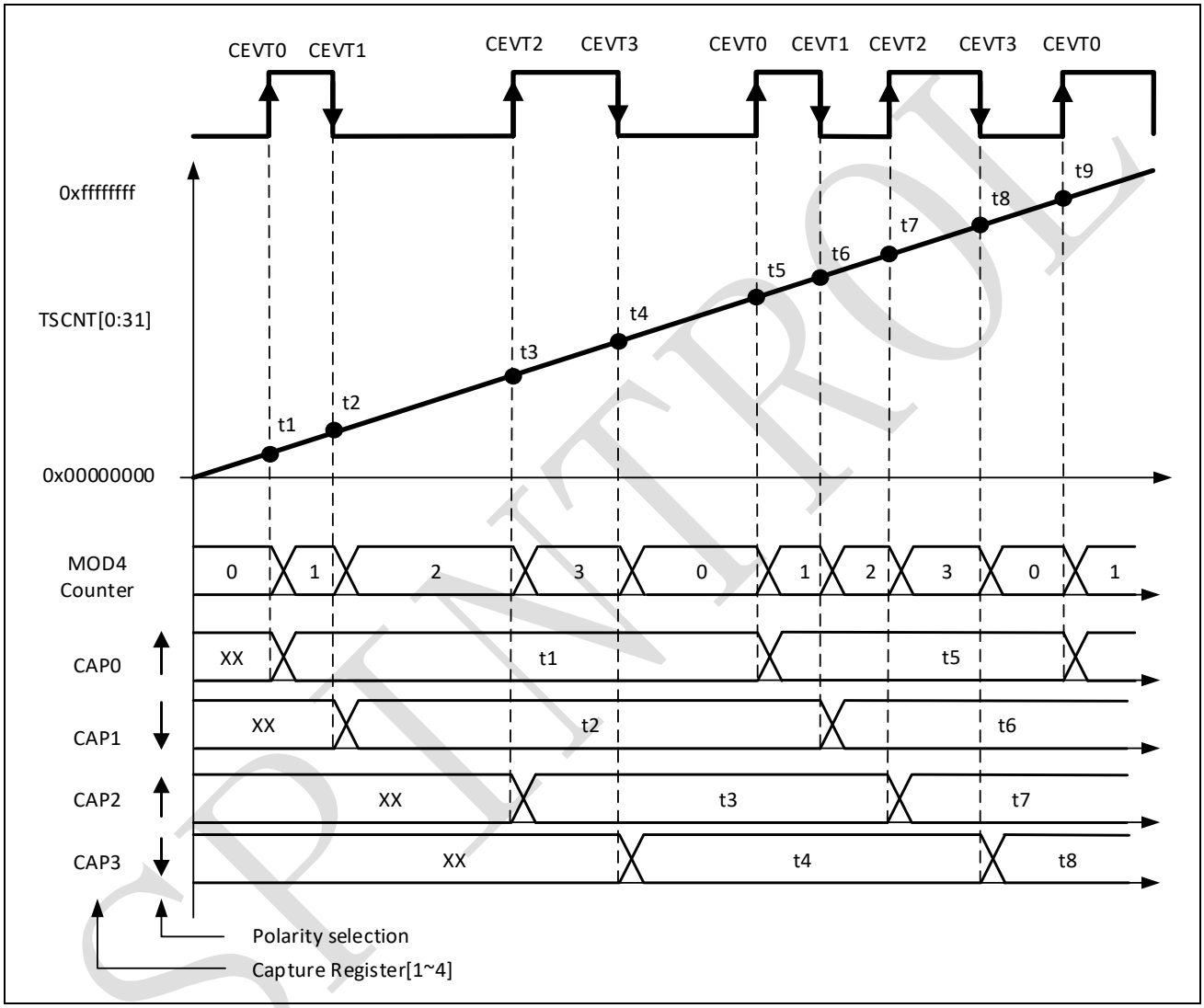
Example Code

```
// Code snippet for CAP mode Absolute Time, Rising edge trigger
// Initialization Time
//======================
// Spintrol Co., Ltd.. All rights reserved.

ECAP->CAPCTL.bit.CAP0POL = EC_RISING;
ECAP->CAPCTL.bit.CAP1POL = EC_RISING;
ECAP->CAPCTL.bit.CAP2POL = EC_RISING;
ECAP->CAPCTL.bit.CAP3POL = EC_RISING;
ECAP->CAPCTL.bit.CNTRST0 = EC_ABS_MODE;
ECAP->CAPCTL.bit.CNTRST1 = EC_ABS_MODE;
ECAP->CAPCTL.bit.CNTRST2 = EC_ABS_MODE;
ECAP->CAPCTL.bit.CNTRST3 = EC_ABS_MODE;
ECAP->CAPCTL.bit.CAPLDEN = EC_ENABLE;
ECAP->CAPCTL.bit.EVTDIV = EC_DIV1;
ECAP->CAPCTL.bit.CAPAPWM = EC_CAP_MODE;
ECAP->CAPCTL.bit.MODE = EC_CONTINUOUS;
ECAP->CAPCTL.bit.SYNCOSEL = EC_SYNCO_DIS;
ECAP->CAPCTL.bit.SYNCIEN = EC_DISABLE;
ECAP->CAPCTL.bit.TSCNTRUN = EC_RUN; // Allow TSCNT to run

// Run Time (e.g. CEVT3 triggered ISR call)
//=========================================
TSt1 = ECAP->CAP0; // Fetch Time-Stamp captured at t1
TSt2 = ECAP->CAP1; // Fetch Time-Stamp captured at t2
TSt3 = ECAP->CAP2; // Fetch Time-Stamp captured at t3
TSt4 = ECAP->CAP3; // Fetch Time-Stamp captured at t4
Period1 = TSt2-TSt1; // Calculate 1st period
Period2 = TSt3-TSt2; // Calculate 2nd period
Period3 = TSt4-TSt3; // Calculate 3rd period
```

Example 2 – Absolute time-stamp operation for duty and period

The ECAP operating mode shown in Figure 11-12 is almost the same as in the previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information, i.e: Period1 = $t_3 - t_1$, Period2 = $t_5 - t_3$, etc. Duty Cycle1 (on-time %) = ($t_2 - t_1$) / Period1 x 100%, etc. Duty Cycle1 (off-time %) = ($t_3 - t_2$) / Period1 x 100%, etc.

Figure 11-12: Capture sequence for period and duty without counter reset



Example Code

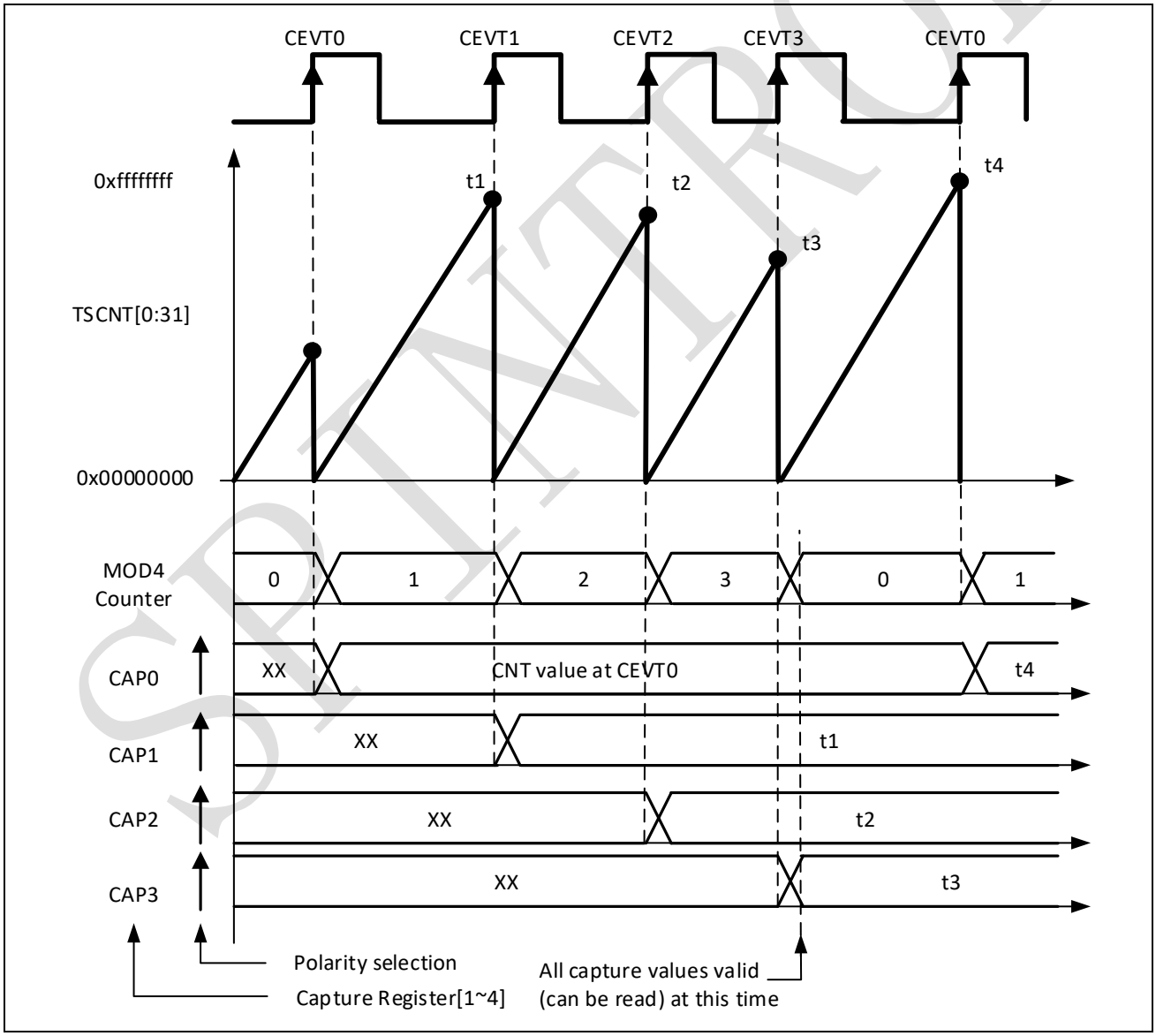
```
// Initialization Time
//=======================
// Spintrol Co., Ltd.. All rights reserved.

ECAP->CAPCTL.bit.CAP0POL = EC_RISING;
ECAP->CAPCTL.bit.CAP1POL = EC_FALLING;
ECAP->CAPCTL.bit.CAP2POL = EC_RISING;
ECAP->CAPCTL.bit.CAP3POL = EC_FALLING;
ECAP->CAPCTL.bit.CNTRST0 = EC_ABS_MODE;
ECAP->CAPCTL.bit.CNTRST1 = EC_ABS_MODE;
ECAP->CAPCTL.bit.CNTRST2 = EC_ABS_MODE;
ECAP->CAPCTL.bit.CNTRST3 = EC_ABS_MODE;
ECAP->CAPCTL.bit.CAPLDEN = EC_ENABLE;
ECAP->CAPCTL.bit.EVTDIV = EC_DIV1;
ECAP->CAPCTL.bit.CAPAPWM = EC_CAP_MODE;
ECAP->CAPCTL.bit.MODE = EC_CONTINUOUS;
ECAP->CAPCTL.bit.SYNCOSEL = EC_SYNCO_DIS;
ECAP->CAPCTL.bit.SYNCIEN = EC_DISABLE;
ECAP->CAPCTL.bit.TSCNTRUN = EC_RUN; // Allow TSCNT to run

// Run Time (e.g. CEVT3 triggered ISR call)
//=========================================
TSt1 = ECAP->CAP0; // Fetch Time-Stamp captured at t1
TSt2 = ECAP->CAP1; // Fetch Time-Stamp captured at t2
TSt3 = ECAP->CAP2; // Fetch Time-Stamp captured at t3
TSt4 = ECAP->CAP3; // Fetch Time-Stamp captured at t4
Period1 = TSt3-TSt1; // Calculate 1st period
DutyOnTime1 = TSt2-TSt1; // Calculate On time
DutyOffTime1 = TSt3-TSt2; // Calculate Off time
```

Example 3 – Time difference (Delta) operation with counter reset

Figure 11-13 shows how the ECAP module can be used to collect Delta timing data from pulse train waveforms. Here Continuous Capture mode (TSCNT counts-up without resetting, and Mod4 counter wraps around) is used. In Delta-time mode, TSCNT is reset back to Zero on every valid event. Here Capture events are qualified as rising edge only. On an event, TSCNT contents (i.e. Time-Stamp) is captured first, and then TSCNT is reset to Zero. The Mod4 counter then increments to the next state.

If TSCNT reaches 0xFFFFFFFF (i.e. Max value), before the next event, it wraps around to 0x00000000 and continues, a CNTOVF (counter overflow) Flag is set, and an Interrupt (if enabled) occurs. The advantage of Delta-time Mode is that the CAPx contents directly give timing data without the need for CPU calculations, i.e. Period1 = T_1, Period2 = T_2, etc. As shown in the diagram, the CEVT0 event is a good trigger point to read the timing data, T_1, T_2, T_3, T_4 are all valid here.

Figure 11-13: Capture period with counter reset



Example Code

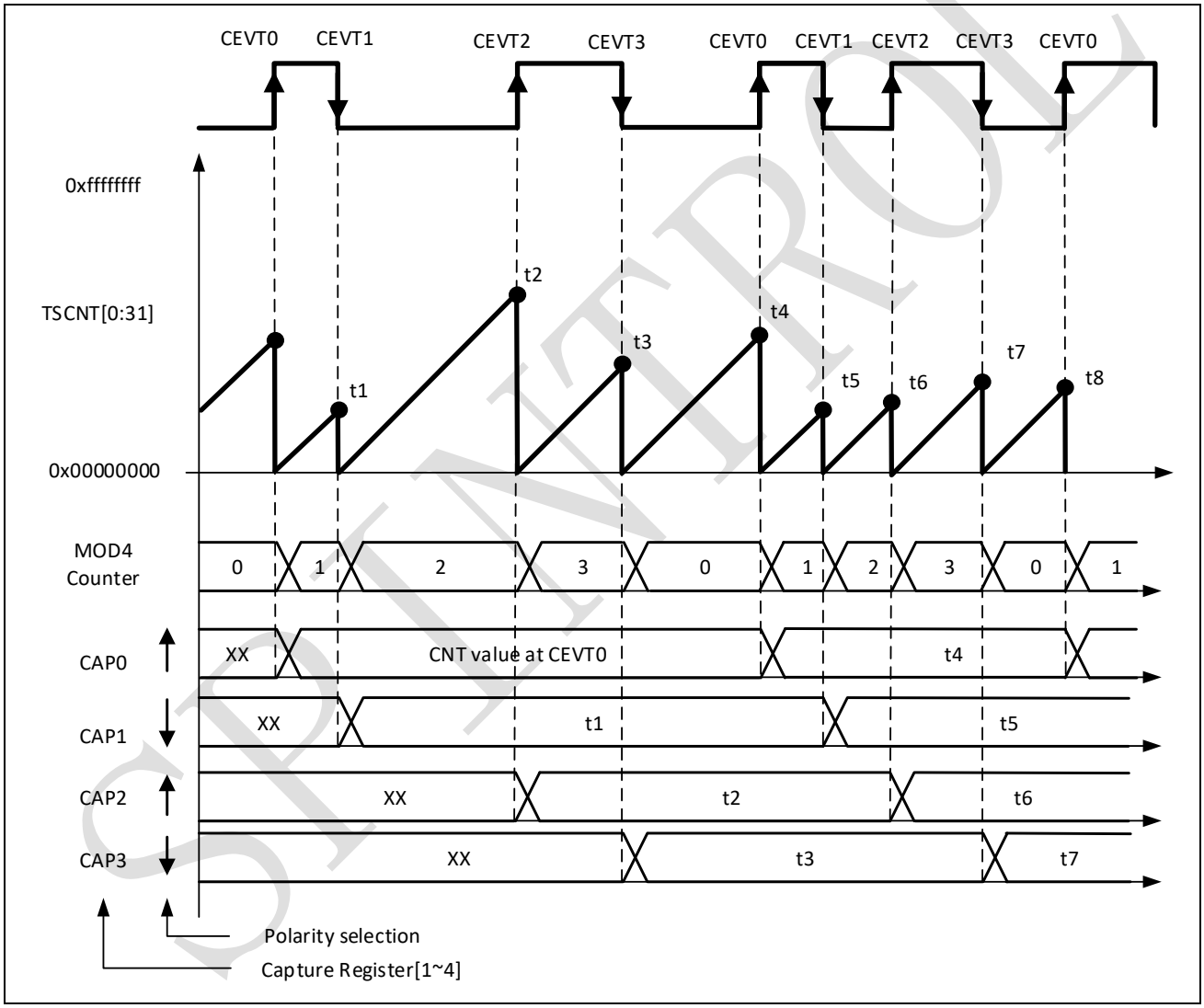
```
// Initialization Time
//=======================
//Spintrol Co., Ltd.. All rights reserved.

ECAP->CAPCTL.bit.CAP0POL = EC_RISING;
ECAP->CAPCTL.bit.CAP1POL = EC_RISING;
ECAP->CAPCTL.bit.CAP2POL = EC_RISING;
ECAP->CAPCTL.bit.CAP3POL = EC_RISING;
ECAP->CAPCTL.bit.CNTRST0 = EC_DELTA_MODE;
ECAP->CAPCTL.bit.CNTRST1 = EC_DELTA_MODE;
ECAP->CAPCTL.bit.CNTRST2 = EC_DELTA_MODE;
ECAP->CAPCTL.bit.CNTRST3 = EC_DELTA_MODE;
ECAP->CAPCTL.bit.CAPLDEN = EC_ENABLE;
ECAP->CAPCTL.bit.EVTDIV = EC_DIV1;
ECAP->CAPCTL.bit.CAPAPWM = EC_CAP_MODE;
ECAP->CAPCTL.bit.MODE = EC_CONTINUOUS;
ECAP->CAPCTL.bit.SYNCOSEL = EC_SYNCO_DIS;
ECAP->CAPCTL.bit.SYNCIEN = EC_DISABLE;
ECAP->CAPCTL.bit.TSCNTRUN = EC_RUN; // Allow TSCNT to run

// Run Time (e.g. CEVT0 triggered ISR call)
//=========================================
// Note: here Time-stamp directly represents the Period value.
Period3 = ECAP->CAP0; // Fetch Time-Stamp captured at T1
Period0 = ECAP->CAP1; // Fetch Time-Stamp captured at T2
Period1 = ECAP->CAP2; // Fetch Time-Stamp captured at T3
Period2 = ECAP->CAP3; // Fetch Time-Stamp captured at T4
```

Example 4 – Time difference operation rising and falling edge trigger

In Figure 11-14 the ECAP operating mode is almost the same as in previous section except Capture events are qualified as either Rising or Falling edge, this now gives both Period and Duty cycle information, i.e:Period1 = T_1+T_2, Period2 = T_3+T_4,etc Duty Cycle1 (on-time %) = T_1 / Period1 x 100%, etc Duty Cycle1(off-time %) = T_2 / Period1 x 100%, etc.

During initialization, you must access the active registers for both period and compare. This will then automatically copy the init values into the shadow values. During run-time, only the shadow registers must be used.

Figure 11-14: Capture duty and period with counter reset



Example Code

```
// Initialization Time
//=======================
// Spintrol Co., Ltd.. All rights reserved.

ECAP->CAPCTL.bit.CAP0POL = EC_RISING;
ECAP->CAPCTL.bit.CAP1POL = EC_FALLING;
ECAP->CAPCTL.bit.CAP2POL = EC_RISING;
ECAP->CAPCTL.bit.CAP3POL = EC_FALLING;
ECAP->CAPCTL.bit.CNTRST0 = EC_DELTA_MODE;
ECAP->CAPCTL.bit.CNTRST1 = EC_DELTA_MODE;
ECAP->CAPCTL.bit.CNTRST2 = EC_DELTA_MODE;
ECAP->CAPCTL.bit.CNTRST3 = EC_DELTA_MODE;
ECAP->CAPCTL.bit.CAPLDEN = EC_ENABLE;
ECAP->CAPCTL.bit.EVTDIV = EC_DIV1;
ECAP->CAPCTL.bit.CAPAPWM = EC_CAP_MODE;
ECAP->CAPCTL.bit.MODE = EC_CONTINUOUS;
ECAP->CAPCTL.bit.SYNCOSEL = EC_SYNCO_DIS;
ECAP->CAPCTL.bit.SYNCIEN = EC_DISABLE;
ECAP->CAPCTL.bit.TSCNTRUN = EC_RUN; // Allow TSCNT to run

// Run Time (e.g. CEVT0 triggered ISR call)
//=========================================
// Note: here Time-stamp directly represents the Duty cycle values.
DutyOnTime1 = ECAP->CAP1; // Fetch Time-Stamp captured at T2
DutyOffTime1 = ECAP->CAP2; // Fetch Time-Stamp captured at T3
DutyOnTime2 = ECAP->CAP3; // Fetch Time-Stamp captured at T4
DutyOffTime2 = ECAP->CAP0; // Fetch Time-Stamp captured at T1
Period1 = DutyOnTime1 + DutyOffTime1;
Period2 = DutyOnTime2 + DutyOffTime2;
```

11.7 Registers

11.7.1 ECAP register map

Table 11-1: ECAP Module Base Address

Peripheral Module	Base Address
ECAP	0x4000 A000

Table 11-2: ECAP Register Map

Register	Offset	Description	Reset Value
CAPSRCCTL	0x0	Capture Source Input Control Register	0x0000007F
CAPSYNCICTL	0x4	Capture Synchronization Source Input Control Register	0x0000007F
CAPTSCNT	0x8	Time-Stamp Counter Register	0x00000000
CAPCNTPHS	0xC	Counter Phase Offset Value Register	0x00000000
CAP0	0x10	Capture Register 0	0x00000000
CAP1	0x14	Capture Register 1	0x00000000
CAP2	0x18	Capture Register 2	0x00000000
CAP3	0x1C	Capture Register 3	0x00000000
CAPCTL	0x20	Capture Control Register	0x000C0000
CAPIF	0x24	Capture Interrupt Flag Register	0x00000000
CAPIE	0x28	Capture Interrupt Enable Register	0x00000000
CAPIC	0x2C	Capture Interrupt Clear Register	0x00000000
CAPIFRC	0x30	Capture Interrupt Force Register	0x00000000
CAPREGKEY	0x34	Capture Register Write-Allow Key Register	0x1ACCE551

11.7.2 ECAP registers

Table 11-3: Capture Source Input Control Register (CAPSRCCTL) Layout

CAPSRCCTL (Capture Source Input Control Register) Offset: 0x0 Default: 0x0000007F Access: ECAP -> CAPSRCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 11-4: Capture Source Input Control Register (CAPSRCCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	Source input polarity 0: Active low 1: Active high
5:0	IOSEL	RW	0x3F	GPIO number to be selected as capture input

Table 11-5: Capture Synchronization Source Input Control Register (CAPSYNCICTL) Layout

CAPSYNCICTL (Capture Synchronization Source Input Control Register) Offset: 0x4 Default: 0x0000007F Access: ECAP -> CAPSYNCICTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 11-6: Capture Synchronization Source Input Control Register (CAPSYNCICTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	Source input polarity 0: Active low 1: Active high
5:0	IOSEL	RW	0x3F	GPIO number to be selected as sync input

Table 11-7: Time-Stamp Counter Register (CAPTSCNT) Layout

CAPTSCNT (Time-Stamp Counter Register) Offset: 0x8 Default: 0x00000000 Access: ECAP -> CAPTSCNT.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 11-8: Time-Stamp Counter Register (CAPTSCNT) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Active 32-bit counter register that is used as the capture time-base.

Table 11-9: Counter Phase Offset Value Register (CAPCNTPHS) Layout

CAPCNTPHS (Counter Phase Offset Value Register) Offset: 0xC Default: 0x00000000 Access: ECAP -> CAPCNTPHS.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 11-10: Counter Phase Offset Value Register (CAPCNTPHS) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Counter phase value register that can be programmed for phase lag/lead.

Table 11-11: Capture Register 0 (CAP0) Layout

CAP0 (Capture Register 0) Offset: 0x10 Default: 0x00000000 Access: ECAP -> CAP0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 11-12: Capture Register 0 (CAP0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	This register can be loaded by: - Time-Stamp(i.e., counter value) during a capture event - Software - may be useful for test purposes - APRD shadow register(i.e., CAP2) when used in APWM mode NOTE: In APWM mode, writing to CAP0/CAP1 active registers also writes the same value to the corresponding shadow registers CAP2/CAP3. This emulates immediate mode. Writing to the shadow registers CAP2/CAP3 invokes the shadow mode.

Table 11-13: Capture Register 1 (CAP1) Layout

CAP1 (Capture Register 1) Offset: 0x14 Default: 0x00000000 Access: ECAP -> CAP1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 11-14: Capture Register 1 (CAP1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	This register can be loaded by: - Time-Stamp(i.e., counter value) during a capture event - Software - may be useful for test purposes - ACMP shadow register(i.e., CAP3) when used in APWM mode NOTE: In APWM mode, writing to CAP0/CAP1 active registers also writes the same value to the corresponding shadow registers CAP2/CAP3. This emulates immediate mode. Writing to the shadow registers CAP2/CAP3 invokes the shadow mode.

Table 11-15: Capture Register 2 (CAP2) Layout

CAP2 (Capture Register 2) Offset: 0x18 Default: 0x00000000 Access: ECAP -> CAP2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 11-16: Capture Register 2 (CAP2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the period shadow(APRD) register. You update the PWM period value through this register. In this mode, CAP2(APRD) shadows CAP0.

Table 11-17: Capture Register 3 (CAP3) Layout

CAP3 (Capture Register 3) Offset: 0x1C Default: 0x00000000 Access: ECAP -> CAP3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 11-18: Capture Register 3 (CAP3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the compare shadow(ACMP) register. You update the PWM compare value via this register. In this mode, CAP3(ACMP) shadows CAP1.

Table 11-19: Capture Control Register (CAPCTL) Layout

CAPCTL (Capture Control Register) Offset: 0x20 Default: 0x000C0000 Access: ECAP -> CAPCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_30		DBGRUN		APWMPOL	APWMMODE	FRCSYNC	SYNCOSEL
23	22	21	20	19	18	17	16
SYNCOSEL	PHSEN	TSCNTRUN	REARM	STOPWRAP		ONESHOT	EVTDIV
15	14	13	12	11	10	9	8
EVTDIV							CAPLDEN
7	6	5	4	3	2	1	0
CNTRST3	CAP3POL	CNTRST2	CAP2POL	CNTRST1	CAP1POL	CNTRST0	CAP0POL

Table 11-20: Capture Control Register (CAPCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29:28	DBGRUN	RW	0x0	ECAP counter behaviour when CPU is halted Note: Debug operation and CPU fault exceptions both can cause CPU halted. 00: TSCNT counter stops immediately when CPU is halted 01: TSCNT counter runs until = 0 when CPU is halted 10: TSCNT counter is unaffected when CPU is halted 11: TSCNT counter is unaffected when CPU is halted
27	APWMPOL	RW	0x0	APWM output polarity select. This is applicable only in APWM operating mode. 0: Output is active low (i.e., Compare value defines low time) 1: Output is active high (i.e., Compare value defines high time)

Bits	Field Name	Type	Reset	Description
26	APWMMODE	RW	0x0	CAP/APWM operating mode select 0: ECAP module operates in capture mode. This mode forces the following configuration: - Inhibits TSCNT resets via CNT = PRD event - Inhibits shadow loads on CAP0 and CAP1 registers - Permits user to enable CAP0-3 register load - CAPx/APWMx pin operates as a capture input 1: ECAP module operates in APWM mode. This mode forces the following configuration: - Resets TSCNT on CNT = PRD event (period boundary - Permits shadow loading on CAP0 and CAP1 registers - Disables loading of time-stamps into CAP0-3 registers - CAPx/APWMx pin operates as a APWM output
25	FRCSYNC	W1S	0x0	Software-forced Counter (TSCNT) Synchronizing. This provides a convenient software method to synchronize some or all ECAP time bases. In APWM mode, the synchronizing can also be done via the CNT = PRD event. Note: Selection CNT = PRD is meaningful only in APWM mode; however, you can choose it in CAP mode if you find doing so useful. 0: Writing a zero has no effect. Reading always returns a zero 1: Writing a one forces a TSCNT shadow load of current ECAP module and any ECAP modules down-stream providing the SYNCOSEL bits are 00. After writing a 1, this bit returns to a zero.
24:23	SYNCOSEL	RW	0x0	Sync-Out Select 00: Disable sync out signal 01: Disable sync out signal 10: Select sync-in event to be the sync-out signal (pass through) 11: Select CNT = PRD event to be the sync-out signal
22	PHSEN	RW	0x0	Phase synchronization enable 0: Disable synchronization 1: Enable counter (CAPTSCNT) to be loaded from CAPCNTPHS register upon either a SYNCI signal or a S/W force event

Bits	Field Name	Type	Reset	Description
21	TSCNTRUN	RW	0x0	Time stamp counter (TSCNT) run/stop control 0: TSCNT stopped 1: TSCNT free-running
20	REARM	W1S	0x0	One-Shot Re-Arming Control, i.e. wait for stop trigger. Note: The re-arm function is valid in one shot or continuous mode. 0: Write a 0 has no effect. Always reads back a 0. 1: Arms the one-shot sequence as follows: 1) Resets the Mod4 counter to zero 2) Unfreezes the Mod4 counter 3) Enables capture register loads
19:18	STOPWRAP	RW	0x3	Stop value for one-shot mode This is the number (between 0-3) of captures allowed to occur before the CAP(0-3) registers are frozen, i.e., capture sequence is stopped. Wrap value for continuous mode. This is the number (between 0-3) of the capture register in which the circular buffer wraps around and starts again. Notes: STOP_WRAP is compared to Mod4 counter and, when equal, 2 actions occur: - Mod4 counter is stopped (frozen) - Capture register loads are inhibited In one-shot mode, further interrupt events are blocked until re-armed. 00: Stop after capture event 0 in one-shot mode. Wrap after capture event 0 in continuous mode. 01: Stop after capture event 1 in one-shot mode. Wrap after capture event 1 in continuous mode. 10: Stop after capture event 2 in one-shot mode. Wrap after capture event 2 in continuous mode. 11: Stop after capture event 3 in one-shot mode. Wrap after capture event 3 in continuous mode.
17	ONESHOT	RW	0x0	One-shot mode enable (applicable only in capture mode) 0: Operate in continuous mode 1: Operate in one-shot mode
16:9	EVTDIV	RW	0x0	Event Filter prescale select

Bits	Field Name	Type	Reset	Description
8	CAPLDEN	RW	0x0	Enable CAP0-3 loading on a capture event 0: Disable CAP0-3 register loads at capture event time 1: Enable CAP0-3 register loads at capture event time
7	CNTRST3	RW	0x0	Counter reset on capture event 3 0: Do not reset counter on Capture Event 3 (absolute time stamp operation) 1: Reset counter after Capture Event 3 time-stamp has been captured (used in difference mode operation)
6	CAP3POL	RW	0x0	Capture event 3 polarity select 0: Capture Event 3 triggered on a falling edge (FE) 1: Capture Event 3 triggered on a rising edge (RE)
5	CNTRST2	RW	0x0	Counter reset on capture event 2 0: Do not reset counter on Capture Event 2 (absolute time stamp) 1: Reset counter after Event 2 time-stamp has been captured (used in difference mode operation)
4	CAP2POL	RW	0x0	Capture event 2 polarity select 0: Capture Event 2 triggered on a falling edge (FE) 1: Capture Event 2 triggered on a rising edge (RE)
3	CNTRST1	RW	0x0	Counter reset on capture event 1 0: Do not reset counter on Capture Event 1 (absolute time stamp) 1: Reset counter after Event 1 time-stamp has been captured (used in difference mode operation)
2	CAP1POL	RW	0x0	Capture event 1 polarity select 0: Capture Event 1 triggered on a falling edge (FE) 1: Capture Event 1 triggered on a rising edge (RE)
1	CNTRST0	RW	0x0	Counter reset on capture event 0 0: Do not reset counter on Capture Event 0 (absolute time stamp) 1: Reset counter after Event 0 time-stamp has

Bits	Field Name	Type	Reset	Description
				been captured (used in difference mode operation)
0	CAP0POL	RW	0x0	Capture event 0 polarity select 0: Capture Event 0 triggered on a falling edge (FE) 1: Capture Event 0 triggered on a rising edge (RE)

Table 11-21: Capture Interrupt Flag Register (CAPIF) Layout

CAPIF (Capture Interrupt Flag Register) Offset: 0x24 Default: 0x00000000 Access: ECAP -> CAPIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7 INT	6 CMP	5 PRD	4 CNTOVF	3 CEVT3	2 CEVT2	1 CEVT1	0 CEVT0

Table 11-22: Capture Interrupt Flag Register (CAPIF) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	INT	RO	0x0	Global interrupt flag 0: Indicates no interrupt generated 1: Indicates that an interrupt was generated
6	CMP	RO	0x0	Counter comparison equal flag. This flag is active only in APWM mode. 0: Indicates no event occurred 1: Indicates the counter (TSCNT) reached the compare register value (ACMP)
5	PRD	RO	0x0	Counter equal period flag. This flag is only active in APWM mode. 0: Indicates no event occurred 1: Indicates the counter (TSCNT) reached the period register value (APRD) and was reset
4	CNTOVF	RO	0x0	Counter overflow status Flag. This flag is active in CAP and APWM mode. 0: Indicates no event occurred 1: Indicates the counter (TSCNT) has made the transition from FFFFFFFF to 00000000
3	CEVT3	RO	0x0	Capture event 3 flag. This flag is only active in CAP mode. 0: Indicates no event occurred 1: Indicates the fourth event occurred at ECAPx pin
2	CEVT2	RO	0x0	Capture event 2 flag. This flag is active only in CAP mode. 0: Indicates no event occurred 1: Indicates the third event occurred at ECAPx pin

Bits	Field Name	Type	Reset	Description
1	CEVT1	RO	0x0	Capture event 1 flag. This flag is only active in CAP mode. 0: Indicates no event occurred 1: Indicates the second event occurred at ECAPx pin
0	CEVT0	RO	0x0	Capture event 0 flag. This flag is only active in CAP mode. 0: Indicates no event occurred 1: Indicates the first event occurred at ECAPx pin

Table 11-23: Capture Interrupt Enable Register (CAPIE) Layout

CAPIE (Capture Interrupt Enable Register) Offset: 0x28 Default: 0x00000000 Access: ECAP -> CAPIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	CMP	PRD	CNTOVF	CEVT3	CEVT2	CEVT1	CEVT0

Table 11-24: Capture Interrupt Enable Register (CAPIE) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	CMP	RW	0x0	Counter comparison equal interrupt enable 0: Disable Compare Equal as an Interrupt source 1: Enable Compare Equal as an Interrupt source
5	PRD	RW	0x0	Counter equal period interrupt enable 0: Disable Period Equal as an Interrupt source 1: Enable Period Equal as an Interrupt source
4	CNTOVF	RW	0x0	Counter overflow interrupt enable 0: Disabled counter Overflow as an Interrupt source 1: Enable counter Overflow as an Interrupt source
3	CEVT3	RW	0x0	Capture event 3 interrupt enable 0: Disable Capture Event 3 as an Interrupt source 1: Enable Capture Event 3 as an Interrupt source

Bits	Field Name	Type	Reset	Description
2	CEVT2	RW	0x0	Capture event 2 interrupt enable 0: Disable Capture Event 2 as an Interrupt source 1: Enable Capture Event 2 as an Interrupt source
1	CEVT1	RW	0x0	Capture event 1 interrupt enable 0: Disable Capture Event 1 as an Interrupt source 1: Enable Capture Event 1 as an Interrupt source
0	CEVT0	RW	0x0	Capture event 0 interrupt enable 0: Disable Capture Event 0 as an Interrupt source 1: Enable Capture Event 0 as an Interrupt source

Table 11-25: Capture Interrupt Clear Register (CAPIC) Layout

CAPIC (Capture Interrupt Clear Register) Offset: 0x2C Default: 0x00000000 Access: ECAP -> CAPIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
INT	CMP	PRD	CNTOVF	CEVT3	CEVT2	CEVT1	CEVT0

Table 11-26: Capture Interrupt Clear Register (CAPIC) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	INT	W1C	0x0	Global interrupt clear 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 clears the INT flag and enable further interrupts to be generated if any of the event flags are set to 1. This bit is self-cleared to 0.
6	CMP	W1C	0x0	Counter comparison equal flag clear 0: Writing a 0 has no effect. Always reads back a 0 1: Writing a 1 clears the CNT=CMP flag. This bit is self-cleared to 0.

Bits	Field Name	Type	Reset	Description
5	PRD	W1C	0x0	Counter equal to period flag clear 0: Writing a 0 has no effect. Always reads back a 0 1: Writing a 1 clears the CNT=PRD flag. This bit is self-cleared to 0.
4	CNTOVF	W1C	0x0	Counter overflow flag clear 0: Writing a 0 has no effect. Always reads back a 0 1: Writing a 1 clears the CNTOVF flag. This bit is self-cleared to 0.
3	CEVT3	W1C	0x0	Capture event 3 flag clear 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 clears the CEVT3 flag. This bit is self-cleared to 0.
2	CEVT2	W1C	0x0	Capture event 2 flag clear 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 clears the CEVT2 flag. This bit is self-cleared to 0.
1	CEVT1	W1C	0x0	Capture event 1 flag clear 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 clears the CEVT1 flag. This bit is self-cleared to 0
0	CEVT0	W1C	0x0	Capture event 0 flag clear 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 clears the CEVT0 flag. This bit is self-cleared to 0.

Table 11-27: Capture Interrupt Force Register (CAPIFRC) Layout

CAPIFRC (Capture Interrupt Force Register) Offset: 0x30 Default: 0x00000000 Access: ECAP -> CAPIFRC.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	CMP	PRD	CNTOVF	CEVT3	CEVT2	CEVT1	CEVT0

Table 11-28: Capture Interrupt Force Register (CAPIFRC) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	CMP	W1S	0x0	Counter comparison equal flag force 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 forces the CNT=CMP flag. This bit is self-cleared to 0.
5	PRD	W1S	0x0	Counter equal to period flag force 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 forces the CNT=PRD flag. This bit is self-cleared to 0.
4	CNTOVF	W1S	0x0	Counter overflow flag force 0: Writing a 0 has no effect. Always reads back a 0 1: Writing a 1 forces the CNTOVF flag. This bit is self-cleared to 0.
3	CEVT3	W1S	0x0	Capture event 3 flag force 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 forces the CEVT3 flag. This bit is self-cleared to 0.
2	CEVT2	W1S	0x0	Capture event 2 flag force 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 forces the CEVT2 flag. This bit is self-cleared to 0.
1	CEVT1	W1S	0x0	Capture event 1 flag force 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 forces the CEVT1 flag. This bit is self-cleared to 0
0	CEVT0	W1S	0x0	Capture event 0 flag force 0: Writing a 0 has no effect. Always reads back a 0. 1: Writing a 1 forces the CEVT0 flag. This bit is self-cleared to 0.

Table 11-29: Capture Register Write-Allow Key Register (CAPREGKEY) Layout

CAPREGKEY (Capture Register Write-Allow Key Register) Offset: 0x34 Default: 0x1ACCE551 Access: ECAP -> CAPREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 11-30: Capture Register Write-Allow Key Register (CAPREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected capture registers

12 ADC

12.1 ADC overview

The SPD1148 implements a 14-bit successive approximation A/D converter (SAR-ADC). There are triple sample and hold circuits in front of ADC. The sample and hold circuits can be sampled sequentially or simultaneously. There are 16 analog input channels to ADC. The converter can be configured with an internal bandgap reference or an external voltage reference.

This ADC is easy for the user to create a series of conversions from a single trigger. The basic principle of operation is based on the configurations of individual conversions, called SOC's, or State-Of-Conversions.

Functions of the ADC module include:

- 14-bit resolution ADC core with built-in triple sample-and-hold (S/H)
  - Up to 4 Million samples per second (Msps)
  - Up to 16-channel, multiplexed inputs
- Support 3 channel simultaneous sampling
- Analog full scale input: 3.657 V fixed, or an external voltage reference ratio-metric from GPIO12
- 16 SOC's, configurable trigger source, sampling mode, channel selection, averaging time and sample time
- 16 result registers (individually addressable) to store conversion values
- Multiple trigger sources
  - Software immediate start
  - Timer 0/1/2
  - GPIO external interrupt
  - PWMxSOCA, PWMxSOCB and PWMxSOCC (x=0~5) with configurable timing
- 16 independent interrupt signals with 16 independent NVIC interrupts, can configure interrupt request after any conversion

12.2 ADC architecture

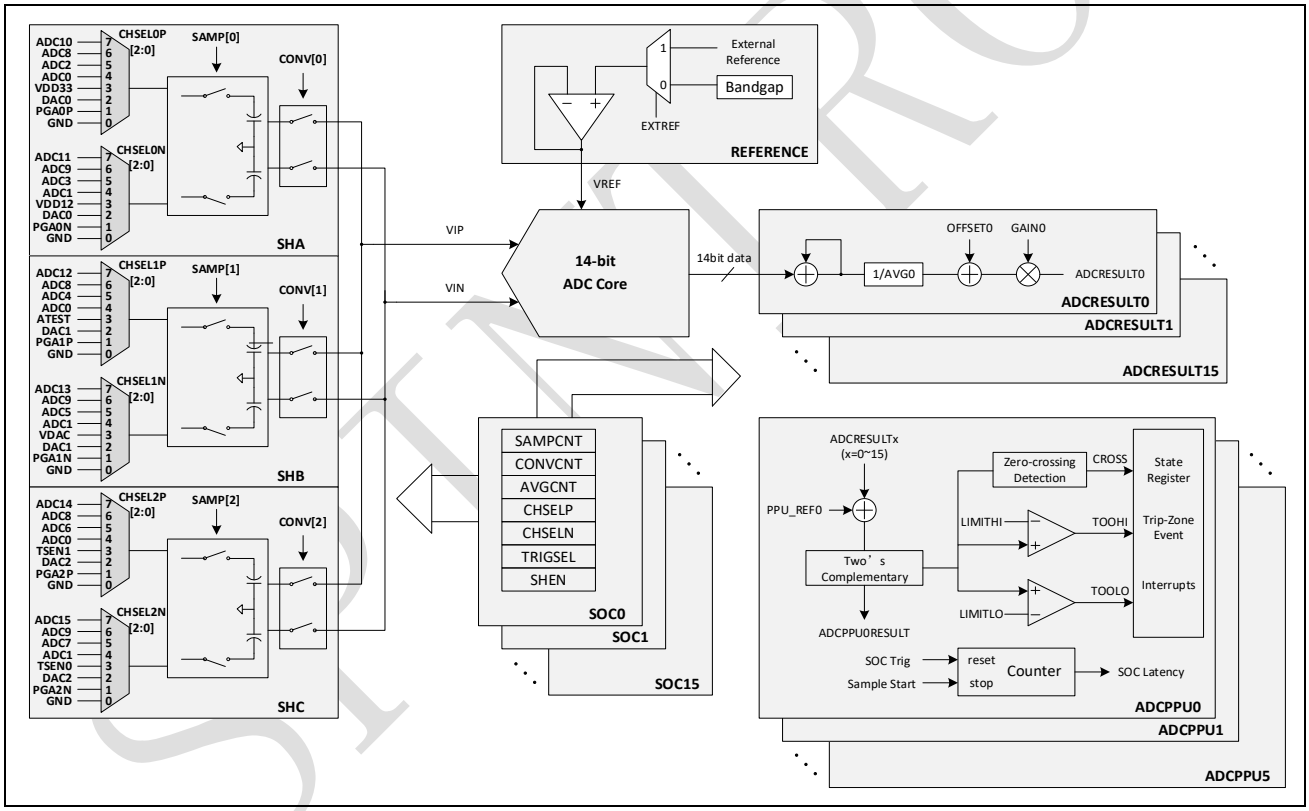
As shown in Figure 12-1, the ADC analog circuit includes:

- 3 ADC samplers (SHA, SHB, SHC)
- 1 ADC Core
- ADC reference, as well as other analog supporting circuits

The digital circuit includes:

- 16 SOC configuration control
- 16 ADC result modules (ADCRESULT) with averaging, gain and offset calibration
- 6 post-processing units (PPU)
- Interface to other on-chip modules (not showed in diagram)

Figure 12-1: ADC block diagram



12.3 SOC principle of operation

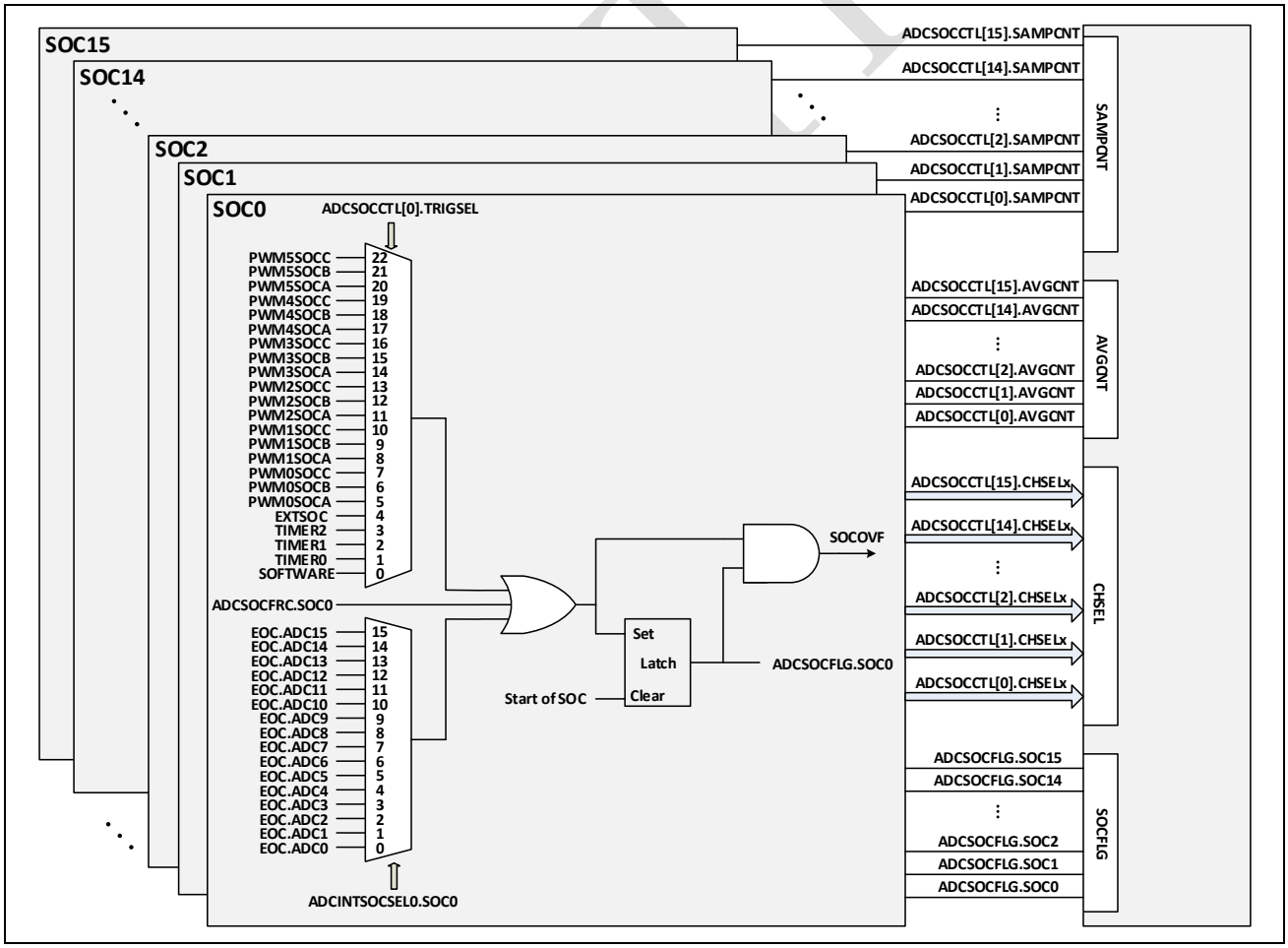
This ADC is SOC-based. The term SOC means a configuration set defining a single conversion. In one SOC there are four configurations: the trigger source that starts the conversion, the channel selection, the average times and the windows size of sample control. In additional, the sampling mode control also is set in SOC which is described in Section 12.3.2. As showed in Figure 12-2, each SOC is independently configured and can have any combination of the trigger, channel selection, averaging times and sample window size available. Multiple SOC's can be configured for the same trigger, channel selection, averaging times and sample window size as desired. It can be implemented for individual samples of different channels with different triggers, or to oversample the same channel using a single trigger, or to create a series of conversions of different channels all from a single trigger.

The trigger source for SOCx is configured by a combination of the TRIGSEL field in the ADCSOCCTL[x] register and the appropriate bits in the ADCINTSOCSEL0~1 register. Software can also force an SOC event with the ADCSOCFRC register.

The channel selection is configured by a combination of SHEN and CHSELx (x=P or N) field in ADCSOCCTL[x] register.

The sample time for SOCx are configured with SAMPCNT fields in ADCSOCCTL[x] register respectively.

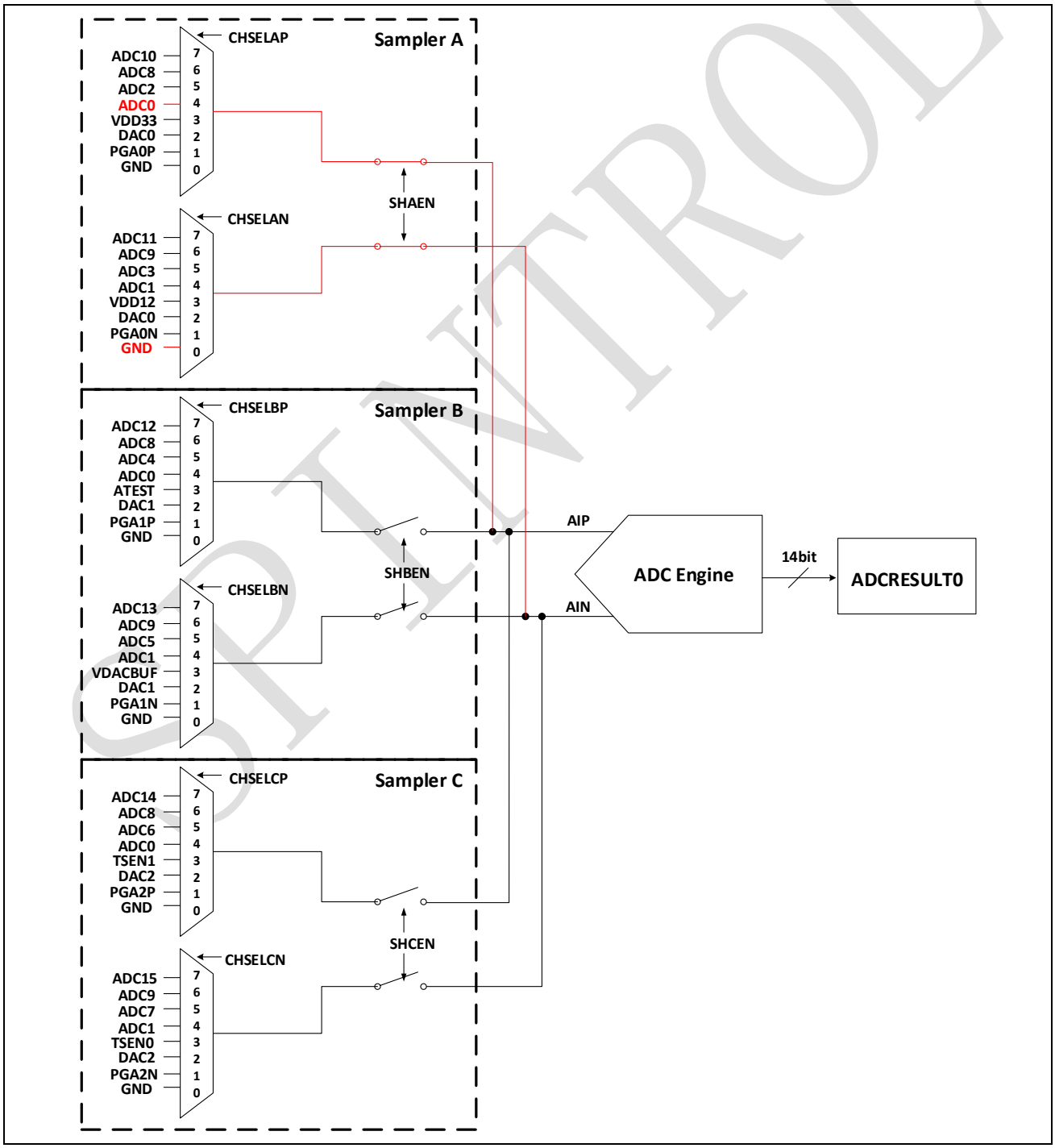
Figure 12-2: SOC block diagram



Example 12.3.1 PWMSOC1A trigger SOC0 to convert ADC0

To configure a single conversion on channel ADC0 to occur when the PWM1SOCA timer reaches its period match, one must first setup PWM1SOCA to output an SOC signal on a period match. Let's configure one of the SOC's using its ADCSOCCTL[x] register. It makes no difference which SOC to choose, so as an example use SOC0. Setting PWMSOC1A for the SOC0 trigger source, select averaging is 4 times, selecting channel ADC0 to convert and choosing 4 cycles and 4 cycles for the sample and convert time. So we'll set the TRIGSEL field to 8, the AVGCNT field to 2(power 2 based), the SHEN=1(only Sampler A on work), CHSELP to 4 (ADC0), the CHSELN to 0 (GND), the SAMPCNT field to 3, the CONVCNT field to 3, respectively.

Figure 12-3: ADC channel selection for Example 12.3.1



It can be divided in to the followed lines:

Example 12.3.1

```
void  ADC_Example12_3_1(void)
{
  ADC->ADCSOCCTL[0].bit.SHEN     = 1;/* Enable Sampler A                          */
  ADC->ADCSOCCTL[0].bit.TRIGSEL = 8;/* Select PWM1SOCA as the trigger source  */
  ADC->ADCSOCCTL[0].bit.CHSELP   = 4;/* Sampler A Positive = ADC0               */
  ADC->ADCSOCCTL[0].bit.CHSELN   = 0;/* Sampler A Negative = GND                */
  ADC->ADCSOCCTL[0].bit.AVGCNT   = 2;/* Averaging 4 times (2^2)                 */
  ADC->ADCSOCCTL[0].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock     */
  ADC->ADCSOCCTL[0].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock   */
}
```

Assume that ADC clock is 32MHz, 31.25ns for one cycle. So the fastest configuration are 125ns sampling time (SAMPCNT=3) and 156.25ns conversion time (CONVCNT=4). Single conversion of ADC0 will be started on a PWM1SOCA event with the resulting value stored in the ADCRESULT[0] register. In this example, we use positive input as ADC channel and negative input as GND. That means the differential input sign is plus. Alternately, we can also use positive input as GND and use negative input as ADC channel. That means the differential input sign is minus.

Example 12.3.2 Oversample ADC0

If ADC0 needed to be oversampled by 3X, then SOC1 and SOC2 could all be given the same configuration as SOC0.

Example 12.3.2

```
void  ADC_Example12_3_2(void)
{
  ADC->ADCSOCCTL[0].bit.SHEN     = 1;/* Enable Sampler A                          */
  ADC->ADCSOCCTL[0].bit.TRIGSEL = 8;/* Select PWM1SOCA as the trigger source  */
  ADC->ADCSOCCTL[0].bit.CHSELP   = 4;/* Sampler A Positive = ADC0               */
  ADC->ADCSOCCTL[0].bit.CHSELN   = 0;/* Sampler A Negative = GND                */
  ADC->ADCSOCCTL[0].bit.AVGCNT   = 2;/* Averaging 4 times (2^2)                 */
  ADC->ADCSOCCTL[0].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock       */
  ADC->ADCSOCCTL[0].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock     */

  ADC->ADCSOCCTL[1].bit.SHEN     = 1;/* Enable Sampler A                          */
  ADC->ADCSOCCTL[1].bit.TRIGSEL = 8;/* Select PWM1SOCA as the trigger source  */
  ADC->ADCSOCCTL[1].bit.CHSELP   = 4;/* Sampler A Positive = ADC0               */
  ADC->ADCSOCCTL[1].bit.CHSELN   = 0;/* Sampler A Negative = GND                */
  ADC->ADCSOCCTL[1].bit.AVGCNT   = 2;/* Averaging 4 times (2^2)                 */
  ADC->ADCSOCCTL[1].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock       */
  ADC->ADCSOCCTL[1].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock     */

  ADC->ADCSOCCTL[2].bit.SHEN     = 1;/* Enable Sampler A                          */
  ADC->ADCSOCCTL[2].bit.TRIGSEL = 8;/* Select PWM1SOCA as the trigger source  */
  ADC->ADCSOCCTL[2].bit.CHSELP   = 4;/* Sampler A Positive = ADC0               */
  ADC->ADCSOCCTL[2].bit.CHSELN   = 0;/* Sampler A Negative = GND                */
  ADC->ADCSOCCTL[2].bit.AVGCNT   = 2;/* Averaging 4 times (2^2)                 */
  ADC->ADCSOCCTL[2].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock       */
  ADC->ADCSOCCTL[2].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock     */
}
```

When configured as such, three conversions of ADC0 will be started in series on a PWM1SOCA event with the resulting values stored in the ADCRESULT[0] ~ ADCRESULT[2] registers.

Example 12.3.3 Same trigger converts different channels (ADC0, ADC2, ADC8)

Example 12.3.3

```
void  ADC_Example12_3_3(void)
{
  ADC->ADCSOCCTL[0].bit.SHEN     = 1;/* Enable Sampler A                          */
  ADC->ADCSOCCTL[0].bit.TRIGSEL = 8;/* Select PWM1SOCA as the trigger source  */
  ADC->ADCSOCCTL[0].bit.CHSELP = 4;/* Sampler A Positive = ADC0                 */
  ADC->ADCSOCCTL[0].bit.CHSELN = 0;/* Sampler A Negative = GND                  */
  ADC->ADCSOCCTL[0].bit.AVGCNT = 2;/* Averaging 4 times (2^2)                   */
  ADC->ADCSOCCTL[0].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock      */
  ADC->ADCSOCCTL[0].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock      */

  ADC->ADCSOCCTL[1].bit.SHEN     = 1;/* Enable Sampler A                          */
  ADC->ADCSOCCTL[1].bit.TRIGSEL = 8;/* Select PWM1SOCA as the trigger source  */
  ADC->ADCSOCCTL[1].bit.CHSELP = 5;/* Sampler A Positive = ADC2                 */
  ADC->ADCSOCCTL[1].bit.CHSELN = 0;/* Sampler A Negative = GND                  */
  ADC->ADCSOCCTL[1].bit.AVGCNT = 2;/* Averaging 4 times (2^2)                   */
  ADC->ADCSOCCTL[1].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock      */
  ADC->ADCSOCCTL[1].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock      */

  ADC->ADCSOCCTL[2].bit.SHEN     = 1;/* Enable Sampler A                          */
  ADC->ADCSOCCTL[2].bit.TRIGSEL = 8;/* Select PWM1SOCA as the trigger source  */
  ADC->ADCSOCCTL[2].bit.CHSELP = 6;/* Sampler A Positive = ADC8                 */
  ADC->ADCSOCCTL[2].bit.CHSELN = 0;/* Sampler A Negative = GND                  */
  ADC->ADCSOCCTL[2].bit.AVGCNT = 2;/* Averaging 4 times (2^2)                   */
  ADC->ADCSOCCTL[2].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock      */
  ADC->ADCSOCCTL[2].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock      */
}
```

When configured this way, three conversions will be started in series on a PWM1SOCA event. The result of the conversion on channel ADC0 will show up in ADCRESULT0. The result of the conversion on channel ADC2 will show up in ADCRESULT[1]. The result of the conversion on channel ADC8 will show up in ADCRESULT[2]. The channel converted and the trigger has no bearing on where the result of the conversion shows up. The RESULT register is associated with the SOC.

If the application requires three different signals to be sampled from the same trigger, this can be done by simply changing the CHSELx field for SOC0-SOC2 while leaving the TRIGSEL field unchanged.

12.3.1 Trigger operation

Each SOC can be configured to start on one of many input triggers. Multiple SOC's can be configured for the same channel if desired. Following is a list of the available input triggers:

- Software
- Timer 0/1/2 interrupts
- External SOC
- PWMxSOCA, PWMxSOCB and PWMxSOCC (x=0~5)

Please see the ADCSOCCTL[x] Register Bit Definitions for the configuration details of these triggers shown in Table 12-1.

Table 12-1 lists the available trigger source. If TRIGSEL = 4, select the EXTSOC (External SOC event) as the SOC trigger source. Please see Table 12-38 and Table 12-39 for how to select GPIO pin as external SOC event source (ADCEXTSOCCTL).

Additionally ADCINT0~15 can be fed back to trigger another conversion. This configuration is controlled in the ADCINTSOCSEL0~1 registers. This mode is useful if a continuous stream of conversions is desired. See Section 12.6 for information on the ADC interrupt signals.

Table 12-1: Trigger source selection

TRIGSEL	Trigger Source
other	Invalid Selection
22	PWM5SOCC
21	PWM5SOCB
20	PWM5SOCA
19	PWM4SOCC
18	PWM4SOCB
17	PWM4SOCA
16	PWM3SOCC
15	PWM3SOCB
14	PWM3SOCA
13	PWM2SOCC
12	PWM2SOCB
11	PWM2SOCA
10	PWM1SOCC
9	PWM1SOCB
8	PWM1SOCA
7	PWM0SOCC
6	PWM0SOCB

5	PWM0SOCA
4	EXTSOC
3	Timer 2
2	Timer 1
1	Timer 0
0	Software

12.3.2 ADC sampling mode and channel selection

The SHEN bit field controls ADC sampling mode. As Table 12-2 shown, when SHEN equals 0~3, all SOCs have the same control for the sequential sampling model. When SHEN equals 4~7, SOC0/SOC3/SOC6/SOC9/SOC12 have their dedicate control for the simultaneous sampling control.

Table 12-2: Sample mode description

ADCSOCCTL[x].SHEN[2:0] (x=0,3,6,9,12)	ADC Sampling Mode	ADCSOC[y]CTL.SHEN[1:0] (y=1,2,4,5,7,8,10,11,13,14,15)	ADC Sampling Mode
7	Sampler A, B, C Simultaneous	/	/
6	Sampler A and C Simultaneous	/	/
5	Sampler B and C Simultaneous	/	/
4	Sampler A and B Simultaneous	/	/
3	Sampler C Single	3	Sampler C Single
2	Sampler B Single	2	Sampler B Single
1	Sampler A Single	1	Sampler A Single
0	Disable	0	Disable

Although 3 sampler need 3 pair channel selection CHSELxP and CHSELxN (x=A,B,C), there are only one pair field CHSELP and CHSELN in each ADCSOCCTL[x] register. The sampler control bit-fields (SHEN) in ADCSOCCTL[x] register is to select which sampler should work and be converted. The CHSELP and CHSELN are associated with SHEN.

In sequential sampling mode, the SHEN control which sampler's channels are related to CHSELP and CHSELN. Please see Table 12-3 for details.

Table 12-3: ADCSOCCTL[x] channel selection for sequential sampler sampling

SHEN	CHSELAP	CHSELAN	CHSELBP	CHSELBN	CHSELCP	CHSELCN
3	Disable	Disable	Disable	Disable	CHSELP[2:0]	CHSELN[2:0]
2	Disable	Disable	CHSELP[2:0]	CHSELN[2:0]	Disable	Disable
1	CHSELP[2:0]	CHSELN[2:0]	Disable	Disable	Disable	Disable
0	Disable	Disable	Disable	Disable	Disable	Disable

The detail channel selection for each sampler are showed in Table 12-4.

Table 12-4: ADC channel selection

Option	CHSELAP	CHSELAN	CHSELBP	CHSELBN	CHSELCP	CHSELCN
7	ADC10	ADC11	ADC12	ADC13	ADC14	ADC15
6	ADC8	ADC9	ADC8	ADC9	ADC8	ADC9
5	ADC2	ADC3	ADC4	ADC5	ADC6	ADC7
4	ADC0	ADC1	ADC0	ADC1	ADC0	ADC1
3	VDD33	VDD12	ATEST	VDCBUF	TSEN1	TSEN0
2	DAC0	DAC0	DAC1	DAC1	DAC2	DAC2
1	PGA0P	PGA0N	PGA1P	PGA1N	PGA2P	PGA2N
0	GND	GND	GND	GND	GND	GND

In simultaneous mode, the CHSELP and CHSELN control the channel related to SHEN and SOC. For example, we can use SOC0/SOC1/SOC2 to make 3 samplers simultaneous sampling according to the settings in Table 12-5. The channel selection for Sampler A is in ADCSOCCTL[0]. The channel selection for Sampler B is in ADCSOCCTL[1]. The channel selection for Sampler C is in ADCSOCCTL[2].

Table 12-5: Use SOC0/SOC1/SOC2 for 3 sampler simultaneous sampling

SHEN	CHSELAP	CHSELAN	CHSELBP	CHSELBN	CHSELCP	CHSELCN
7	ADCSOCCTL[0].CHSELP[2:0]	ADCSOCCTL[0].CHSELN[2:0]	ADCSOCCTL[1].CHSELP[2:0]	ADCSOCCTL[1].CHSELN[2:0]	ADCSOCCTL[2].CHSELP[2:0]	ADCSOCCTL[2].CHSELN[2:0]

And we can also use SOC4/SOC5 to make sampler B and sampler C simultaneous sampling according to the settings in Table 12-6. In this case, sampler A can't be used because simultaneous configuration is set in ADCSOCCTL[3]. The channel selection for Sampler B is in ADCSOCCTL[4]. The channel selection for Sampler C is in ADCSOCCTL[5].

Table 12-6: Use SOC4/SOC5 for 2 sampler simultaneous sampling

SHEN	CHSELAP	CHSELAN	CHSELBP	CHSELBN	CHSELCP	CHSELCN
6	Disable	Disable	ADCSOCCTL[4].CHSELP[2:0]	ADCSOCCTL[4].CHSELN[2:0]	ADCSOCCTL[5].CHSELP[2:0]	ADCSOCCTL[5].CHSELN[2:0]

For more detail description about simultaneous mode, please refer to Section 12.5.

Example 12.3.4 ADC2, ADC4, ADC6 simultaneous sampling

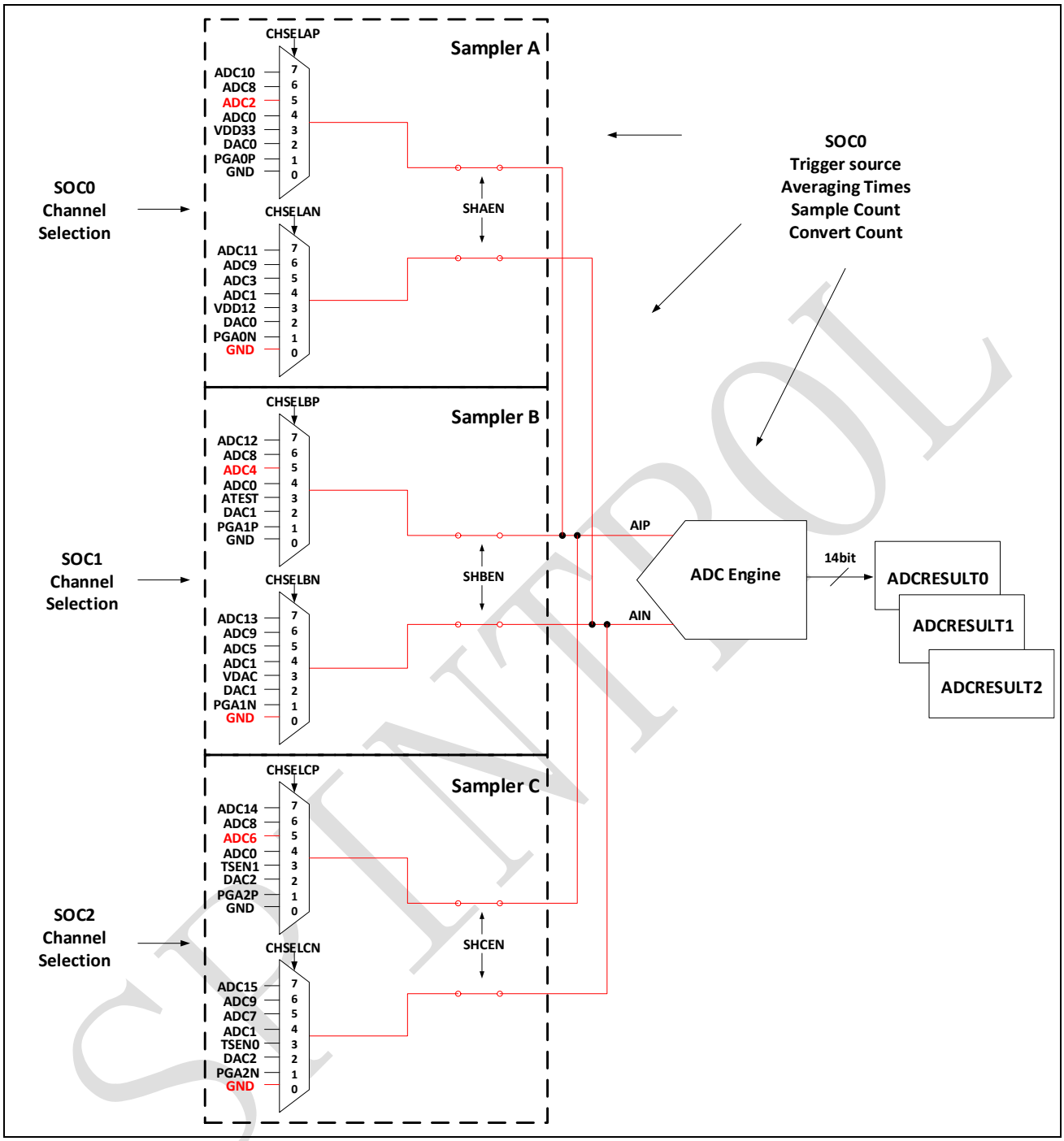
It can be divided in to the followed lines:

Example 12.3.4

```
void  ADC_Example12_3_4(void)
{
  ADC->ADCSOCCTL[0].bit.SHEN     = 7;/* Enable Sampler A/B/C simultaneous     */
  ADC->ADCSOCCTL[0].bit.TRIGSEL = 8;/* Select PWM1SOCA as the trigger source  */
  ADC->ADCSOCCTL[0].bit.CHSELP = 5;/* Sampler A Positive = ADC2               */
  ADC->ADCSOCCTL[0].bit.CHSELN = 0;/* Sampler A Negative = GND                */
  ADC->ADCSOCCTL[0].bit.AVGCNT = 2;/* Averaging 4 times (2^2)                 */
  ADC->ADCSOCCTL[0].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock      */
  ADC->ADCSOCCTL[0].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock       */

  ADC->ADCSOCCTL[1].bit.SHEN     = 1;/* Will be ignore                         */
  ADC->ADCSOCCTL[1].bit.TRIGSEL = 8;/* Will be ignore  */
  ADC->ADCSOCCTL[1].bit.CHSELP = 5;/* Sampler B Positive = ADC4               */
  ADC->ADCSOCCTL[1].bit.CHSELN = 0;/* Sampler B Negative = GND                */
  ADC->ADCSOCCTL[1].bit.AVGCNT = 2;/* Will be ignore                  */
  ADC->ADCSOCCTL[1].bit.SAMPCNT = 3;/* Will be ignore     */
  ADC->ADCSOCCTL[1].bit.CONVCNT = 4;/* Will be ignore      */

  ADC->ADCSOCCTL[2].bit.SHEN     = 1;/* Will be ignore                         */
  ADC->ADCSOCCTL[2].bit.TRIGSEL = 8;/* Will be ignore  */
  ADC->ADCSOCCTL[2].bit.CHSELP = 5;/* Sampler C Positive = ADC6               */
  ADC->ADCSOCCTL[2].bit.CHSELN = 0;/* Sampler C Negative = GND                */
  ADC->ADCSOCCTL[2].bit.AVGCNT = 2;/* Will be ignore                  */
  ADC->ADCSOCCTL[2].bit.SAMPCNT = 3;/* Will be ignore      */
  ADC->ADCSOCCTL[2].bit.CONVCNT = 4;/* Will be ignore       */
}
```

Configure that SOC0 for 3 sampler simultaneous mode, so SOC0/SOC1/SOC2 channel control will be add to sampler A/B/C separately. The trigger source, averaging times, sampling count and conversion count will all use SOC0's configuration. Then the result will be saved to ADCRESULT[0]/[1]/[2], which are correspond to Sampler A/B/C.

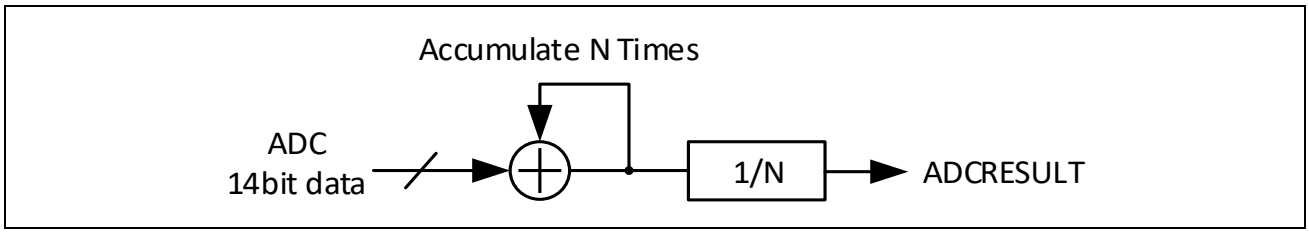
Figure 12-4: ADC2, ADC4 and ADC6 simultaneous sampling



12.3.3 Averaging control

Usually, there are some thermal noise source existing in sampling system such as input signal noise, KT/C noise, comparator noise, reference noise and power supply noise and so on. If use software to do post-averaging, it will make response time longer and waste CPU source. Use averaging control in SOC configuration, ADC can continuously sample many times and get an averaging result in hardware.

Figure 12-5: Averaging control for ADC result



Each SOC use the bit field AVGCNT in ADCSOCCTL[x] to set averaging times, which is a 3-bit field register.

Table 12-7: Averaging time selection

AVGCNT[2:0]	Averaging Times
7	128
6	64
5	32
4	16
3	8
2	4
1	2
0	1

12.3.4 ADC sample and convert window

External drivers vary in their ability to drive an analog signal quickly and effectively. Some circuits require longer times to properly transfer the charge into the sampling capacitor of an ADC. To address this, the ADC supports control over the sample window length for each individual SOC-configuration. Each ADCSOCCTL[x] register has an 8-bit field for SAMPCNT that determines the sampling time and a 7-bit field for CONVCNT that determines the conversion time.

$$Sample\_Time = (\text{Sampcnt} + 1) \times \text{ADC\_Clk\_Cycle}$$
$$Conversion\_Time = (\text{Convcnt} + 1) \times \text{ADC\_Clk\_Cycle}$$

The minimum conversion time is 140 ns. Spintrol suggests that sampling time should be at least 125 ns. The total time to process analog signal is the sample time plus the conversion time. Examples of various sample window size are shown below in Table 12-8.

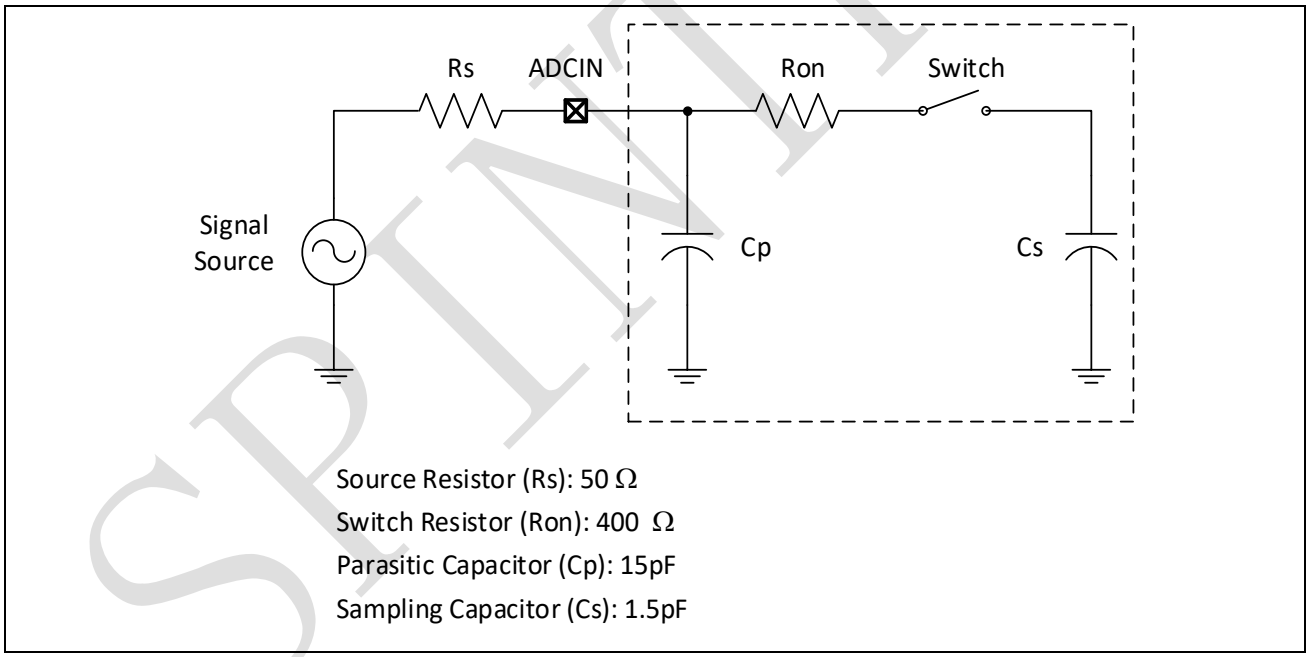
ADC clock cycle and CONVCNT setting must confirm this. For example, if ADC clock frequency is 32 MHz (clock cycle = 31.25ns), CONVCNT should be set above 4 to confirm conversion time over 140 ns.

Table 12-8: Sampling time with different SAMPCNT and CONVCNT (Example)

ADC Frequency	SAMPCNT	Sample Time(ns)	CONVCNT	Conversion Time(ns)	Total time to process analog signal(ns)	Sampling Rate
32 MHz	3	125	4	156.25	281.25	3.56MHz
32 MHz	27	875	4	156.25	1031.15	970kHz
40 MHz	4	125	5	150	275	3.64MHz
40 MHz	34	875	5	150	1025	976kHz
64 MHz	7	125	8	140.625	265.625	3.76MHz
64 MHz	55	875	8	140.625	1015.625	985kHz

The minimum sampling time is decided by signal source impedance, parasitic resistor and parasitic capacitor in signal path. As shown in Figure 12-6, the ADCIN pins can be modeled as an RC circuit. A voltage swing from 0 to 3.3 V on ADCIN needs 70ns settling time for 12 bits precision.

Figure 12-6: ADCx input pin model



12.4 ADC conversion priority

When multiple SOC flags are set at the same time, one of two forms of priority determines the order in which they are converted. The default priority method is round robin. In this scheme, no SOC has an inherent higher priority than another. Priority depends on the round robin pointer (RRPOINTER). The RRPOINTER reflected in the ADCSOCPRICTL register points to the last SOC converted. The highest priority SOC is given to the next value greater than the RRPOINTER value, wrapping around back to SOC0 after SOC15. At reset the value is 0. The RRPOINTER is reset by a device reset, when the ADCCTL0.RST bit is set, or when the ADCSOCPRICTL register is written.

An example of the round robin priority method is given in Figure 12-7.

Step1. After reset, SOC0 is highest priority SOC.

Step2. SOC7 receives trigger; SOC7 configured channel is converted immediately.

Step3. RRPOINTER changes to point to SOC7; SOC8 is now highest priority SOC. SOC2 & SOC12 triggers happen simultaneously; SOC12 is first on round robin wheel; SOC12 configured channel is converted while SOC2 stays pending.

Step4. RRPOINTER changes to point to SOC12; SOC2 configured channel is now converted.

Step5. RRPOINTER changes to point to SOC2; SOC3 is now highest priority SOC.

The PRIORITY field in the ADCSOCPRICTL register can be used to assign high priority, from a single SOC to all of the SOC's. When configured as high priority, an SOC will interrupt the round robin wheel after any current conversion completes and insert itself in as the next conversion. After its conversion completes, the round robin wheel will continue where it was interrupted. If two high priority SOC's are triggered at the same time, the SOC with the lower number will take precedence.

High priority mode is assigned first to SOC0, then in increasing numerical order. The value written in the PRIORITY field defines the first SOC that is not high priority. In other words, if a value of 4 is written into PRIORITY, then SOC0, SOC1, SOC2, and SOC3 are defined as high priority, with SOC0 the highest.

An example using high priority SOC's is given in Figure 12-8.

Example when PRIORITY = 4

Step1. After reset, SOC4 is 1st on round robin wheel; SOC7 receives trigger; SOC7 configured channel is converted immediately.

Step2. RRPOINTER changes to point to SOC7; SOC8 is now 1st on round robin wheel.

Step3. SOC2 & SOC12 triggers happen simultaneously; SOC2 interrupts round robin wheel and SOC2 configured channel is converted while SOC12 stays pending.

Step4. RRPOINTER stays pointing to 7; SOC12 configured channel is now converted.

Step5. RRPOINTER changes to point to SOC12; SOC13 is now 1st on round robin wheel.

Figure 12-7: Round ring priority example

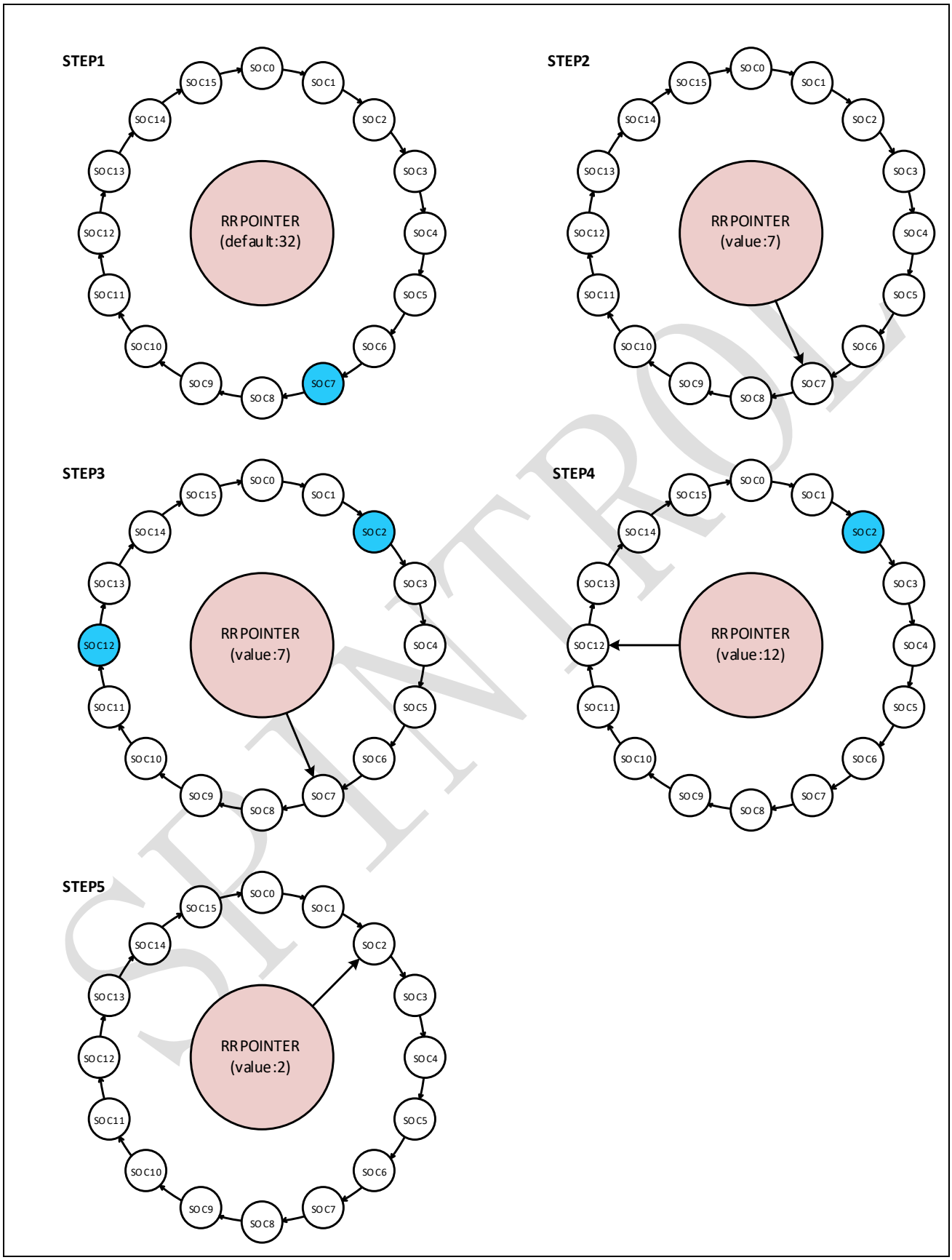
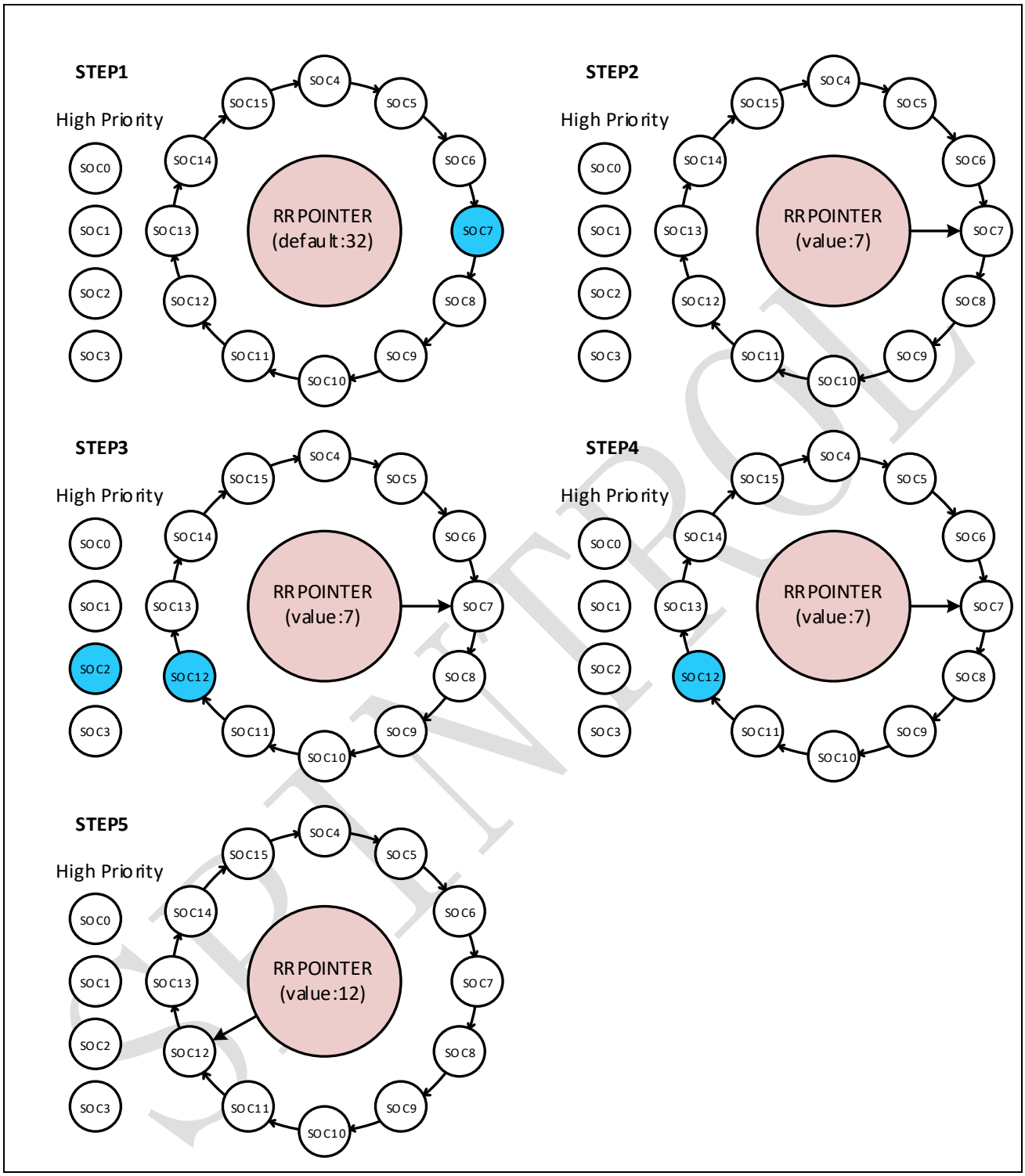


Figure 12-8: High priority example



12.5 Simultaneous sampling mode

In some applications it is important to keep the delay minimal between the sampling of several signals. The ADC contains dual sample and hold circuits to allow two or three different channels to be sampled simultaneously. Simultaneous sampling mode is configured for SHEN with the ADCSOCCTL[x] register.

The behavior is as follows:

- The ADC system will combine SOC0~2, SOC3~SOC5, SCO6~SOC8, SOC9~SOC11, SOC12~SOC14 as 5 simultaneous sampling groups.
- For the simultaneous configuration like simultaneous mode, trigger source, averaging times, sampling count, conversion count will just can be set in SOC0, SOC3, SOC6, SOC9, SOC12.
- For the simultaneous channel selection, it is decided by simultaneous mode and SOC channel together.
- For different simultaneous mode, the channels selection of sampler A/B/C will be set in SOC[x], SOC[x+1], SOC[x+2]. (x=0, 3, 6, 9, 12)
- Sampler A channel will be always convert firstly. Then sampler B will be converted. At last, it's turn for sampler C conversion.
- The results of Sampler A/B/C conversion are placed in the ADCRESULT[x]/ ADCRESULT[x+1]/ ADCRESULT[x+2] register (x=0, 3, 6, 9, 12).
- When SHEN equals 4, Sampler A and B are simultaneous mode. The result will be saved to ADCRESULT[x]/ADCRESULT[x+1] register. Sampler C can be used separately, which can be configured by ADCSOCCTLx (x=0, 3, 6, 9, 12).
- When SHEN equals 5, Sampler B and C are simultaneous mode. Sampler A can't be used separately any more. The result will be saved to ADCRESULT[x+1]/ADCRESULT[x+2] register (x=0, 3, 6, 9, 12).
- When SHEN equal 6, Sampler A and C are simultaneous mode. The result will be saved to ADCRESULT[x]/ADCRESULT[x+2] register. Sampler B can be used separately, which can be configured by ADCSOCCTL[x+1] (x=0, 3, 6, 9, 12).

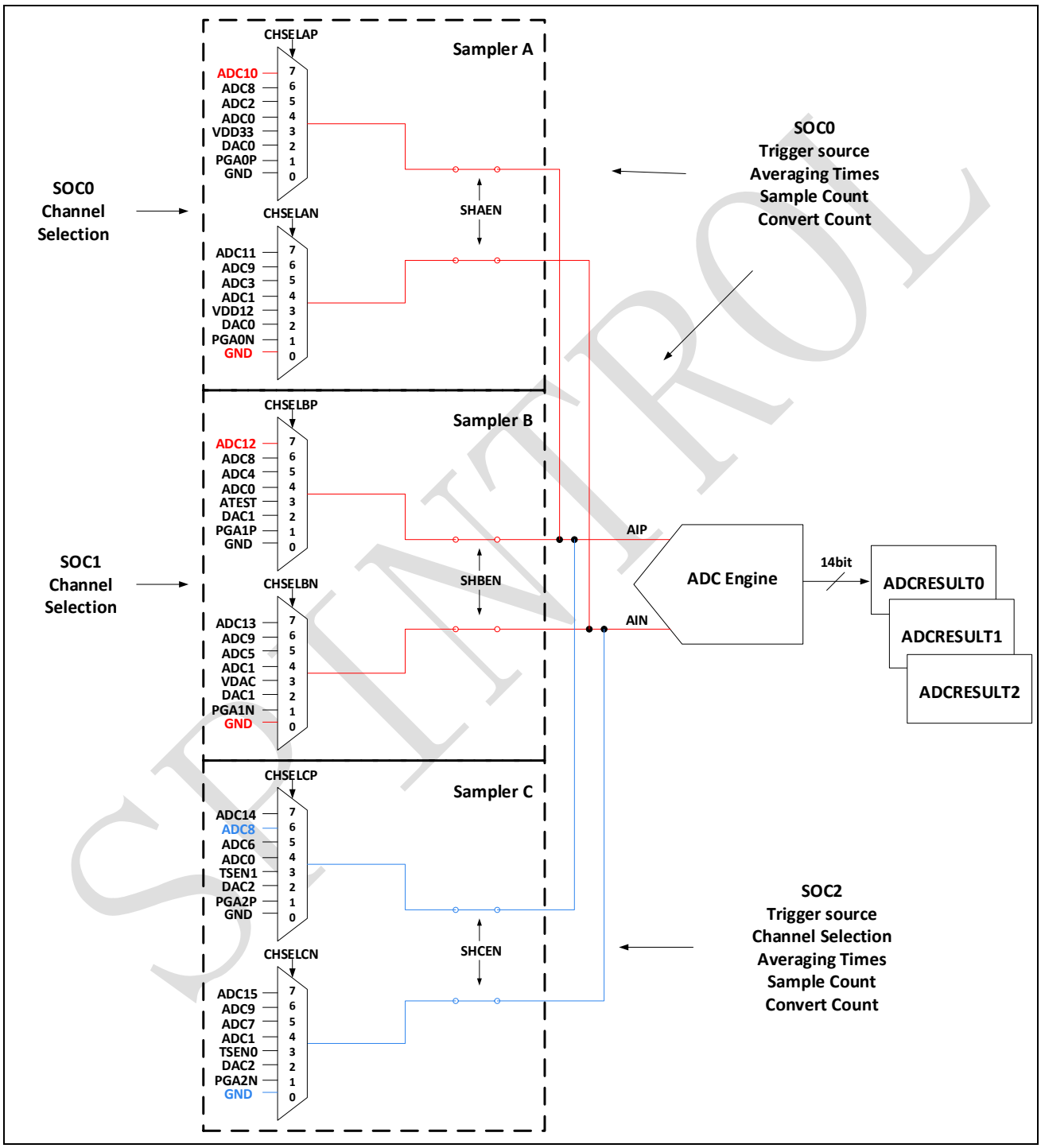
Table 12-9: Simultaneous sampling mode control

ADCSOCCTL[x].SHEN[2:0] (x=0,3,6,9,12)	SOCx	SOCx+1	SOCx+2	Sampling Mode Description
7	Sampler A	Sampler B	Sampler C	Sampler A, B, C Simultaneous
6	Sampler A	/	Sampler C	Sampler A and C Simultaneous
5	/	Sampler B	Sampler C	Sampler B and C Simultaneous
4	Sampler A	Sampler B	/	Sampler A and B Simultaneous

Note: The limitation of simultaneous sampling mode:

1. When ADCSOCCTL[9].bit.SHEN = 2, ADCSOCCTL[0].bit.SHEN can not set to 5, 6 or 7;
2. When ADCSOCCTL[9].bit.SHEN = 7, ADCSOCCTL[0].bit.SHEN must be set to 7.

Figure 12-9: SOC0 simultaneous sampling signal flow and configuration



Example 12.5.1 PWM0SOCA trigger SOC0 to convert ADC10, ADC12 simultaneously

Example 12.5.1

```
void  ADC_Example12_5_1(void)
{
  ADC->ADCSOCCTL[0].bit.SHEN     = 4;/* Enable Sampler A and B simultaneous     */
  ADC->ADCSOCCTL[0].bit.TRIGSEL = 5;/* Select PWM0SOCA as the trigger source  */
  ADC->ADCSOCCTL[0].bit.CHSELP = 7;/* Sampler A Positive = ADC10               */
  ADC->ADCSOCCTL[0].bit.CHSELN = 0;/* Sampler A Negative = GND                 */
  ADC->ADCSOCCTL[0].bit.AVGCNT = 2;/* Averaging 4 times (2^2)                  */
  ADC->ADCSOCCTL[0].bit.SAMPCNT = 3;/* Sampling time = (3+1)*ADC_Clock       */
  ADC->ADCSOCCTL[0].bit.CONVCNT = 4;/* Conversion time = (4+1)*ADC_Clock       */

  ADC->ADCSOCCTL[1].bit.SHEN     = 1;/* Will be ignore                         */
  ADC->ADCSOCCTL[1].bit.TRIGSEL = 6;/* Will be ignore  */
  ADC->ADCSOCCTL[1].bit.CHSELP = 7;/* Sampler B Positive = ADC12               */
  ADC->ADCSOCCTL[1].bit.CHSELN = 0;/* Sampler B Negative = GND                 */
  ADC->ADCSOCCTL[1].bit.AVGCNT = 2;/* Will be ignore                    */
  ADC->ADCSOCCTL[1].bit.SAMPCNT = 3;/* Will be ignore     */
  ADC->ADCSOCCTL[1].bit.CONVCNT = 4;/* Will be ignore      */

  ADC->ADCSOCCTL[2].bit.SHEN     = 3;/* Enable Sampler C single sampling        */
  ADC->ADCSOCCTL[2].bit.TRIGSEL = 9;/* Select PWM1SOCB as the trigger source  */
  ADC->ADCSOCCTL[2].bit.CHSELP = 6;/* Sampler C Positive = ADC8                */
  ADC->ADCSOCCTL[2].bit.CHSELN = 0;/* Sampler C Negative = GND                 */
  ADC->ADCSOCCTL[2].bit.AVGCNT = 3;/* Averaging 8 times (2^3)                  */
  ADC->ADCSOCCTL[2].bit.SAMPCNT = 5;/* Sampling time = (5+1)*ADC_Clock       */
  ADC->ADCSOCCTL[2].bit.CONVCNT = 5;/* Conversion time = (5+1)*ADC_Clock       */
}
```

As Figure 12-9 shown, when the PWM0SOCA sends out a trigger, both ADC10 and ADC12 will be sampled simultaneously (assuming priority). Immediately after, the ADC10 channel will be converted and its value will be stored in the ADCRESULT[0] register. Then the ADC12 channel will be converted and its value will be stored in the ADCRESULT[1] register. The EOC0 pulse will occur when the conversion of ADC12 completes. In addition to that, when PWM1SOCB sends out a trigger, ADC8 will be single sampled and its configuration comes from SOC2. Then its conversion result will be saved to ADCRESULT[2].

The rules of priority for the SOCx's remain the same as In sequential sampling mode.

Section 12.8 shows the timing of simultaneous sampling mode.

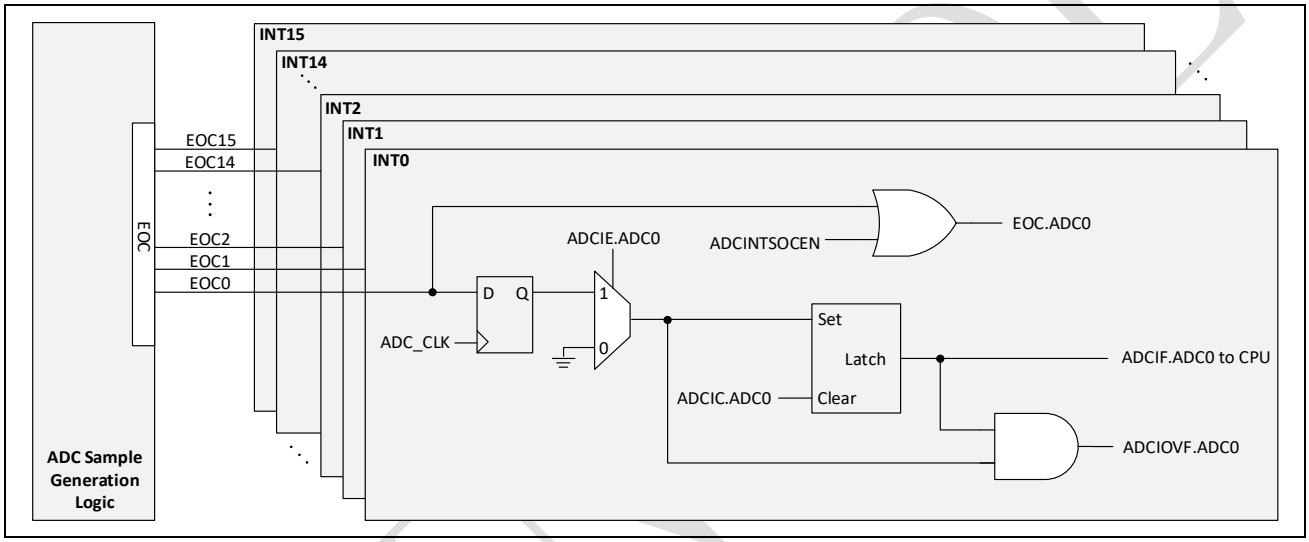
Note: In simultaneous mode, just channels in different samplers can be combined. For example, ADC0 can be simultaneously sampled with ADC4, but not ADC2. Because ADC0 and ADC2 are in the same sampler.

12.6 EOC and interrupt operation

Just as there are 16 independent SOC configuration sets, there are 16 EOCx pulses specifying corresponding end-of-conversion. In sequential sampling mode, the EOCx is associated directly with the SOCx. In simultaneous sampling mode, EOCx group are associated with SOCx group (x=0, 3, 6, 9, 12), as described in Section 12.5. The EOCx pulse will occur at the end of a conversion. See Figure 12-12 for exact timing sequence on the EOCx pulses. Each of EOCx signals can be configured as a trigger source to SOC. Also, this EOC signal can generate the ADC interrupts after 1 clock delay. There are 16 ADC interrupts from 16 EOC pulse. These interrupts can be passed on to the CPU.

Figure 12-10 shows the block diagram of the ADC interrupt structure.

Figure 12-10: Interrupt structure

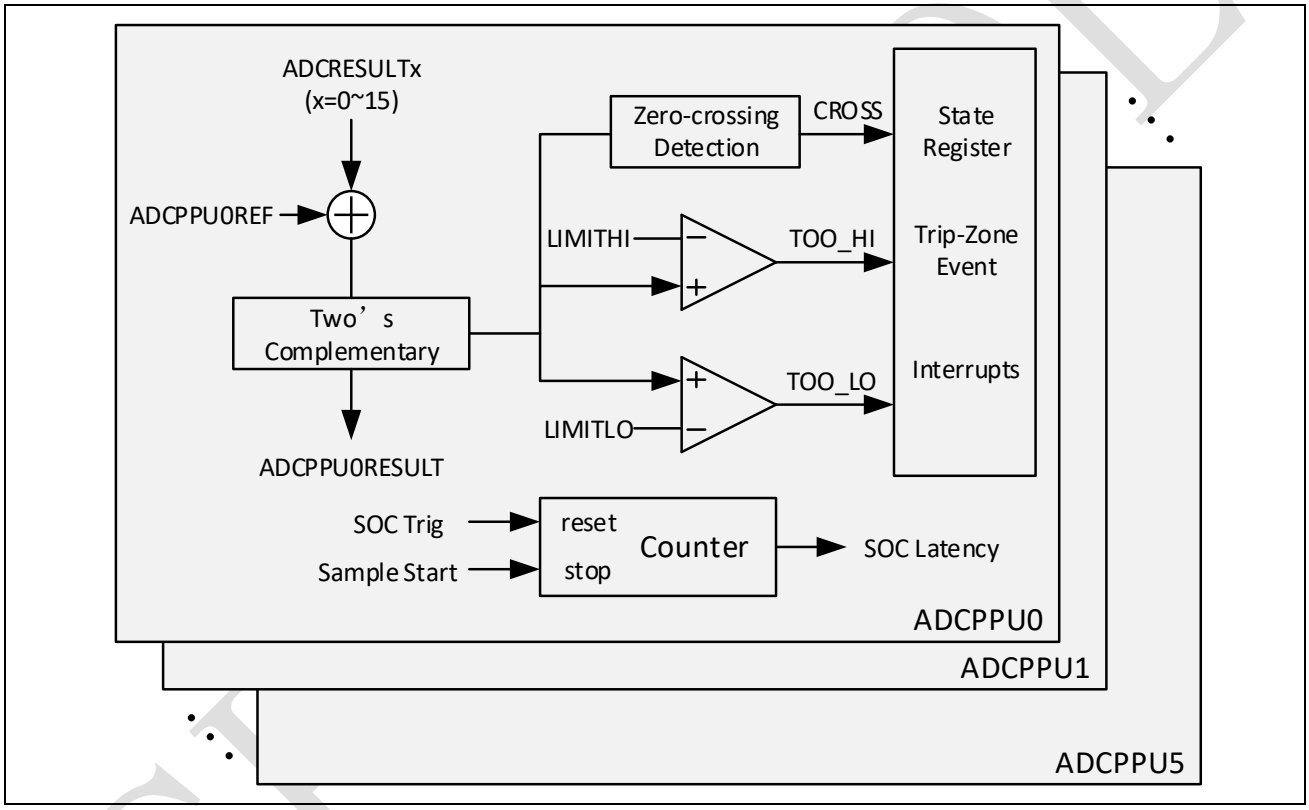


12.7 ADC post-processing units

There are 6 post-processing units (PPU) in ADC digital circuits. These blocks can be associated with any of the 16 RESULT registers using the ADCPPUCTLx.DATASEL bit field. Each PPU can simultaneously remove subtract out a reference value, flag a zero-crossing point, and flag a high or low compare limit. Furthermore, the zero-crossing can generate a trip-zone event to PWM and/or generate an interrupt.

A PPU is also capable of recording the delay between when the SOC associated with the PPU is triggered and when it actually begins to be sampled. Figure 12-11 presents the structure of each PPU. Subsequent sections explain the use of each sub-module.

Figure 12-11: Post-processing unit topology



12.7.1 PPU reference error calculation

In many applications, ADC conversion result need to be subtracted from a reference value. The PPU can perform these function automatically, reducing the sample to output latency and software overhead.

First, configure the register ADCPPUCTLx.DATASEL pointing to the desired SOC result. That means PPU will operation after this SOC result come out. Then, writing a value to the ADCPPUREFx register as the reference value. The post-processing block will automatically subtract the value from the desired SOC result and store it in the ADCPPURESULTx register. This subtraction will produce a sign-extended 32-bit result. It is also possible to selectively invert the calculated value before storing in the ADCPPURESULTx register by setting the ADCPPUCTLx.POL field.

Example 12.7.1 Use PPU0 subtract a reference value from SOC0 result

It can be divided in to the followed lines:

Example 12.7.1

```
void  ADC_Example12_7_1(void)
{
  ADC->ADCPPUCTL[0].bit.EN      = 1;/* Enable ADC PPU0      */
  ADC->ADCPPUCTL[0].bit.DATASEL = 0;/*Select SOC0 result as post-processing data
*/
  ADC->ADCPPUREF[0].all = 100;/* Set the reference value of PPU0 is 100  */

  ADC->ADCIC.bit.INT0 = 1;/* Clear SOC0 interrupt flag */
  ADC->ADCSOCFRC.bit.SOC0 = 1;/*Trig SOC0 to convert and PPU0 to post-processing
*/
  while(ADC->ADCIF.bit.INT0 != 1){}; /* wait for SOC0 result ready*/
}
```

Configure that SOC0's result will subtract a reference value in PPU0, so PPU0 should enable firstly. Then select SOC0 conversion result for PPU to post-process. Set the fixed-point in PPU0. At last trig SOC0, PPU will post-process the result for SOC0 after conversion.

12.7.2 PPU zero-crossing and threshold detection

Many applications perform a threshold detection against the ADC conversion results. The PPU can automatically perform a check against whenever ADCPPURESULTx changes sign or a high and low limit. Based on these comparisons, PPU can generate a trip to the PWM and/or an interrupt automatically, lowering the sample to PWM latency and reducing software overhead. This functionality also enables safety conscious applications to trip the PWM based on an out-of-range ADC conversion without any CPU intervention.

To enable this functionality, configure the register ADCPPUCTLx.DATASEL pointing to the desired SOC result. That means PPU will operation once the SOC is triggered. Then write a value to one or both of the registers ADCPPUTHHx and ADCPPUTHLx (Zero-crossing detection does not require further configuration).

Whenever the ADCPPURESULTx changes signs, the XZRO bit in the ADCPPUTZEx register will be only set when the corresponding EOC signal occurs. Whenever the threshold limits are exceeded, the TZHI and TZLO will be set when the corresponding EOC signal occurs. These event can also generate the corresponding interrupt flag in the ADCPPUIFx register. The ADCPPUIEx register can enable this function. The corresponding bit in ADCPPUICx register can clear the related interrupt flag.

Example 12.7.2 Use PPU0 detect zero-crossing and high/low limit from SOC0 result

It can be divided in to the followed lines:

Example 12.7.2

```
void ADC_Example12_7_2(void)
{
  ADC->ADCPPUCTL[0].bit.EN       = 1;/* Enable ADC PPU0      */
  ADC->ADCPPUIE[0].bit.XZRO    = 1;/* Enable zero-crossing interrupt of ADC PPU0
*/
  ADC->ADCPPUIE[0].bit.TZHI      = 1;/* Enable too high interrupt of ADC PPU0 */
  ADC->ADCPPUIE[0].bit.TZLO      = 1;/* Enable too low interrupt of ADC PPU0 */

  ADC->ADCPPUCTL[0].bit.DATASEL = 0;/*Select SOC0 result as post-processing data
*/
  ADC->ADCPPUTHH[0].all = 3000;/* Set PPU0 high threshold is 3000  */
  ADC->ADCPPUTHL[0].all = 1000;/* Set PPU0 low threshold is 1000  */

  ADC->ADCIC.bit.INT0 = 1;/* Clear SOC0 interrupt flag */
  ADC->ADCSOCFRC.bit.SOC0 = 1;/*Trig SOC0 to convert and PPU0 to post-processing
*/
  while(ADC->ADCIF.bit.INT0 != 1){}; /* wait for SOC0 result ready*/
}
```

Configure that SOC0's result will be detected by PPU0. So enable PPU0 and related interrupt. Set high and low limit threshold for PPU0 detection. At last, trig SOC0. Then PPU0 will make a detection after SOC0 conversion done.

12.7.3 Sample delay capture

When multiple control loops are running asynchronously on the same ADC, there is a chance that an ADC request from two or more loops will collide, causing one of the samples to be delayed. This shows up as a measurement error in the system. By knowing when this delay occurs and the amount of delay that has occurred, software can employ extrapolation techniques to reduce the error.

To this effect, each PPU has the field SOCSEL in the ADCPPUCTLx register. This field contains the number of ADC clock cycles between the moment the associate SOC was triggered and that it began converting. This is achieved by having a global 32-bit free running counter based on ADC clock, which is in the ADCPPUSOCDLYx register. When the trigger for the associated SOC arrives, this counter value will reset and start to count. When the actual sample window for that SOC begins, the counter will stop and the current counter value will store in the ADCPPUSOCDLYx register.

Example 12.7.3 Use PPU0 to counter the sampling delay for SOC0

It can be divided in to the followed lines:

Example 12.7.3

```
void  ADC_Example12_7_3(void)
{
  ADC->ADCPPUCTL[0].bit.EN      = 1;/* Enable ADC PPU0     */
  ADC->ADCPPUCTL[0].bit.SOCSEL = 0;/* Select SOC0 to count sample delay */

  ADC->ADCIC.bit.INT0 = 1;      /* Clear SOC0 interrupt flag */
  ADC->ADCSOCFRC.bit.SOC0 = 1; /* Trig SOC0 and reset PPU0 delay capture counter
*/
  while(ADC->ADCIF.bit.INT0 != 1){}; /* wait for SOC0 result ready*/
}
```

Enable PPU0 for delay capture and configure SOC0 as tracking target. Then, trig SOC0. Then PPU0 will counter the delay from trigger time to SOC0 real sampling.

12.8 ADC timing

Figure 12-12: Timing example for sequential mode interrupt

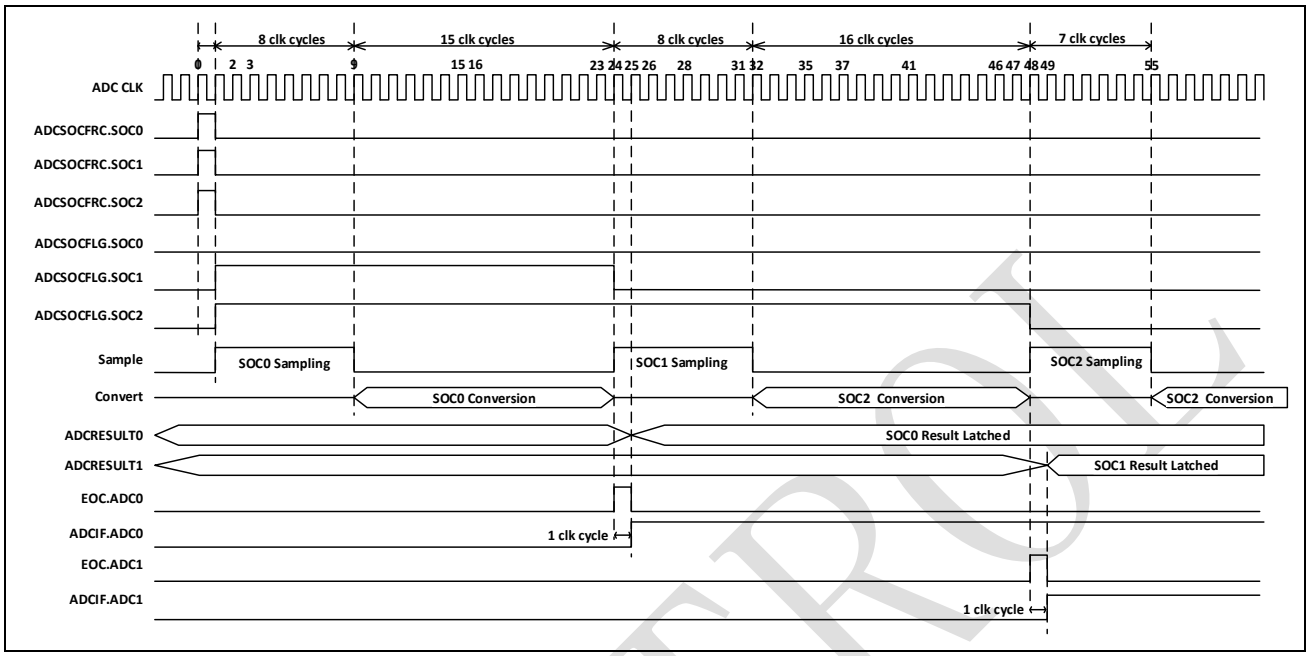
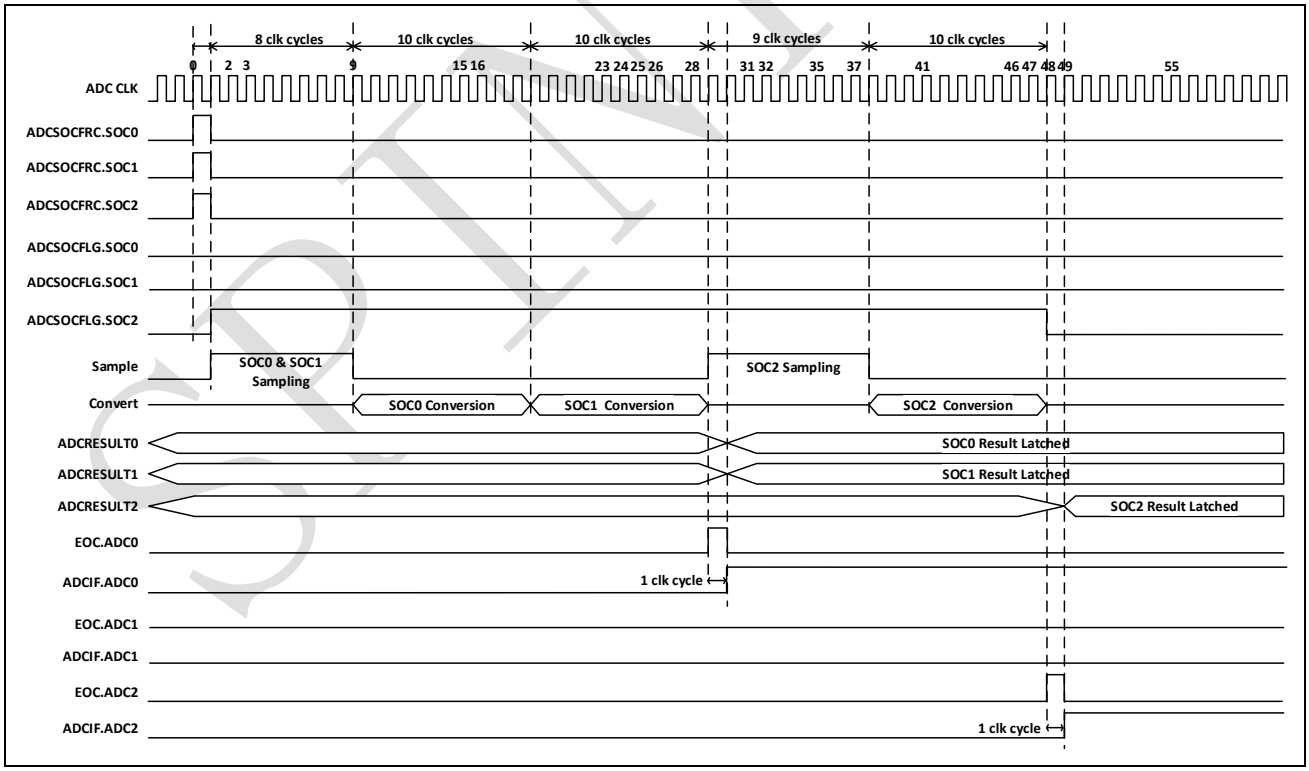


Figure 12-13: Timing example for simultaneous mode interrupt



12.9 Power-up sequence

When powering up the ADC, the following sequence must be used:

Step 1: Power up ADC bandgap

Step 2: After 200us, power up ADC reference buffer

Step 3: After 200us, power up the analog circuits of ADC

Example 12.9.1 Power up sequence

It can be divided in to the followed lines:

Example 12.9.1

```
void  ADC_Example12_9_1(void)
{
  ADC->ADCBGCTL.bit.EN = 1;  /* Enable on Bandgap           */
  Delay_ms(1);
  ADC->ADCREFCTL.bit.EN = 1; /* Enable Reference Buffer       */
  Delay_ms(1);
  ADC->ADCCTL.bit.EN = 1; /* Enable ADC          */
  ADC->ADCCTL.bit.RST = 1; /* ADC logic state-machine reset, will be self-clear
*/
}
```

When powering down the ADC, all three bits in step 1~3 can be cleared simultaneously. The ADC power levels must be controlled via software and they are independent of the state of the device power modes.

When power-up, all ADC registers will be reset. After that, all ADC registers can be reset also when ADCCTL.RST is enabled. This reset will be self-clear.

12.10 ADC calibration

Inherent in any converter is an offset error and a full scale gain error. The ADC is factory calibrated at 25-degrees Celsius to correct both of these while allowing the user to modify the offset and gain correction register for any application environmental effects, such as the ambient temperature. Except under certain emulation conditions, or unless a modification from the factory settings is desired, the user is not required to perform any specific action. The ADC will be properly calibrated in every conversion, which is completed by hardware.

12.10.1 Factory settings and hardware calibration

During the fabrication and test process Spintrol calibrates several ADC settings which are stored in Flash memory. During the SPD1148 startup, boot code will write the factory settings into their respective active registers. When ADC in normal operation, the active registers will be used for hardware calibration.

12.10.2 Channel calibration

As Figure 12-1 shown, the respective active registers are OFFSET0~15 and GAIN0~15. They are correspond to SOC0~15. That allowed customer to modify the offset and gain parameter for special channel.

The channel calibration formula is:

$$ADC\_RESULT[i] = (\text{ADC\_RAW\_CODE[i]} - \text{OFFSET[i]}) \times \text{GAIN[i]}$$

In this formula, the parameter I is related to SOC and its range is from 0 to 15. This calibration function will be done in hardware automatically.

12.11 ADC result

12.11.1 Quantization range

The quantization range of ADC is from -FS to +FS. (FS: full scale)

$$FS = 3.657\text{V}$$

However, for each channel, its voltage can't be over AVDD. If AVDD is 3.3V, that means the input voltage of each channel can't be over 3.3V.

12.11.2 ADC result calculation formula

As Figure 12-14 showed, ADC input is differential. Each end input range is 0~3.657V. As Figure 12-15 showed, its quantization range is -3.657V to 3.657V; digital code range is -8192~8191.

For differential mode,

$$ADC\_RESULT = \frac{(AIP - AIN)}{3.657} \times 8192$$

All fractional values are truncated.

Figure 12-14: ADC core description

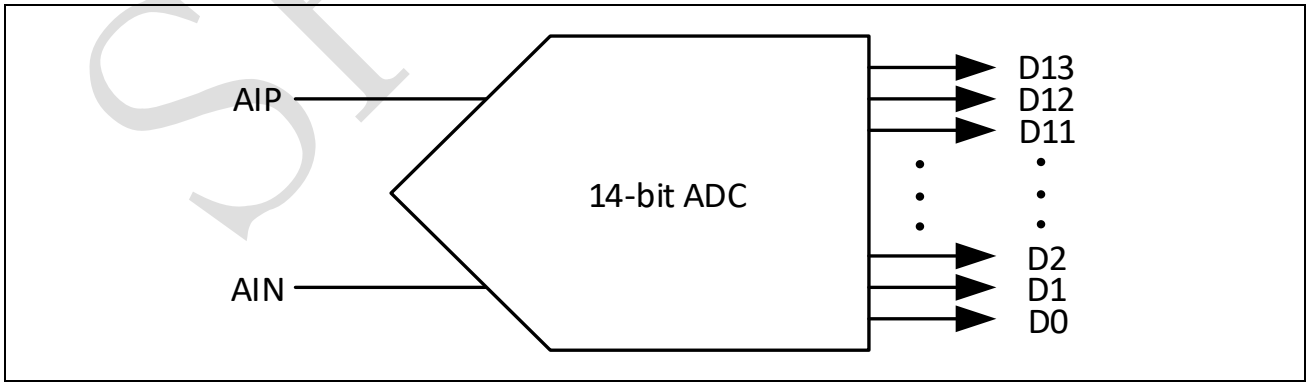
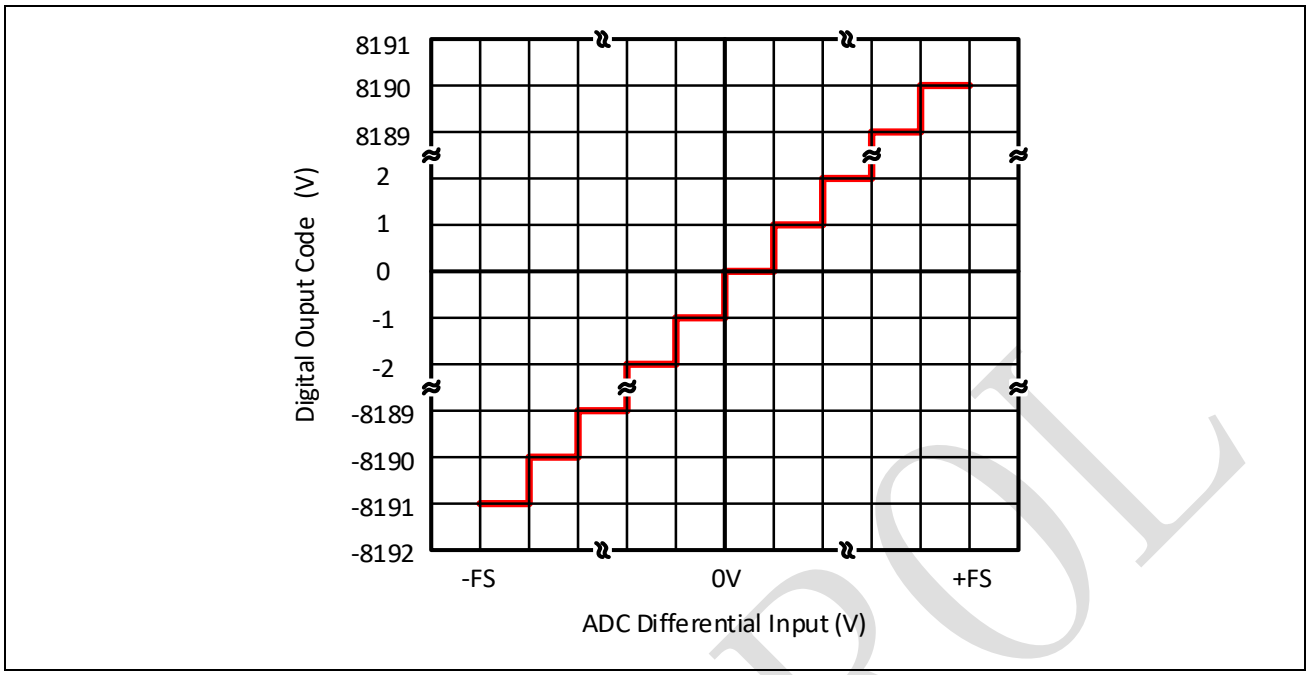


Figure 12-15: ADC transfer curve



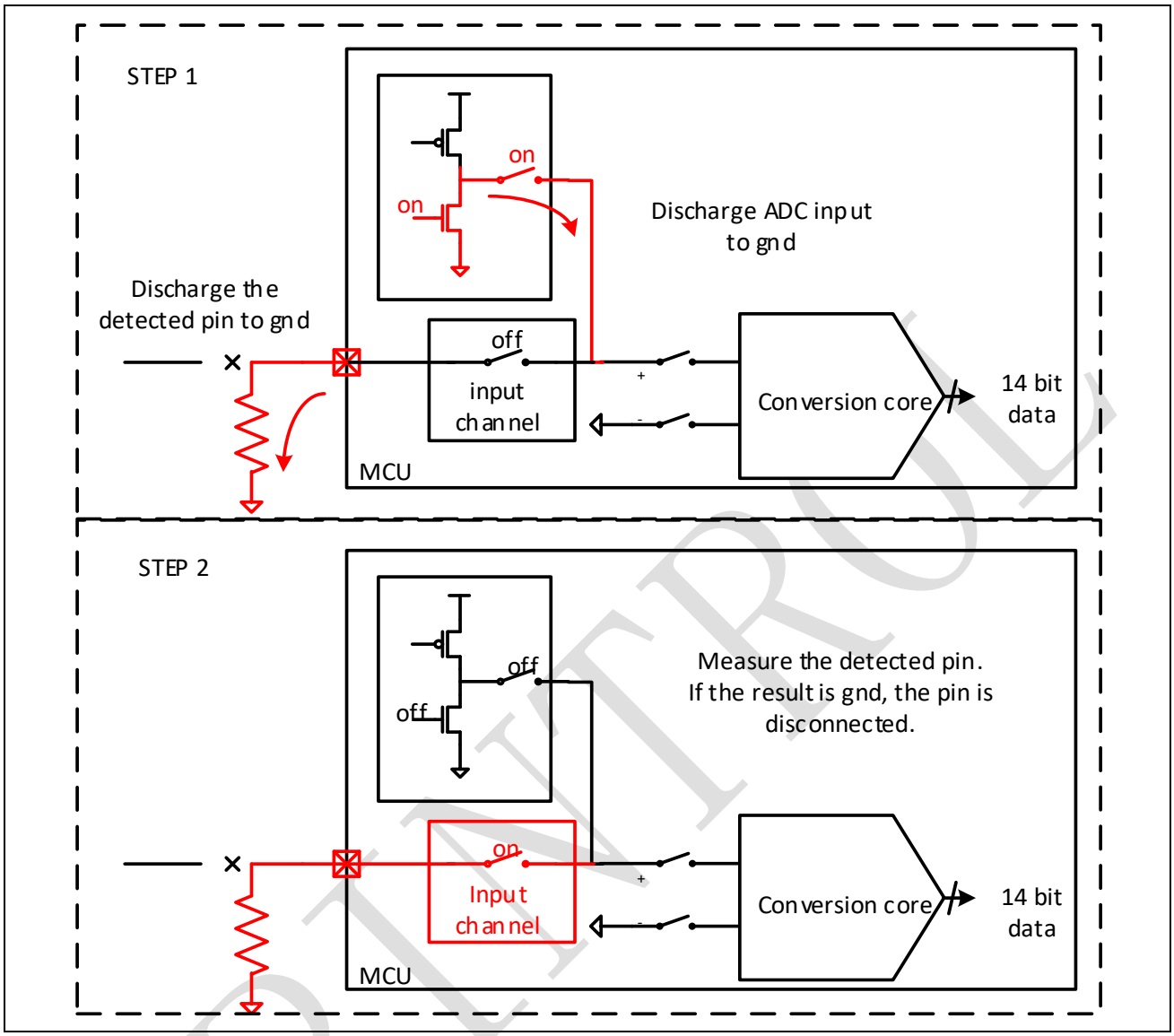
12.12 ADC safety feature

12.12.1 Analog disconnection detection

When the analog input from a sensor to the MCU is disconnected, the position of rotation, temperature and so on cannot be sensed correctly and depending on the application could cause serious damage. The ADC Input disconnection circuit detects an error under such circumstances which provides opportunity to activate fail-safe procedures.

The disconnection detection function can be enabled by configuring the ADCCTL register. In this ADC IP, it supports two optional scenarios to detect the disconnection. They are discharge and precharge cases.

Figure 12-16: Detect the disconnection of ADC input pin (Discharge)

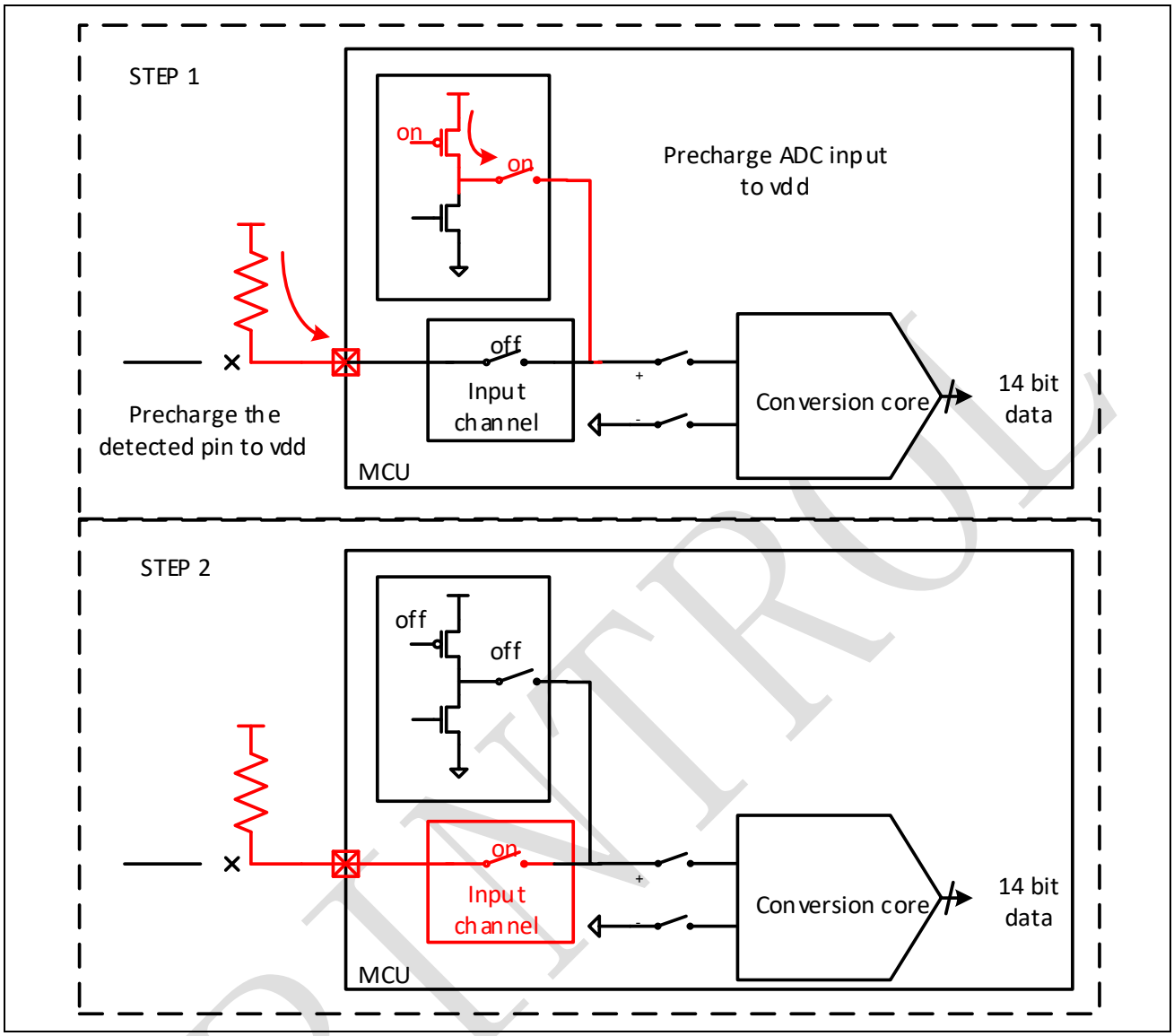


As Figure 12-16 shown the discharge case as below, two procedures are carried on the detection.

- Step1. Keep the input channel mux switch is disable. In the same time, discharge the input pin to gnd, and discharge internal ADC sampling node to gnd separately.
- Step2. Enable input channel mux switch. Use ADC to measure the input of pin. If the data show around 0x0000h, that means the input pin always keep around ground without external driven source.

In this case, we think the pin is disconnected.

Figure 12-17: Detect the disconnection of ADC input pin (Precharge)



As Figure 12-17 shown the precharge case as below, two procedures are carried on the detection.

- Step1. Keep the input channel mux switch is disable. In the same time, precharge the input pin to vdd, and precharge internal ADC sampling node to vdd separately.
- Step2. Enable input channel mux switch. Use ADC to measure the input of pin. If the data show around 0x1CE0h (VDD=3.3V), that means the input pin always keep around vdd without external driven source.

In this case, we think the pin is disconnected.

12.12.2 Input short detection

When the analog input from a sensor to the MCU is anomalously shorted, the ADC Input disconnection circuit can detect a short, which provides opportunity to activate fail-safe procedures.

The input-short detection function can be enabled by configuring the ADCCTL register. The procedures of the detection are shown below:

Step1. ADC measures the input of pin. Save the result as DATA1.

Step2. Force this input pin to GND. ADC measures the input of pin. Save the result as DATA2.

Step3. Force this input pin to AVDD. ADC measures the input of pin. Save the result as DATA3.

Step4. If DATA1= DATA2=DATA3, the pin is shorted.

Note that the short can only be determined if the driving impedance strength is high enough. Assume ADC resolves about 1mV voltage, and given pull-up and pull-down current of 8uA, maximum source impedance can be calculated as: Rsource,max = 1mV/8uA = 125Ω. If the ADC is driven by the source whose impedance is larger than this value, input short cannot be detected.

12.13 Registers

12.13.1 ADC register map

Table 12-10: ADC Module Base Address

Peripheral Module	Base Address
ADC	0x4000 8C00

Table 12-11: ADC Register Map

Register	Offset	Description	Reset Value
ADCIF	0x0	ADC Interrupt Flag Register	0x00000000
ADCIC	0x4	ADC Interrupt Flag Clear Register	0x00000000
ADCIOVF	0x8	ADC Interrupt Overflow Flag Register	0x00000000
ADCIOVFC	0xC	ADC Interrupt Overflow Flag Clear Register	0x00000000
ADCIE*	0x10	ADC Interrupt Enable Register	0x00000000
ADCSOCPRICTL*	0x14	ADC Start of Conversion Priority Control Register	0x00000000
ADCSOCFLG	0x18	ADC SOC Flag Register	0x00000000
ADCSOCFRC	0x1C	ADC SOC Force Register	0x00000000
ADCSOCOVF	0x20	ADC SOC Overflow Register	0x00000000
ADCSOCOVFC	0x24	ADC SOC Overflow Clear Register	0x00000000
ADCINTSOCEN*	0x28	ADC Interrupt Trigger SOC Enable Register	0x00000000
ADCINTSOCSEL0*	0x2C	ADC Interrupt Trigger SOC Select Register 0	0x00000000
ADCINTSOCSEL1*	0x30	ADC Interrupt Trigger SOC Select Register 1	0x00000000
ADCEXTSOCCTL*	0x34	ADC External SOC Input Control Register	0x0000007F
ADCSOCCTL0*	0x38	ADC SOC0 Control Register	0x00000000
ADCSOCCTL1*	0x3C	ADC SOC1 Control Register	0x00000000
ADCSOCCTL2*	0x40	ADC SOC2 Control Register	0x00000000
ADCSOCCTL3*	0x44	ADC SOC3 Control Register	0x00000000
ADCSOCCTL4*	0x48	ADC SOC4 Control Register	0x00000000
ADCSOCCTL5*	0x4C	ADC SOC5 Control Register	0x00000000
ADCSOCCTL6*	0x50	ADC SOC6 Control Register	0x00000000
ADCSOCCTL7*	0x54	ADC SOC7 Control Register	0x00000000
ADCSOCCTL8*	0x58	ADC SOC8 Control Register	0x00000000
ADCSOCCTL9*	0x5C	ADC SOC9 Control Register	0x00000000

Register	Offset	Description	Reset Value
ADCSOCCTL10*	0x60	ADC SOC10 Control Register	0x00000000
ADCSOCCTL11*	0x64	ADC SOC11 Control Register	0x00000000
ADCSOCCTL12*	0x68	ADC SOC12 Control Register	0x00000000
ADCSOCCTL13*	0x6C	ADC SOC13 Control Register	0x00000000
ADCSOCCTL14*	0x70	ADC SOC14 Control Register	0x00000000
ADCSOCCTL15*	0x74	ADC SOC15 Control Register	0x00000000
ADCOFFSET0*	0x78	ADC Offset Trim Register 0	0x00000000
ADCOFFSET1*	0x7C	ADC Offset Trim Register 1	0x00000000
ADCOFFSET2*	0x80	ADC Offset Trim Register 2	0x00000000
ADCOFFSET3*	0x84	ADC Offset Trim Register 3	0x00000000
ADCOFFSET4*	0x88	ADC Offset Trim Register 4	0x00000000
ADCOFFSET5*	0x8C	ADC Offset Trim Register 5	0x00000000
ADCOFFSET6*	0x90	ADC Offset Trim Register 6	0x00000000
ADCOFFSET7*	0x94	ADC Offset Trim Register 7	0x00000000
ADCOFFSET8*	0x98	ADC Offset Trim Register 8	0x00000000
ADCOFFSET9*	0x9C	ADC Offset Trim Register 9	0x00000000
ADCOFFSET10*	0xA0	ADC Offset Trim Register 10	0x00000000
ADCOFFSET11*	0xA4	ADC Offset Trim Register 11	0x00000000
ADCOFFSET12*	0xA8	ADC Offset Trim Register 12	0x00000000
ADCOFFSET13*	0xAC	ADC Offset Trim Register 13	0x00000000
ADCOFFSET14*	0xB0	ADC Offset Trim Register 14	0x00000000
ADCOFFSET15*	0xB4	ADC Offset Trim Register 15	0x00000000
ADCGAIN0*	0xB8	ADC Gain Trim Register 0	0x00008000
ADCGAIN1*	0xBC	ADC Gain Trim Register 1	0x00008000
ADCGAIN2*	0xC0	ADC Gain Trim Register 2	0x00008000
ADCGAIN3*	0xC4	ADC Gain Trim Register 3	0x00008000
ADCGAIN4*	0xC8	ADC Gain Trim Register 4	0x00008000
ADCGAIN5*	0xCC	ADC Gain Trim Register 5	0x00008000
ADCGAIN6*	0xD0	ADC Gain Trim Register 6	0x00008000
ADCGAIN7*	0xD4	ADC Gain Trim Register 7	0x00008000
ADCGAIN8*	0xD8	ADC Gain Trim Register 8	0x00008000
ADCGAIN9*	0xDC	ADC Gain Trim Register 9	0x00008000

Register	Offset	Description	Reset Value
ADCGAIN10*	0xE0	ADC Gain Trim Register 10	0x00008000
ADCGAIN11*	0xE4	ADC Gain Trim Register 11	0x00008000
ADCGAIN12*	0xE8	ADC Gain Trim Register 12	0x00008000
ADCGAIN13*	0xEC	ADC Gain Trim Register 13	0x00008000
ADCGAIN14*	0xF0	ADC Gain Trim Register 14	0x00008000
ADCGAIN15*	0xF4	ADC Gain Trim Register 15	0x00008000
ADCOFFSETA*	0xF8	ADC SHA Offset Trim Register	0x00000000
ADCOFFSETB*	0xFC	ADC SHB Offset Trim Register	0x00000000
ADCOFFSETC*	0x100	ADC SHC Offset Trim Register	0x00000000
ADCGAINA*	0x104	ADC SHA Gain Trim Register	0x00008000
ADCGAINB*	0x108	ADC SHB Gain Trim Register	0x00008000
ADCGAINC*	0x10C	ADC SHC Gain Trim Register	0x00008000
ADCSTS	0x110	ADC Status Register	0x00000000
ADCSTSCLR	0x114	ADC Status Clear Register	0x00000000
ADCCTL*	0x118	ADC Control Register	0x00000008
ADCBGCTL*	0x11C	ADC Bandgap Control Register	0x00000000
ADCREFCTL*	0x120	ADC Reference Control Register	0x00003D3C
ADCRAWCODEA	0x124	ADC SHA Raw Code Register	0x00000000
ADCRAWCODEB	0x128	ADC SHB Raw Code Register	0x00000000
ADCRAWCODEC	0x12C	ADC SHC Raw Code Register	0x00000000
ADCRESULT0	0x130	ADC Result Register 0	0x00000000
ADCRESULT1	0x134	ADC Result Register 1	0x00000000
ADCRESULT2	0x138	ADC Result Register 2	0x00000000
ADCRESULT3	0x13C	ADC Result Register 3	0x00000000
ADCRESULT4	0x140	ADC Result Register 4	0x00000000
ADCRESULT5	0x144	ADC Result Register 5	0x00000000
ADCRESULT6	0x148	ADC Result Register 6	0x00000000
ADCRESULT7	0x14C	ADC Result Register 7	0x00000000
ADCRESULT8	0x150	ADC Result Register 8	0x00000000
ADCRESULT9	0x154	ADC Result Register 9	0x00000000
ADCRESULT10	0x158	ADC Result Register 10	0x00000000
ADCRESULT11	0x15C	ADC Result Register 11	0x00000000

Register	Offset	Description	Reset Value
ADCRESULT12	0x160	ADC Result Register 12	0x00000000
ADCRESULT13	0x164	ADC Result Register 13	0x00000000
ADCRESULT14	0x168	ADC Result Register 14	0x00000000
ADCRESULT15	0x16C	ADC Result Register 15	0x00000000
ADCPPURESULT0	0x170	ADCPPU0 Comparison Result Register	0x00000000
ADCPPURESULT1	0x174	ADCPPU1 Comparison Result Register	0x00000000
ADCPPURESULT2	0x178	ADCPPU2 Comparison Result Register	0x00000000
ADCPPURESULT3	0x17C	ADCPPU3 Comparison Result Register	0x00000000
ADCPPURESULT4	0x180	ADCPPU4 Comparison Result Register	0x00000000
ADCPPURESULT5	0x184	ADCPPU5 Comparison Result Register	0x00000000
ADCPPUSOCDLY0	0x190	ADCPPU0 SOC Delay Register	0x00000000
ADCPPUSOCDLY1	0x194	ADCPPU1 SOC Delay Register	0x00000000
ADCPPUSOCDLY2	0x198	ADCPPU2 SOC Delay Register	0x00000000
ADCPPUSOCDLY3	0x19C	ADCPPU3 SOC Delay Register	0x00000000
ADCPPUSOCDLY4	0x1A0	ADCPPU4 SOC Delay Register	0x00000000
ADCPPUSOCDLY5	0x1A4	ADCPPU5 SOC Delay Register	0x00000000
ADCPPUIF0	0x1B0	ADCPPU0 Interrupt Flag Register	0x00000000
ADCPPUIF1	0x1B4	ADCPPU1 Interrupt Flag Register	0x00000000
ADCPPUIF2	0x1B8	ADCPPU2 Interrupt Flag Register	0x00000000
ADCPPUIF3	0x1BC	ADCPPU3 Interrupt Flag Register	0x00000000
ADCPPUIF4	0x1C0	ADCPPU4 Interrupt Flag Register	0x00000000
ADCPPUIF5	0x1C4	ADCPPU5 Interrupt Flag Register	0x00000000
ADCPPUIC0	0x1D0	ADCPPU0 Interrupt Clear Register	0x00000000
ADCPPUIC1	0x1D4	ADCPPU1 Interrupt Clear Register	0x00000000
ADCPPUIC2	0x1D8	ADCPPU2 Interrupt Clear Register	0x00000000
ADCPPUIC3	0x1DC	ADCPPU3 Interrupt Clear Register	0x00000000
ADCPPUIC4	0x1E0	ADCPPU4 Interrupt Clear Register	0x00000000
ADCPPUIC5	0x1E4	ADCPPU5 Interrupt Clear Register	0x00000000
ADCPPUIE0*	0x1F0	ADCPPU0 Interrupt Enable Register	0x00000000
ADCPPUIE1*	0x1F4	ADCPPU1 Interrupt Enable Register	0x00000000
ADCPPUIE2*	0x1F8	ADCPPU2 Interrupt Enable Register	0x00000000
ADCPPUIE3*	0x1FC	ADCPPU3 Interrupt Enable Register	0x00000000

Register	Offset	Description	Reset Value
ADCPPUIE4*	0x200	ADCPPU4 Interrupt Enable Register	0x00000000
ADCPPUIE5*	0x204	ADCPPU5 Interrupt Enable Register	0x00000000
ADCPPUTZE0*	0x210	ADCPPU0 Trip-Zone Event Enable Register	0x00000000
ADCPPUTZE1*	0x214	ADCPPU1 Trip-Zone Event Enable Register	0x00000000
ADCPPUTZE2*	0x218	ADCPPU2 Trip-Zone Event Enable Register	0x00000000
ADCPPUTZE3*	0x21C	ADCPPU3 Trip-Zone Event Enable Register	0x00000000
ADCPPUTZE4*	0x220	ADCPPU4 Trip-Zone Event Enable Register	0x00000000
ADCPPUTZE5*	0x224	ADCPPU5 Trip-Zone Event Enable Register	0x00000000
ADCPPUCTL0*	0x230	ADCPPU0 Control Register	0x00000400
ADCPPUCTL1*	0x234	ADCPPU1 Control Register	0x00000400
ADCPPUCTL2*	0x238	ADCPPU2 Control Register	0x00000400
ADCPPUCTL3*	0x23C	ADCPPU3 Control Register	0x00000400
ADCPPUCTL4*	0x240	ADCPPU4 Control Register	0x00000400
ADCPPUCTL5*	0x244	ADCPPU5 Control Register	0x00000400
ADCPPUREF0*	0x250	ADCPPU0 Reference Register	0x00000000
ADCPPUREF1*	0x254	ADCPPU1 Reference Register	0x00000000
ADCPPUREF2*	0x258	ADCPPU2 Reference Register	0x00000000
ADCPPUREF3*	0x25C	ADCPPU3 Reference Register	0x00000000
ADCPPUREF4*	0x260	ADCPPU4 Reference Register	0x00000000
ADCPPUREF5*	0x264	ADCPPU5 Reference Register	0x00000000
ADCPPUTHH0*	0x270	ADCPPU0 Trip-Zone High-Side Threshold Register	0x00000000
ADCPPUTHH1*	0x274	ADCPPU1 Trip-Zone High-Side Threshold Register	0x00000000
ADCPPUTHH2*	0x278	ADCPPU2 Trip-Zone High-Side Threshold Register	0x00000000
ADCPPUTHH3*	0x27C	ADCPPU3 Trip-Zone High-Side Threshold Register	0x00000000
ADCPPUTHH4*	0x280	ADCPPU4 Trip-Zone High-Side Threshold Register	0x00000000
ADCPPUTHH5*	0x284	ADCPPU5 Trip-Zone High-Side Threshold Register	0x00000000
ADCPPUTHL0*	0x290	ADCPPU0 Trip-Zone Low-Side Threshold Register	0x00000000
ADCPPUTHL1*	0x294	ADCPPU1 Trip-Zone Low-Side Threshold Register	0x00000000
ADCPPUTHL2*	0x298	ADCPPU2 Trip-Zone Low-Side Threshold Register	0x00000000
ADCPPUTHL3*	0x29C	ADCPPU3 Trip-Zone Low-Side Threshold Register	0x00000000
ADCPPUTHL4*	0x2A0	ADCPPU4 Trip-Zone Low-Side Threshold Register	0x00000000
ADCPPUTHL5*	0x2A4	ADCPPU5 Trip-Zone Low-Side Threshold Register	0x00000000

Register	Offset	Description	Reset Value
TSENSCTL*	0x2B0	Temperature Sensor Control Register	0x00000000
ADCREGKEY	0x2B4	ADC Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the ADCREGKEY=0x1ACCE551.

12.13.2 ADC registers

Table 12-12: ADC Interrupt Flag Register (ADCIF) Layout

ADCIF (ADC Interrupt Flag Register) Offset: 0x0 Default: 0x00000000 Access: ADC -> ADCIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 INT15	14 INT14	13 INT13	12 INT12	11 INT11	10 INT10	9 INT9	8 INT8
7 INT7	6 INT6	5 INT5	4 INT4	3 INT3	2 INT2	1 INT1	0 INT0

Table 12-13: ADC Interrupt Flag Register (ADCIF) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	INT15	RO	0x0	ADC interrupt 15 flag 0: ADC interrupt 15 does not happen 1: ADC interrupt 15 happened
14	INT14	RO	0x0	ADC interrupt 14 flag 0: ADC interrupt 14 does not happen 1: ADC interrupt 14 happened
13	INT13	RO	0x0	ADC interrupt 13 flag 0: ADC interrupt 13 does not happen 1: ADC interrupt 13 happened
12	INT12	RO	0x0	ADC interrupt 12 flag 0: ADC interrupt 12 does not happen 1: ADC interrupt 12 happened
11	INT11	RO	0x0	ADC interrupt 11 flag 0: ADC interrupt 11 does not happen 1: ADC interrupt 11 happened
10	INT10	RO	0x0	ADC interrupt 10 flag 0: ADC interrupt 10 does not happen 1: ADC interrupt 10 happened
9	INT9	RO	0x0	ADC interrupt 9 flag 0: ADC interrupt 9 does not happen 1: ADC interrupt 9 happened
8	INT8	RO	0x0	ADC interrupt 8 flag 0: ADC interrupt 8 does not happen 1: ADC interrupt 8 happened

Bits	Field Name	Type	Reset	Description
7	INT7	RO	0x0	ADC interrupt 7 flag 0: ADC interrupt 7 does not happen 1: ADC interrupt 7 happened
6	INT6	RO	0x0	ADC interrupt 6 flag 0: ADC interrupt 6 does not happen 1: ADC interrupt 6 happened
5	INT5	RO	0x0	ADC interrupt 5 flag 0: ADC interrupt 5 does not happen 1: ADC interrupt 5 happened
4	INT4	RO	0x0	ADC interrupt 4 flag 0: ADC interrupt 4 does not happen 1: ADC interrupt 4 happened
3	INT3	RO	0x0	ADC interrupt 3 flag 0: ADC interrupt 3 does not happen 1: ADC interrupt 3 happened
2	INT2	RO	0x0	ADC interrupt 2 flag 0: ADC interrupt 2 does not happen 1: ADC interrupt 2 happened
1	INT1	RO	0x0	ADC interrupt 1 flag 0: ADC interrupt 1 does not happen 1: ADC interrupt 1 happened
0	INT0	RO	0x0	ADC interrupt 0 flag 0: ADC interrupt 0 does not happen 1: ADC interrupt 0 happened

Table 12-14: ADC Interrupt Flag Clear Register (ADCIC) Layout

ADCIC (ADC Interrupt Flag Clear Register) Offset: 0x4 Default: 0x00000000 Access: ADC -> ADCIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 INT15	14 INT14	13 INT13	12 INT12	11 INT11	10 INT10	9 INT9	8 INT8
7 INT7	6 INT6	5 INT5	4 INT4	3 INT3	2 INT2	1 INT1	0 INT0

Table 12-15: ADC Interrupt Flag Clear Register (ADCIC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
15	INT15	W1S	0x0	ADC Interrupt 15 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT15]. This bit is self-cleared.
14	INT14	W1S	0x0	ADC Interrupt 14 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT14]. This bit is self-cleared.
13	INT13	W1S	0x0	ADC Interrupt 13 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT13]. This bit is self-cleared.
12	INT12	W1S	0x0	ADC Interrupt 12 flag clear 0: Write a 0 has no effect. Always readback 1. 1: Write a 1 clears ADCIF[INT12]. This bit is self-cleared.
11	INT11	W1S	0x0	ADC Interrupt 11 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT11]. This bit is self-cleared.
10	INT10	W1S	0x0	ADC Interrupt 10 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT10]. This bit is self-cleared.
9	INT9	W1S	0x0	ADC Interrupt 9 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT9]. This bit is self-cleared.
8	INT8	W1S	0x0	ADC Interrupt 8 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT8]. This bit is self-cleared.
7	INT7	W1S	0x0	ADC Interrupt 7 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT7]. This bit is self-cleared.
6	INT6	W1S	0x0	ADC Interrupt 6 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT6]. This bit is self-cleared.

Bits	Field Name	Type	Reset	Description
5	INT5	W1S	0x0	ADC Interrupt 5 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT5]. This bit is self-cleared.
4	INT4	W1S	0x0	ADC Interrupt 4 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT4]. This bit is self-cleared.
3	INT3	W1S	0x0	ADC Interrupt 3 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT3]. This bit is self-cleared.
2	INT2	W1S	0x0	ADC Interrupt 2 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT2]. This bit is self-cleared.
1	INT1	W1S	0x0	ADC Interrupt 1 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT1]. This bit is self-cleared.
0	INT0	W1S	0x0	ADC Interrupt 0 flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clears ADCIF[INT0]. This bit is self-cleared.

Table 12-16: ADC Interrupt Overflow Flag Register (ADCIOVF) Layout

ADCIOVF (ADC Interrupt Overflow Flag Register) Offset: 0x8 Default: 0x00000000 Access: ADC -> ADCIOVF.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 INT15	14 INT14	13 INT13	12 INT12	11 INT11	10 INT10	9 INT9	8 INT8
7 INT7	6 INT6	5 INT5	4 INT4	3 INT3	2 INT2	1 INT1	0 INT0

Table 12-17: ADC Interrupt Overflow Flag Register (ADCIOVF) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
15	INT15	RO	0x0	ADC interrupt 15 overflow flag 0: No overflow 1: ADC interrupt 15 overflow
14	INT14	RO	0x0	ADC interrupt 14 overflow flag 0: No overflow 1: ADC interrupt 14 overflow
13	INT13	RO	0x0	ADC interrupt 13 overflow flag 0: No overflow 1: ADC interrupt 13 overflow
12	INT12	RO	0x0	ADC interrupt 12 overflow flag 0: No overflow 1: ADC interrupt 12 overflow
11	INT11	RO	0x0	ADC interrupt 11 overflow flag 0: No overflow 1: ADC interrupt 11 overflow
10	INT10	RO	0x0	ADC interrupt 10 overflow flag 0: No overflow 1: ADC interrupt 10 overflow
9	INT9	RO	0x0	ADC interrupt 9 overflow flag 0: No overflow 1: ADC interrupt 9 overflow
8	INT8	RO	0x0	ADC interrupt 8 overflow flag 0: No overflow 1: ADC interrupt 8 overflow
7	INT7	RO	0x0	ADC interrupt 7 overflow flag 0: No overflow 1: ADC interrupt 7 overflow
6	INT6	RO	0x0	ADC interrupt 6 overflow flag 0: No overflow 1: ADC interrupt 6 overflow
5	INT5	RO	0x0	ADC interrupt 5 overflow flag 0: No overflow 1: ADC interrupt 5 overflow
4	INT4	RO	0x0	ADC interrupt 4 overflow flag 0: No overflow 1: ADC interrupt 4 overflow
3	INT3	RO	0x0	ADC interrupt 3 overflow flag 0: No overflow 1: ADC interrupt 3 overflow

Bits	Field Name	Type	Reset	Description
2	INT2	RO	0x0	ADC interrupt 2 overflow flag 0: No overflow 1: ADC interrupt 2 overflow
1	INT1	RO	0x0	ADC interrupt 1 overflow flag 0: No overflow 1: ADC interrupt 1 overflow
0	INT0	RO	0x0	ADC interrupt 0 overflow flag 0: No overflow 1: ADC interrupt 0 overflow

Table 12-18: ADC Interrupt Overflow Flag Clear Register (ADCIOVFC) Layout

ADCIOVFC (ADC Interrupt Overflow Flag Clear Register) Offset: 0xC Default: 0x00000000 Access: ADC -> ADCIOVFC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 INT15	14 INT14	13 INT13	12 INT12	11 INT11	10 INT10	9 INT9	8 INT8
7 INT7	6 INT6	5 INT5	4 INT4	3 INT3	2 INT2	1 INT1	0 INT0

Table 12-19: ADC Interrupt Overflow Flag Clear Register (ADCIOVFC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	INT15	W1S	0x0	ADC interrupt 15 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT15]. This bit is self-cleared.
14	INT14	W1S	0x0	ADC interrupt 14 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT14]. This bit is self-cleared.
13	INT13	W1S	0x0	ADC interrupt 13 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT13]. This bit is self-cleared.
12	INT12	W1S	0x0	ADC interrupt 12 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT12]. This bit is self-cleared.

Bits	Field Name	Type	Reset	Description
11	INT11	W1S	0x0	ADC interrupt 11 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT11]. This bit is self-cleared.
10	INT10	W1S	0x0	ADC interrupt 10 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT10]. This bit is self-cleared.
9	INT9	W1S	0x0	ADC interrupt 9 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT9]. This bit is self-cleared.
8	INT8	W1S	0x0	ADC interrupt 8 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT8]. This bit is self-cleared.
7	INT7	W1S	0x0	ADC interrupt 7 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT7]. This bit is self-cleared.
6	INT6	W1S	0x0	ADC interrupt 6 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT6]. This bit is self-cleared.
5	INT5	W1S	0x0	ADC interrupt 5 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT5]. This bit is self-cleared.
4	INT4	W1S	0x0	ADC interrupt 4 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT4]. This bit is self-cleared.
3	INT3	W1S	0x0	ADC interrupt 3 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT3]. This bit is self-cleared.
2	INT2	W1S	0x0	ADC interrupt 2 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT2]. This bit is self-cleared.

Bits	Field Name	Type	Reset	Description
1	INT1	W1S	0x0	ADC interrupt 1 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT1]. This bit is self-cleared.
0	INT0	W1S	0x0	ADC interrupt 0 overflow flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears the ADCIOVF[INT0]. This bit is self-cleared.

Table 12-20: ADC Interrupt Enable Register (ADCIE) Layout

ADCIE (ADC Interrupt Enable Register) Offset: 0x10 Default: 0x00000000 Access: ADC -> ADCIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 INT15	14 INT14	13 INT13	12 INT12	11 INT11	10 INT10	9 INT9	8 INT8
7 INT7	6 INT6	5 INT5	4 INT4	3 INT3	2 INT2	1 INT1	0 INT0

Table 12-21: ADC Interrupt Enable Register (ADCIE) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	INT15	RW	0x0	ADC interrupt 15 enable 0: Disable ADC interrupt 15 1: Enable ADC interrupt 15
14	INT14	RW	0x0	ADC interrupt 14 enable 0: Disable ADC interrupt 14 1: Enable ADC interrupt 14
13	INT13	RW	0x0	ADC interrupt 13 enable 0: Disable ADC interrupt 13 1: Enable ADC interrupt 13
12	INT12	RW	0x0	ADC interrupt 12 enable 0: Disable ADC interrupt 12 1: Enable ADC interrupt 12
11	INT11	RW	0x0	ADC interrupt 11 enable 0: Disable ADC interrupt 11 1: Enable ADC interrupt 11

Bits	Field Name	Type	Reset	Description
10	INT10	RW	0x0	ADC interrupt 10 enable 0: Disable ADC interrupt 10 1: Enable ADC interrupt 10
9	INT9	RW	0x0	ADC interrupt 9 enable 0: Disable ADC interrupt 9 1: Enable ADC interrupt 9
8	INT8	RW	0x0	ADC interrupt 8 enable 0: Disable ADC interrupt 8 1: Enable ADC interrupt 8
7	INT7	RW	0x0	ADC interrupt 7 enable 0: Disable ADC interrupt 7 1: Enable ADC interrupt 7
6	INT6	RW	0x0	ADC interrupt 6 enable 0: Disable ADC interrupt 6 1: Enable ADC interrupt 6
5	INT5	RW	0x0	ADC interrupt 5 enable 0: Disable ADC interrupt 5 1: Enable ADC interrupt 5
4	INT4	RW	0x0	ADC interrupt 4 enable 0: Disable ADC interrupt 4 1: Enable ADC interrupt 4
3	INT3	RW	0x0	ADC interrupt 3 enable 0: Disable ADC interrupt 3 1: Enable ADC interrupt 3
2	INT2	RW	0x0	ADC interrupt 2 enable 0: Disable ADC interrupt 2 1: Enable ADC interrupt 2
1	INT1	RW	0x0	ADC interrupt 1 enable 0: Disable ADC interrupt 1 1: Enable ADC interrupt 1
0	INT0	RW	0x0	ADC interrupt 0 enable 0: Disable ADC interrupt 0 1: Enable ADC interrupt 0

Table 12-22: ADC Start of Conversion Priority Control Register (ADCSOCPRICTL) Layout

ADCSOCPRICTL (ADC Start of Conversion Priority Control Register) Offset: 0x14 Default: 0x00000000 Access: ADC -> ADCSOCPRICTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
RESERVED_7_4				PRIORITY			

Table 12-23: ADC Start of Conversion Priority Control Register (ADCSOCPRICTL) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:4	RESERVED_7_4	RO	0x0	Reserved.
3:0	PRIORITY	RW	0x0	SOC (Start of Conversion) Priority Determineds the cutoff point for priority mode and round arbitration of SOCs 0000: All SOCs priority follow the rule of round mode 0001: SOC0 are high priority, SOC1-SOC15 is in round mode 0010: SOC0-SOC1 are high priority, SOC2-SOC15 is in round mode 0011: SOC0-SOC2 are high priority, SOC3-SOC15 is in round mode 0100: SOC0-SOC3 are high priority, SOC4-SOC15 is in round mode 0101: SOC0-SOC4 are high priority, SOC5-SOC15 is in round mode 0110: SOC0-SOC5 are high priority, SOC6-SOC15 is in round mode 0111: SOC0-SOC6 are high priority, SOC7-SOC15 is in round mode 1000: SOC0-SOC7 are high priority, SOC8-SOC15 is in round mode 1001: SOC0-SOC8 are high priority, SOC9-SOC15 is in round mode 1010: SOC0-SOC9 are high priority, SOC10-SOC15 is in round mode 1011: SOC0-SOC10 are high priority, SOC11-SOC15 is in round mode 1100: SOC0-SOC11 are high priority, SOC12-SOC15 is in round mode 1101: SOC0-SOC12 are high priority, SOC13-

Bits	Field Name	Type	Reset	Description
				SOC15 is in round mode 1110: SOC0-SOC13 are high priority, SOC14-SOC15 is in round mode 1111: SOC0-SOC14 are high priority, SOC15 is in round mode

Table 12-24: ADC SOC Flag Register (ADCSOCFLG) Layout

ADCSOCFLG (ADC SOC Flag Register) Offset: 0x18 Default: 0x00000000 Access: ADC -> ADCSOCFLG.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 SOC15	14 SOC14	13 SOC13	12 SOC12	11 SOC11	10 SOC10	9 SOC9	8 SOC8
7 SOC7	6 SOC6	5 SOC5	4 SOC4	3 SOC3	2 SOC2	1 SOC1	0 SOC0

Table 12-25: ADC SOC Flag Register (ADCSOCFLG) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	SOC15	RO	0x0	SOC15 (Start of Conversion) flag 0: No sample pending for SOC15 1: Trigger has been received and sample is pending for SOC15. Self-cleared when sample starts.
14	SOC14	RO	0x0	SOC14 (Start of Conversion) flag 0: No sample pending for SOC14 1: Trigger has been received and sample is pending for SOC14. Self-cleared when sample starts.
13	SOC13	RO	0x0	SOC13 (Start of Conversion) flag 0: No sample pending for SOC13 1: Trigger has been received and sample is pending for SOC13. Self-cleared when sample starts.
12	SOC12	RO	0x0	SOC12 (Start of Conversion) flag 0: No sample pending for SOC12 1: Trigger has been received and sample is pending for SOC12. Self-cleared when sample starts.
11	SOC11	RO	0x0	SOC11 (Start of Conversion) flag 0: No sample pending for SOC11 1: Trigger has been received and sample is pending for SOC11. Self-cleared when sample starts.
10	SOC10	RO	0x0	SOC10 (Start of Conversion) flag 0: No sample pending for SOC10 1: Trigger has been received and sample is

Bits	Field Name	Type	Reset	Description
				pending for SOC10. Self-cleared when sample starts.
9	SOC9	RO	0x0	SOC9 (Start of Conversion) flag 0: No sample pending for SOC9 1: Trigger has been received and sample is pending for SOC9. Self-cleared when sample starts.
8	SOC8	RO	0x0	SOC8 (Start of Conversion) flag 0: No sample pending for SOC8 1: Trigger has been received and sample is pending for SOC8. Self-cleared when sample starts.
7	SOC7	RO	0x0	SOC7 (Start of Conversion) flag 0: No sample pending for SOC7 1: Trigger has been received and sample is pending for SOC7. Self-cleared when sample starts.
6	SOC6	RO	0x0	SOC6 (Start of Conversion) flag 0: No sample pending for SOC6 1: Trigger has been received and sample is pending for SOC6. Self-cleared when sample starts.
5	SOC5	RO	0x0	SOC5 (Start of Conversion) flag 0: No sample pending for SOC5 1: Trigger has been received and sample is pending for SOC5. Self-cleared when sample starts.
4	SOC4	RO	0x0	SOC4 (Start of Conversion) flag 0: No sample pending for SOC4 1: Trigger has been received and sample is pending for SOC4. Self-cleared when sample starts.
3	SOC3	RO	0x0	SOC3 (Start of Conversion) flag 0: No sample pending for SOC3 1: Trigger has been received and sample is pending for SOC3. Self-cleared when sample starts.
2	SOC2	RO	0x0	SOC2 (Start of Conversion) flag 0: No sample pending for SOC2 1: Trigger has been received and sample is pending for SOC2. Self-cleared when sample starts.

Bits	Field Name	Type	Reset	Description
1	SOC1	RO	0x0	SOC1 (Start of Conversion) flag 0: No sample pending for SOC1 1: Trigger has been received and sample is pending for SOC1. Self-cleared when sample starts.
0	SOC0	RO	0x0	SOC0 (Start of Conversion) flag 0: No sample pending for SOC0 1: Trigger has been received and sample is pending for SOC0. Self-cleared when sample starts.

Table 12-26: ADC SOC Force Register (ADCSOCFRC) Layout

ADCSOCFRC (ADC SOC Force Register) Offset: 0x1C Default: 0x00000000 Access: ADC -> ADCSOCFRC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 SOC15	14 SOC14	13 SOC13	12 SOC12	11 SOC11	10 SOC10	9 SOC9	8 SOC8
7 SOC7	6 SOC6	5 SOC5	4 SOC4	3 SOC3	2 SOC2	1 SOC1	0 SOC0

Table 12-27: ADC SOC Force Register (ADCSOCFRC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	SOC15	W1S	0x0	SOC15 (Start of Conversion) flag force 0: No action 1: Force high to SOC15 bit in ADCSOCFLG register
14	SOC14	W1S	0x0	SOC14 (Start of Conversion) flag force 0: No action 1: Force high to SOC14 bit in ADCSOCFLG register
13	SOC13	W1S	0x0	SOC13 (Start of Conversion) flag force 0: No action 1: Force high to SOC13 bit in ADCSOCFLG register
12	SOC12	W1S	0x0	SOC12 (Start of Conversion) flag force 0: No action 1: Force high to SOC12 bit in ADCSOCFLG register

Bits	Field Name	Type	Reset	Description
11	SOC11	W1S	0x0	SOC11 (Start of Conversion) flag force 0: No action 1: Force high to SOC11 bit in ADCSOCFLG register
10	SOC10	W1S	0x0	SOC10 (Start of Conversion) flag force 0: No action 1: Force high to SOC10 bit in ADCSOCFLG register
9	SOC9	W1S	0x0	SOC9 (Start of Conversion) flag force 0: No action 1: Force high to SOC9 bit in ADCSOCFLG register
8	SOC8	W1S	0x0	SOC8 (Start of Conversion) flag force 0: No action 1: Force high to SOC8 bit in ADCSOCFLG register
7	SOC7	W1S	0x0	SOC7 (Start of Conversion) flag force 0: No action 1: Force high to SOC7 bit in ADCSOCFLG register
6	SOC6	W1S	0x0	SOC6 (Start of Conversion) flag force 0: No action 1: Force high to SOC6 bit in ADCSOCFLG register
5	SOC5	W1S	0x0	SOC5 (Start of Conversion) flag force 0: No action 1: Force high to SOC5 bit in ADCSOCFLG register
4	SOC4	W1S	0x0	SOC4 (Start of Conversion) flag force 0: No action 1: Force high to SOC4 bit in ADCSOCFLG register
3	SOC3	W1S	0x0	SOC3 (Start of Conversion) flag force 0: No action 1: Force high to SOC3 bit in ADCSOCFLG register
2	SOC2	W1S	0x0	SOC2 (Start of Conversion) flag force 0: No action 1: Force high to SOC2 bit in ADCSOCFLG register
1	SOC1	W1S	0x0	SOC1 (Start of Conversion) flag force 0: No action 1: Force high to SOC1 bit in ADCSOCFLG register
0	SOC0	W1S	0x0	SOC0 (Start of Conversion) flag force 0: No action 1: Force high to SOC0 bit in ADCSOCFLG register

Table 12-28: ADC SOC Overflow Register (ADCSOCOVF) Layout

ADCSOCOVF (ADC SOC Overflow Register) Offset: 0x20 Default: 0x00000000 Access: ADC -> ADCSOCOVF.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 SOC15	14 SOC14	13 SOC13	12 SOC12	11 SOC11	10 SOC10	9 SOC9	8 SOC8
7 SOC7	6 SOC6	5 SOC5	4 SOC4	3 SOC3	2 SOC2	1 SOC1	0 SOC0

Table 12-29: ADC SOC Overflow Register (ADCSOCOVF) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	SOC15	RO	0x0	SOC15 (Start of Conversion) overflow flag This indicates an SOC15 interrupt was generated while an existing SOC15 interrupt was already pending. This overflow flag can't stop SOC15 interrupts from being processing. It just show that a trigger was missed. 0: No SOC15 interrupt overflow 1: SOC15 interrupt overflow
14	SOC14	RO	0x0	SOC14 (Start of Conversion) overflow flag This indicates an SOC14 interrupt was generated while an existing SOC14 interrupt was already pending. This overflow flag can't stop SOC14 interrupts from being processing. It just show that a trigger was missed. 0: No SOC14 interrupt overflow 1: SOC14 interrupt overflow
13	SOC13	RO	0x0	SOC13 (Start of Conversion) overflow flag This indicates an SOC13 interrupt was generated while an existing SOC13 interrupt was already pending. This overflow flag can't stop SOC13 interrupts from being processing. It just show that a trigger was missed. 0: No SOC13 interrupt overflow 1: SOC13 interrupt overflow
12	SOC12	RO	0x0	SOC12 (Start of Conversion) overflow flag This indicates an SOC12 interrupt was generated while an existing SOC12 interrupt was already pending. This overflow flag can't stop SOC12 interrupts from being processing. It just show that a trigger was missed.

Bits	Field Name	Type	Reset	Description
				0: No SOC12 interrupt overflow 1: SOC12 interrupt overflow
11	SOC11	RO	0x0	SOC11 (Start of Conversion) overflow flag This indicates an SOC11 interrupt was generated while an existing SOC11 interrupt was already pending. This overflow flag can't stop SOC11 interrupts from being processing. It just show that a trigger was missed. 0: No SOC11 interrupt overflow 1: SOC11 interrupt overflow
10	SOC10	RO	0x0	SOC10 (Start of Conversion) overflow flag This indicates an SOC10 interrupt was generated while an existing SOC10 interrupt was already pending. This overflow flag can't stop SOC10 interrupts from being processing. It just show that a trigger was missed. 0: No SOC10 interrupt overflow 1: SOC10 interrupt overflow
9	SOC9	RO	0x0	SOC9 (Start of Conversion) overflow flag This indicates an SOC9 interrupt was generated while an existing SOC9 interrupt was already pending. This overflow flag can't stop SOC9 interrupts from being processing. It just show that a trigger was missed. 0: No SOC9 interrupt overflow 1: SOC9 interrupt overflow
8	SOC8	RO	0x0	SOC8 (Start of Conversion) overflow flag This indicates an SOC8 interrupt was generated while an existing SOC8 interrupt was already pending. This overflow flag can't stop SOC8 interrupts from being processing. It just show that a trigger was missed. 0: No SOC8 interrupt overflow 1: SOC8 interrupt overflow
7	SOC7	RO	0x0	SOC7 (Start of Conversion) overflow flag This indicates an SOC7 interrupt was generated while an existing SOC7 interrupt was already pending. This overflow flag can't stop SOC7 interrupts from being processing. It just show that a trigger was missed. 0: No SOC7 interrupt overflow 1: SOC7 interrupt overflow
6	SOC6	RO	0x0	SOC6 (Start of Conversion) overflow flag This indicates an SOC6 interrupt was generated

Bits	Field Name	Type	Reset	Description
				while an existing SOC6 interrupt was already pending. This overflow flag can't stop SOC6 interrupts from being processing. It just show that a trigger was missed. 0: No SOC6 interrupt overflow 1: SOC6 interrupt overflow
5	SOC5	RO	0x0	SOC5 (Start of Conversion) overflow flag This indicates an SOC5 interrupt was generated while an existing SOC5 interrupt was already pending. This overflow flag can't stop SOC5 interrupts from being processing. It just show that a trigger was missed. 0: No SOC5 interrupt overflow 1: SOC5 interrupt overflow
4	SOC4	RO	0x0	SOC4 (Start of Conversion) overflow flag This indicates an SOC4 interrupt was generated while an existing SOC4 interrupt was already pending. This overflow flag can't stop SOC4 interrupts from being processing. It just show that a trigger was missed. 0: No SOC4 interrupt overflow 1: SOC4 interrupt overflow
3	SOC3	RO	0x0	SOC3 (Start of Conversion) overflow flag This indicates an SOC3 interrupt was generated while an existing SOC3 interrupt was already pending. This overflow flag can't stop SOC3 interrupts from being processing. It just show that a trigger was missed. 0: No SOC3 interrupt overflow 1: SOC3 interrupt overflow
2	SOC2	RO	0x0	SOC2 (Start of Conversion) overflow flag This indicates an SOC2 interrupt was generated while an existing SOC2 interrupt was already pending. This overflow flag can't stop SOC2 interrupts from being processing. It just show that a trigger was missed. 0: No SOC2 interrupt overflow 1: SOC2 interrupt overflow
1	SOC1	RO	0x0	SOC1 (Start of Conversion) overflow flag This indicates an SOC1 interrupt was generated while an existing SOC1 interrupt was already pending. This overflow flag can't stop SOC1 interrupts from being processing. It just show that a trigger was missed.

Bits	Field Name	Type	Reset	Description
				0: No SOC1 interrupt overflow 1: SOC1 interrupt overflow
0	SOC0	RO	0x0	SOC0 (Start of Conversion) overflow flag This indicates an SOC0 interrupt was generated while an existing SOC0 interrupt was already pending. This overflow flag can't stop SOC0 interrupts from being processing. It just show that a trigger was missed. 0: No SOC0 interrupt overflow 1: SOC0 interrupt overflow

Table 12-30: ADC SOC Overflow Clear Register (ADCSOCOVFC) Layout

ADCSOCOVFC (ADC SOC Overflow Clear Register) Offset: 0x24 Default: 0x00000000 Access: ADC -> ADCSOCOVFC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 SOC15	14 SOC14	13 SOC13	12 SOC12	11 SOC11	10 SOC10	9 SOC9	8 SOC8
7 SOC7	6 SOC6	5 SOC5	4 SOC4	3 SOC3	2 SOC2	1 SOC1	0 SOC0

Table 12-31: ADC SOC Overflow Clear Register (ADCSOCOVFC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	SOC15	W1S	0x0	Clear SOC15 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC15 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC15 Overflow Flag
14	SOC14	W1S	0x0	Clear SOC14 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC14 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC14 Overflow Flag
13	SOC13	W1S	0x0	Clear SOC13 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC13 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC13 Overflow Flag
12	SOC12	W1S	0x0	Clear SOC12 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC12 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC12 Overflow Flag
11	SOC11	W1S	0x0	Clear SOC11 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC11 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC11 Overflow Flag
10	SOC10	W1S	0x0	Clear SOC10 overflow flag If this bit receive a request to set from software

Bits	Field Name	Type	Reset	Description
				and hardware try to set respective SOC10 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC10 Overflow Flag
9	SOC9	W1S	0x0	Clear SOC9 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC9 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC9 Overflow Flag
8	SOC8	W1S	0x0	Clear SOC8 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC8 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC8 Overflow Flag
7	SOC7	W1S	0x0	Clear SOC7 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC7 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC7 Overflow Flag
6	SOC6	W1S	0x0	Clear SOC6 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC6 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC6 Overflow Flag
5	SOC5	W1S	0x0	Clear SOC5 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC5 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set.

Bits	Field Name	Type	Reset	Description
				0: No clear 1: Clear SOC5 Overflow Flag
4	SOC4	W1S	0x0	Clear SOC4 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC4 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC4 Overflow Flag
3	SOC3	W1S	0x0	Clear SOC3 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC3 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC3 Overflow Flag
2	SOC2	W1S	0x0	Clear SOC2 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC2 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC2 Overflow Flag
1	SOC1	W1S	0x0	Clear SOC1 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC1 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC1 Overflow Flag
0	SOC0	W1S	0x0	Clear SOC0 overflow flag If this bit receive a request to set from software and hardware try to set respective SOC0 overflow bit in ADCSOCOVF register simultaneously, then hardware has priority and ADCSOCOVF register will be set. 0: No clear 1: Clear SOC0 Overflow Flag

Table 12-32: ADC Interrupt Trigger SOC Enable Register (ADCINTSOCEN) Layout

ADCINTSOCEN (ADC Interrupt Trigger SOC Enable Register) Offset: 0x28 Default: 0x00000000 Access: ADC -> ADCINTSOCEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 SOC15	14 SOC14	13 SOC13	12 SOC12	11 SOC11	10 SOC10	9 SOC9	8 SOC8
7 SOC7	6 SOC6	5 SOC5	4 SOC4	3 SOC3	2 SOC2	1 SOC1	0 SOC0

Table 12-33: ADC Interrupt Trigger SOC Enable Register (ADCINTSOCEN) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	SOC15	RW	0x0	ADC interrupt trigger SOC15 enable 0: ADCINTSOCSEL1[SOC15] is ignored 1: ADCSOCCTL15[TRIGSEL] is 417orrespond by ADCINTSOCSEL1[SOC15]
14	SOC14	RW	0x0	ADC interrupt trigger SOC14 enable 0: ADCINTSOCSEL1[SOC14] is ignored 1: ADCSOCCTL14[TRIGSEL] is 417orrespond by ADCINTSOCSEL1[SOC14]
13	SOC13	RW	0x0	ADC interrupt trigger SOC13 enable 0: ADCINTSOCSEL1[SOC13] is ignored 1: ADCSOCCTL13[TRIGSEL] is 417orrespond by ADCINTSOCSEL1[SOC13]
12	SOC12	RW	0x0	ADC interrupt trigger SOC12 enable 0: ADCINTSOCSEL1[SOC12] is ignored 1: ADCSOCCTL12[TRIGSEL] is 417orrespond by ADCINTSOCSEL1[SOC12]
11	SOC11	RW	0x0	ADC interrupt trigger SOC11 enable 0: ADCINTSOCSEL1[SOC11] is ignored 1: ADCSOCCTL11[TRIGSEL] is 417orrespond by ADCINTSOCSEL1[SOC11]
10	SOC10	RW	0x0	ADC interrupt trigger SOC10 enable 0: ADCINTSOCSEL1[SOC10] is ignored 1: ADCSOCCTL10[TRIGSEL] is 417orrespond by ADCINTSOCSEL1[SOC10]
9	SOC9	RW	0x0	ADC interrupt trigger SOC9 enable 0: ADCINTSOCSEL1[SOC9] is ignored 1: ADCSOCCTL9[TRIGSEL] is 417orrespond by ADCINTSOCSEL1[SOC9]

Bits	Field Name	Type	Reset	Description
8	SOC8	RW	0x0	ADC interrupt trigger SOC8 enable 0: ADCINTSOCSEL0[SOC8] is ignored 1: ADCSOCCTL8[TRIGSEL] is 418orrespond by ADCINTSOCSEL1[SOC8]
7	SOC7	RW	0x0	ADC interrupt trigger SOC7 enable 0: ADCINTSOCSEL0[SOC7] is ignored 1: ADCSOCCTL7[TRIGSEL] is 418orrespond by ADCINTSOCSEL0[SOC7]
6	SOC6	RW	0x0	ADC interrupt trigger SOC6 enable 0: ADCINTSOCSEL0[SOC6] is ignored 1: ADCSOCCTL6[TRIGSEL] is 418orrespond by ADCINTSOCSEL0[SOC6]
5	SOC5	RW	0x0	ADC interrupt trigger SOC5 enable 0: ADCINTSOCSEL0[SOC5] is ignored 1: ADCSOCCTL5[TRIGSEL] is 418orrespond by ADCINTSOCSEL0[SOC5]
4	SOC4	RW	0x0	ADC interrupt trigger SOC4 enable 0: ADCINTSOCSEL0[SOC4] is ignored 1: ADCSOCCTL4[TRIGSEL] is 418orrespond by ADCINTSOCSEL0[SOC4]
3	SOC3	RW	0x0	ADC interrupt trigger SOC3 enable 0: ADCINTSOCSEL0[SOC3] is ignored 1: ADCSOCCTL3[TRIGSEL] is 418orrespond by ADCINTSOCSEL0[SOC3]
2	SOC2	RW	0x0	ADC interrupt trigger SOC2 enable 0: ADCINTSOCSEL0[SOC2] is ignored 1: ADCSOCCTL2[TRIGSEL] is 418orrespond by ADCINTSOCSEL0[SOC2]
1	SOC1	RW	0x0	ADC interrupt trigger SOC1 enable 0: ADCINTSOCSEL0[SOC1] is ignored 1: ADCSOCCTL1[TRIGSEL] is 418orrespond by ADCINTSOCSEL0[SOC1]
0	SOC0	RW	0x0	ADC interrupt trigger SOC0 enable 0: ADCINTSOCSEL0[SOC0] is ignored 1: ADCSOCCTL0[TRIGSEL] is 418orrespond by ADCINTSOCSEL0[SOC0]

Table 12-34: ADC Interrupt Trigger SOC Select Register 0 (ADCINTSOCSEL0) Layout

ADCINTSOCSEL0 (ADC Interrupt Trigger SOC Select Register 0) Offset: 0x2C Default: 0x00000000 Access: ADC -> ADCINTSOCSEL0.all							
31	30	29	28	27	26	25	24
SOC7				SOC6			
23	22	21	20	19	18	17	16
SOC5				SOC4			
15	14	13	12	11	10	9	8
SOC3				SOC2			
7	6	5	4	3	2	1	0
SOC1				SOC0			

Table 12-35: ADC Interrupt Trigger SOC Select Register 0 (ADCINTSOCSEL0) Description

Bits	Field Name	Type	Reset	Description
31:28	SOC7	RW	0x0	ADC interrupt source for SOC7
27:24	SOC6	RW	0x0	ADC interrupt source for SOC6
23:20	SOC5	RW	0x0	ADC interrupt source for SOC5
19:16	SOC4	RW	0x0	ADC interrupt source for SOC4
15:12	SOC3	RW	0x0	ADC interrupt source for SOC3
11:8	SOC2	RW	0x0	ADC interrupt source for SOC2
7:4	SOC1	RW	0x0	ADC interrupt source for SOC1
3:0	SOC0	RW	0x0	ADC interrupt source for SOC0

Table 12-36: ADC Interrupt Trigger SOC Select Register 1 (ADCINTSOCSEL1) Layout

ADCINTSOCSEL1 (ADC Interrupt Trigger SOC Select Register 1) Offset: 0x30 Default: 0x00000000 Access: ADC -> ADCINTSOCSEL1.all							
31	30	29	28	27	26	25	24
SOC15				SOC14			
23	22	21	20	19	18	17	16
SOC13				SOC12			
15	14	13	12	11	10	9	8
SOC11				SOC10			
7	6	5	4	3	2	1	0
SOC9				SOC8			

Table 12-37: ADC Interrupt Trigger SOC Select Register 1 (ADCINTSOCSEL1) Description

Bits	Field Name	Type	Reset	Description
31:28	SOC15	RW	0x0	ADC interrupt source for SOC15
27:24	SOC14	RW	0x0	ADC interrupt source for SOC14
23:20	SOC13	RW	0x0	ADC interrupt source for SOC13

Bits	Field Name	Type	Reset	Description
19:16	SOC12	RW	0x0	ADC interrupt source for SOC12
15:12	SOC11	RW	0x0	ADC interrupt source for SOC11
11:8	SOC10	RW	0x0	ADC interrupt source for SOC10
7:4	SOC9	RW	0x0	ADC interrupt source for SOC9
3:0	SOC8	RW	0x0	ADC interrupt source for SOC8

Table 12-38: ADC External SOC Input Control Register (ADCEXTSOCCTL) Layout

ADCEXTSOCCTL (ADC External SOC Input Control Register) Offset: 0x34 Default: 0x0000007F Access: ADC -> ADCEXTSOCCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	POL	IOSEL					

Table 12-39: ADC External SOC Input Control Register (ADCEXTSOCCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	POL	RW	0x1	EXTSOC polarity 0: Active when low 1: Active when high
5:0	IOSEL	RW	0x3F	GPIO number for EXTSOC

Table 12-40: ADC SOC0 Control Register (ADCSOCCTL0) Layout

ADCSOCCTL0 (ADC SOC0 Control Register) Offset: 0x38 Default: 0x00000000 Access: ADC -> ADCSOCCTL0.all							
31	30	29	28	27	26	25	24
SHEN			TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-41: ADC SOC0 Control Register (ADCSOCCTL0) Description

Bits	Field Name	Type	Reset	Description
31:29	SHEN	RW	0x0	SH enable 000: Disable all SHs 001: Enable SHA 010: Enable SHB 011: Enable SHC 100: Enable SHA and SHB simultaneously 101: Enable SHB and SHC simultaneously 110: Enable SHA and SHC simultaneously 111: Enable SHA, SHB, SHC simultaneously
28:24	TRIGSEL	RW	0x0	Select SOC0 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC0]. The flag will initiate a conversion to start once and the priority is given to SOC0. This setting can be overridden by the ADCINTSOCSEL0[SOC0]. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101:

Bits	Field Name	Type	Reset	Description
				11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC0 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC0 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average

Bits	Field Name	Type	Reset	Description
				111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC0 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC0 sample window size

Table 12-42: ADC SOC1 Control Register (ADCSOCCTL1) Layout

ADCSOCCTL1 (ADC SOC1 Control Register) Offset: 0x3C Default: 0x00000000 Access: ADC -> ADCSOCCTL1.all							
31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-43: ADC SOC1 Control Register (ADCSOCCTL1) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL0[SHEN] is set to 4,5 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC1 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC1]. The flag will initiate a conversion to start once and the priority is given to SOC1. This setting can be overridden by the ADCINTSOCSEL0[SOC1]. This field is not used when ADCSOCCTL0[SHEN] is set to 4,5 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC

Bits	Field Name	Type	Reset	Description
				10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC1 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL0[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC0 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC1 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL0[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC0 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations

Bits	Field Name	Type	Reset	Description
				and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC1 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC1 sample window size

Table 12-44: ADC SOC2 Control Register (ADCSOCCTL2) Layout

ADCSOCCTL2 (ADC SOC2 Control Register) Offset: 0x40 Default: 0x00000000
Access: ADC -> ADCSOCCTL2.all

31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-45: ADC SOC2 Control Register (ADCSOCCTL2) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL0[SHEN] is set to 5,6 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC2 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC2]. The flag will initiate a conversion to start once and the priority is given to SOC2. This setting can be overridden by the

Bits	Field Name	Type	Reset	Description
				ADCINTSOCSEL0[SOC2]. This field is not used when ADCSOCCTL0[SHEN] is set to 5,6 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC2 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL0[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC0 000: 001: 010: 011:

Bits	Field Name	Type	Reset	Description
				100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC2 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL0[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC0 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC2 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC2 sample window size

Table 12-46: ADC SOC3 Control Register (ADCSOCCTL3) Layout

ADCSOCCTL3 (ADC SOC3 Control Register) Offset: 0x44 Default: 0x00000000 Access: ADC -> ADCSOCCTL3.all							
31	30	29	28	27	26	25	24
SHEN			TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-47: ADC SOC3 Control Register (ADCSOCCTL3) Description

Bits	Field Name	Type	Reset	Description
31:29	SHEN	RW	0x0	SH enable 000: Disable all SHs 001: Enable SHA 010: Enable SHB 011: Enable SHC 100: Enable SHA and SHB simultaneously 101: Enable SHB and SHC simultaneously 110: Enable SHA and SHC simultaneously 111: Enable SHA, SHB, SHC simultaneously
28:24	TRIGSEL	RW	0x0	Select SOC3 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC3]. The flag will initiate a conversion to start once and the priority is given to SOC3. This setting can be overridden by the ADCINTSOCSEL0[SOC3]. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC

Bits	Field Name	Type	Reset	Description
				10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC3 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC3 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC

Bits	Field Name	Type	Reset	Description
				operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC3 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC3 sample window size

Table 12-48: ADC SOC4 Control Register (ADCSOCCTL4) Layout

ADCSOCCTL4 (ADC SOC4 Control Register) Offset: 0x48 Default: 0x00000000
Access: ADC -> ADCSOCCTL4.all

31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-49: ADC SOC4 Control Register (ADCSOCCTL4) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL3[SHEN] is set to 4,5 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC4 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC4]. The flag will initiate a conversion to start once and the priority is given to SOC4. This setting can be overridden by the

Bits	Field Name	Type	Reset	Description
				ADCINTSOCSEL0[SOC4]. This field is not used when ADCSOCCTL3[SHEN] is set to 4,5 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC4 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL3[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC3 000: 001: 010: 011:

Bits	Field Name	Type	Reset	Description
				100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC4 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL3[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC3 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC4 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC4 sample window size

Table 12-50: ADC SOC5 Control Register (ADCSOCCTL5) Layout

ADCSOCCTL5 (ADC SOC5 Control Register) Offset: 0x4C Default: 0x00000000
Access: ADC -> ADCSOCCTL5.all

31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-51: ADC SOC5 Control Register (ADCSOCCTL5) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL3[SHEN] is set to 5,6 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC5 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC5]. The flag will initiate a conversion to start once and the priority is given to SOC5. This setting can be overridden by the ADCINTSOCSEL0[SOC5]. This field is not used when ADCSOCCTL3[SHEN] is set to 5,6 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB

Bits	Field Name	Type	Reset	Description
				10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC5 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL3[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC3 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC5 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL3[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC3 000: 001: 010: 011: 100: 101: 110: 111:

Bits	Field Name	Type	Reset	Description
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC5 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC5 sample window size

Table 12-52: ADC SOC6 Control Register (ADCSOCCTL6) Layout

ADCSOCCTL6 (ADC SOC6 Control Register) Offset: 0x50 Default: 0x00000000
Access: ADC -> ADCSOCCTL6.all

31	30	29	28	27	26	25	24
SHEN			TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-53: ADC SOC6 Control Register (ADCSOCCTL6) Description

Bits	Field Name	Type	Reset	Description
31:29	SHEN	RW	0x0	SH enable 000: Disable all SHs 001: Enable SHA 010: Enable SHB 011: Enable SHC 100: Enable SHA and SHB simultaneously 101: Enable SHB and SHC simultaneously 110: Enable SHA and SHC simultaneously 111: Enable SHA, SHB, SHC simultaneously

Bits	Field Name	Type	Reset	Description
28:24	TRIGSEL	RW	0x0	Select SOC6 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC6]. The flag will initiate a conversion to start once and the priority is given to SOC6. This setting can be overridden by the ADCINTSOCSEL0[SOC6]. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC6 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001:

Bits	Field Name	Type	Reset	Description
				010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC6 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC6 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC6 sample window size

Table 12-54: ADC SOC7 Control Register (ADCSOCCTL7) Layout

ADCSOCCTL7 (ADC SOC7 Control Register) Offset: 0x54 Default: 0x00000000 Access: ADC -> ADCSOCCTL7.all							
31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-55: ADC SOC7 Control Register (ADCSOCCTL7) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL6[SHEN] is set to 4,5 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC7 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC7]. The flag will initiate a conversion to start once and the priority is given to SOC7. This setting can be overridden by the ADCINTSOCSEL0[SOC7]. This field is not used when ADCSOCCTL6[SHEN] is set to 4,5 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB

Bits	Field Name	Type	Reset	Description
				10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC7 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL6[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC6 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC6 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL6[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC6 000: 001: 010: 011: 100: 101: 110: 111:

Bits	Field Name	Type	Reset	Description
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC7 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC7 sample window size

Table 12-56: ADC SOC8 Control Register (ADCSOCCTL8) Layout

ADCSOCCTL8 (ADC SOC8 Control Register) Offset: 0x58 Default: 0x00000000
Access: ADC -> ADCSOCCTL8.all

31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-57: ADC SOC8 Control Register (ADCSOCCTL8) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL6[SHEN] is set to 5,6 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC8 Trigger Source Configures which trigger source will set

Bits	Field Name	Type	Reset	Description
				ADCSOCFLG[SOC8]. The flag will initiate a conversion to start once and the priority is given to SOC8. This setting can be overridden by the ADCINTSOCSEL1[SOC8]. This field is not used when ADCSOCCTL6[SHEN] is set to 5,6 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC8 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL6[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC6 000:

Bits	Field Name	Type	Reset	Description
				001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC8 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL6[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC6 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC8 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC8 sample window size

Table 12-58: ADC SOC9 Control Register (ADCSOCCTL9) Layout

ADCSOCCTL9 (ADC SOC9 Control Register) Offset: 0x5C Default: 0x00000000 Access: ADC -> ADCSOCCTL9.all							
31	30	29	28	27	26	25	24
SHEN			TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-59: ADC SOC9 Control Register (ADCSOCCTL9) Description

Bits	Field Name	Type	Reset	Description
31:29	SHEN	RW	0x0	SH enable 000: Disable all SHs 001: Enable SHA 010: Enable SHB 011: Enable SHC 100: Enable SHA and SHB simultaneously 101: Enable SHB and SHC simultaneously 110: Enable SHA and SHC simultaneously 111: Enable SHA, SHB, SHC simultaneously
28:24	TRIGSEL	RW	0x0	Select SOC9 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC9]. The flag will initiate a conversion to start once and the priority is given to SOC9. This setting can be overridden by the ADCINTSOCSEL1[SOC9]. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC

Bits	Field Name	Type	Reset	Description
				10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC9 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC9 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC

Bits	Field Name	Type	Reset	Description
				operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC9 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC9 sample window size

Table 12-60: ADC SOC10 Control Register (ADCSOCCTL10) Layout

ADCSOCCTL10 (ADC SOC10 Control Register) Offset: 0x60 Default: 0x00000000
Access: ADC -> ADCSOCCTL10.all

31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-61: ADC SOC10 Control Register (ADCSOCCTL10) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL9[SHEN] is set to 4,5 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC10 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC10]. The flag will initiate a conversion to start once and the priority is given to SOC10. This setting can be overridden by the

Bits	Field Name	Type	Reset	Description
				ADCINTSOCSEL1[SOC10]. This field is not used when ADCSOCCTL9[SHEN] is set to 4,5 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC10 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL9[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC9 000: 001: 010: 011:

Bits	Field Name	Type	Reset	Description
				100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC10 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL9[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC9 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC10 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC10 sample window size

Table 12-62: ADC SOC11 Control Register (ADCSOCCTL11) Layout

ADCSOCCTL11 (ADC SOC11 Control Register) Offset: 0x64 Default: 0x00000000 Access: ADC -> ADCSOCCTL11.all							
31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-63: ADC SOC11 Control Register (ADCSOCCTL11) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL9[SHEN] is set to 5,6 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC11 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC11]. The flag will initiate a conversion to start once and the priority is given to SOC11. This setting can be overridden by the ADCINTSOCSEL1[SOC11]. This field is not used when ADCSOCCTL9[SHEN] is set to 5,6 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB

Bits	Field Name	Type	Reset	Description
				10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC11 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL9[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC9 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC11 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL9[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC9 000: 001: 010: 011: 100: 101: 110: 111:

Bits	Field Name	Type	Reset	Description
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC9 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC9 sample window size

Table 12-64: ADC SOC12 Control Register (ADCSOCCTL12) Layout

ADCSOCCTL12 (ADC SOC12 Control Register) Offset: 0x68 Default: 0x00000000
Access: ADC -> ADCSOCCTL12.all

31	30	29	28	27	26	25	24
SHEN			TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-65: ADC SOC12 Control Register (ADCSOCCTL12) Description

Bits	Field Name	Type	Reset	Description
31:29	SHEN	RW	0x0	SH enable 000: Disable all SHs 001: Enable SHA 010: Enable SHB 011: Enable SHC 100: Enable SHA and SHB simultaneously 101: Enable SHB and SHC simultaneously 110: Enable SHA and SHC simultaneously 111: Enable SHA, SHB, SHC simultaneously

Bits	Field Name	Type	Reset	Description
28:24	TRIGSEL	RW	0x0	Select SOC12 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC12]. The flag will initiate a conversion to start once and the priority is given to SOC12. This setting can be overridden by the ADCINTSOCSEL1[SOC12]. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC12 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001:

Bits	Field Name	Type	Reset	Description
				010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC12 When SHEN=1, 4, 6 or 7, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When SHEN=5, this field is not used. 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC12 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC12 sample window size

Table 12-66: ADC SOC13 Control Register (ADCSOCCTL13) Layout

ADCSOCCTL13 (ADC SOC13 Control Register) Offset: 0x6C Default: 0x00000000 Access: ADC -> ADCSOCCTL13.all							
31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-67: ADC SOC13 Control Register (ADCSOCCTL13) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL12[SHEN] is set to 4,5 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC13 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC13]. The flag will initiate a conversion to start once and the priority is given to SOC13. This setting can be overridden by the ADCINTSOCSEL1[SOC13]. This field is not used when ADCSOCCTL12[SHEN] is set to 4,5 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB

Bits	Field Name	Type	Reset	Description
				10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC13 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL12[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC12 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC13 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL12[SHEN] is set to 4, 5 or 7, SHB is controlled for SOC12 000: 001: 010: 011: 100: 101: 110: 111:

Bits	Field Name	Type	Reset	Description
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC13 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC13 sample window size

Table 12-68: ADC SOC14 Control Register (ADCSOCCTL14) Layout

ADCSOCCTL14 (ADC SOC14 Control Register) Offset: 0x70 Default: 0x00000000
Access: ADC -> ADCSOCCTL14.all

31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-69: ADC SOC14 Control Register (ADCSOCCTL14) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable This field is not used when ADCSOCCTL12[SHEN] is set to 5,6 or 7. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC14 Trigger Source Configures which trigger source will set

Bits	Field Name	Type	Reset	Description
				ADCSOCFLG[SOC14]. The flag will initiate a conversion to start once and the priority is given to SOC14. This setting can be overridden by the ADCINTSOCSEL1[SOC14]. This field is not used when ADCSOCCTL12[SHEN] is set to 5,6 or 7. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC 10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC14 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL12[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC12 000:

Bits	Field Name	Type	Reset	Description
				001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC14 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. When ADCSOCCTL12[SHEN] is set to 5, 6 or 7, SHC is controlled for SOC12 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average 100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC14 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC14 sample window size

Table 12-70: ADC SOC15 Control Register (ADCSOCCTL15) Layout

ADCSOCCTL15 (ADC SOC15 Control Register) Offset: 0x74 Default: 0x00000000 Access: ADC -> ADCSOCCTL15.all							
31	30	29	28	27	26	25	24
RESERVED_31	SHEN		TRIGSEL				
23	22	21	20	19	18	17	16
CHSELN			CHSELP			AVGCNT	
15	14	13	12	11	10	9	8
AVGCNT	CONVCNT						
7	6	5	4	3	2	1	0
SAMPCNT							

Table 12-71: ADC SOC15 Control Register (ADCSOCCTL15) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RO	0x0	Reserved.
30:29	SHEN	RW	0x0	SH enable. 00: Disable all SHs 01: Enable SHA 10: Enable SHB 11: Enable SHC
28:24	TRIGSEL	RW	0x0	Select SOC15 Trigger Source Configures which trigger source will set ADCSOCFLG[SOC15]. The flag will initiate a conversion to start once and the priority is given to SOC15. This setting can be overridden by the ADCINTSOCSEL1[SOC15]. 00000: ADCTRIG0 – Software only 00001: ADCTRIG1 – CPU Timer 0 00010: ADCTRIG2 – CPU Timer 1 00011: ADCTRIG3 – CPU Timer 2 00100: ADCTRIG4 – EXTSOC 00101: ADCTRIG5 – PWM0SOCA 00110: ADCTRIG6 – PWM0SOCB 00111: ADCTRIG7 – PWM0SOCC 01000: ADCTRIG8 – PWM1SOCA 01001: ADCTRIG9 – PWM1SOCB 01010: ADCTRIG10 – PWM1SOCC 01011: ADCTRIG11 – PWM2SOCA 01100: ADCTRIG12 – PWM2SOCB 01101: ADCTRIG13 – PWM2SOCC 01110: ADCTRIG14 – PWM3SOCA 01111: ADCTRIG15 – PWM3SOCB 10000: ADCTRIG16 – PWM3SOCC 10001: ADCTRIG17 – PWM4SOCA 10010: ADCTRIG18 – PWM4SOCB 10011: ADCTRIG19 – PWM4SOCC

Bits	Field Name	Type	Reset	Description
				10100: ADCTRIG20 – PWM5SOCA 10101: ADCTRIG21 – PWM5SOCB 10110: ADCTRIG22 – PWM5SOCC 10111: 11000: 11001: 11010: 11011: 11100: 11101: 11110: 11111:
23:21	CHSELN	RW	0x0	Select channel to negative terminal of SH for SOC15 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. 000: 001: 010: 011: 100: 101: 110: 111:
20:18	CHSELP	RW	0x0	Select channel to positive terminal of SH for SOC15 When SHEN=1, SHA is controlled. When SHEN=2, SHB is controlled. When SHEN=3, SHC is controlled. 000: 001: 010: 011: 100: 101: 110: 111:
17:15	AVGCNT	RW	0x0	Select average counts for the ADC result 000: No average 001: Each SOC will trigger 2 consecutive ADC operations and then average 010: Each SOC will trigger 4 consecutive ADC operations and then average 011: Each SOC will trigger 8 consecutive ADC operations and then average

Bits	Field Name	Type	Reset	Description
				100: Each SOC will trigger 16 consecutive ADC operations and then average 101: Each SOC will trigger 32 consecutive ADC operations and then average 110: Each SOC will trigger 64 consecutive ADC operations and then average 111: Each SOC will trigger 128 consecutive ADC operations and then average
14:8	CONVCNT	RW	0x0	Select SOC15 convert window size
7:0	SAMPCNT	RW	0x0	Select SOC15 sample window size

Table 12-72: ADC Offset Trim Register 0 (ADCOFFSET0) Layout

ADCOFFSET0 (ADC Offset Trim Register 0) Offset: 0x78 Default: 0x00000000 Access: ADC -> ADCOFFSET0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-73: ADC Offset Trim Register 0 (ADCOFFSET0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT0 Should be within [-8192, 8191]

Table 12-74: ADC Offset Trim Register 1 (ADCOFFSET1) Layout

ADCOFFSET1 (ADC Offset Trim Register 1) Offset: 0x7C Default: 0x00000000 Access: ADC -> ADCOFFSET1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-75: ADC Offset Trim Register 1 (ADCOFFSET1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT1 Should be within [-8192, 8191]

Table 12-76: ADC Offset Trim Register 2 (ADCOFFSET2) Layout

ADCOFFSET2 (ADC Offset Trim Register 2) Offset: 0x80 Default: 0x00000000 Access: ADC -> ADCOFFSET2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-77: ADC Offset Trim Register 2 (ADCOFFSET2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT2 Should be within [-8192, 8191]

Table 12-78: ADC Offset Trim Register 3 (ADCOFFSET3) Layout

ADCOFFSET3 (ADC Offset Trim Register 3) Offset: 0x84 Default: 0x00000000 Access: ADC -> ADCOFFSET3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-79: ADC Offset Trim Register 3 (ADCOFFSET3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT3 Should be within [-8192, 8191]

Table 12-80: ADC Offset Trim Register 4 (ADCOFFSET4) Layout

ADCOFFSET4 (ADC Offset Trim Register 4) Offset: 0x88 Default: 0x00000000 Access: ADC -> ADCOFFSET4.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-81: ADC Offset Trim Register 4 (ADCOFFSET4) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT4 Should be within [-8192, 8191]

Table 12-82: ADC Offset Trim Register 5 (ADCOFFSET5) Layout

ADCOFFSET5 (ADC Offset Trim Register 5) Offset: 0x8C Default: 0x00000000 Access: ADC -> ADCOFFSET5.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-83: ADC Offset Trim Register 5 (ADCOFFSET5) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT5 Should be within [-8192, 8191]

Table 12-84: ADC Offset Trim Register 6 (ADCOFFSET6) Layout

ADCOFFSET6 (ADC Offset Trim Register 6) Offset: 0x90 Default: 0x00000000 Access: ADC -> ADCOFFSET6.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-85: ADC Offset Trim Register 6 (ADCOFFSET6) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT6 Should be within [-8192, 8191]

Table 12-86: ADC Offset Trim Register 7 (ADCOFFSET7) Layout

ADCOFFSET7 (ADC Offset Trim Register 7) Offset: 0x94 Default: 0x00000000 Access: ADC -> ADCOFFSET7.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-87: ADC Offset Trim Register 7 (ADCOFFSET7) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT7 Should be within [-8192, 8191]

Table 12-88: ADC Offset Trim Register 8 (ADCOFFSET8) Layout

ADCOFFSET8 (ADC Offset Trim Register 8) Offset: 0x98 Default: 0x00000000 Access: ADC -> ADCOFFSET8.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-89: ADC Offset Trim Register 8 (ADCOFFSET8) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT8 Should be within [-8192, 8191]

Table 12-90: ADC Offset Trim Register 9 (ADCOFFSET9) Layout

ADCOFFSET9 (ADC Offset Trim Register 9) Offset: 0x9C Default: 0x00000000 Access: ADC -> ADCOFFSET9.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-91: ADC Offset Trim Register 9 (ADCOFFSET9) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT9 Should be within [-8192, 8191]

Table 12-92: ADC Offset Trim Register 10 (ADCOFFSET10) Layout

ADCOFFSET10 (ADC Offset Trim Register 10) Offset: 0xA0 Default: 0x00000000 Access: ADC -> ADCOFFSET10.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-93: ADC Offset Trim Register 10 (ADCOFFSET10) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT10 Should be within [-8192, 8191]

Table 12-94: ADC Offset Trim Register 11 (ADCOFFSET11) Layout

ADCOFFSET11 (ADC Offset Trim Register 11) Offset: 0xA4 Default: 0x00000000 Access: ADC -> ADCOFFSET11.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-95: ADC Offset Trim Register 11 (ADCOFFSET11) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT11 Should be within [-8192, 8191]

Table 12-96: ADC Offset Trim Register 12 (ADCOFFSET12) Layout

ADCOFFSET12 (ADC Offset Trim Register 12) Offset: 0xA8 Default: 0x00000000 Access: ADC -> ADCOFFSET12.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-97: ADC Offset Trim Register 12 (ADCOFFSET12) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT12 Should be within [-8192, 8191]

Table 12-98: ADC Offset Trim Register 13 (ADCOFFSET13) Layout

ADCOFFSET13 (ADC Offset Trim Register 13) Offset: 0xAC Default: 0x00000000 Access: ADC -> ADCOFFSET13.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-99: ADC Offset Trim Register 13 (ADCOFFSET13) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT13 Should be within [-8192, 8191]

Table 12-100: ADC Offset Trim Register 14 (ADCOFFSET14) Layout

ADCOFFSET14 (ADC Offset Trim Register 14) Offset: 0xB0 Default: 0x00000000 Access: ADC -> ADCOFFSET14.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-101: ADC Offset Trim Register 14 (ADCOFFSET14) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT14 Should be within [-8192, 8191]

Table 12-102: ADC Offset Trim Register 15 (ADCOFFSET15) Layout

ADCOFFSET15 (ADC Offset Trim Register 15) Offset: 0xB4 Default: 0x00000000 Access: ADC -> ADCOFFSET15.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-103: ADC Offset Trim Register 15 (ADCOFFSET15) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for ADCRESULT15 Should be within [-8192, 8191]

Table 12-104: ADC Gain Trim Register 0 (ADCGAIN0) Layout

ADCGAIN0 (ADC Gain Trim Register 0) Offset: 0xB8 Default: 0x00008000 Access: ADC -> ADCGAIN0.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-105: ADC Gain Trim Register 0 (ADCGAIN0) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT0

Table 12-106: ADC Gain Trim Register 1 (ADCGAIN1) Layout

ADCGAIN1 (ADC Gain Trim Register 1) Offset: 0xBC Default: 0x00008000 Access: ADC -> ADCGAIN1.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-107: ADC Gain Trim Register 1 (ADCGAIN1) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT1

Table 12-108: ADC Gain Trim Register 2 (ADCGAIN2) Layout

ADCGAIN2 (ADC Gain Trim Register 2) Offset: 0xC0 Default: 0x00008000 Access: ADC -> ADCGAIN2.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-109: ADC Gain Trim Register 2 (ADCGAIN2) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT2

Table 12-110: ADC Gain Trim Register 3 (ADCGAIN3) Layout

ADCGAIN3 (ADC Gain Trim Register 3) Offset: 0xC4 Default: 0x00008000 Access: ADC -> ADCGAIN3.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-111: ADC Gain Trim Register 3 (ADCGAIN3) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT3

Table 12-112: ADC Gain Trim Register 4 (ADCGAIN4) Layout

ADCGAIN4 (ADC Gain Trim Register 4) Offset: 0xC8 Default: 0x00008000 Access: ADC -> ADCGAIN4.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-113: ADC Gain Trim Register 4 (ADCGAIN4) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT4

Table 12-114: ADC Gain Trim Register 5 (ADCGAIN5) Layout

ADCGAIN5 (ADC Gain Trim Register 5) Offset: 0xCC Default: 0x00008000 Access: ADC -> ADCGAIN5.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-115: ADC Gain Trim Register 5 (ADCGAIN5) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT5

Table 12-116: ADC Gain Trim Register 6 (ADCGAIN6) Layout

ADCGAIN6 (ADC Gain Trim Register 6) Offset: 0xD0 Default: 0x00008000 Access: ADC -> ADCGAIN6.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-117: ADC Gain Trim Register 6 (ADCGAIN6) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT6

Table 12-118: ADC Gain Trim Register 7 (ADCGAIN7) Layout

ADCGAIN7 (ADC Gain Trim Register 7) Offset: 0xD4 Default: 0x00008000 Access: ADC -> ADCGAIN7.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-119: ADC Gain Trim Register 7 (ADCGAIN7) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT7

Table 12-120: ADC Gain Trim Register 8 (ADCGAIN8) Layout

ADCGAIN8 (ADC Gain Trim Register 8) Offset: 0xD8 Default: 0x00008000 Access: ADC -> ADCGAIN8.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-121: ADC Gain Trim Register 8 (ADCGAIN8) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT8

Table 12-122: ADC Gain Trim Register 9 (ADCGAIN9) Layout

ADCGAIN9 (ADC Gain Trim Register 9) Offset: 0xDC Default: 0x00008000 Access: ADC -> ADCGAIN9.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-123: ADC Gain Trim Register 9 (ADCGAIN9) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT9

Table 12-124: ADC Gain Trim Register 10 (ADCGAIN10) Layout

ADCGAIN10 (ADC Gain Trim Register 10) Offset: 0xE0 Default: 0x00008000 Access: ADC -> ADCGAIN10.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-125: ADC Gain Trim Register 10 (ADCGAIN10) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT10

Table 12-126: ADC Gain Trim Register 11 (ADCGAIN11) Layout

ADCGAIN11 (ADC Gain Trim Register 11) Offset: 0xE4 Default: 0x00008000 Access: ADC -> ADCGAIN11.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-127: ADC Gain Trim Register 11 (ADCGAIN11) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT11

Table 12-128: ADC Gain Trim Register 12 (ADCGAIN12) Layout

ADCGAIN12 (ADC Gain Trim Register 12) Offset: 0xE8 Default: 0x00008000 Access: ADC -> ADCGAIN12.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-129: ADC Gain Trim Register 12 (ADCGAIN12) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT12

Table 12-130: ADC Gain Trim Register 13 (ADCGAIN13) Layout

ADCGAIN13 (ADC Gain Trim Register 13) Offset: 0xEC Default: 0x00008000 Access: ADC -> ADCGAIN13.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-131: ADC Gain Trim Register 13 (ADCGAIN13) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT13

Table 12-132: ADC Gain Trim Register 14 (ADCGAIN14) Layout

ADCGAIN14 (ADC Gain Trim Register 14) Offset: 0xF0 Default: 0x00008000 Access: ADC -> ADCGAIN14.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-133: ADC Gain Trim Register 14 (ADCGAIN14) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT14

Table 12-134: ADC Gain Trim Register 15 (ADCGAIN15) Layout

ADCGAIN15 (ADC Gain Trim Register 15) Offset: 0xF4 Default: 0x00008000 Access: ADC -> ADCGAIN15.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-135: ADC Gain Trim Register 15 (ADCGAIN15) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for ADCRESULT15

Table 12-136: ADC SHA Offset Trim Register (ADCOFFSETA) Layout

ADCOFFSETA (ADC SHA Offset Trim Register) Offset: 0xF8 Default: 0x00000000 Access: ADC -> ADCOFFSETA.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-137: ADC SHA Offset Trim Register (ADCOFFSETA) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for SHA Should be within [-8192, 8191]

Table 12-138: ADC SHB Offset Trim Register (ADCOFFSETB) Layout

ADCOFFSETB (ADC SHB Offset Trim Register) Offset: 0xFC Default: 0x00000000 Access: ADC -> ADCOFFSETB.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-139: ADC SHB Offset Trim Register (ADCOFFSETB) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for SHB Should be within [-8192, 8191]

Table 12-140: ADC SHC Offset Trim Register (ADCOFFSETC) Layout

ADCOFFSETC (ADC SHC Offset Trim Register) Offset: 0x100 Default: 0x00000000 Access: ADC -> ADCOFFSETC.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-141: ADC SHC Offset Trim Register (ADCOFFSETC) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Offset trimming (signed number) for SHC Should be within [-8192, 8191]

Table 12-142: ADC SHA Gain Trim Register (ADCGAINA) Layout

ADCGAINA (ADC SHA Gain Trim Register) Offset: 0x104 Default: 0x00008000 Access: ADC -> ADCGAINA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-143: ADC SHA Gain Trim Register (ADCGAINA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for SHA

Table 12-144: ADC SHB Gain Trim Register (ADCGAINB) Layout

ADCGAINB (ADC SHB Gain Trim Register) Offset: 0x108 Default: 0x00008000 Access: ADC -> ADCGAINB.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-145: ADC SHB Gain Trim Register (ADCGAINB) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for SHB

Table 12-146: ADC SHC Gain Trim Register (ADCGAINC) Layout

ADCGAINC (ADC SHC Gain Trim Register) Offset: 0x10C Default: 0x00008000 Access: ADC -> ADCGAINC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-147: ADC SHC Gain Trim Register (ADCGAINC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8000	Gain trimming (VAL/32768) for SHC

Table 12-148: ADC Status Register (ADCSTS) Layout

ADCSTS (ADC Status Register) Offset: 0x110 Default: 0x00000000 Access: ADC -> ADCSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					CODECRDY	CODEBRDY	CODEARDY

Table 12-149: ADC Status Register (ADCSTS) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	CODECRDY	RO	0x0	ADCRAWCODEC ready flag 0: ADCRAWCODEC not ready 1: ADCRAWCODEC ready
1	CODEBRDY	RO	0x0	ADCRAWCODEB ready flag 0: ADCRAWCODEB not ready 1: ADCRAWCODEB ready
0	CODEARDY	RO	0x0	ADCRAWCODEA ready flag 0: ADCRAWCODEA not ready 1: ADCRAWCODEA ready

Table 12-150: ADC Status Clear Register (ADCSTSCLR) Layout

ADCSTSCLR (ADC Status Clear Register) Offset: 0x114 Default: 0x00000000 Access: ADC -> ADCSTSCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					CODECRDY	CODEBRDY	CODEARDY

Table 12-151: ADC Status Clear Register (ADCSTSCLR) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
2	CODECRDY	W1C	0x0	ADCRAWCODEC ready flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCSTS[CODECRDY]. This bit is self-cleared.
1	CODEBRDY	W1C	0x0	ADCRAWCODEB ready flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCSTS[CODEBRDY]. This bit is self-cleared.
0	CODEARDY	W1C	0x0	ADCRAWCODEA ready flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCSTS[CODEARDY]. This bit is self-cleared.

Table 12-152: ADC Control Register (ADCCTL) Layout

ADCCTL (ADC Control Register) Offset: 0x118 Default: 0x00000008
Access: ADC -> ADCCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
TIECN		TIECP		TIEBN		TIEBP	
7	6	5	4	3	2	1	0
TIEAN		TIEAP		SYNCEDGE	SYNCEN	RST	EN

Table 12-153: ADC Control Register (ADCCTL) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:14	TIECN	RW	0x0	Force SHC negative input connection 00: Disable 01: Disable 10: Force SHC negative input tied low 11: Force SHC negative input tied high
13:12	TIECP	RW	0x0	Force SHC positive input connection 00: Disable 01: Disable 10: Force SHC 481orrespo input tied low 11: Force SHC 481orrespo input tied high

Bits	Field Name	Type	Reset	Description
11:10	TIEBN	RW	0x0	Force SHB negative input connection 00: Disable 01: Disable 10: Force SHB negative input tied low 11: Force SHB negative input tied high
9:8	TIEBP	RW	0x0	Force SHB positive input connection 00: Disable 01: Disable 10: Force SHB 482orrespo input tied low 11: Force SHB 482orrespo input tied high
7:6	TIEAN	RW	0x0	Force SHA negative input connection 00: Disable 01: Disable 10: Force SHA negative input tied low 11: Force SHA negative input tied high
5:4	TIEAP	RW	0x0	Force SHA positive input connection 00: Disable 01: Disable 10: Force SHA 482orrespo input tied low 11: Force SHA 482orrespo input tied high
3	SYNCEDGE	RW	0x1	Clock edge for sample/convert control re-synchronization 0: Negative edge 1: 482orrespo edge
2	SYNCEN	RW	0x0	Enable sample/convert control re-synchronization with local clean clock 0: Disable 1: Enable
1	RST	W1S	0x0	ADC digital reset 0: Write a 0 has no effect. Always readback 0 1: Write a 1 reset the ADC digital part. This bit is self-cleared
0	EN	RW	0x0	ADC enable 0: Disable ADC 1: Enable ADC

Table 12-154: ADC Bandgap Control Register (ADCBGCTL) Layout

ADCBGCTL (ADC Bandgap Control Register) Offset: 0x11C Default: 0x00000000 Access: ADC -> ADCBGCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							EN

Table 12-155: ADC Bandgap Control Register (ADCBGCTL) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	EN	RW	0x0	ADC bandgap enable 0: Disable ADC bandgap 1: Enable ADC bandgap

Table 12-156: ADC Reference Control Register (ADCREFCTL) Layout

ADCREFCTL (ADC Reference Control Register) Offset: 0x120 Default: 0x00003D3C Access: ADC -> ADCREFCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_15							
23	22	21	20	19	18	17	16
RESERVED_31_15							
15	14	13	12	11	10	9	8
ESERVED_31_1	VDDTRIM				VANATRIM		VDIGTRIM
7	6	5	4	3	2	1	0
VDIGTRIM	VREFTRIM					EXTREF	EN

Table 12-157: ADC Reference Control Register (ADCREFCTL) Description

Bits	Field Name	Type	Reset	Description
31:15	RESERVED_31_15	RO	0x0	Reserved.
14:11	VDDTRIM	RW	0x7	VDD trimming
10:9	VANATRIM	RW	0x2	VANA trimming
8:7	VDIGTRIM	RW	0x2	VDIG trimming
6:2	VREFTRIM	RW	0xF	VREF trimming
1	EXTREF	RW	0x0	External reference enable 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
0	EN	RW	0x0	Reference enable 0: Disable 1: Enable

Table 12-158: ADC SHA Raw Code Register (ADCRAWCODEA) Layout

ADCRAWCODEA (ADC SHA Raw Code Register) Offset: 0x124 Default: 0x00000000 Access: ADC -> ADCRAWCODEA.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-159: ADC SHA Raw Code Register (ADCRAWCODEA) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Raw code (Signed number from -8192 to 8191) for SHA.

Table 12-160: ADC SHB Raw Code Register (ADCRAWCODEB) Layout

ADCRAWCODEB (ADC SHB Raw Code Register) Offset: 0x128 Default: 0x00000000 Access: ADC -> ADCRAWCODEB.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-161: ADC SHB Raw Code Register (ADCRAWCODEB) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Raw code (Signed number from -8192 to 8191) for SHB.

Table 12-162: ADC SHC Raw Code Register (ADCRAWCODEC) Layout

ADCRAWCODEC (ADC SHC Raw Code Register) Offset: 0x12C Default: 0x00000000 Access: ADC -> ADCRAWCODEC.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-163: ADC SHC Raw Code Register (ADCRAWCODEC) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Raw code (Signed number from -8192 to 8191) for SHC.

Table 12-164: ADC Result Register 0 (ADCRESULT0) Layout

ADCRESULT0 (ADC Result Register 0) Offset: 0x130 Default: 0x00000000 Access: ADC -> ADCRESULT0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-165: ADC Result Register 0 (ADCRESULT0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL0. In simultaneous sampling mode enabled by ADCSOCCTL0[SHEN], it is the result of SHA configured in ADCSOCCTL0

Table 12-166: ADC Result Register 1 (ADCRESULT1) Layout

ADCRESULT1 (ADC Result Register 1) Offset: 0x134 Default: 0x00000000 Access: ADC -> ADCRESULT1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-167: ADC Result Register 1 (ADCRESULT1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL1. In simultaneous sampling mode enabled by ADCSOCCTL0[SHBEN], it is the result of SHB configured in ADCSOCCTL1

Table 12-168: ADC Result Register 2 (ADCRESULT2) Layout

ADCRESULT2 (ADC Result Register 2) Offset: 0x138 Default: 0x00000000 Access: ADC -> ADCRESULT2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-169: ADC Result Register 2 (ADCRESULT2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL2. In simultaneous sampling mode enabled by ADCSOCCTL0[SHCEN], it is the result of SHC configured in ADCSOCCTL2

Table 12-170: ADC Result Register 3 (ADCRESULT3) Layout

ADCRESULT3 (ADC Result Register 3) Offset: 0x13C Default: 0x00000000 Access: ADC -> ADCRESULT3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-171: ADC Result Register 3 (ADCRESULT3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL3. In simultaneous sampling mode enabled by ADCSOCCTL3[SHAEN], it is the result of SHA configured in ADCSOCCTL3

Table 12-172: ADC Result Register 4 (ADCRESULT4) Layout

ADCRESULT4 (ADC Result Register 4) Offset: 0x140 Default: 0x00000000 Access: ADC -> ADCRESULT4.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-173: ADC Result Register 4 (ADCRESULT4) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL4. In simultaneous sampling mode enabled by ADCSOCCTL3[SHBEN], it is the result of SHB configured in ADCSOCCTL4

Table 12-174: ADC Result Register 5 (ADCRESULT5) Layout

ADCRESULT5 (ADC Result Register 5) Offset: 0x144 Default: 0x00000000 Access: ADC -> ADCRESULT5.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-175: ADC Result Register 5 (ADCRESULT5) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL5. In simultaneous sampling mode enabled by ADCSOCCTL3[SHCEN], it is the result of SHC configured in ADCSOCCTL5

Table 12-176: ADC Result Register 6 (ADCRESULT6) Layout

ADCRESULT6 (ADC Result Register 6) Offset: 0x148 Default: 0x00000000 Access: ADC -> ADCRESULT6.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-177: ADC Result Register 6 (ADCRESULT6) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL6. In simultaneous sampling mode enabled by ADCSOCCTL6[SHAEN], it is the result of SHA configured in ADCSOCCTL6

Table 12-178: ADC Result Register 7 (ADCRESULT7) Layout

ADCRESULT7 (ADC Result Register 7) Offset: 0x14C Default: 0x00000000 Access: ADC -> ADCRESULT7.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-179: ADC Result Register 7 (ADCRESULT7) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL7. In simultaneous sampling mode enabled by ADCSOCCTL6[SHBEN], it is the result of SHB configured in ADCSOCCTL7

Table 12-180: ADC Result Register 8 (ADCRESULT8) Layout

ADCRESULT8 (ADC Result Register 8) Offset: 0x150 Default: 0x00000000 Access: ADC -> ADCRESULT8.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-181: ADC Result Register 8 (ADCRESULT8) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL8. In simultaneous sampling mode enabled by ADCSOCCTL6[SHCEN], it is the result of SHC configured in ADCSOCCTL8

Table 12-182: ADC Result Register 9 (ADCRESULT9) Layout

ADCRESULT9 (ADC Result Register 9) Offset: 0x154 Default: 0x00000000 Access: ADC -> ADCRESULT9.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-183: ADC Result Register 9 (ADCRESULT9) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL9. In simultaneous sampling mode enabled by ADCSOCCTL9[SHAEN], it is the result of SHA configured in ADCSOCCTL9

Table 12-184: ADC Result Register 10 (ADCRESULT10) Layout

ADCRESULT10 (ADC Result Register 10) Offset: 0x158 Default: 0x00000000 Access: ADC -> ADCRESULT10.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-185: ADC Result Register 10 (ADCRESULT10) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL10. In simultaneous sampling mode enabled by ADCSOCCTL9[SHBEN], it is the result of SHB configured in ADCSOCCTL10

Table 12-186: ADC Result Register 11 (ADCRESULT11) Layout

ADCRESULT11 (ADC Result Register 11) Offset: 0x15C Default: 0x00000000 Access: ADC -> ADCRESULT11.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-187: ADC Result Register 11 (ADCRESULT11) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL11. In simultaneous sampling mode enabled by ADCSOCCTL9[SHCEN], it is the result of SHC configured in ADCSOCCTL11

Table 12-188: ADC Result Register 12 (ADCRESULT12) Layout

ADCRESULT12 (ADC Result Register 12) Offset: 0x160 Default: 0x00000000 Access: ADC -> ADCRESULT12.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-189: ADC Result Register 12 (ADCRESULT12) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL12. In simultaneous sampling mode enabled by ADCSOCCTL12[SHAEN], it is the result of SHA configured in ADCSOCCTL12

Table 12-190: ADC Result Register 13 (ADCRESULT13) Layout

ADCRESULT13 (ADC Result Register 13) Offset: 0x164 Default: 0x00000000 Access: ADC -> ADCRESULT13.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-191: ADC Result Register 13 (ADCRESULT13) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL13. In simultaneous sampling mode enabled by ADCSOCCTL12[SHBEN], it is the result of SHB configured in ADCSOCCTL13

Table 12-192: ADC Result Register 14 (ADCRESULT14) Layout

ADCRESULT14 (ADC Result Register 14) Offset: 0x168 Default: 0x00000000 Access: ADC -> ADCRESULT14.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-193: ADC Result Register 14 (ADCRESULT14) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL14. In simultaneous sampling mode enabled by ADCSOCCTL12[SHCEN], it is the result of SHC configured in ADCSOCCTL14

Table 12-194: ADC Result Register 15 (ADCRESULT15) Layout

ADCRESULT15 (ADC Result Register 15) Offset: 0x16C Default: 0x00000000 Access: ADC -> ADCRESULT15.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-195: ADC Result Register 15 (ADCRESULT15) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	14-bit ADC result with extended sign bit In sequential mode, it is the result of the corresponding SH enabled in ADCSOCCTL15. Simultaneous sample mode is not supported.

Table 12-196: ADCPPU0 Comparison Result Register (ADCPPURESULT0) Layout

ADCPPURESULT0 (ADCPPU0 Comparison Result Register) Offset: 0x170 Default: 0x00000000 Access: ADC -> ADCPPURESULT0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-197: ADCPPU0 Comparison Result Register (ADCPPURESULT0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Comparison result with extended sign bit

Table 12-198: ADCPPU1 Comparison Result Register (ADCPPURESULT1) Layout

ADCPPURESULT1 (ADCPPU1 Comparison Result Register) Offset: 0x174 Default: 0x00000000 Access: ADC -> ADCPPURESULT1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-199: ADCPPU1 Comparison Result Register (ADCPPURESULT1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Comparison result with extended sign bit

Table 12-200: ADCPPU2 Comparison Result Register (ADCPPURESULT2) Layout

ADCPPURESULT2 (ADCPPU2 Comparison Result Register) Offset: 0x178 Default: 0x00000000 Access: ADC -> ADCPPURESULT2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-201: ADCPPU2 Comparison Result Register (ADCPPURESULT2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Comparison result with extended sign bit

Table 12-202: ADCPPU3 Comparison Result Register (ADCPPURESULT3) Layout

ADCPPURESULT3 (ADCPPU3 Comparison Result Register) Offset: 0x17C Default: 0x00000000 Access: ADC -> ADCPPURESULT3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-203: ADCPPU3 Comparison Result Register (ADCPPURESULT3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Comparison result with extended sign bit

Table 12-204: ADCPPU4 Comparison Result Register (ADCPPURESULT4) Layout

ADCPPURESULT4 (ADCPPU4 Comparison Result Register) Offset: 0x180 Default: 0x00000000 Access: ADC -> ADCPPURESULT4.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-205: ADCPPU4 Comparison Result Register (ADCPPURESULT4) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Comparison result with extended sign bit

Table 12-206: ADCPPU5 Comparison Result Register (ADCPPURESULT5) Layout

ADCPPURESULT5 (ADCPPU5 Comparison Result Register) Offset: 0x184 Default: 0x00000000 Access: ADC -> ADCPPURESULT5.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-207: ADCPPU5 Comparison Result Register (ADCPPURESULT5) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Comparison result with extended sign bit

Table 12-208: ADCPPU0 SOC Delay Register (ADCPPUSOCDLY0) Layout

ADCPPUSOCDLY0 (ADCPPU0 SOC Delay Register) Offset: 0x190 Default: 0x00000000 Access: ADC -> ADCPPUSOCDLY0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-209: ADCPPU0 SOC Delay Register (ADCPPUSOCDLY0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Delay from trigger to execution for SOC selected by ADCPPU0CTL[SOCSEL]

Table 12-210: ADCPPU1 SOC Delay Register (ADCPPUSOCDLY1) Layout

ADCPPUSOCDLY1 (ADCPPU1 SOC Delay Register) Offset: 0x194 Default: 0x00000000 Access: ADC -> ADCPPUSOCDLY1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-211: ADCPPU1 SOC Delay Register (ADCPPUSOCDLY1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Delay from trigger to execution for SOC selected by ADCPPU1CTL[SOCSEL]

Table 12-212: ADCPPU2 SOC Delay Register (ADCPPUSOCDLY2) Layout

ADCPPUSOCDLY2 (ADCPPU2 SOC Delay Register) Offset: 0x198 Default: 0x00000000 Access: ADC -> ADCPPUSOCDLY2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-213: ADCPPU2 SOC Delay Register (ADCPPUSOCDLY2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Delay from trigger to execution for SOC selected by ADCPPU2CTL[SOCSEL]

Table 12-214: ADCPPU3 SOC Delay Register (ADCPPUSOCDLY3) Layout

ADCPPUSOCDLY3 (ADCPPU3 SOC Delay Register) Offset: 0x19C Default: 0x00000000 Access: ADC -> ADCPPUSOCDLY3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-215: ADCPPU3 SOC Delay Register (ADCPPUSOCDLY3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Delay from trigger to execution for SOC selected by ADCPPU3CTL[SOCSEL]

Table 12-216: ADCPPU4 SOC Delay Register (ADCPPUSOCDLY4) Layout

ADCPPUSOCDLY4 (ADCPPU4 SOC Delay Register) Offset: 0x1A0 Default: 0x00000000 Access: ADC -> ADCPPUSOCDLY4.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-217: ADCPPU4 SOC Delay Register (ADCPPUSOCDLY4) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Delay from trigger to execution for SOC selected by ADCPPU4CTL[SOCSEL]

Table 12-218: ADCPPU5 SOC Delay Register (ADCPPUSOCDLY5) Layout

ADCPPUSOCDLY5 (ADCPPU5 SOC Delay Register) Offset: 0x1A4 Default: 0x00000000 Access: ADC -> ADCPPUSOCDLY5.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-219: ADCPPU5 SOC Delay Register (ADCPPUSOCDLY5) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	Delay from trigger to execution for SOC selected by ADCPPU5CTL[SOCSEL]

Table 12-220: ADCPPU0 Interrupt Flag Register (ADCPPUIF0) Layout

ADCPPUIF0 (ADCPPU0 Interrupt Flag Register) Offset: 0x1B0 Default: 0x00000000 Access: ADC -> ADCPPUIF0.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-221: ADCPPU0 Interrupt Flag Register (ADCPPUIF0) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	RO	0x0	ADCPPU0 interrupt flag 0: ADCPPU0 interrupt does not happen 1: ADCPPU0 interrupt happened
2	XZRO	RO	0x0	Latched ADCPPU0 zero-cross status 0: ADCPPU0 zero-cross does not happen 1: ADCPPU0 zero-cross happened
1	TZHI	RO	0x0	Latched ADCPPU0 too-high status 0: ADCPPU0 too-high does not happen 1: ADCPPU0 too-high happened
0	TZLO	RO	0x0	Latched ADCPPU0 too-low status 0: ADCPPU0 too-low does not happen 1: ADCPPU0 too-low happened

Table 12-222: ADCPPU1 Interrupt Flag Register (ADCPPUIF1) Layout

ADCPPUIF1 (ADCPPU1 Interrupt Flag Register) Offset: 0x1B4 Default: 0x00000000 Access: ADC -> ADCPPUIF1.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-223: ADCPPU1 Interrupt Flag Register (ADCPPUIF1) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
3	INT	RO	0x0	ADCPPU1 interrupt flag 0: ADCPPU1 interrupt does not happen 1: ADCPPU1 interrupt happened
2	XZRO	RO	0x0	Latched ADCPPU1 zero-cross status 0: ADCPPU1 zero-cross does not happen 1: ADCPPU1 zero-cross happened
1	TZHI	RO	0x0	Latched ADCPPU1 too-high status 0: ADCPPU1 too-high does not happen 1: ADCPPU1 too-high happened
0	TZLO	RO	0x0	Latched ADCPPU1 too-low status 0: ADCPPU1 too-low does not happen 1: ADCPPU1 too-low happened

Table 12-224: ADCPPU2 Interrupt Flag Register (ADCPPUIF2) Layout

ADCPPUIF2 (ADCPPU2 Interrupt Flag Register) Offset: 0x1B8 Default: 0x00000000
Access: ADC -> ADCPPUIF2.all

31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-225: ADCPPU2 Interrupt Flag Register (ADCPPUIF2) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	RO	0x0	ADCPPU2 interrupt flag 0: ADCPPU2 interrupt does not happen 1: ADCPPU2 interrupt happened
2	XZRO	RO	0x0	Latched ADCPPU2 zero-cross status 0: ADCPPU2 zero-cross does not happen 1: ADCPPU2 zero-cross happened
1	TZHI	RO	0x0	Latched ADCPPU2 too-high status 0: ADCPPU2 too-high does not happen 1: ADCPPU2 too-high happened
0	TZLO	RO	0x0	Latched ADCPPU2 too-low status 0: ADCPPU2 too-low does not happen 1: ADCPPU2 too-low happened

Table 12-226: ADCPPU3 Interrupt Flag Register (ADCPPUIF3) Layout

ADCPPUIF3 (ADCPPU3 Interrupt Flag Register) Offset: 0x1BC Default: 0x00000000 Access: ADC -> ADCPPUIF3.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-227: ADCPPU3 Interrupt Flag Register (ADCPPUIF3) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	RO	0x0	ADCPPU3 interrupt flag 0: ADCPPU3 interrupt does not happen 1: ADCPPU3 interrupt happened
2	XZRO	RO	0x0	Latched ADCPPU3 zero-cross status 0: ADCPPU3 zero-cross does not happen 1: ADCPPU3 zero-cross happened
1	TZHI	RO	0x0	Latched ADCPPU3 too-high status 0: ADCPPU3 too-high does not happen 1: ADCPPU3 too-high happened
0	TZLO	RO	0x0	Latched ADCPPU3 too-low status 0: ADCPPU3 too-low does not happen 1: ADCPPU3 too-low happened

Table 12-228: ADCPPU4 Interrupt Flag Register (ADCPPUIF4) Layout

ADCPPUIF4 (ADCPPU4 Interrupt Flag Register) Offset: 0x1C0 Default: 0x00000000
Access: ADC -> ADCPPUIF4.all

31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-229: ADCPPU4 Interrupt Flag Register (ADCPPUIF4) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	RO	0x0	ADCPPU4 interrupt flag 0: ADCPPU4 interrupt does not happen 1: ADCPPU4 interrupt happened
2	XZRO	RO	0x0	Latched ADCPPU4 zero-cross status 0: ADCPPU4 zero-cross does not happen 1: ADCPPU4 zero-cross happened
1	TZHI	RO	0x0	Latched ADCPPU4 too-high status 0: ADCPPU4 too-high does not happen 1: ADCPPU4 too-high happened

Bits	Field Name	Type	Reset	Description
0	TZLO	RO	0x0	Latched ADCPPU4 too-low status 0: ADCPPU4 too-low does not happen 1: ADCPPU4 too-low happened

Table 12-230: ADCPPU5 Interrupt Flag Register (ADCPPUIF5) Layout

ADCPPUIF5 (ADCPPU5 Interrupt Flag Register) Offset: 0x1C4 Default: 0x00000000 Access: ADC -> ADCPPUIF5.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-231: ADCPPU5 Interrupt Flag Register (ADCPPUIF5) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	RO	0x0	ADCPPU5 interrupt flag 0: ADCPPU5 interrupt does not happen 1: ADCPPU5 interrupt happened
2	XZRO	RO	0x0	Latched ADCPPU5 zero-cross status 0: ADCPPU5 zero-cross does not happen 1: ADCPPU5 zero-cross happened
1	TZHI	RO	0x0	Latched ADCPPU5 too-high status 0: ADCPPU5 too-high does not happen 1: ADCPPU5 too-high happened
0	TZLO	RO	0x0	Latched ADCPPU5 too-low status 0: ADCPPU5 too-low does not happen 1: ADCPPU5 too-low happened

Table 12-232: ADCPPU0 Interrupt Clear Register (ADCPPUIC0) Layout

ADCPPUIC0 (ADCPPU0 Interrupt Clear Register) Offset: 0x1D0 Default: 0x00000000 Access: ADC -> ADCPPUIC0.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-233: ADCPPU0 Interrupt Clear Register (ADCPPUIC0) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	W1C	0x0	ADCPPU0 interrupt flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU0IF[INT]. This bit is self-cleared.
2	XZRO	W1C	0x0	Latched ADCPPU0 zero-cross status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU0IF[XZRO]. This bit is self-cleared.
1	TZHI	W1C	0x0	Latched ADCPPU0 too-high status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU0IF[TZHI]. This bit is self-cleared.
0	TZLO	W1C	0x0	Latched ADCPPU0 too-low status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU0IF[TZLO]. This bit is self-cleared.

Table 12-234: ADCPPU1 Interrupt Clear Register (ADCPPUIC1) Layout

ADCPPUIC1 (ADCPPU1 Interrupt Clear Register) Offset: 0x1D4 Default: 0x00000000 Access: ADC -> ADCPPUIC1.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-235: ADCPPU1 Interrupt Clear Register (ADCPPUIC1) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	W1C	0x0	ADCPPU1 interrupt flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU1IF[INT]. This bit is self-cleared.
2	XZRO	W1C	0x0	Latched ADCPPU1 zero-cross status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU1IF[XZRO]. This bit is self-cleared.
1	TZHI	W1C	0x0	Latched ADCPPU1 too-high status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU1IF[TZHI]. This bit is self-cleared.
0	TZLO	W1C	0x0	Latched ADCPPU1 too-low status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU1IF[TZLO]. This bit is self-cleared.

Table 12-236: ADCPPU2 Interrupt Clear Register (ADCPPUIC2) Layout

ADCPPUIC2 (ADCPPU2 Interrupt Clear Register) Offset: 0x1D8 Default: 0x00000000
Access: ADC -> ADCPPUIC2.all

31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-237: ADCPPU2 Interrupt Clear Register (ADCPPUIC2) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	W1C	0x0	ADCPPU2 interrupt flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU2IF[INT]. This bit is self-cleared.
2	XZRO	W1C	0x0	Latched ADCPPU2 zero-cross status clear 0: Write a 0 has no effect. Always readback 0

Bits	Field Name	Type	Reset	Description
				1: Write a 1 clears ADCPPU2IF[XZRO]. This bit is self-cleared.
1	TZHI	W1C	0x0	Latched ADCPPU2 too-high status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU2IF[TZHI]. This bit is self-cleared.
0	TZLO	W1C	0x0	Latched ADCPPU2 too-low status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU2IF[TZLO]. This bit is self-cleared.

Table 12-238: ADCPPU3 Interrupt Clear Register (ADCPPUIC3) Layout

ADCPPUIC3 (ADCPPU3 Interrupt Clear Register) Offset: 0x1DC Default: 0x00000000 Access: ADC -> ADCPPUIC3.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-239: ADCPPU3 Interrupt Clear Register (ADCPPUIC3) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	W1C	0x0	ADCPPU3 interrupt flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU3IF[INT]. This bit is self-cleared.
2	XZRO	W1C	0x0	Latched ADCPPU3 zero-cross status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU3IF[XZRO]. This bit is self-cleared.
1	TZHI	W1C	0x0	Latched ADCPPU3 too-high status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU3IF[TZHI]. This bit is self-cleared.
0	TZLO	W1C	0x0	Latched ADCPPU3 too-low status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU3IF[TZLO]. This bit is self-cleared.

Table 12-240: ADCPPU4 Interrupt Clear Register (ADCPPUIC4) Layout

ADCPPUIC4 (ADCPPU4 Interrupt Clear Register) Offset: 0x1E0 Default: 0x00000000 Access: ADC -> ADCPPUIC4.all							
31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-241: ADCPPU4 Interrupt Clear Register (ADCPPUIC4) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	W1C	0x0	ADCPPU4 interrupt flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU4IF[INT]. This bit is self-cleared.
2	XZRO	W1C	0x0	Latched ADCPPU4 zero-cross status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU4IF[XZRO]. This bit is self-cleared.
1	TZHI	W1C	0x0	Latched ADCPPU4 too-high status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU4IF[TZHI]. This bit is self-cleared.
0	TZLO	W1C	0x0	Latched ADCPPU4 too-low status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU4IF[TZLO]. This bit is self-cleared.

Table 12-242: ADCPPU5 Interrupt Clear Register (ADCPPUIC5) Layout

ADCPPUIC5 (ADCPPU5 Interrupt Clear Register) Offset: 0x1E4 Default: 0x00000000
Access: ADC -> ADCPPUIC5.all

31	30	29	28	27	26	25	24
RESERVED_31_4							
23	22	21	20	19	18	17	16
RESERVED_31_4							
15	14	13	12	11	10	9	8
RESERVED_31_4							
7	6	5	4	3	2	1	0
RESERVED_31_4				INT	XZRO	TZHI	TZLO

Table 12-243: ADCPPU5 Interrupt Clear Register (ADCPPUIC5) Description

Bits	Field Name	Type	Reset	Description
31:4	RESERVED_31_4	RO	0x0	Reserved.
3	INT	W1C	0x0	ADCPPU5 interrupt flag clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU5IF[INT]. This bit is self-cleared.
2	XZRO	W1C	0x0	Latched ADCPPU5 zero-cross status clear 0: Write a 0 has no effect. Always readback 0

Bits	Field Name	Type	Reset	Description
				1: Write a 1 clears ADCPPU5IF[XZRO]. This bit is self-cleared.
1	TZHI	W1C	0x0	Latched ADCPPU5 too-high status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU5IF[TZHI]. This bit is self-cleared.
0	TZLO	W1C	0x0	Latched ADCPPU5 too-low status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears ADCPPU5IF[TZLO]. This bit is self-cleared.

Table 12-244: ADCPPU0 Interrupt Enable Register (ADCPPUIE0) Layout

ADCPPUIE0 (ADCPPU0 Interrupt Enable Register) Offset: 0x1F0 Default: 0x00000000
Access: ADC -> ADCPPUIE0.all

31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-245: ADCPPU0 Interrupt Enable Register (ADCPPUIE0) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU0 zero-cross as ADCPPU0 interrupt source 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU0 too-high as ADCPPU0 interrupt source 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU0 too-low as ADCPPU0 interrupt source 0: Disable 1: Enable

Table 12-246: ADCPPU1 Interrupt Enable Register (ADCPPUIE1) Layout

ADCPPUIE1 (ADCPPU1 Interrupt Enable Register) Offset: 0x1F4 Default: 0x00000000 Access: ADC -> ADCPPUIE1.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-247: ADCPPU1 Interrupt Enable Register (ADCPPUIE1) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU1 zero-cross as ADCPPU1 interrupt source 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU1 too-high as ADCPPU1 interrupt source 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU1 too-low as ADCPPU1 interrupt source 0: Disable 1: Enable

Table 12-248: ADCPPU2 Interrupt Enable Register (ADCPPUIE2) Layout

ADCPPUIE2 (ADCPPU2 Interrupt Enable Register) Offset: 0x1F8 Default: 0x00000000 Access: ADC -> ADCPPUIE2.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-249: ADCPPU2 Interrupt Enable Register (ADCPPUIE2) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
2	XZRO	RW	0x0	Enable ADCPPU2 zero-cross as ADCPPU2 interrupt source 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU2 too-high as ADCPPU2 interrupt source 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU2 too-low as ADCPPU2 interrupt source 0: Disable 1: Enable

Table 12-250: ADCPPU3 Interrupt Enable Register (ADCPPUIE3) Layout

ADCPPUIE3 (ADCPPU3 Interrupt Enable Register) Offset: 0x1FC Default: 0x00000000 Access: ADC -> ADCPPUIE3.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-251: ADCPPU3 Interrupt Enable Register (ADCPPUIE3) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU3 zero-cross as ADCPPU3 interrupt source 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU3 too-high as ADCPPU3 interrupt source 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU3 too-low as ADCPPU3 interrupt source 0: Disable 1: Enable

Table 12-252: ADCPPU4 Interrupt Enable Register (ADCPPUIE4) Layout

ADCPPUIE4 (ADCPPU4 Interrupt Enable Register) Offset: 0x200 Default: 0x00000000 Access: ADC -> ADCPPUIE4.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-253: ADCPPU4 Interrupt Enable Register (ADCPPUIE4) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU4 zero-cross as ADCPPU4 interrupt source 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU4 too-high as ADCPPU4 interrupt source 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU4 too-low as ADCPPU4 interrupt source 0: Disable 1: Enable

Table 12-254: ADCPPU5 Interrupt Enable Register (ADCPPUIE5) Layout

ADCPPUIE5 (ADCPPU5 Interrupt Enable Register) Offset: 0x204 Default: 0x00000000 Access: ADC -> ADCPPUIE5.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-255: ADCPPU5 Interrupt Enable Register (ADCPPUIE5) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
2	XZRO	RW	0x0	Enable ADCPPU5 zero-cross as ADCPPU5 interrupt source 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU5 too-high as ADCPPU5 interrupt source 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU5 too-low as ADCPPU5 interrupt source 0: Disable 1: Enable

Table 12-256: ADCPPU0 Trip-Zone Event Enable Register (ADCPPUTZE0) Layout

ADCPPUTZE0 (ADCPPU0 Trip-Zone Event Enable Register) Offset: 0x210 Default: 0x00000000
Access: ADC -> ADCPPUTZE0.all

31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-257: ADCPPU0 Trip-Zone Event Enable Register (ADCPPUTZE0) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU0 zero-cross as ADCPPU0 trip-zone event 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU0 too-high as ADCPPU0 trip-zone event 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU0 too-low as ADCPPU0 trip-zone event 0: Disable 1: Enable

Table 12-258: ADCPPU1 Trip-Zone Event Enable Register (ADCPPUTZE1) Layout

ADCPPUTZE1 (ADCPPU1 Trip-Zone Event Enable Register) Offset: 0x214 Default: 0x00000000 Access: ADC -> ADCPPUTZE1.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-259: ADCPPU1 Trip-Zone Event Enable Register (ADCPPUTZE1) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU1 zero-cross as ADCPPU1 trip-zone event 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU1 too-high as ADCPPU1 trip-zone event 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU1 too-low as ADCPPU1 trip-zone event 0: Disable 1: Enable

Table 12-260: ADCPPU2 Trip-Zone Event Enable Register (ADCPPUTZE2) Layout

ADCPPUTZE2 (ADCPPU2 Trip-Zone Event Enable Register) Offset: 0x218 Default: 0x00000000 Access: ADC -> ADCPPUTZE2.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-261: ADCPPU2 Trip-Zone Event Enable Register (ADCPPUTZE2) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
2	XZRO	RW	0x0	Enable ADCPPU2 zero-cross as ADCPPU2 trip-zone event 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU2 too-high as ADCPPU2 trip-zone event 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU2 too-low as ADCPPU2 trip-zone event 0: Disable 1: Enable

Table 12-262: ADCPPU3 Trip-Zone Event Enable Register (ADCPPUTZE3) Layout

ADCPPUTZE3 (ADCPPU3 Trip-Zone Event Enable Register) Offset: 0x21C Default: 0x00000000 Access: ADC -> ADCPPUTZE3.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-263: ADCPPU3 Trip-Zone Event Enable Register (ADCPPUTZE3) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU3 zero-cross as ADCPPU3 trip-zone event 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU3 too-high as ADCPPU3 trip-zone event 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU3 too-low as ADCPPU3 trip-zone event 0: Disable 1: Enable

Table 12-264: ADCPPU4 Trip-Zone Event Enable Register (ADCPPUTZE4) Layout

ADCPPUTZE4 (ADCPPU4 Trip-Zone Event Enable Register) Offset: 0x220 Default: 0x00000000 Access: ADC -> ADCPPUTZE4.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-265: ADCPPU4 Trip-Zone Event Enable Register (ADCPPUTZE4) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU4 zero-cross as ADCPPU4 trip-zone event 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU4 too-high as ADCPPU4 trip-zone event 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU4 too-low as ADCPPU4 trip-zone event 0: Disable 1: Enable

Table 12-266: ADCPPU5 Trip-Zone Event Enable Register (ADCPPUTZE5) Layout

ADCPPUTZE5 (ADCPPU5 Trip-Zone Event Enable Register) Offset: 0x224 Default: 0x00000000 Access: ADC -> ADCPPUTZE5.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					XZRO	TZHI	TZLO

Table 12-267: ADCPPU5 Trip-Zone Event Enable Register (ADCPPUTZE5) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	XZRO	RW	0x0	Enable ADCPPU5 zero-cross as ADCPPU5 trip-zone event 0: Disable 1: Enable
1	TZHI	RW	0x0	Enable ADCPPU5 too-high as ADCPPU5 trip-zone event 0: Disable 1: Enable
0	TZLO	RW	0x0	Enable ADCPPU5 too-low as ADCPPU5 trip-zone event 0: Disable 1: Enable

Table 12-268: ADCPPU0 Control Register (ADCPPUCTL0) Layout

ADCPPUCTL0 (ADCPPU0 Control Register) Offset: 0x230 Default: 0x00000400
Access: ADC -> ADCPPUCTL0.all

31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					POL	DATASEL	
7	6	5	4	3	2	1	0
DATASEL		SOCSEL				CBCEN	EN

Table 12-269: ADCPPU0 Control Register (ADCPPUCTL0) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	POL	RW	0x1	Polarity for comparison 0: ADCPPU0 result is the reference minus the selected result 1: ADCPPU0 result is the selected result minus the reference
9:6	DATASEL	RW	0x0	Select the ADC result to compare with the reference
5:2	SOCSEL	RW	0x0	Select the SOC signal to derive the ADCPPU0SOCDLY

Bits	Field Name	Type	Reset	Description
1	CBCEN	RW	0x0	ADCPPU0 cycle-by-cycle clear enable 0: Disable ADCPPU0 cycle-by-cycle clear 1: Enable ADCPPU0 cycle-by-cycle clear
0	EN	RW	0x0	ADCPPU0 enable 0: Disable ADCPPU0 1: Enable ADCPPU0

Table 12-270: ADCPPU1 Control Register (ADCPPUCTL1) Layout

ADCPPUCTL1 (ADCPPU1 Control Register) Offset: 0x234 Default: 0x00000400
Access: ADC -> ADCPPUCTL1.all

31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					POL	DATASEL	
7	6	5	4	3	2	1	0
DATASEL		SOCSEL				CBCEN	EN

Table 12-271: ADCPPU1 Control Register (ADCPPUCTL1) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	POL	RW	0x1	Polarity for comparison 0: ADCPPU1 result is the reference minus the selected result 1: ADCPPU1 result is the selected result minus the reference
9:6	DATASEL	RW	0x0	Select the ADC result to compare with the reference
5:2	SOCSEL	RW	0x0	Select the SOC signal to derive the ADCPPU0SOCDLY
1	CBCEN	RW	0x0	ADCPPU1 cycle-by-cycle clear enable 0: Disable ADCPPU1 cycle-by-cycle clear 1: Enable ADCPPU1 cycle-by-cycle clear
0	EN	RW	0x0	ADCPPU1 enable 0: Disable ADCPPU1 1: Enable ADCPPU1

Table 12-272: ADCPPU2 Control Register (ADCPPUCTL2) Layout

ADCPPUCTL2 (ADCPPU2 Control Register) Offset: 0x238 Default: 0x00000400 Access: ADC -> ADCPPUCTL2.all							
31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					POL	DATASEL	
7	6	5	4	3	2	1	0
DATASEL		SOCSEL				CBCEN	EN

Table 12-273: ADCPPU2 Control Register (ADCPPUCTL2) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	POL	RW	0x1	Polarity for comparison 0: ADCPPU2 result is the reference minus the selected result 1: ADCPPU2 result is the selected result minus the reference
9:6	DATASEL	RW	0x0	Select the ADC result to compare with the reference
5:2	SOCSEL	RW	0x0	Select the SOC signal to derive the ADCPPU0SOCDLY
1	CBCEN	RW	0x0	ADCPPU2 cycle-by-cycle clear enable 0: Disable ADCPPU2 cycle-by-cycle clear 1: Enable ADCPPU2 cycle-by-cycle clear
0	EN	RW	0x0	ADCPPU2 enable 0: Disable ADCPPU2 1: Enable ADCPPU2

Table 12-274: ADCPPU3 Control Register (ADCPPUCTL3) Layout

ADCPPUCTL3 (ADCPPU3 Control Register) Offset: 0x23C Default: 0x00000400 Access: ADC -> ADCPPUCTL3.all							
31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					POL	DATASEL	
7	6	5	4	3	2	1	0
DATASEL		SOCSEL				CBCEN	EN

Table 12-275: ADCPPU3 Control Register (ADCPPUCTL3) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	POL	RW	0x1	Polarity for comparison 0: ADCPPU3 result is the reference minus the selected result 1: ADCPPU3 result is the selected result minus the reference
9:6	DATASEL	RW	0x0	Select the ADC result to compare with the reference
5:2	SOCSEL	RW	0x0	Select the SOC signal to derive the ADCPPU0SOCDLY
1	CBCEN	RW	0x0	ADCPPU3 cycle-by-cycle clear enable 0: Disable ADCPPU3 cycle-by-cycle clear 1: Enable ADCPPU3 cycle-by-cycle clear
0	EN	RW	0x0	ADCPPU3 enable 0: Disable ADCPPU3 1: Enable ADCPPU3

Table 12-276: ADCPPU4 Control Register (ADCPPUCTL4) Layout

ADCPPUCTL4 (ADCPPU4 Control Register) Offset: 0x240 Default: 0x00000400
Access: ADC -> ADCPPUCTL4.all

31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					POL	DATASEL	
7	6	5	4	3	2	1	0
DATASEL		SOCSEL				CBCEN	EN

Table 12-277: ADCPPU4 Control Register (ADCPPUCTL4) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	POL	RW	0x1	Polarity for comparison 0: ADCPPU4 result is the reference minus the selected result 1: ADCPPU4 result is the selected result minus the reference
9:6	DATASEL	RW	0x0	Select the ADC result to compare with the reference

Bits	Field Name	Type	Reset	Description
5:2	SOCSEL	RW	0x0	Select the SOC signal to derive the ADCPPU0SOCDLY
1	CBCEN	RW	0x0	ADCPPU4 cycle-by-cycle clear enable 0: Disable ADCPPU4 cycle-by-cycle clear 1: Enable ADCPPU4 cycle-by-cycle clear
0	EN	RW	0x0	ADCPPU4 enable 0: Disable ADCPPU4 1: Enable ADCPPU4

Table 12-278: ADCPPU5 Control Register (ADCPPUCTL5) Layout

ADCPPUCTL5 (ADCPPU5 Control Register) Offset: 0x244 Default: 0x00000400
Access: ADC -> ADCPPUCTL5.all

31	30	29	28	27	26	25	24
RESERVED_31_11							
23	22	21	20	19	18	17	16
RESERVED_31_11							
15	14	13	12	11	10	9	8
RESERVED_31_11					POL	DATASEL	
7	6	5	4	3	2	1	0
DATASEL		SOCSEL				CBCEN	EN

Table 12-279: ADCPPU5 Control Register (ADCPPUCTL5) Description

Bits	Field Name	Type	Reset	Description
31:11	RESERVED_31_11	RO	0x0	Reserved.
10	POL	RW	0x1	Polarity for comparison 0: ADCPPU5 result is the reference minus the selected result 1: ADCPPU5 result is the selected result minus the reference
9:6	DATASEL	RW	0x0	Select the ADC result to compare with the reference
5:2	SOCSEL	RW	0x0	Select the SOC signal to derive the ADCPPU0SOCDLY
1	CBCEN	RW	0x0	ADCPPU5 cycle-by-cycle clear enable 0: Disable ADCPPU5 cycle-by-cycle clear 1: Enable ADCPPU5 cycle-by-cycle clear
0	EN	RW	0x0	ADCPPU5 enable 0: Disable ADCPPU5 1: Enable ADCPPU5

Table 12-280: ADCPPU0 Reference Register (ADCPPUREF0) Layout

ADCPPUREF0 (ADCPPU0 Reference Register) Offset: 0x250 Default: 0x00000000 Access: ADC -> ADCPPUREF0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-281: ADCPPU0 Reference Register (ADCPPUREF0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Reference with extended sign bit Should be within [-8192, 8191]

Table 12-282: ADCPPU1 Reference Register (ADCPPUREF1) Layout

ADCPPUREF1 (ADCPPU1 Reference Register) Offset: 0x254 Default: 0x00000000 Access: ADC -> ADCPPUREF1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-283: ADCPPU1 Reference Register (ADCPPUREF1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Reference with extended sign bit Should be within [-8192, 8191]

Table 12-284: ADCPPU2 Reference Register (ADCPPUREF2) Layout

ADCPPUREF2 (ADCPPU2 Reference Register) Offset: 0x258 Default: 0x00000000 Access: ADC -> ADCPPUREF2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-285: ADCPPU2 Reference Register (ADCPPUREF2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Reference with extended sign bit Should be within [-8192, 8191]

Table 12-286: ADCPPU3 Reference Register (ADCPPUREF3) Layout

ADCPPUREF3 (ADCPPU3 Reference Register) Offset: 0x25C Default: 0x00000000 Access: ADC -> ADCPPUREF3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-287: ADCPPU3 Reference Register (ADCPPUREF3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Reference with extended sign bit Should be within [-8192, 8191]

Table 12-288: ADCPPU4 Reference Register (ADCPPUREF4) Layout

ADCPPUREF4 (ADCPPU4 Reference Register) Offset: 0x260 Default: 0x00000000 Access: ADC -> ADCPPUREF4.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-289: ADCPPU4 Reference Register (ADCPPUREF4) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Reference with extended sign bit Should be within [-8192, 8191]

Table 12-290: ADCPPU5 Reference Register (ADCPPUREF5) Layout

ADCPPUREF5 (ADCPPU5 Reference Register) Offset: 0x264 Default: 0x00000000 Access: ADC -> ADCPPUREF5.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-291: ADCPPU5 Reference Register (ADCPPUREF5) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Reference with extended sign bit Should be within [-8192, 8191]

Table 12-292: ADCPPU0 Trip-Zone High-Side Threshold Register (ADCPPUTHH0) Layout

ADCPPUTHH0 (ADCPPU0 Trip-Zone High-Side Threshold Register) Offset: 0x270 Default: 0x00000000 Access: ADC -> ADCPPUTHH0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-293: ADCPPU0 Trip-Zone High-Side Threshold Register (ADCPPUTHH0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-294: ADCPPU1 Trip-Zone High-Side Threshold Register (ADCPPUTHH1) Layout

ADCPPUTHH1 (ADCPPU1 Trip-Zone High-Side Threshold Register) Offset: 0x274 Default: 0x00000000 Access: ADC -> ADCPPUTHH1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-295: ADCPPU1 Trip-Zone High-Side Threshold Register (ADCPPUTHH1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-296: ADCPPU2 Trip-Zone High-Side Threshold Register (ADCPPUTHH2) Layout

ADCPPUTHH2 (ADCPPU2 Trip-Zone High-Side Threshold Register) Offset: 0x278 Default: 0x00000000 Access: ADC -> ADCPPUTHH2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-297: ADCPPU2 Trip-Zone High-Side Threshold Register (ADCPPUTHH2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-298: ADCPPU3 Trip-Zone High-Side Threshold Register (ADCPPUTHH3) Layout

ADCPPUTHH3 (ADCPPU3 Trip-Zone High-Side Threshold Register) Offset: 0x27C Default: 0x00000000 Access: ADC -> ADCPPUTHH3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-299: ADCPPU3 Trip-Zone High-Side Threshold Register (ADCPPUTHH3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-300: ADCPPU4 Trip-Zone High-Side Threshold Register (ADCPPUTHH4) Layout

ADCPPUTHH4 (ADCPPU4 Trip-Zone High-Side Threshold Register) Offset: 0x280 Default: 0x00000000 Access: ADC -> ADCPPUTHH4.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-301: ADCPPU4 Trip-Zone High-Side Threshold Register (ADCPPUTHH4) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-302: ADCPPU5 Trip-Zone High-Side Threshold Register (ADCPPUTHH5) Layout

ADCPPUTHH5 (ADCPPU5 Trip-Zone High-Side Threshold Register) Offset: 0x284 Default: 0x00000000 Access: ADC -> ADCPPUTHH5.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-303: ADCPPU5 Trip-Zone High-Side Threshold Register (ADCPPUTHH5) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-304: ADCPPU0 Trip-Zone Low-Side Threshold Register (ADCPPUTHL0) Layout

ADCPPUTHL0 (ADCPPU0 Trip-Zone Low-Side Threshold Register) Offset: 0x290 Default: 0x00000000 Access: ADC -> ADCPPUTHL0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-305: ADCPPU0 Trip-Zone Low-Side Threshold Register (ADCPPUTHL0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-306: ADCPPU1 Trip-Zone Low-Side Threshold Register (ADCPPUTHL1) Layout

ADCPPUTHL1 (ADCPPU1 Trip-Zone Low-Side Threshold Register) Offset: 0x294 Default: 0x00000000 Access: ADC -> ADCPPUTHL1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-307: ADCPPU1 Trip-Zone Low-Side Threshold Register (ADCPPUTHL1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-308: ADCPPU2 Trip-Zone Low-Side Threshold Register (ADCPPUTHL2) Layout

ADCPPUTHL2 (ADCPPU2 Trip-Zone Low-Side Threshold Register) Offset: 0x298 Default: 0x00000000 Access: ADC -> ADCPPUTHL2.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-309: ADCPPU2 Trip-Zone Low-Side Threshold Register (ADCPPUTHL2) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-310: ADCPPU3 Trip-Zone Low-Side Threshold Register (ADCPPUTHL3) Layout

ADCPPUTHL3 (ADCPPU3 Trip-Zone Low-Side Threshold Register) Offset: 0x29C Default: 0x00000000 Access: ADC -> ADCPPUTHL3.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-311: ADCPPU3 Trip-Zone Low-Side Threshold Register (ADCPPUTHL3) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-312: ADCPPU4 Trip-Zone Low-Side Threshold Register (ADCPPUTHL4) Layout

ADCPPUTHL4 (ADCPPU4 Trip-Zone Low-Side Threshold Register) Offset: 0x2A0 Default: 0x00000000 Access: ADC -> ADCPPUTHL4.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-313: ADCPPU4 Trip-Zone Low-Side Threshold Register (ADCPPUTHL4) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-314: ADCPPU5 Trip-Zone Low-Side Threshold Register (ADCPPUTHL5) Layout

ADCPPUTHL5 (ADCPPU5 Trip-Zone Low-Side Threshold Register) Offset: 0x2A4 Default: 0x00000000 Access: ADC -> ADCPPUTHL5.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 12-315: ADCPPU5 Trip-Zone Low-Side Threshold Register (ADCPPUTHL5) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Signed trip-zone threshold within [-16384, 16383]

Table 12-316: Temperature Sensor Control Register (TSENSCTL) Layout

TSENSCTL (Temperature Sensor Control Register) Offset: 0x2B0 Default: 0x00000000 Access: ADC -> TSENSCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	DEMSEL				SWAPBJT	OUTINV	EN

Table 12-317: Temperature Sensor Control Register (TSENSCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6:3	DEMSEL	RW	0x0	Current branch select for dynamic element match
2	SWAPBJT	RW	0x0	Swap BJT 0: Not swap bi-polar 1: Swap bi-polar
1	OUTINV	RW	0x0	Invert T-Sensor output 0 and output 1 0: Normal connect 1: Invert T-Sensor output 0 and output 1
0	EN	RW	0x0	Temperature sensor enable 0: Disable 1: Enable

Table 12-318: ADC Register Write-Allow Key Register (ADCREGKEY) Layout

ADCREGKEY (ADC Register Write-Allow Key Register) Offset: 0x2B4 Default: 0x1ACCE551 Access: ADC -> ADCREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 12-319: ADC Register Write-Allow Key Register (ADCREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected ADC registers

13 Temperature sensor

13.1 T-sensor overview

The SPD1148 implements an internal temperature sensor to measure the junction temperature of the device. The sensor output can be sampled with the ADC on Sampler C, which was shown in Figure 12-1.

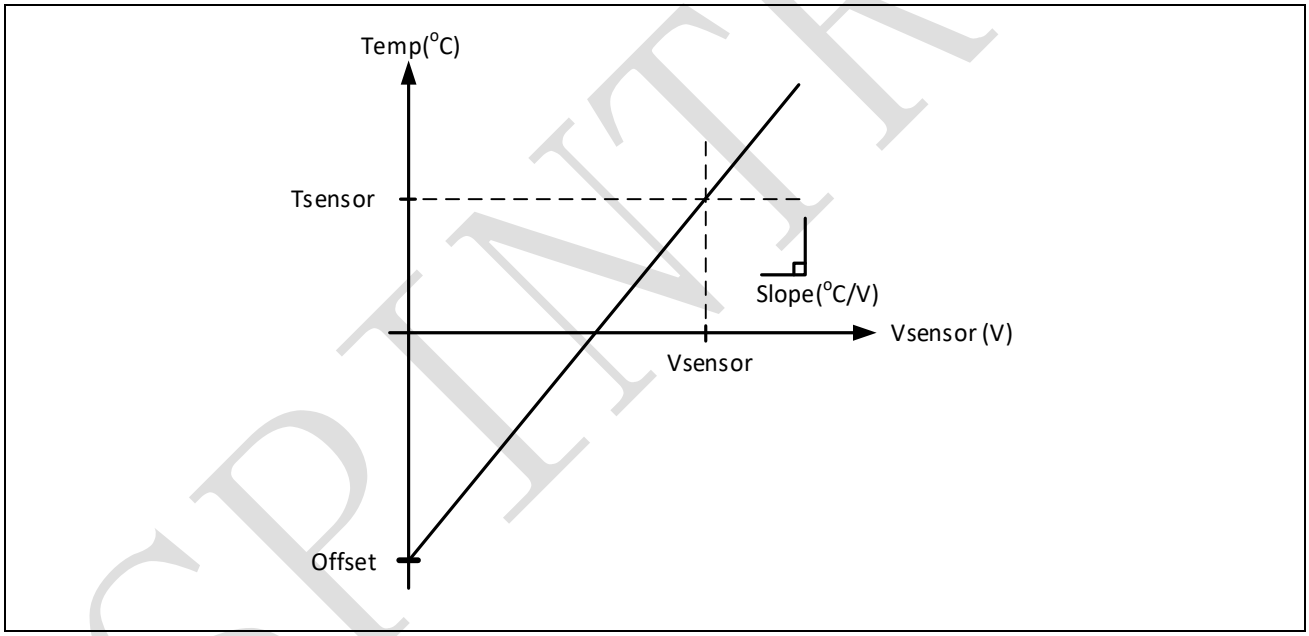
13.2 Transfer function

The temperature sensor output and the resulting ADC values increase with increasing junction temperature. As shown in Figure 13-1, the transfer function of the sensor is defined as:

$$\text{Temperature} = \text{Slope} \times \text{Vsensor} + \text{Offset}$$

Where Slope = 4238.97 °C/V, Offset = -283.36 °C.

Figure 13-1: Temperature sensor transfer function



So the measure temperature step is shown below:

Step 1. In target temperature T, measure the sensor output as Vsensor(Vsensor=TSEN1-TSEN2). The Vsensor should be calibrated using ADC gain and offset trim value.

Step 2. Using the above formula to calculate the target temperature.

Note: One ADC code is 533orresponding 1.9°C. ADC sampling time should be 2us at least to meet the settling requirement of temperature sensor. SPD1148 support the function named ADC_CalculateTemperature in SDK to get the temperature directly.

13.3 Power-up sequence

When powering up the T-sensor, the following sequence must be used:

Step 1: Power up ADC bandgap

Step 2: After 200us, power up ADC reference buffer

Step 3: After 200us, power up the analog circuits of ADC

13.4 Precision improvement

The accurary of temperature sensor will be degraded by inherent mismatch from internal device. Because this temperature sensor is always used for over-temperature protection that didn't need high precision case, the temperature sensor is not factory trimmed.

However, SPD1148 uses dynamic element matching (DEM) to correct this mismatch and get the better accuary for tempearature sensor. The function named ADC_CalculateTemperature in SDK has used this method to get tempearature value.

13.5 Registers

Please see Table 12-316 and Table 12-317 for T-sensor related register definitions.

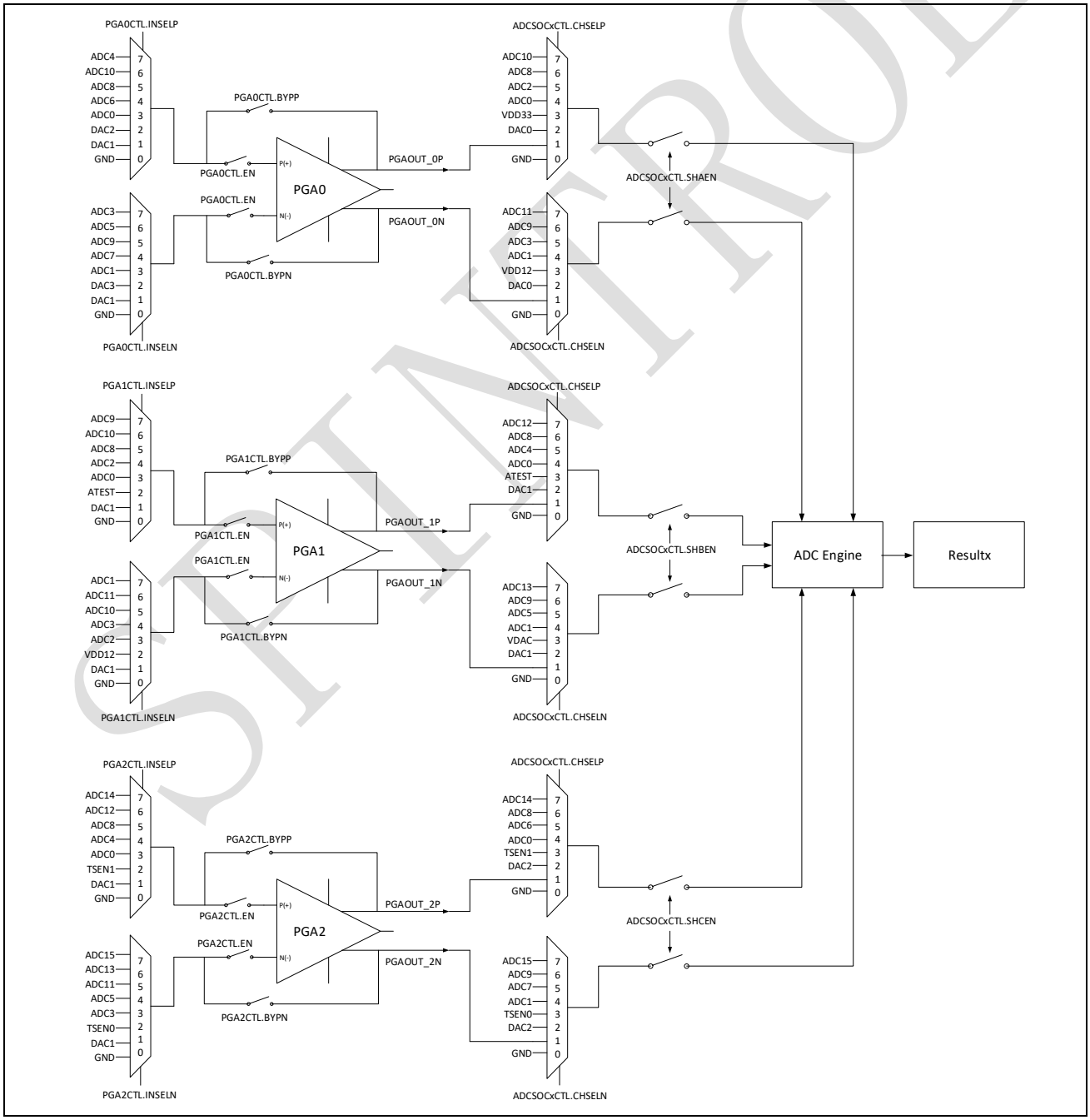
14 PGA

14.1 PGA overview

The SPD1148 implements three Programmable-Gain Amplifiers (PGA). The three PGAs can support 3-channel amplifying simultaneously. Each PGA can support two mode: one is differential mode with gain up to 64, the other is single ended mode with gain up to 32. In single ended mode, one PGA can act like two single-ended PGA.

14.2 PGA architecture

Figure 14-1: PGA block architecture



14.3 PGA channel selection

For each PGA, there are 3 different options groups to control 8 to 1 channel MUX. In one group, the selection contain positive and negative (PGAxCTL.INSELP/N, x = 0, 1, 2 for PGA0/1/2). The channel selection table is shown below.

Table 14-1: PGA MUX selection

PGAxCTL.INSELP AND PGAxCTL.INSELN	in_p to PGA0	in_n to PGA0	in_p to PGA1	in_n to PGA1	in_p to PGA2	in_n to PGA2
7	ADC4	ADC3	ADC9	ADC1	ADC14	ADC15
6	ADC10	ADC5	ADC10	ADC11	ADC12	ADC13
5	ADC8	ADC9	ADC8	ADC10	ADC8	ADC11
4	ADC6	ADC7	ADC2	ADC3	ADC4	ADC5
3	ADC0	ADC1	ADC0	ADC2	ADC0	ADC3
2	DAC2	DAC3	ATEST	VDD12	TSEN1	TSEN0
1	DAC1	DAC1	DAC1	DAC1	DAC1	DAC1
0	GND	GND	GND	GND	GND	GND

Example 13.3.1 Examples of PGA MUX usage

- Connect temperature sensor differential output to PGA2
- Connect input ADC10 voltage (referred to ground) into PGA1
- Connect difference between ADC8 and DAC1 into PGA0

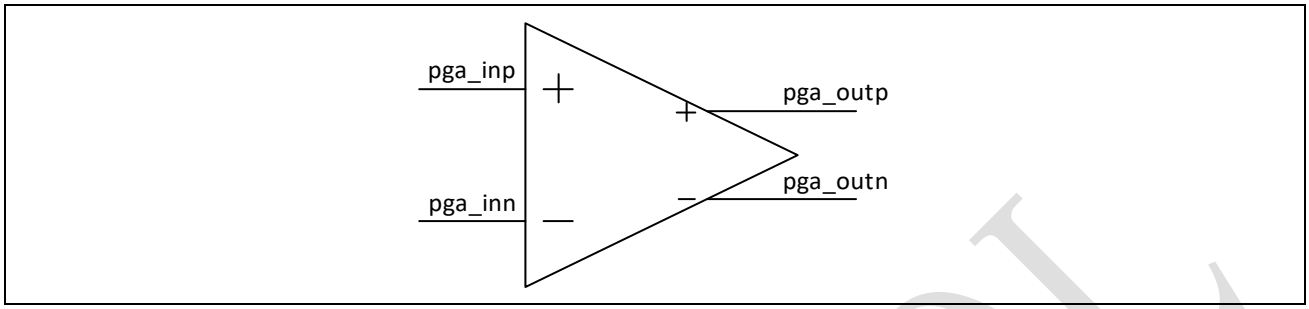
Example 13.3.1

```
void PGA_Example13_3_1(void)
{
  PGA->PGA2CTL.bit.INSELP=2;/* Select TSENS1 to PGA2_INP           */
  PGA->PGA2CTL.bit.INSELN=2;/* Select TSENS0 to PGA2_INN*/
  PGA->PGA1CTL.bit.INSELP=6;/* Select ADC10 to PGA1_INP              */
  PGA->PGA1CTL.bit.INSELN=0;/* Select GND to PGA1_INN           */
  PGA->PGA0CTL.bit.INSELP=5;/* Select ADC8 to PGA0_INP            */
  PGA->PGA0CTL.bit.INSELN=1;/* Select DAC1 to PGA0_INN            */
}
```

14.4 PGA mode and gain

Each PGA can be configured to work in single-ended mode or differential-mode.

Figure 14-2: PGA input and output



In single-ended mode, for each PGA, either inp or inn can be chose as input. Correspondingly, the output is outp or outn. And inp and inn can be simultaneously used in single-ended mode for same PGA. It depends on PGAxCTL.MODE settings shown as below:

Table 14-2: PGA mode selection

PGAxCTL.MODE	Description
0	Differential mode
1	Single-ended mode with positive path only
2	Single-ended mode with negative path only
3	Single-ended mode with both negative path and positive path

The gain selection for single-ended mode and differential mode is shown as below:

Table 14-3: PGA gain selection

PGAxCTL.GAINP AND PGAxCTL.GAINN	Single-ended mode gain	Differential mode gain
0	1	2
1	2	4
2	4	8
3	8	16
4	12	24
5	16	32
6	24	48
7	32	64

For single-ended mode, positive path gain is set by GAINP, and negative path gain is set by GAINN, shown as the following equations:

$$\mathrm{OUTP} = \mathrm{GAINP} \times \mathrm{INP}$$

$$\mathrm{OUTN} = \mathrm{GAINN} \times \mathrm{INN}$$

For each PGA, both inp and inn are rail to rail input range. Output range in each-end is 0.3V ~ AVDD-0.3V, to avoid saturation in PGA, different gain is corresponding to different input range. Assume AVDD is 3.3V, input range in different gain for single-ended mode is shown as below:

Table 14-4: PGA input range in different gain for single-ended mode

PGAxCTL.GAINP(N)	Gain	Input range (V)	Output range (V)
0	1	0.3~3	0.3~3
1	2	0.15~1.5	0.3~3
2	4	0.075~0.75	0.3~3
3	8	0.0375~0.375	0.3~3
4	12	0.025~0.25	0.3~3
5	16	0.01875~0.1875	0.3~3
6	24	0.0125~0.125	0.3~3
7	32	0.009375~0.09375	0.3~3

The GAINP and GAINN need to be set as the same value for differential mode. For example, GAINP = GAINN = GAIN, then the differential output is shown as below:

$$\mathrm{OUTP} - \mathrm{OUTN} = \mathrm{GAIN} \times (INP - INN)$$

Common mode voltage of output can be set by either inp or inn, depends on the PGAxCTL.CMSEL setting. By default, PGAxCTL.CMSEL = 0, and inn is set as output common voltage. If PGAxCTL.CMSEL = 1, inp is set as output common mode voltage. For example, if PGAxCTL.CMSEL = 0,

$$\mathrm{OUT\_CM} = \mathrm{INN}$$

$$\mathrm{OUTP} = \mathrm{INN} + (INP - INN) \times \frac{Gain}{2}$$

$$\mathrm{OUTN} = \mathrm{INN} - (INP - INN) \times \frac{Gain}{2}$$

If PGAxCTL.CMSEL = 1,

$$\mathrm{OUT\_CM} = \mathrm{INP}$$

$$\mathrm{OUTP} = \mathrm{INP} + (INP - INN) \times \frac{Gain}{2}$$

$$\mathrm{OUTN} = \mathrm{INP} - (INP - INN) \times \frac{Gain}{2}$$

Example 13.4.1 PGA in single-ended mode with positive path and set gain

Only use PGA2 positive path, inp = 0.5V, and expected outp = 1V

- Set PGA2 positive path in single-ended mode
- Set single mode gain = 2

Example 13.4.1

```
void PGA_Example13_4_1(void)
{
  PGA->PGA2CTL.bit.MODE=1;/* Set single mode with positive path only   */
  PGA->PGA2CTL.bit.GAINP=1;/* Set single mode gain =2 */
}
```

Example 13.4.2 PGA in single-ended mode with positive and negative path and set gain

Use PGA1 both positive and negative path in single-ended mode, if inp range is 0.3V to 0.5V, and inn range is 0.06V to 0.14V, set positive path gain to 4, so outp range is from 1.2V to 2V; and set negative path gain to 16, so outn range is from 0.96V to 2.24V.

- Set PGA1 positive path and negative path in single mode
- Set PGA1 positive path gain is 4
- Set PGA1 negative path gain is 16

Example 13.4.2

```
void PGA_Example13_4_2(void)
{
PGA->PGA1CTL.bit.MODE=3;/* Set single mode with positive path and negative path  */
PGA->PGA1CTL.bit.GAINP=2;/* Set single mode positive path gain = 4 */
PGA->PGA1CTL.bit.GAINN=5;/* Set single mode negative path gain = 16 */
}
```

Example 13.4.3 PGA in differential mode and gain set

If AVDD=3.3V, gain=4X, inp=2.05V, inn=1.65V, use inn as output common voltage:

outp = 1.65 + (2.05-1.65) * 4/2 = 2.45V

outn = 1.65 – (2.05-1.65) * 4/2 = 0.85V

- Set PGA1 in differential mode
- Set INN as output common mode voltage
- Set gain = 4

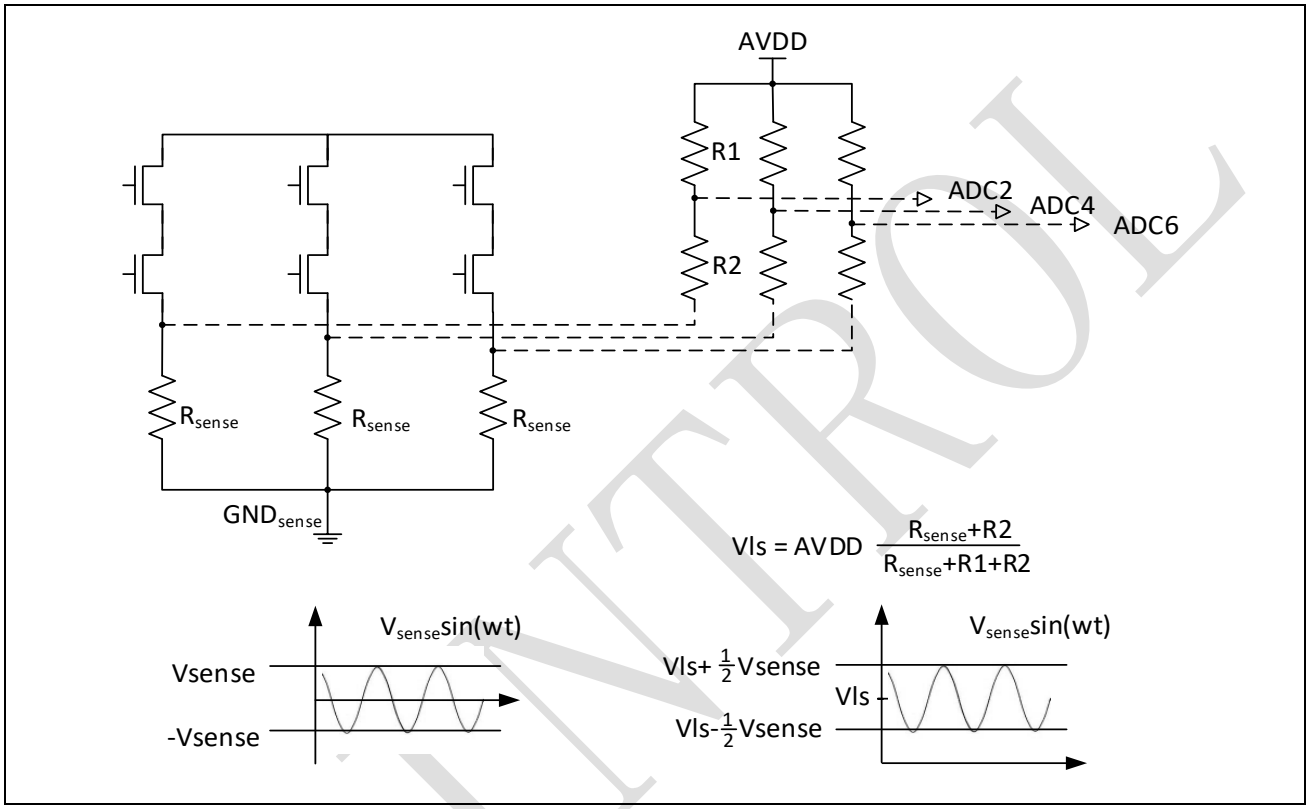
Example 13.4.3

```
void PGA_Example13_4_3(void)
{
PGA->PGA2CTL.bit.MODE=0;/* Set differential mode   */
PGA->PGA2CTL.bit.CMSEL=1;/* Use INN as output common mode voltage  */
PGA->PGA2CTL.bit.GAINP=1;/* Set differential mode gain = 4 */
PGA->PGA2CTL.bit.GAINN=1;/* Set differential mode gain = 4 */
}
```

14.5 PGA single-ended application in current sensing

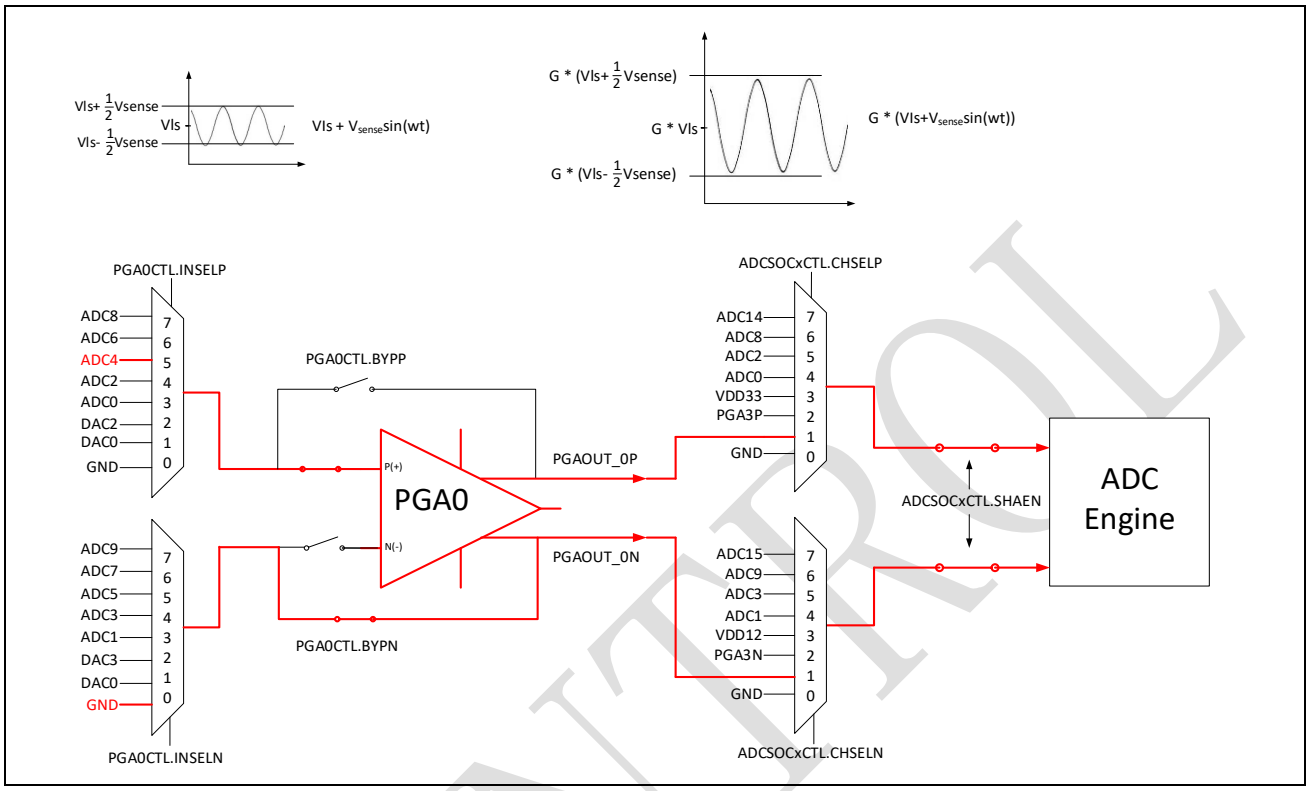
For driver FET current sensing, as Figure 14-3 shows, assume use ADC2, ADC4, ADC6 to sample. PGA configuration is described in Table 14-1. It is important to level-shift V_{sense} signal just enough to guarantee V_{sense} does not go below ground, so that the level-shifted signal could be brought up on-chip to PGA.

Figure 14-3: Single-ended sensing of driver FET currents



As Figure 14-4 shows, PGA0 amplifies ADC4 and its outputs are brought to ADC for conversion via ADC MUX. Input signal Vls+Vsense*sin(wt) is multiplied by G.

Figure 14-4: PGA0 sample ADC4 in single-ended mode



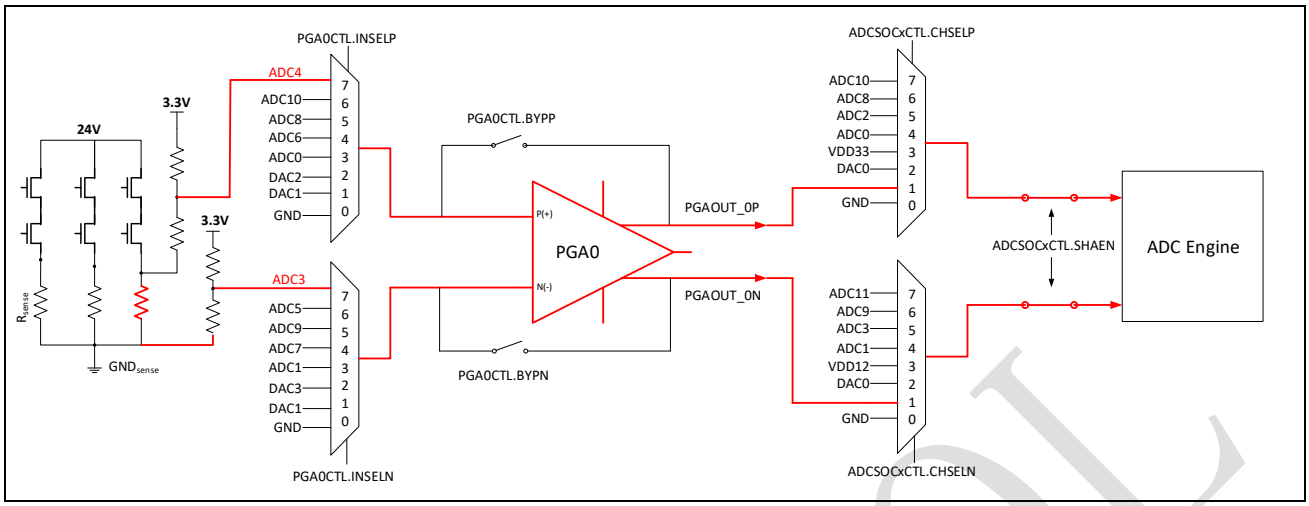
14.6 PGA differential application in current sensing

Motor control is achieved usually by sensing current through a driver FET as depicted in Figure 14-5. The current is usually measured by measuring voltage in the sense resistor. The resulting voltage is a sinusoid with average 0V. The negative voltage on the sensing resistor Rsense cannot be fed into the chip and is level-shifted up by using resistor divider consisting of two resistors R. The most optimal design has the two resistors equal and top resistor connected to the voltage level equal to high-voltage power of the chip AVDD, as will be seen from further discussion.

The level-shifted signal has average of AVDD/2 and amplitude equal to half of the amplitude of voltage on the sense resistor. In addition to the level-shifted motor current signals, a level-shifted sensing ground is also set up for conversion.

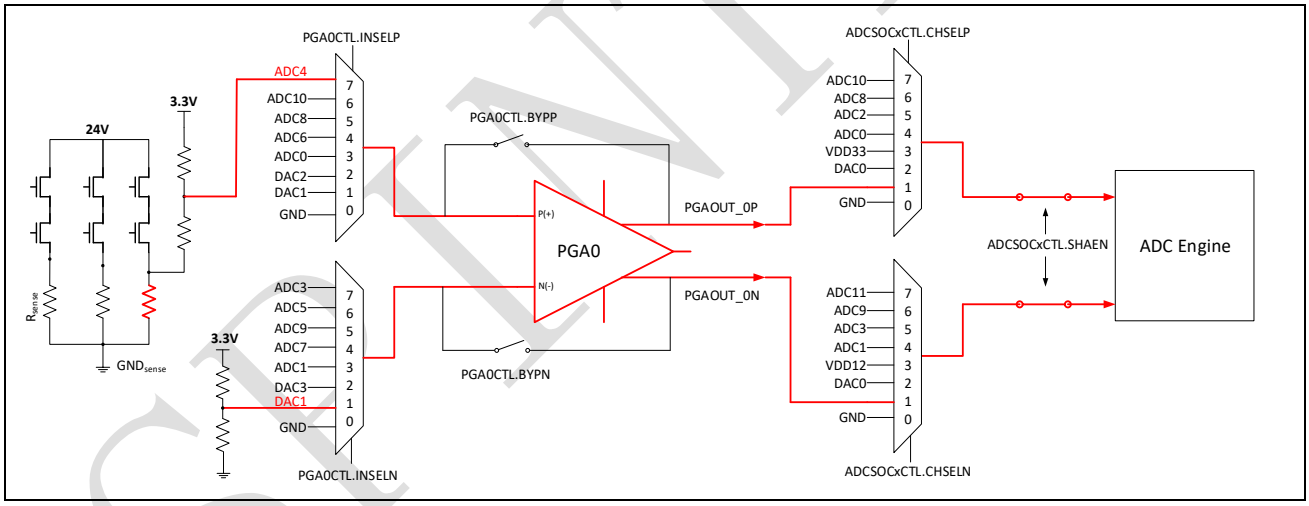
Arrange PGA0 in differential mode with top branch receiving level-shifted sensed signal ADC4, and bottom branch receiving ADC3 (level-shifted ground, optimally AVDD/2). The output common mode of PGA0 is equal to the input of the bottom branch, i.e. equal to AVDD/2. This creates the optimal range for PGA0 output signal differential swing, because it places common mode voltage directly in the middle of 0 to AVDD output range of PGA0. The PGA0 output differential signal is equal to the input differential signal multiplied by gain, as shown in Figure 14-5.

Figure 14-5: Motor sensing arrangement (Differential gain = 8)



Instead of using outside resistor divider to generate AVDD/2, we can choose an internal 10-bit DAC as common mode, as Figure 14-6 shown. The advantage of this method is to save a pin of feeding level-shifting ground.

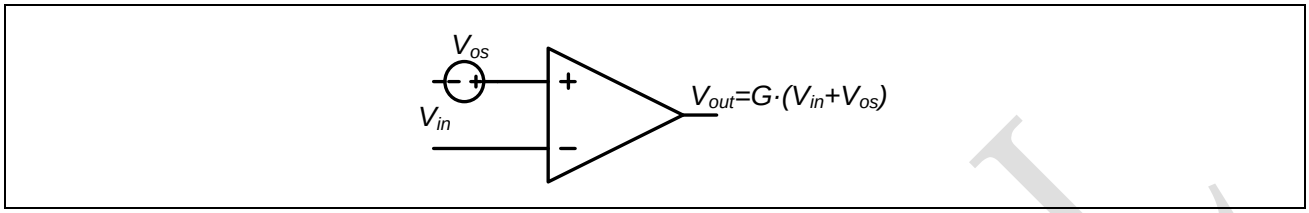
Figure 14-6: Three-pin differential sensing of three currents (Differential gain = 8)



14.7 PGA offset calibration

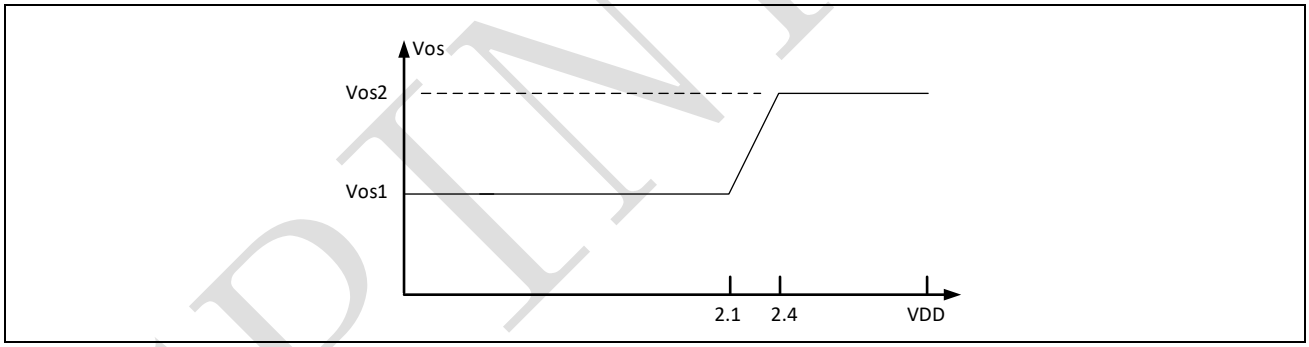
Due to natural mismatch of transistors on the chip, every amplifier has a natural input-referred offset which distorts output. This concept is shown in Figure 14-7. Here PGA output is equal to gain multiplying input plus an extra voltage Vos, called input-referred offset or simply offset of PGA.

Figure 14-7: PGA offset



In order to calibrate for offset, the offset must be independent of input, i.e. Vos cannot depend on Vin, both its differential and common mode component. The offset dynamics of SPD1148 amplifiers is shown in Figure 14-8. For input common mode voltage less than 2.1V, the offset has one value, for the input common mode range high enough (larger than 2.4V), the offset has another value, and a transition region between 2.1 and 2.4V. Note that for AVDD>3V, the input is centered on 1.5V or more, as long as input common mode voltage is less than 2.1V, the constant-Vos scenario holds and offset can be calibrated.

Figure 14-8: Offset behavior with input common-mode voltage



If the offset is determined, the software can account for it, by subtracting from measured signal offset multiplied by gain, to get ideally-amplified signal information. Therefore, calibrating for offset is an integral part of accurate signal amplification and measurement.

In order to calibrate for offset, equal signal must apply to both inputs of PGA (Vin=0 in Figure 14-8). Then the measured output, divided by gain, is the desired offset value Vos. It is recommended to apply equal 10bit DAC value to PGAn (use DAC1 see Table 14-1 and related discussion), with DAC1 output being equal to value less than 2.1V. The PGA will output Vos1 times gain. Since Vos1 is small to begin with (<2mV), within largest gain of 64 the PGA differential output is less than 200mV and the PGA output will not saturate.

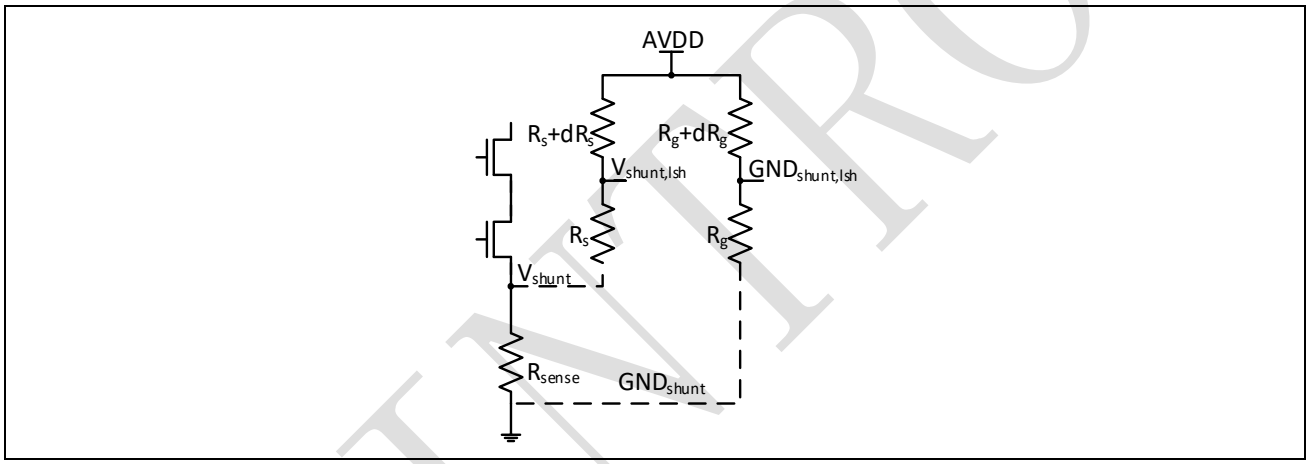
It is not recommended to apply external level shifters on board because level shifters mismatch probably more than any on-chip elements. Moreover, since the same DAC1 voltage point is applied to both p and n inputs of PGA, even if internal DAC resistors have minor mismatch, differential input signal is exactly zero.

There are three different offsets for three different modes of PGA operation: PGA operating in differential mode, PGA operating with single-ended mode for p-branch (n-branch disabled and bypassed), PGA operating with single-ended mode for n-branch (p-branch disabled and bypassed) has different offset which should be independently determined and accounted for by post-ADC processing.

14.8 Resistive level shifter offset calibration

As can be seen from Figure 14-5, each on-board resistive level shifter consists of two resistors. For conventional differential mode operation, both resistors should be equal, in order to level shift signal to come out differentially centered on mid-rail AVDD/2 level. The two resistors usually mismatch and this mismatch also has to be calibrated, in order to correctly determine voltage across current-sensing shunt resistor.

Figure 14-9: Level-shifting with resistor mismatch



For simplicity, but without reducing discussion generality, assume the case of Figure 14-9. There the voltages are level-shifted to the mid-rail AVDD/2 level, and nominally each two level-shifting resistors are equal. Then we can write for the level-shifted outputs:

$$V_{shunt,lsh} = V_{ddx}\frac{R_s}{2\cdot R_s + \delta R_s} + V_{shunt}\frac{R_s + \delta R_s}{2\cdot R_s + \delta R_s} = \frac{V_{ddx}}{2}\left[1 - \frac{\delta_s}{2}\right] + \frac{V_{shunt}}{2}\left[1 + \frac{\delta_s}{2}\right]$$

$$GND_{shunt,lsh} = V_{ddx}\frac{R_g}{2\cdot R_g + \delta R_g} + GND_{shunt}\frac{R_g + \delta R_g}{2\cdot R_g + \delta R_g} = \frac{V_{ddx}}{2}\left[1 - \frac{\delta_g}{2}\right] + \frac{GND_{shunt}}{2}\left[1 + \frac{\delta_g}{2}\right]$$

Here Rs/Rs=δ_s and Rg/Rg=δ_g. The differential signal can be written:

$$V_{ind} = V_{shunt,lsh} - GND_{shunt,lsh} = \frac{V_{ddx}}{2}\cdot\frac{\delta_g - \delta_s}{2} + \frac{V_{shunt} - GND_{shunt}}{2} + \frac{V_{shunt}}{2}\cdot\frac{\delta_s}{2}$$

Here we neglect term $GND_{shunt}\delta_g/4$ because both GND_{shunt} and δ_g are small. We see that mismatch is separated in gain error (V_{shunt}-multiplying term δ_s /2) and offset (V_{ind} at V_{shunt}=0 term). Gain error term is equivalent to effective difference in R_{sense} resistor, because $V_{shunt}=I_{FET}xR_{sense}$. Both terms can be determined from measuring signal when the motor is not operating, i.e. when V_{shunt}=0. The following procedure can be observed:

- Measure AVDD. It can be routed to ADC via ADC MUX, as can be seen from Table 14-1.

- With zero current, when V_{shunt}=0, measure $V_{shunt,lsh}$ and determine δ_s (remember, AVDD is known already) using the first equation of this chapter.

Finally, with zero current, measure V_{ind} and determine $\delta_g - \delta_s$ term, thereby deriving δ_g using the third equation of this chapter.

14.9 Power-up sequence

When powering up the PGA, the following sequence must be used:

Step 1: Power up ADC bandgap

Step 2: Power up the analog circuits of PGA

14.10 Registers

14.10.1 PGA register map

Table 14-5: PGA Module Base Address

Peripheral Module	Base Address
PGA	0x4000 8C00

Table 14-6: PGA Register Map

Register	Offset	Description	Reset Value
PGA0CTL*	0x2C0	PGA0 Control Register	0x00000000
PGA1CTL*	0x2C4	PGA1 Control Register	0x00000000
PGA2CTL*	0x2C8	PGA2 Control Register	0x00000000
PGAREGKEY	0x2E0	PGA Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the PGAREGKEY=0x1ACCE551.

14.10.2 PGA registers

Table 14-7: PGA0 Control Register (PGA0CTL) Layout

PGA0CTL (PGA0 Control Register) Offset: 0x2C0 Default: 0x00000000
Access: PGA -> PGA0CTL.all

31	30	29	28	27	26	25	24
RESERVED_31_18							
23	22	21	20	19	18	17	16
RESERVED_31_18						BYPN	BYPP
15	14	13	12	11	10	9	8
GAINN			GAINP			INSELN	
7	6	5	4	3	2	1	0
INSELN	INSELP			CMSEL	MODE		EN

Table 14-8: PGA0 Control Register (PGA0CTL) Description

Bits	Field Name	Type	Reset	Description
31:18	RESERVED_31_18	RO	0x0	Reserved.
17	BYPN	RW	0x0	PGA0 negative path bypass 0: Not bypass 1: Bypass

Bits	Field Name	Type	Reset	Description
16	BYPP	RW	0x0	PGA0 positive path bypass 0: Not bypass 1: Bypass
15:13	GAINN	RW	0x0	PGA0 negative path gain 000: 1x for single-ended mode and 2x for differential mode 001: 2x for single-ended mode and 4x for differential mode 010: 4x for single-ended mode and 8x for differential mode 011: 8x for single-ended mode and 16x for differential mode 100: 12x for single-ended mode and 24x for differential mode 101: 16x for single-ended mode and 32x for differential mode 110: 24x for single-ended mode and 48x for differential mode 111: 32x for single-ended mode and 64x for differential mode
12:10	GAINP	RW	0x0	PGA0 positive path gain 000: 1x for single-ended mode and 2x for differential mode 001: 2x for single-ended mode and 4x for differential mode 010: 4x for single-ended mode and 8x for differential mode 011: 8x for single-ended mode and 16x for differential mode 100: 12x for single-ended mode and 24x for differential mode 101: 16x for single-ended mode and 32x for differential mode 110: 24x for single-ended mode and 48x for differential mode 111: 32x for single-ended mode and 64x for differential mode
9:7	INSELN	RW	0x0	PGA0 negative input select 000: Analog ground 001: DAC1 for common input 010: DAC3 output 011: ADC1 input from GPIO1 100: ADC7 input from GPIO7 101: ADC9 input from GPIO9

Bits	Field Name	Type	Reset	Description
				110: ADC5 input from GPIO5 111: ADC3 input from GPIO3
6:4	INSELP	RW	0x0	PGA0 positive input select 000: Analog ground 001: DAC1 for common input 010: DAC2 output 011: ADC0 input from GPIO0 100: ADC6 input from GPIO6 101: ADC8 input from GPIO8 110: ADC10 input from GPIO10 111: ADC4 input from GPIO4
3	CMSEL	RW	0x0	PGA0 output common voltage selection for differential mode 0: Select negative input voltage as output common voltage 1: Select positive input voltage as output common voltage
2:1	MODE	RW	0x0	PGA0 mode 00: Differential mode 01: Single-ended mode with positive path only 10: Single-ended mode with negative path only 11: Single-ended mode with both paths
0	EN	RW	0x0	PGA0 enable 0: Disable PGA0 1: Enable PGA0

Table 14-9: PGA1 Control Register (PGA1CTL) Layout

PGA1CTL (PGA1 Control Register) Offset: 0x2C4 Default: 0x00000000
Access: PGA -> PGA1CTL.all

31	30	29	28	27	26	25	24
RESERVED_31_18							
23	22	21	20	19	18	17	16
RESERVED_31_18						BYPN	BYPP
15	14	13	12	11	10	9	8
GAINN			GAINP			INSELN	
7	6	5	4	3	2	1	0
INSELN	INSELP			CMSEL	MODE		EN

Table 14-10: PGA1 Control Register (PGA1CTL) Description

Bits	Field Name	Type	Reset	Description
31:18	RESERVED_31_18	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
17	BYPN	RW	0x0	PGA1 negative path bypass 0: Not bypass 1: Bypass
16	BYPP	RW	0x0	PGA1 positive path bypass 0: Not bypass 1: Bypass
15:13	GAINN	RW	0x0	PGA1 negative path gain 000: 1x for single-ended mode and 2x for differential mode 001: 2x for single-ended mode and 4x for differential mode 010: 4x for single-ended mode and 8x for differential mode 011: 8x for single-ended mode and 16x for differential mode 100: 12x for single-ended mode and 24x for differential mode 101: 16x for single-ended mode and 32x for differential mode 110: 24x for single-ended mode and 48x for differential mode 111: 32x for single-ended mode and 64x for differential mode
12:10	GAINP	RW	0x0	PGA1 positive path gain 000: 1x for single-ended mode and 2x for differential mode 001: 2x for single-ended mode and 4x for differential mode 010: 4x for single-ended mode and 8x for differential mode 011: 8x for single-ended mode and 16x for differential mode 100: 12x for single-ended mode and 24x for differential mode 101: 16x for single-ended mode and 32x for differential mode 110: 24x for single-ended mode and 48x for differential mode 111: 32x for single-ended mode and 64x for differential mode
9:7	INSELN	RW	0x0	PGA1 negative input select 000: Analog ground 001: DAC1 for common input 010: Digital 1.2V power 011: ADC2 input from GPIO2

Bits	Field Name	Type	Reset	Description
				100: ADC3 input from GPIO3 101: ADC10 input from GPIO10 110: ADC11 input from GPIO11 111: ADC1 input from GPIO1
6:4	INSELP	RW	0x0	PGA1 positive input select 000: Analog ground 001: DAC1 for common input 010: ATEST output voltage 011: ADC0 input from GPIO0 100: ADC2 input from GPIO2 101: ADC8 input from GPIO8 110: ADC10 input from GPIO10 111: ADC9 input from GPIO9
3	CMSEL	RW	0x0	PGA1 output common voltage selection for differential mode 0: Select negative input voltage as output common voltage 1: Select positive input voltage as output common voltage
2:1	MODE	RW	0x0	PGA1 mode 00: Differential mode 01: Single-ended mode with positive path only 10: Single-ended mode with negative path only 11: Single-ended mode with both paths
0	EN	RW	0x0	PGA1 enable 0: Disable PGA1 1: Enable PGA1

Table 14-11: PGA2 Control Register (PGA2CTL) Layout

PGA2CTL (PGA2 Control Register) Offset: 0x2C8 Default: 0x00000000 Access: PGA -> PGA2CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_18							
23	22	21	20	19	18	17	16
RESERVED_31_18						BYPN	BYPP
15	14	13	12	11	10	9	8
GAINN			GAINP			INSELN	
7	6	5	4	3	2	1	0
INSELN	INSELP			CMSEL	MODE		EN

Table 14-12: PGA2 Control Register (PGA2CTL) Description

Bits	Field Name	Type	Reset	Description
31:18	RESERVED_31_18	RO	0x0	Reserved.
17	BYPN	RW	0x0	PGA2 negative path bypass 0: Not bypass 1: Bypass
16	BYPP	RW	0x0	PGA2 positive path bypass 0: Not bypass 1: Bypass
15:13	GAINN	RW	0x0	PGA2 negative path gain 000: 1x for single-ended mode and 2x for differential mode 001: 2x for single-ended mode and 4x for differential mode 010: 4x for single-ended mode and 8x for differential mode 011: 8x for single-ended mode and 16x for differential mode 100: 12x for single-ended mode and 24x for differential mode 101: 16x for single-ended mode and 32x for differential mode 110: 24x for single-ended mode and 48x for differential mode 111: 32x for single-ended mode and 64x for differential mode
12:10	GAINP	RW	0x0	PGA2 positive path gain 000: 1x for single-ended mode and 2x for differential mode 001: 2x for single-ended mode and 4x for differential mode 010: 4x for single-ended mode and 8x for differential mode

Bits	Field Name	Type	Reset	Description
				011: 8x for single-ended mode and 16x for differential mode 100: 12x for single-ended mode and 24x for differential mode 101: 16x for single-ended mode and 32x for differential mode 110: 24x for single-ended mode and 48x for differential mode 111: 32x for single-ended mode and 64x for differential mode
9:7	INSELN	RW	0x0	PGA2 negative input select 000: Analog ground 001: DAC1 for common input 010: T-sensor output 0 011: ADC3 input from GPIO3 100: ADC5 input from GPIO5 101: ADC11 input from GPIO11 110: ADC13 input from GPIO13 111: ADC15 input from GPIO15
6:4	INSELP	RW	0x0	PGA2 positive input select 000: Analog ground 001: DAC1 for common input 010: T-sensor output 1 011: ADC0 input from GPIO0 100: ADC4 input from GPIO4 101: ADC8 input from GPIO8 110: ADC12 input from GPIO12 111: ADC14 input from GPIO14
3	CMSEL	RW	0x0	PGA2 output common voltage selection for differential mode 0: Select negative input voltage as output common voltage 1: Select positive input voltage as output common voltage
2:1	MODE	RW	0x0	PGA2 mode 00: Differential mode 01: Single-ended mode with positive path only 10: Single-ended mode with negative path only 11: Single-ended mode with both paths
0	EN	RW	0x0	PGA2 enable 0: Disable PGA2 1: Enable PGA2

Table 14-13: PGA Register Write-Allow Key Register (PGAREGKEY) Layout

PGAREGKEY (PGA Register Write-Allow Key Register) Offset: 0x2E0 Default: 0x1ACCE551 Access: PGA -> PGAREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 14-14: PGA Register Write-Allow Key Register (PGAREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected PGA registers

15 Comparator

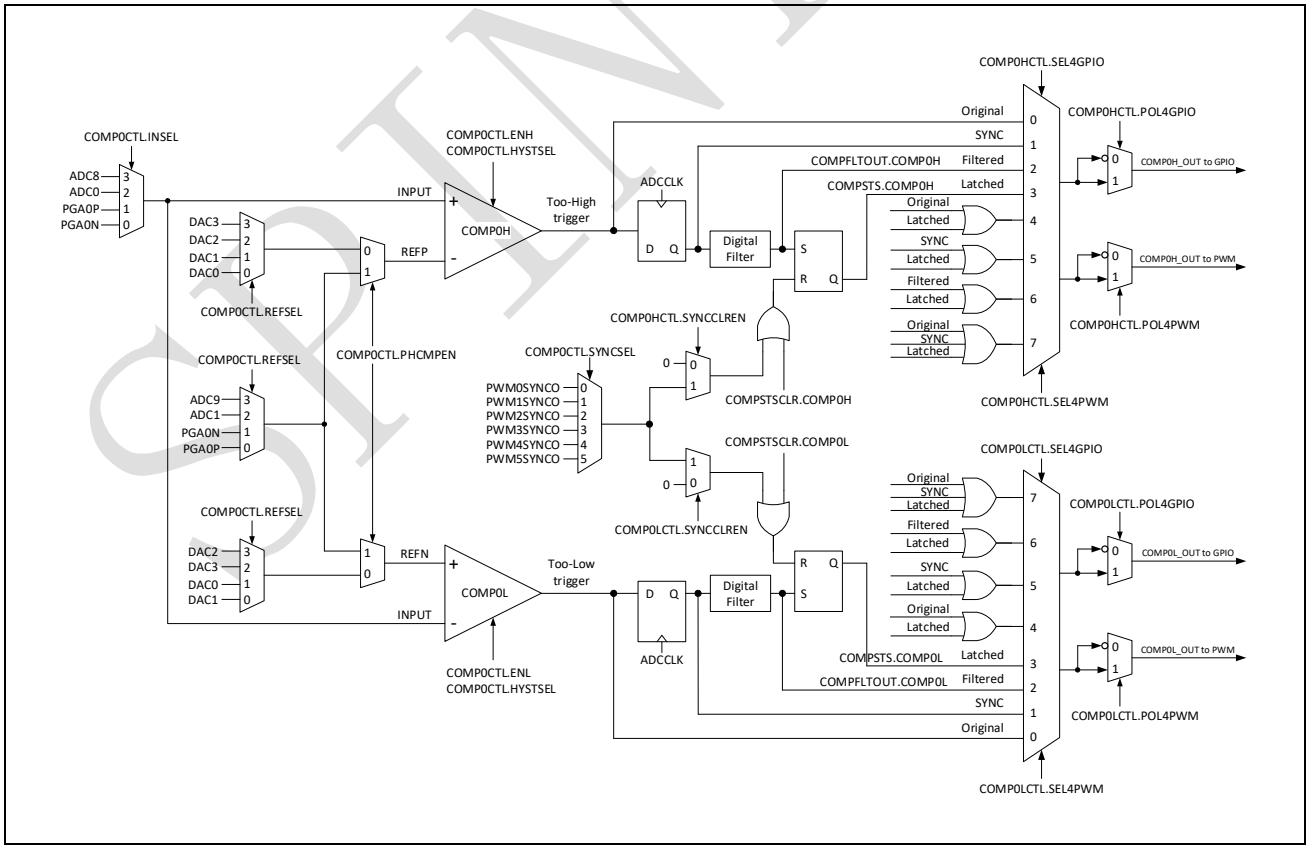
15.1 Comparator overview

For each PGA in SPD1148, there are 2 high speed rail-to-rail input comparators, check whether the voltage is too high or too low for over-current detection resulting in overall 6 comparators. There are two more pairs of comparators reserved for additional applications, such as two motor control support or PFC. Totally, SPD1148 has 10 comparators.

15.2 Comparator architecture

Figure 15-1 shows comparator0 block diagram, which includes two comparators, one is for monitoring voltage is too high, the other is to check voltage too low, and also includes input MUX to the comparators. SPD1148 has phase comparator feature, when COMP0CTL.PHCMPEN = 1, this feature is enabled, for example, if INSEL=1, REFSEL=1, PGA0P will be sent to positive input of the too high comparator and negative input of the too low comparator, PGA0N will be sent to negative input of the too high comparator and positive input of the too low comparator. When COMP0CTL.PHCMPEN = 0, the comparator references are from one of the four 10-bit DACs. Hysteresis can be enabled for each comparator with a typical value of 30 mV. The output of each comparator can be used as the PWM trip-zone event and has a typical delay of 50 ns.

Figure 15-1: Comparator 0 block diagram



15.3 Comparator MUX selection

The input selections for 5 comparator pairs (comparator0~4) are shown in Table 15-1.

Table 15-1: Comparator 0~4 MUX selection

Comparator0 MUX								
INSEL	**INPUT**	**PHCOMPEN**	**REFSEL**	**REFP/N**	**PHCOMPEN**	**REFSEL**	**REFP**	**REFN**
3	ADC8	1	3	ADC9	0	3	DAC3	DAC2
2	ADC0	1	2	ADC1	0	2	DAC2	DAC3
1	PGA0P	1	1	PGA0N	0	1	DAC1	DAC0
0	PGA0N	1	0	PGA0P	0	0	DAC0	DAC1
Comparator1 MUX								
INSEL	**INPUT**	**PHCOMPEN**	**REFSEL**	**REFP/N**	**PHCOMPEN**	**REFSEL**	**REFP**	**REFN**
3	ADC8	1	3	ADC10	0	3	DAC3	DAC2
2	ADC0	1	2	ADC2	0	2	DAC2	DAC3
1	PGA1P	1	1	PGA1N	0	1	DAC1	DAC0
0	PGA1N	1	0	PGA1P	0	0	DAC0	DAC1
Comparator2 MUX								
INSEL	**INPUT**	**PHCOMPEN**	**REFSEL**	**REFP/N**	**PHCOMPEN**	**REFSEL**	**REFP**	**REFN**
3	ADC8	1	3	ADC11	0	3	DAC3	DAC2
2	ADC0	1	2	ADC3	0	2	DAC2	DAC3
1	PGA2P	1	1	PGA2N	0	1	DAC1	DAC0
0	PGA2N	1	0	PGA2P	0	0	DAC0	DAC1
Comparator3 MUX								
INSEL	**INPUT**	**PHCOMPEN**	**REFSEL**	**REFP/N**	**PHCOMPEN**	**REFSEL**	**REFP**	**REFN**
3	ADC12	1	3	ADC13	0	3	DAC3	DAC2
2	ADC8	1	2	ADC9	0	2	DAC2	DAC3
1	ADC4	1	1	ADC5	0	1	DAC1	DAC0
0	ADC0	1	0	ADC1	0	0	DAC0	DAC1
Comparator4 MUX								
INSEL	**INPUT**	**PHCOMPEN**	**REFSEL**	**REFP/N**	**PHCOMPEN**	**REFSEL**	**REFP**	**REFN**
3	ADC14	1	3	ADC15	0	3	DAC3	DAC2
2	ADC10	1	2	ADC11	0	2	DAC2	DAC3
1	ADC6	1	1	ADC7	0	1	DAC1	DAC0
0	ADC2	1	0	ADC3	0	0	DAC0	DAC1

Example 14.3.1 Example of comparator MUX selection

- Set PGA2 positive output to comparator
- Set DAC1 output as too high threshold value
- Set DAC0 output as too low threshold value

Example 14.3.1

```
void COMP_Example14_3_1(void)
{
COMP->COMP2CTL.bit.INSEL=1;/* Set PGA2P as input to comparator */
COMP->COMP2CTL.bit.PHCMPEN=0;/* disable phase comparator function */
COMP->COMP2CTL.bit.REFSEL=1;/* Set DAC1 as REFP, and DAC0 as REFN */
COMP->COMP2CTL.bit.ENH=1;/* Enable too high comparator */
COMP->COMP2CTL.bit.ENL=1;/* Enable too low comparator */
}
```

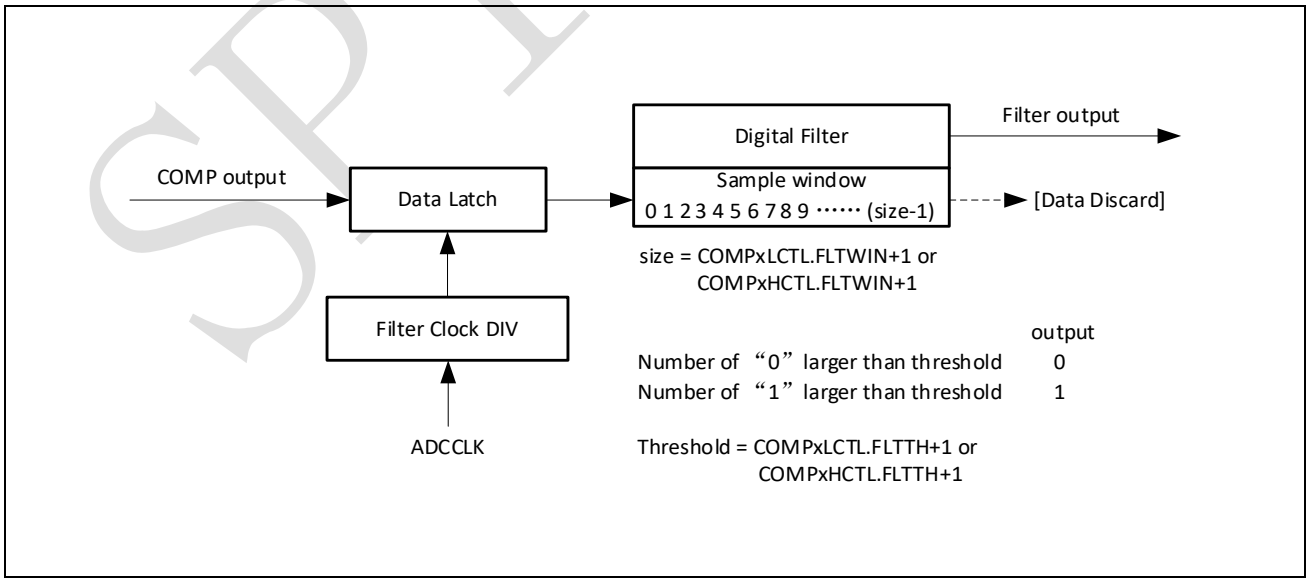
15.4 Digital filter

The comparator's output is sent to the sample window of the digital filter. The sampled clock is from ADCCLK, and can be divided by COMPxLCTL.FLTDIV/COMPxHCTL.FLTDIV, the window size is defined by COMPxLCTL.FLTWIN/COMPxHCTL.FLTWIN. The digital filter can count the number of majority value of the sample window. If the number is larger than the pre-defined threshold value COMPxLCTL.FLTTH/COMPxHCTL.FLTTH, the filter output will change to the majority value; otherwise, the filter output will not be changed.

For proper operation, the threshold values should be greater than COMPxLCTL.FLTWIN/2 and COMPxHCTL.FLTWIN/2 respectively.

A conceptual model of the digital filter is shown in Figure 15-2.

Figure 15-2: Digital filter behavior



15.5 Power-up sequence

When powering up the COMP, the following sequence must be used:

Step 1: Power up ADC bandgap

Step 2: Power up the analog circuits of COMP

15.6 Registers

15.6.1 Comparator register map

Table 15-2: Comparator Module Base Address

Peripheral Module	Base Address
COMP	0x4000 8C00

Table 15-3: COMP Register Map

Register	Offset	Description	Reset Value
COMPFLTOUT	0x2F0	Comparator Filter Output Register	0x00000000
COMPSTS	0x2F4	Comparator Status Register	0x00000000
COMPSTSCLR	0x2F8	Comparator Status Clear Register	0x00000000
COMP0CTL*	0x2FC	Comparator 0 Control Register	0x00000000
COMP0LCTL*	0x300	COMP0L Control Register	0x00000088
COMP0HCTL*	0x304	COMP0H Control Register	0x00000088
COMP1CTL*	0x308	Comparator 1 Control Register	0x00000000
COMP1LCTL*	0x30C	COMP1L Control Register	0x00000088
COMP1HCTL*	0x310	COMP1H Control Register	0x00000088
COMP2CTL*	0x314	Comparator 2 Control Register	0x00000000
COMP2LCTL*	0x318	COMP2L Control Register	0x00000088
COMP2HCTL*	0x31C	COMP2H Control Register	0x00000088
COMP3CTL*	0x320	Comparator 3 Control Register	0x00000000
COMP3LCTL*	0x324	COMP3L Control Register	0x00000088
COMP3HCTL*	0x328	COMP3H Control Register	0x00000088
COMP4CTL*	0x32C	Comparator 4 Control Register	0x00000000
COMP4LCTL*	0x330	COMP4L Control Register	0x00000088
COMP4HCTL*	0x334	COMP4H Control Register	0x00000088
DAC0CTL*	0x35C	DAC0 Control Register	0x00000002

Register	Offset	Description	Reset Value
DAC0CODE	0x360	DAC0 Code Register	0x00000000
DAC0CODEA	0x364	DAC0 Active Code Register	0x00000000
RAMP0DLY*	0x368	RAMP0 Delay Shadow Register	0x00000000
RAMP0DLYA	0x36C	RAMP0 Delay Active Register	0x00000000
RAMP0DEC*	0x370	RAMP0 Decrement Shadow Register	0x00000000
RAMP0DECA	0x374	RAMP0 Decrement Active Register	0x00000000
RAMP0MAX*	0x378	RAMP0 Maximum Value Shadow Register	0x00000000
RAMP0MAXA	0x37C	RAMP0 Maximum Value Active Register	0x00000000
RAMP0CNT	0x380	RAMP0 Count Register	0x00000000
DAC1CTL*	0x384	DAC1 Control Register	0x00000002
DAC1CODE	0x388	DAC1 Code Register	0x00000000
DAC1CODEA	0x38C	DAC1 Active Code Register	0x00000000
RAMP1DLY*	0x390	RAMP1 Delay Shadow Register	0x00000000
RAMP1DLYA	0x394	RAMP1 Delay Active Register	0x00000000
RAMP1DEC*	0x398	RAMP1 Decrement Shadow Register	0x00000000
RAMP1DECA	0x39C	RAMP1 Decrement Active Register	0x00000000
RAMP1MAX*	0x3A0	RAMP1 Maximum Value Shadow Register	0x00000000
RAMP1MAXA	0x3A4	RAMP1 Maximum Value Active Register	0x00000000
RAMP1CNT	0x3A8	RAMP1 Count Register	0x00000000
DAC2CTL*	0x3AC	DAC2 Control Register	0x00000002
DAC2CODE	0x3B0	DAC2 Code Register	0x00000000
DAC2CODEA	0x3B4	DAC2 Active Code Register	0x00000000
DAC3CTL*	0x3B8	DAC3 Control Register	0x00000002
DAC3CODE	0x3BC	DAC3 Code Register	0x00000000
DAC3CODEA	0x3C0	DAC3 Active Code Register	0x00000000
DACBUFCTL*	0x3F4	DAC Buffer Control Register	0x00000000
COMPREGKEY	0x3FC	COMP Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the COMPREGKEY=0x1ACCE551.

15.6.2 Comparator registers

Table 15-4: Comparator Filter Output Register (COMPFLTOUT) Layout

COMPFLTOUT (Comparator Filter Output Register) Offset: 0x2F0 Default: 0x00000000 Access: COMP -> COMPFLTOUT.all							
31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						COMP4H	COMP4L
7	6	5	4	3	2	1	0
COMP3H	COMP3L	COMP2H	COMP2L	COMP1H	COMP1L	COMP0H	COMP0L

Table 15-5: Comparator Filter Output Register (COMPFLTOUT) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9	COMP4H	RO	0x0	COMP4H filtered status without latch 0: Input is lower than the high-boundary 1: Input is higher than the high-boundary
8	COMP4L	RO	0x0	COMP4L filtered status without latch 0: Input is higher than the low-boundary 1: Input is lower than the low-boundary
7	COMP3H	RO	0x0	COMP3H filtered status without latch 0: Input is lower than the high-boundary 1: Input is higher than the high-boundary
6	COMP3L	RO	0x0	COMP3L filtered status without latch 0: Input is higher than the low-boundary 1: Input is lower than the low-boundary
5	COMP2H	RO	0x0	COMP2H filtered status without latch 0: Input is lower than the high-boundary 1: Input is higher than the high-boundary
4	COMP2L	RO	0x0	COMP2L filtered status without latch 0: Input is higher than the low-boundary 1: Input is lower than the low-boundary
3	COMP1H	RO	0x0	COMP1H filtered status without latch 0: Input is lower than the high-boundary 1: Input is higher than the high-boundary
2	COMP1L	RO	0x0	COMP1L filtered status without latch 0: Input is higher than the low-boundary 1: Input is lower than the low-boundary

Bits	Field Name	Type	Reset	Description
1	COMP0H	RO	0x0	COMP0H filtered status without latch 0: Input is lower than the high-boundary 1: Input is higher than the high-boundary
0	COMP0L	RO	0x0	COMP0L filtered status without latch 0: Input is higher than the low-boundary 1: Input is lower than the low-boundary

Table 15-6: Comparator Status Register (COMPSTS) Layout

COMPSTS (Comparator Status Register) Offset: 0x2F4 Default: 0x00000000 Access: COMP -> COMPSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						COMP4H	COMP4L
7	6	5	4	3	2	1	0
COMP3H	COMP3L	COMP2H	COMP2L	COMP1H	COMP1L	COMP0H	COMP0L

Table 15-7: Comparator Status Register (COMPSTS) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9	COMP4H	RO	0x0	Latched COMPFLTOUT[COMP4H] 0: Input has been lower than the high-boundary 1: Input has been higher than the high-boundary
8	COMP4L	RO	0x0	Latched COMPFLTOUT[COMP4L] 0: Input has been higher than the low-boundary 1: Input has been lower than the low-boundary
7	COMP3H	RO	0x0	Latched COMPFLTOUT[COMP3H] 0: Input has been lower than the high-boundary 1: Input has been higher than the high-boundary
6	COMP3L	RO	0x0	Latched COMPFLTOUT[COMP3L] 0: Input has been higher than the low-boundary 1: Input has been lower than the low-boundary
5	COMP2H	RO	0x0	Latched COMPFLTOUT[COMP2H] 0: Input has been lower than the high-boundary 1: Input has been higher than the high-boundary

Bits	Field Name	Type	Reset	Description
4	COMP2L	RO	0x0	Latched COMPFLTOUT[COMP2L] 0: Input has been higher than the low-boundary 1: Input has been lower than the low-boundary
3	COMP1H	RO	0x0	Latched COMPFLTOUT[COMP1H] 0: Input has been lower than the high-boundary 1: Input has been higher than the high-boundary
2	COMP1L	RO	0x0	Latched COMPFLTOUT[COMP1L] 0: Input has been higher than the low-boundary 1: Input has been lower than the low-boundary
1	COMP0H	RO	0x0	Latched COMPFLTOUT[COMP0H] 0: Input has been lower than the high-boundary 1: Input has been higher than the high-boundary
0	COMP0L	RO	0x0	Latched COMPFLTOUT[COMP0L] 0: Input has been higher than the low-boundary 1: Input has been lower than the low-boundary

Table 15-8: Comparator Status Clear Register (COMPSTSCLR) Layout

COMPSTSCLR (Comparator Status Clear Register) Offset: 0x2F8 Default: 0x00000000 Access: COMP -> COMPSTSCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						COMP4H	COMP4L
7	6	5	4	3	2	1	0
COMP3H	COMP3L	COMP2H	COMP2L	COMP1H	COMP1L	COMP0H	COMP0L

Table 15-9: Comparator Status Clear Register (COMPSTSCLR) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9	COMP4H	W1C	0x0	Latched COMP4H status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP4H]. This bit is self-cleared.
8	COMP4L	W1C	0x0	Latched COMP4L status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP4L]. This bit is self-cleared.

Bits	Field Name	Type	Reset	Description
7	COMP3H	W1C	0x0	Latched COMP3H status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP3H]. This bit is self-cleared.
6	COMP3L	W1C	0x0	Latched COMP3L status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP3L]. This bit is self-cleared.
5	COMP2H	W1C	0x0	Latched COMP2H status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP2H]. This bit is self-cleared.
4	COMP2L	W1C	0x0	Latched COMP2L status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP2L]. This bit is self-cleared.
3	COMP1H	W1C	0x0	Latched COMP1H status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP1H]. This bit is self-cleared.
2	COMP1L	W1C	0x0	Latched COMP1L status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP1L]. This bit is self-cleared.
1	COMP0H	W1C	0x0	Latched COMP0H status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP0H]. This bit is self-cleared.
0	COMP0L	W1C	0x0	Latched COMP0L status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears COMPSTS[COMP0L]. This bit is self-cleared.

Table 15-10: Comparator 0 Control Register (COMP0CTL) Layout

COMP0CTL (Comparator 0 Control Register) Offset: 0x2FC Default: 0x00000000 Access: COMP -> COMP0CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				SYNCSEL			REFSEL
7	6	5	4	3	2	1	0
REFSEL	INSEL		HYSTSEL		PHCMPEN	ENH	ENL

Table 15-11: Comparator 0 Control Register (COMP0CTL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11:9	SYNCSEL	RW	0x0	Select the synchronous output from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output 101: PWM5 synchronous output 110: 111:
8:7	REFSEL	RW	0x0	COMP0 reference select 00: PGA0 positive output when PHCMPEN = 1; DAC0 as too-high reference and DAC1 as too-low reference when PHCMPEN = 0 01: PGA0 negative output when PHCMPEN = 1; DAC1 as too-high reference and DAC0 as too-low reference when PHCMPEN = 0 10: ADC1 input from GPIO1 when PHCMPEN = 1; DAC2 as too-high reference and DAC3 as too-low reference when PHCMPEN = 0 11: ADC9 input from GPIO9 when PHCMPEN = 1; DAC3 as too-high reference and DAC2 as too-low reference when PHCMPEN = 0
6:5	INSEL	RW	0x0	COMP0 input select 00: PGA0 negative output 01: PGA0 positive output 10: ADC0 input from GPIO0 11: ADC8 input from GPIO8
4:3	HYSTSEL	RW	0x0	COMP0 hysteresis select 00: 0 mV 01: 12 mV 10: 24 mV 11: 36 mV
2	PHCMPEN	RW	0x0	COMP0 phase comparison enable 0: Normal operation 1: Phase comparison mode
1	ENH	RW	0x0	COMP0H enable 0: Disable COMP0H 1: Enable COMP0H

Bits	Field Name	Type	Reset	Description
0	ENL	RW	0x0	COMP0L enable 0: Disable COMP0L 1: Enable COMP0L

Table 15-12: COMP0L Control Register (COMP0LCTL) Layout

COMP0LCTL (COMP0L Control Register) Offset: 0x300 Default: 0x00000088 Access: COMP -> COMP0LCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-13: COMP0L Control Register (COMP0LCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP0L status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP0L status 1: PWMSYNC will clear the latched COMP0L status
7	POL4GPIO	RW	0x1	COMP0L output polarity for GPIO 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
6:4	SEL4GPIO	RW	0x0	COMP0L output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output

Bits	Field Name	Type	Reset	Description
				011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP0L output polarity for PWM 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
2:0	SEL4PWM	RW	0x0	COMP0L output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-14: COMP0H Control Register (COMP0HCTL) Layout

COMP0HCTL (COMP0H Control Register) Offset: 0x304 Default: 0x00000088 Access: COMP -> COMP0HCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-15: COMP0H Control Register (COMP0HCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP0H status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP0H status 1: PWMSYNC will clear the latched COMP0H status
7	POL4GPIO	RW	0x1	COMP0H output polarity for GPIO 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
6:4	SEL4GPIO	RW	0x0	COMP0H output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP0H output polarity for PWM 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
2:0	SEL4PWM	RW	0x0	COMP0H output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output

Bits	Field Name	Type	Reset	Description
				011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-16: Comparator 1 Control Register (COMP1CTL) Layout

COMP1CTL (Comparator 1 Control Register) Offset: 0x308 Default: 0x00000000 Access: COMP -> COMP1CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				SYNCSEL			REFSEL
7	6	5	4	3	2	1	0
REFSEL	INSEL		HYSTSEL		PHCMPEN	ENH	ENL

Table 15-17: Comparator 1 Control Register (COMP1CTL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11:9	SYNCSEL	RW	0x0	Select the synchronous output from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output 101: PWM5 synchronous output 110: 111:
8:7	REFSEL	RW	0x0	COMP1 reference select 00: PGA1 positive output when PHCMPEN = 1; DAC0 as too-high reference and DAC1 as too-low reference when PHCMPEN = 0 01: PGA1 negative output when PHCMPEN = 1; DAC1 as too-high reference and DAC0 as too-low reference when PHCMPEN = 0 10: ADC2 input from GPIO2 when PHCMPEN = 1; DAC2 as too-high reference and DAC3 as too-

Bits	Field Name	Type	Reset	Description
				low reference when PHCMPEN = 0 11: ADC10 input from GPIO10 when PHCMPEN = 1; DAC3 as too-high reference and DAC2 as too-low reference when PHCMPEN = 0
6:5	INSEL	RW	0x0	COMP1 input select 00: PGA1 negative output 01: PGA1 positive output 10: ADC0 input from GPIO0 11: ADC8 input from GPIO8
4:3	HYSTSEL	RW	0x0	COMP1 hysterisis select 00: 0 mV 01: 12 mV 10: 24 mV 11: 36 mV
2	PHCMPEN	RW	0x0	COMP1 phase comparison enable 0: Normal operation 1: Phase comparison mode
1	ENH	RW	0x0	COMP1H enable 0: Disable COMP1H 1: Enable COMP1H
0	ENL	RW	0x0	COMP1L enable 0: Disable COMP1L 1: Enable COMP1L

Table 15-18: COMP1L Control Register (COMP1LCTL) Layout

COMP1LCTL (COMP1L Control Register) Offset: 0x30C Default: 0x00000088 Access: COMP -> COMP1LCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-19: COMP1L Control Register (COMP1LCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP1L status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP1L status 1: PWMSYNC will clear the latched COMP1L status
7	POL4GPIO	RW	0x1	COMP1L output polarity for GPIO 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
6:4	SEL4GPIO	RW	0x0	COMP1L output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP1L output polarity for PWM 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
2:0	SEL4PWM	RW	0x0	COMP1L output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output

Bits	Field Name	Type	Reset	Description
				011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-20: COMP1H Control Register (COMP1HCTL) Layout

COMP1HCTL (COMP1H Control Register) Offset: 0x310 Default: 0x00000088
Access: COMP -> COMP1HCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-21: COMP1H Control Register (COMP1HCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP1H status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP1H status 1: PWMSYNC will clear the latched COMP1H status
7	POL4GPIO	RW	0x1	COMP1H output polarity for GPIO 0: Output 0 when the comparator input is higher than the reference

Bits	Field Name	Type	Reset	Description
				1: Output 1 when the comparator input is higher than the reference
6:4	SEL4GPIO	RW	0x0	COMP1H output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP1H output polarity for PWM 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
2:0	SEL4PWM	RW	0x0	COMP1H output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-22: Comparator 2 Control Register (COMP2CTL) Layout

COMP2CTL (Comparator 2 Control Register) Offset: 0x314 Default: 0x00000000
Access: COMP -> COMP2CTL.all

31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				SYNCSEL			REFSEL
7	6	5	4	3	2	1	0
REFSEL	INSEL		HYSTSEL		PHCMPEN	ENH	ENL

Table 15-23: Comparator 2 Control Register (COMP2CTL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11:9	SYNCSEL	RW	0x0	Select the synchronous output from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output 101: PWM5 synchronous output 110: 111:
8:7	REFSEL	RW	0x0	COMP2 reference select 00: PGA2 positive output when PHCMPEN = 1; DAC0 as too-high reference and DAC1 as too-low reference when PHCMPEN = 0 01: PGA2 negative output when PHCMPEN = 1; DAC1 as too-high reference and DAC0 as too-low reference when PHCMPEN = 0 10: ADC3 input from GPIO3 when PHCMPEN = 1; DAC2 as too-high reference and DAC3 as too-low reference when PHCMPEN = 0 11: ADC11 input from GPIO11 when PHCMPEN = 1; DAC3 as too-high reference and DAC2 as too-low reference when PHCMPEN = 0
6:5	INSEL	RW	0x0	COMP2 input select 00: PGA2 negative output 01: PGA2 positive output 10: ADC0 input from GPIO0 11: ADC8 input from GPIO8
4:3	HYSTSEL	RW	0x0	COMP2 hysteresis select 00: 0 mV 01: 12 mV 10: 24 mV 11: 36 mV
2	PHCMPEN	RW	0x0	COMP2 phase comparison enable 0: Normal operation 1: Phase comparison mode
1	ENH	RW	0x0	COMP2H enable 0: Disable COMP2H 1: Enable COMP2H
0	ENL	RW	0x0	COMP2L enable 0: Disable COMP2L 1: Enable COMP2L

Table 15-24: COMP2L Control Register (COMP2LCTL) Layout

COMP2LCTL (COMP2L Control Register) Offset: 0x318 Default: 0x00000088 Access: COMP -> COMP2LCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-25: COMP2L Control Register (COMP2LCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP2L status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP2L status 1: PWMSYNC will clear the latched COMP2L status
7	POL4GPIO	RW	0x1	COMP2L output polarity for GPIO 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
6:4	SEL4GPIO	RW	0x0	COMP2L output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one

Bits	Field Name	Type	Reset	Description
				111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP2L output polarity for PWM 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
2:0	SEL4PWM	RW	0x0	COMP2L output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-26: COMP2H Control Register (COMP2HCTL) Layout

COMP2HCTL (COMP2H Control Register) Offset: 0x31C Default: 0x00000088 Access: COMP -> COMP2HCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-27: COMP2H Control Register (COMP2HCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples

Bits	Field Name	Type	Reset	Description
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP2H status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP2H status 1: PWMSYNC will clear the latched COMP2H status
7	POL4GPIO	RW	0x1	COMP2H output polarity for GPIO 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
6:4	SEL4GPIO	RW	0x0	COMP2H output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP2H output polarity for PWM 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
2:0	SEL4PWM	RW	0x0	COMP2H output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-28: Comparator 3 Control Register (COMP3CTL) Layout

COMP3CTL (Comparator 3 Control Register) Offset: 0x320 Default: 0x00000000 Access: COMP -> COMP3CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				SYNCSEL			REFSEL
7	6	5	4	3	2	1	0
REFSEL	INSEL		HYSTSEL		PHCMPEN	ENH	ENL

Table 15-29: Comparator 3 Control Register (COMP3CTL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11:9	SYNCSEL	RW	0x0	Select the synchronous output from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output 101: PWM5 synchronous output 110: 111:
8:7	REFSEL	RW	0x0	COMP3 reference select 00: ADC1 input from GPIO1 when PHCMPEN = 1; DAC0 as too-high reference and DAC1 as too-low reference when PHCMPEN = 0 01: ADC5 input from GPIO5 when PHCMPEN = 1; DAC1 as too-high reference and DAC0 as too-low reference when PHCMPEN = 0 10: ADC9 input from GPIO9 when PHCMPEN = 1; DAC2 as too-high reference and DAC3 as too-low reference when PHCMPEN = 0 11: ADC13 input from GPIO13 when PHCMPEN = 1; DAC3 as too-high reference and DAC2 as too-low reference when PHCMPEN = 0
6:5	INSEL	RW	0x0	COMP3 input select 00: ADC0 input from GPIO0 01: ADC4 input from GPIO4 10: ADC8 input from GPIO8 11: ADC12 input from GPIO12
4:3	HYSTSEL	RW	0x0	COMP3 hysteresis select 00: 0 mV

Bits	Field Name	Type	Reset	Description
				01: 12 mV 10: 24 mV 11: 36 mV
2	PHCMPEN	RW	0x0	COMP3 phase comparison enable 0: Normal operation 1: Phase comparison mode
1	ENH	RW	0x0	COMP3H enable 0: Disable COMP3H 1: Enable COMP3H
0	ENL	RW	0x0	COMP3L enable 0: Disable COMP3L 1: Enable COMP3L

Table 15-30: COMP3L Control Register (COMP3LCTL) Layout

COMP3LCTL (COMP3L Control Register) Offset: 0x324 Default: 0x00000088
Access: COMP -> COMP3LCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-31: COMP3L Control Register (COMP3LCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP3L status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP3L status

Bits	Field Name	Type	Reset	Description
				1: PWMSYNC will clear the latched COMP3L status
7	POL4GPIO	RW	0x1	COMP3L output polarity for GPIO 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
6:4	SEL4GPIO	RW	0x0	COMP3L output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP3L output polarity for PWM 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
2:0	SEL4PWM	RW	0x0	COMP3L output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-32: COMP3H Control Register (COMP3HCTL) Layout

COMP3HCTL (COMP3H Control Register) Offset: 0x328 Default: 0x00000088 Access: COMP -> COMP3HCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-33: COMP3H Control Register (COMP3HCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP3H status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP3H status 1: PWMSYNC will clear the latched COMP3H status
7	POL4GPIO	RW	0x1	COMP3H output polarity for GPIO 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
6:4	SEL4GPIO	RW	0x0	COMP3H output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one

Bits	Field Name	Type	Reset	Description
				111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP3H output polarity for PWM 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
2:0	SEL4PWM	RW	0x0	COMP3H output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-34: Comparator 4 Control Register (COMP4CTL) Layout

COMP4CTL (Comparator 4 Control Register) Offset: 0x32C Default: 0x00000000 Access: COMP -> COMP4CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				SYNCSEL			REFSEL
7	6	5	4	3	2	1	0
REFSEL	INSEL		HYSTSEL		PHCMPEN	ENH	ENL

Table 15-35: Comparator 4 Control Register (COMP4CTL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11:9	SYNCSEL	RW	0x0	Select the synchronous output from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output 101: PWM5 synchronous output

Bits	Field Name	Type	Reset	Description
				110: 111:
8:7	REFSEL	RW	0x0	COMP4 reference select 00: ADC3 input from GPIO3 when PHCMPEN = 1; DAC0 as too-high reference and DAC1 as too-low reference when PHCMPEN = 0 01: ADC7 input from GPIO7 when PHCMPEN = 1; DAC1 as too-high reference and DAC0 as too-low reference when PHCMPEN = 0 10: ADC11 input from GPIO11 when PHCMPEN = 1; DAC2 as too-high reference and DAC3 as too-low reference when PHCMPEN = 0 11: ADC15 input from GPIO15 when PHCMPEN = 1; DAC3 as too-high reference and DAC2 as too-low reference when PHCMPEN = 0
6:5	INSEL	RW	0x0	COMP4 input select 00: ADC2 input from GPIO2 01: ADC6 input from GPIO6 10: ADC10 input from GPIO10 11: ADC14 input from GPIO14
4:3	HYSTSEL	RW	0x0	COMP4 hysteresis select 00: 0 mV 01: 12 mV 10: 24 mV 11: 36 mV
2	PHCMPEN	RW	0x0	COMP4 phase comparison enable 0: Normal operation 1: Phase comparison mode
1	ENH	RW	0x0	COMP4H enable 0: Disable COMP4H 1: Enable COMP4H
0	ENL	RW	0x0	COMP4L enable 0: Disable COMP4L 1: Enable COMP4L

Table 15-36: COMP4L Control Register (COMP4LCTL) Layout

COMP4LCTL (COMP4L Control Register) Offset: 0x330 Default: 0x00000088 Access: COMP -> COMP4LCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-37: COMP4L Control Register (COMP4LCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP4L status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP4L status 1: PWMSYNC will clear the latched COMP4L status
7	POL4GPIO	RW	0x1	COMP4L output polarity for GPIO 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
6:4	SEL4GPIO	RW	0x0	COMP4L output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one

Bits	Field Name	Type	Reset	Description
				111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP4L output polarity for PWM 0: Output 0 when the comparator input is lower than the reference 1: Output 1 when the comparator input is lower than the reference
2:0	SEL4PWM	RW	0x0	COMP4L output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-38: COMP4H Control Register (COMP4HCTL) Layout

COMP4HCTL (COMP4H Control Register) Offset: 0x334 Default: 0x00000088
Access: COMP -> COMP4HCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_30		FLTRST	FLTTH				
23	22	21	20	19	18	17	16
FLTWIN					FLTDIV		
15	14	13	12	11	10	9	8
FLTDIV							SYNCCLREN
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 15-39: COMP4H Control Register (COMP4HCTL) Description

Bits	Field Name	Type	Reset	Description
31:30	RESERVED_31_30	RO	0x0	Reserved.
29	FLTRST	W1S	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
28:24	FLTTH	RW	0x0	Filter's threshold is (FLTTH+1) 1's/0's for output to toggle
23:19	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples

Bits	Field Name	Type	Reset	Description
18:9	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
8	SYNCCLREN	RW	0x0	Enable latched COMP4H status clear by PWMSYNC 0: PWMSYNC does not affect the latched COMP4H status 1: PWMSYNC will clear the latched COMP4H status
7	POL4GPIO	RW	0x1	COMP4H output polarity for GPIO 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
6:4	SEL4GPIO	RW	0x0	COMP4H output select for GPIO 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output
3	POL4PWM	RW	0x1	COMP4H output polarity for PWM 0: Output 0 when the comparator input is higher than the reference 1: Output 1 when the comparator input is higher than the reference
2:0	SEL4PWM	RW	0x0	COMP4H output select for PWM 000: Original output 001: Synchronous output with ADC clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with ADC clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with ADC clock or Latched digital filter output

Table 15-40: DAC0 Control Register (DAC0CTL) Layout

DAC0CTL (DAC0 Control Register) Offset: 0x35C Default: 0x00000002 Access: COMP -> DAC0CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				DBGRUN		COMPHSEL	
7	6	5	4	3	2	1	0
COMPHSEL	RAMPLOAD	RAMPEN	SYNCSEL			CODELOAD	EN

Table 15-41: DAC0 Control Register (DAC0CTL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11:10	DBGRUN	RW	0x0	Behavior on CPU halted or lockup 00: Stop immediately 01: Stop at PWM synchronous event 10: Free running 11:
9:7	COMPHSEL	RW	0x0	Select COMPH from COMP[COMPHSEL] Note: This field is used for RAMP function. 000: COMP0H 001: COMP1H 010: COMP2H 011: COMP3H 100: COMP4H 101: 110: 111:
6	RAMPLOAD	RW	0x0	DAC Ramp load mode 0: Register shadowing is applied for ramping counter load from RAMPMAX 1: RAMP counter is directly loaded from the latest RAMPMAX
5	RAMPEN	RW	0x0	DAC Ramping enable 0: Disable ramping 1: Enable ramping. DAC0CODEA is equivalent to the 10 MSBs of RAMP0CNT
4:2	SYNCSEL	RW	0x0	PWMSYNC is selected from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output

Bits	Field Name	Type	Reset	Description
				101: PWM5 synchronous output 110: 111:
1	CODELOAD	RW	0x1	DAC code load mode 0: DAC code is updated upon PWMSYNC event 1: DAC code is immediately updated by writing to DAC0CODE
0	EN	RW	0x0	DAC enable 0: Disable 1: Enable

Table 15-42: DAC0 Code Register (DAC0CODE) Layout

DAC0CODE (DAC0 Code Register) Offset: 0x360 Default: 0x00000000
Access: COMP -> DAC0CODE.all

31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 15-43: DAC0 Code Register (DAC0CODE) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RW	0x0	DAC code When DAC0CTL[CODELOAD]=0, write to this field only affects the shadow code. When DAC0CTL[CODELOAD]=1, write to this field affects both the shadow code and the active code.

Table 15-44: DAC0 Active Code Register (DAC0CODEA) Layout

DAC0CODEA (DAC0 Active Code Register) Offset: 0x364 Default: 0x00000000 Access: COMP -> DAC0CODEA.all							
31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 15-45: DAC0 Active Code Register (DAC0CODEA) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RO	0x0	Active DAC code

Table 15-46: RAMP0 Delay Shadow Register (RAMP0DLY) Layout

RAMP0DLY (RAMP0 Delay Shadow Register) Offset: 0x368 Default: 0x00000000 Access: COMP -> RAMP0DLY.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-47: RAMP0 Delay Shadow Register (RAMP0DLY) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Delay from PWMSYNC to start of ramping

Table 15-48: RAMP0 Delay Active Register (RAMP0DLYA) Layout

RAMP0DLYA (RAMP0 Delay Active Register) Offset: 0x36C Default: 0x00000000 Access: COMP -> RAMP0DLYA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-49: RAMP0 Delay Active Register (RAMP0DLYA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Delay from PWMSYNC to start of ramping

Table 15-50: RAMP0 Decrement Shadow Register (RAMP0DEC) Layout

RAMP0DEC (RAMP0 Decrement Shadow Register) Offset: 0x370 Default: 0x00000000 Access: COMP -> RAMP0DEC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-51: RAMP0 Decrement Shadow Register (RAMP0DEC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Step size for decremental ramping

Table 15-52: RAMP0 Decrement Active Register (RAMP0DECA) Layout

RAMP0DECA (RAMP0 Decrement Active Register) Offset: 0x374 Default: 0x00000000 Access: COMP -> RAMP0DECA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-53: RAMP0 Decrement Active Register (RAMP0DECA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Step size for decremental ramping

Table 15-54: RAMP0 Maximum Value Shadow Register (RAMP0MAX) Layout

RAMP0MAX (RAMP0 Maximum Value Shadow Register) Offset: 0x378 Default: 0x00000000 Access: COMP -> RAMP0MAX.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-55: RAMP0 Maximum Value Shadow Register (RAMP0MAX) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Maximum value when reset ramping

Table 15-56: RAMP0 Maximum Value Active Register (RAMP0MAXA) Layout

RAMP0MAXA (RAMP0 Maximum Value Active Register) Offset: 0x37C Default: 0x00000000 Access: COMP -> RAMP0MAXA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-57: RAMP0 Maximum Value Active Register (RAMP0MAXA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Maximum value when reset ramping

Table 15-58: RAMP0 Count Register (RAMP0CNT) Layout

RAMP0CNT (RAMP0 Count Register) Offset: 0x380 Default: 0x00000000 Access: COMP -> RAMP0CNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-59: RAMP0 Count Register (RAMP0CNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Count number of RAMP0

Table 15-60: DAC1 Control Register (DAC1CTL) Layout

DAC1CTL (DAC1 Control Register) Offset: 0x384 Default: 0x00000002 Access: COMP -> DAC1CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				DBGRUN		COMPHSEL	
7	6	5	4	3	2	1	0
COMPHSEL	RAMPLOAD	RAMPEN	SYNCSEL			CODELOAD	EN

Table 15-61: DAC1 Control Register (DAC1CTL) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11:10	DBGRUN	RW	0x0	Behavior on CPU halted or lockup 00: Stop immediately 01: Stop at PWM synchronous event 10: Free running 11:
9:7	COMPHSEL	RW	0x0	Select COMPH from COMP[COMPHSEL] Note: This field is used for RAMP function. 000: COMP0H 001: COMP1H 010: COMP2H 011: COMP3H 100: COMP4H 101: 110: 111:
6	RAMPLOAD	RW	0x0	DAC Ramp load mode 0: Register shadowing is applied for ramping counter load from RAMPMAX 1: RAMP counter is directly loaded from the latest RAMPMAX
5	RAMPEN	RW	0x0	DAC Ramping enable 0: Disable ramping 1: Enable ramping. DAC1CODEA is equivalent to the 10 MSBs of RAMP1CNT
4:2	SYNCSEL	RW	0x0	PWMSYNC is selected from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output

Bits	Field Name	Type	Reset	Description
				101: PWM5 synchronous output 110: 111:
1	CODELOAD	RW	0x1	DAC code load mode 0: DAC code is updated upon PWMSYNC event 1: DAC code is immediately updated by writing to DAC1CODE
0	EN	RW	0x0	DAC enable 0: Disable 1: Enable

Table 15-62: DAC1 Code Register (DAC1CODE) Layout

DAC1CODE (DAC1 Code Register) Offset: 0x388 Default: 0x00000000
Access: COMP -> DAC1CODE.all

31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 15-63: DAC1 Code Register (DAC1CODE) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RW	0x0	DAC code When DAC1CTL[CODELOAD]=0, write to this field only affects the shadow code. When DAC1CTL[CODELOAD]=1, write to this field affects both the shadow code and the active code.

Table 15-64: DAC1 Active Code Register (DAC1CODEA) Layout

DAC1CODEA (DAC1 Active Code Register) Offset: 0x38C Default: 0x00000000 Access: COMP -> DAC1CODEA.all							
31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 15-65: DAC1 Active Code Register (DAC1CODEA) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RO	0x0	Active DAC code

Table 15-66: RAMP1 Delay Shadow Register (RAMP1DLY) Layout

RAMP1DLY (RAMP1 Delay Shadow Register) Offset: 0x390 Default: 0x00000000 Access: COMP -> RAMP1DLY.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-67: RAMP1 Delay Shadow Register (RAMP1DLY) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Delay from PWMSYNC to start of ramping

Table 15-68: RAMP1 Delay Active Register (RAMP1DLYA) Layout

RAMP1DLYA (RAMP1 Delay Active Register) Offset: 0x394 Default: 0x00000000 Access: COMP -> RAMP1DLYA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-69: RAMP1 Delay Active Register (RAMP1DLYA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Delay from PWMSYNC to start of ramping

Table 15-70: RAMP1 Decrement Shadow Register (RAMP1DEC) Layout

RAMP1DEC (RAMP1 Decrement Shadow Register) Offset: 0x398 Default: 0x00000000 Access: COMP -> RAMP1DEC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-71: RAMP1 Decrement Shadow Register (RAMP1DEC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Step size for decremental ramping

Table 15-72: RAMP1 Decrement Active Register (RAMP1DECA) Layout

RAMP1DECA (RAMP1 Decrement Active Register) Offset: 0x39C Default: 0x00000000 Access: COMP -> RAMP1DECA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-73: RAMP1 Decrement Active Register (RAMP1DECA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Step size for decremental ramping

Table 15-74: RAMP1 Maximum Value Shadow Register (RAMP1MAX) Layout

RAMP1MAX (RAMP1 Maximum Value Shadow Register) Offset: 0x3A0 Default: 0x00000000 Access: COMP -> RAMP1MAX.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-75: RAMP1 Maximum Value Shadow Register (RAMP1MAX) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x0	Maximum value when reset ramping

Table 15-76: RAMP1 Maximum Value Active Register (RAMP1MAXA) Layout

RAMP1MAXA (RAMP1 Maximum Value Active Register) Offset: 0x3A4 Default: 0x00000000 Access: COMP -> RAMP1MAXA.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-77: RAMP1 Maximum Value Active Register (RAMP1MAXA) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Maximum value when reset ramping

Table 15-78: RAMP1 Count Register (RAMP1CNT) Layout

RAMP1CNT (RAMP1 Count Register) Offset: 0x3A8 Default: 0x00000000 Access: COMP -> RAMP1CNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 15-79: RAMP1 Count Register (RAMP1CNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RO	0x0	Count number of RAMP1

Table 15-80: DAC2 Control Register (DAC2CTL) Layout

DAC2CTL (DAC2 Control Register) Offset: 0x3AC Default: 0x00000002 Access: COMP -> DAC2CTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_5							
23	22	21	20	19	18	17	16
RESERVED_31_5							
15	14	13	12	11	10	9	8
RESERVED_31_5							
7	6	5	4	3	2	1	0
RESERVED_31_5			SYNCSEL			CODELOAD	EN

Table 15-81: DAC2 Control Register (DAC2CTL) Description

Bits	Field Name	Type	Reset	Description
31:5	RESERVED_31_5	RO	0x0	Reserved.
4:2	SYNCSEL	RW	0x0	PWMSYNC is selected from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output 101: PWM5 synchronous output 110: 111:
1	CODELOAD	RW	0x1	DAC code load mode 0: DAC code is updated upon PWMSYNC event 1: DAC code is immediately updated by writing to DAC2CODE
0	EN	RW	0x0	DAC enable 0: Disable 1: Enable

Table 15-82: DAC2 Code Register (DAC2CODE) Layout

DAC2CODE (DAC2 Code Register) Offset: 0x3B0 Default: 0x00000000 Access: COMP -> DAC2CODE.all							
31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 15-83: DAC2 Code Register (DAC2CODE) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RW	0x0	DAC code When DAC2CTL[CODELOAD]=0, write to this field only affects the shadow code. When DAC2CTL[CODELOAD]=1, write to this field affects both the shadow code and the active code.

Table 15-84: DAC2 Active Code Register (DAC2CODEA) Layout

DAC2CODEA (DAC2 Active Code Register) Offset: 0x3B4 Default: 0x00000000
Access: COMP -> DAC2CODEA.all

31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 15-85: DAC2 Active Code Register (DAC2CODEA) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RO	0x0	Active DAC code

Table 15-86: DAC3 Control Register (DAC3CTL) Layout

DAC3CTL (DAC3 Control Register) Offset: 0x3B8 Default: 0x00000002
Access: COMP -> DAC3CTL.all

31	30	29	28	27	26	25	24
RESERVED_31_5							
23	22	21	20	19	18	17	16
RESERVED_31_5							
15	14	13	12	11	10	9	8
RESERVED_31_5							
7	6	5	4	3	2	1	0
RESERVED_31_5			SYNCSEL			CODELOAD	EN

Table 15-87: DAC3 Control Register (DAC3CTL) Description

Bits	Field Name	Type	Reset	Description
31:5	RESERVED_31_5	RO	0x0	Reserved.
4:2	SYNCSEL	RW	0x0	PWMSYNC is selected from PWM[SYNCSEL] 000: PWM0 synchronous output 001: PWM1 synchronous output 010: PWM2 synchronous output 011: PWM3 synchronous output 100: PWM4 synchronous output 101: PWM5 synchronous output 110: 111:
1	CODELOAD	RW	0x1	DAC code load mode 0: DAC code is updated upon PWMSYNC event 1: DAC code is immediately updated by writing to DAC3CODE
0	EN	RW	0x0	DAC enable 0: Disable 1: Enable

Table 15-88: DAC3 Code Register (DAC3CODE) Layout

DAC3CODE (DAC3 Code Register) Offset: 0x3BC Default: 0x00000000
Access: COMP -> DAC3CODE.all

31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 15-89: DAC3 Code Register (DAC3CODE) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RW	0x0	DAC code When DAC3CTL[CODELOAD]=0, write to this field only affects the shadow code. When DAC3CTL[CODELOAD]=1, write to this field affects both the shadow code and the active code.

Table 15-90: DAC3 Active Code Register (DAC3CODEA) Layout

DAC3CODEA (DAC3 Active Code Register) Offset: 0x3C0 Default: 0x00000000 Access: COMP -> DAC3CODEA.all							
31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 15-91: DAC3 Active Code Register (DAC3CODEA) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RO	0x0	Active DAC code

Table 15-92: DAC Buffer Control Register (DACBUFCTL) Layout

DACBUFCTL (DAC Buffer Control Register) Offset: 0x3F4 Default: 0x00000000 Access: COMP -> DACBUFCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_5							
23	22	21	20	19	18	17	16
RESERVED_31_5							
15	14	13	12	11	10	9	8
RESERVED_31_5							
7	6	5	4	3	2	1	0
RESERVED_31_5			OE13	OE10	INSEL		EN

Table 15-93: DAC Buffer Control Register (DACBUFCTL) Description

Bits	Field Name	Type	Reset	Description
31:5	RESERVED_31_5	RO	0x0	Reserved.
4	OE13	RW	0x0	Enable output to GPIO13 pin 0: Disable 1: Enable
3	OE10	RW	0x0	Enable output to GPIO10 pin 0: Disable 1: Enable
2:1	INSEL	RW	0x0	DAC buffer input select 00: DAC0 01: DAC1 10: DAC2 11: DAC3

Bits	Field Name	Type	Reset	Description
0	EN	RW	0x0	DAC buffer enable 0: Disable 1: Enable

Table 15-94: COMP Register Write-Allow Key Register (COMPREGKEY) Layout

COMPREGKEY (COMP Register Write-Allow Key Register) Offset: 0x3FC Default: 0x1ACCE551 Access: COMP -> COMPREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 15-95: COMP Register Write-Allow Key Register (COMPREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected COMP registers

16 Buffered 10-bit DAC

16.1 Buffered DAC overview

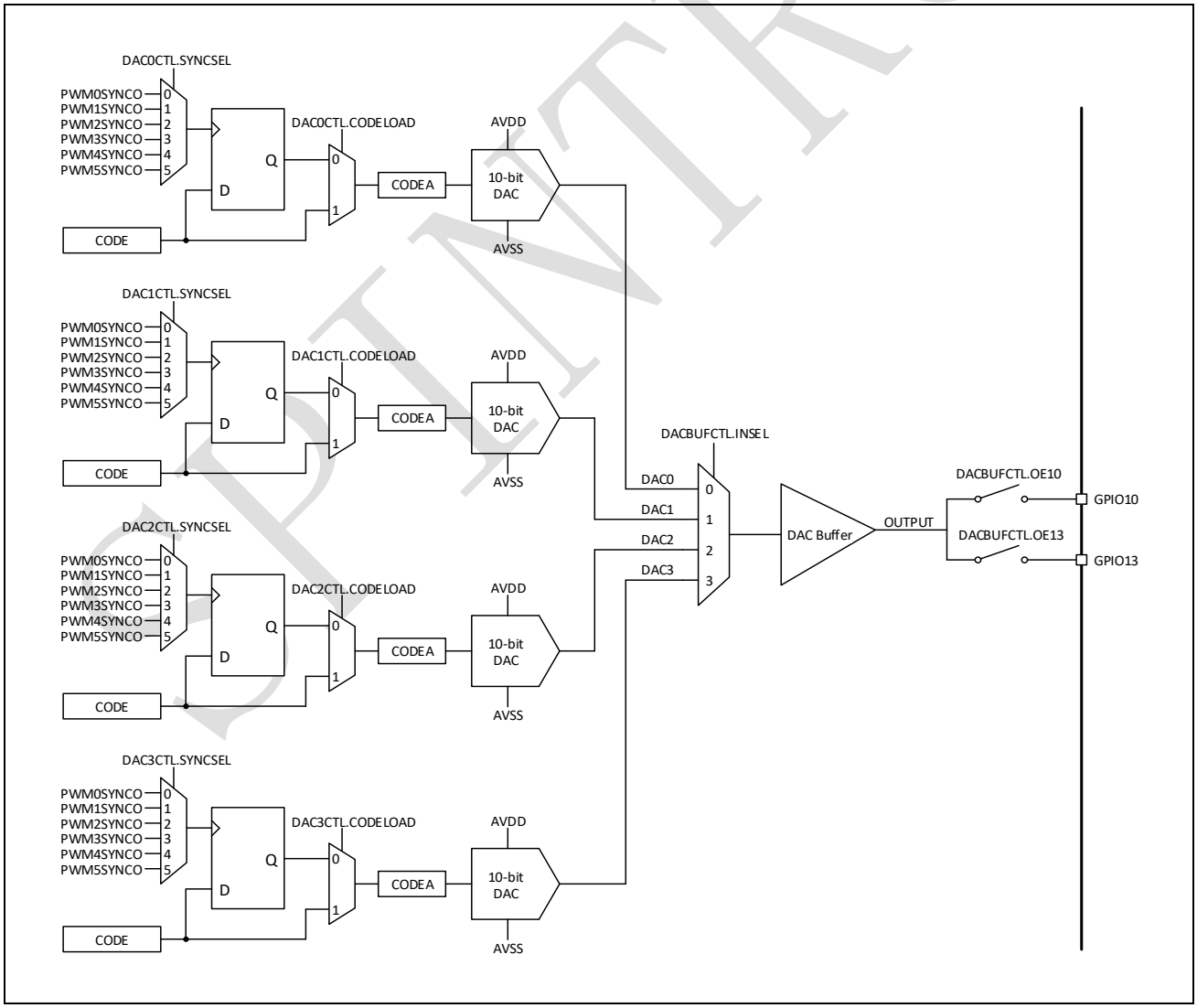
The buffered DAC module consists of four internal 10-bit DAC and an analog output buffer that is capable of driving external load. The buffered DAC is a general purpose DAC that can be configured to generate a DC voltage and AC waveforms such as triangle, square waves and so forth. Each DAC output is divided from AVDD pin with 10-bit resolution, shown as following equation:

$$\mathrm{VOUT} = \frac{AVDD}{2^{10}} * CODE \quad (Typically, AVDD = 3.3V)$$

16.2 Buffered DAC architecture

The block diagram for the buffered DAC is shown in Figure 16-1.

Figure 16-1: Buffered DAC block diagram



Buffered DAC module has two sets of DAC code registers: CODE and CODEA. CODEA is a read-only register that directly control the buffered DAC value, and CODE is a writable shadow register that loads into CODEA either immediately or synchronized with the next PWMSYNC event, depends on the DACxCTL.CODELOAD setting. If DACxCTL.CODELOAD=0, it's in shadow mode, the value written to CODE register will load to CODEA register on next PWMSYNC signal. And if DACxCTL.CODELOAD=1, it's in direct mode, the value written to CODE register will load into CODEA register immediately, then 10-bit DAC output will also change immediately.

The high and low thresholds of the comparator are generated by two DACs, selected from the Table 15-1.

SPD1148 implements one internal DAC buffer, each DAC's output can be sent to the DAC buffer depending on the DAC buffer MUX setting (DACBUFCTL.INSEL), shown as Table 16-1.

Table 16-1: DAC buffer MUX

DACBUFCTL.INSEL	DAC buffer output
0	DAC0
1	DAC1
2	DAC2
3	DAC3

DAC buffer output can be directly sent to the ADC through ADC input MUX (See in ADC channel selection: Sampler B, CHSEL1N=3). DAC buffer output can also be sent to the ADC10 pin or ADC13 pin, depends on the setting DACBUFCTL.OE10 and DACBUFCTL.OE13.

Example 15.2.1 DAC and DAC buffer configuration

Set DAC0 output to 1.65V, (AVDD=3.3V), send it to DAC buffer and use ADC to measure, also send it to ADC10 pin.

- Enable DAC0, set DAC0 output to 1.65V
- Enable DAC buffer, select DAC0 as input
- Set ADC input MUX, select DAC buffer as input
- Send DAC buffer output to ADC10 pin

Example 15.2.1

```
Void COMP_Example15_2_1(void)
{
COMP->DAC0CODE.all=512;/* Set DAC0 output to 1.65V */
COMP->DAC0CTL.bit.CODELOAD=1;/* direct mode, dac code immediately update */
COMP->DAC0CTL.bit.EN=1;/* Enable DAC0 */
COMP->DACBUFCTL.bit.INSEL=0;/* Send DAC0 to DAC buffer input */
ADC->ADCBGCTL.bit.EN=1;/* Enable ADC Bandgap */
COMP->DACBUFCTL.bit.EN=1;/* Enable DAC buffer */
ADC->ADCSOCCTL[0].bit.SHEN=2;/* Enable Sampler B */
ADC->ADCSOCCTL[0].bit.CHSELP=0;/* Sampler B Positive = GND */
ADC->ADCSOCCTL[0].bit.CHSELN=3;/* Sampler B Negative = DAC Buffer */
COMP->DACBUFCTL.bit.OE10=1;/* Send DAC buffer output to ADC10 pin */
}
```

16.3 Ramp generator

The ramp generator can produce a falling ramp reference (REFP voltage) for the too high compactor, see Figure 15-1. Only DAC0 and DAC1 have ramp capability.

Figure 16-2 shows ramp generator diagram. There's two sets of registers for ramp, one is read-only active registers, named RAMPDLYA, RAMPDECA, and RAMPMAXA, which can directly control ramp function. The other is a set of writable shadow registers called RAMPDLY, RAMPDEC, and RAMPMAX. Those shadow registers will load to read-only registers only under certain conditions, for example rising edge of RAMPEN or PWMSYNC, as shown in Figure 16-2.

The most significant 10bits of RAMPCNT will be sent to register CODEA, which controls 10-bit DAC output. RAMPMAXA is initial DAC value, RAMPDLYA is delay counter. For every PWMCLK cycle, RAMPDLYA will decrease one until it reaches zero. RAMPDECA controls DAC decrease step, for each time RAMPDLAYA is zero RAMPCNT value will subtract RAMPDECA.

When ramp generator is enabled (RAMPEN=1), RAMPCNT register will first load from RAMPMAX, and the register remains stable until the PWMSYNC is triggered. Then for every PWMCLK cycle, RAMPDLYA will decrease one. When RAMPDLYA reaches 0, RAMPCNT value will subtract RAMPDECA. Afterwards RAMPDLYA will load initial value for the next PWMCLK cycle and repeat. Therefore, DAC output will decrease RAMPDECA step for each (RAMPDLYA+1) PWMCLK cycles.

When RAMPLOAD=0, ramp generator block is in shadow mode. This means that while COMPxH is asserted, RAMPMAX value will not load to RAMPMAXA immediately. RAMPCNT will be controlled by the previous RAMPMAXA value, and updated RAMPMAX value will load to RAMPMAXA only at the next PWMSYNC rising edge. If RAMPLOAD=1, ramp generator block is in direct mode, which means that when COMPxH is asserted, RAMPMAX value will directly load to RAMPMAXA and control RAMPCNT.

Figure 16-2: Ramp generator diagram

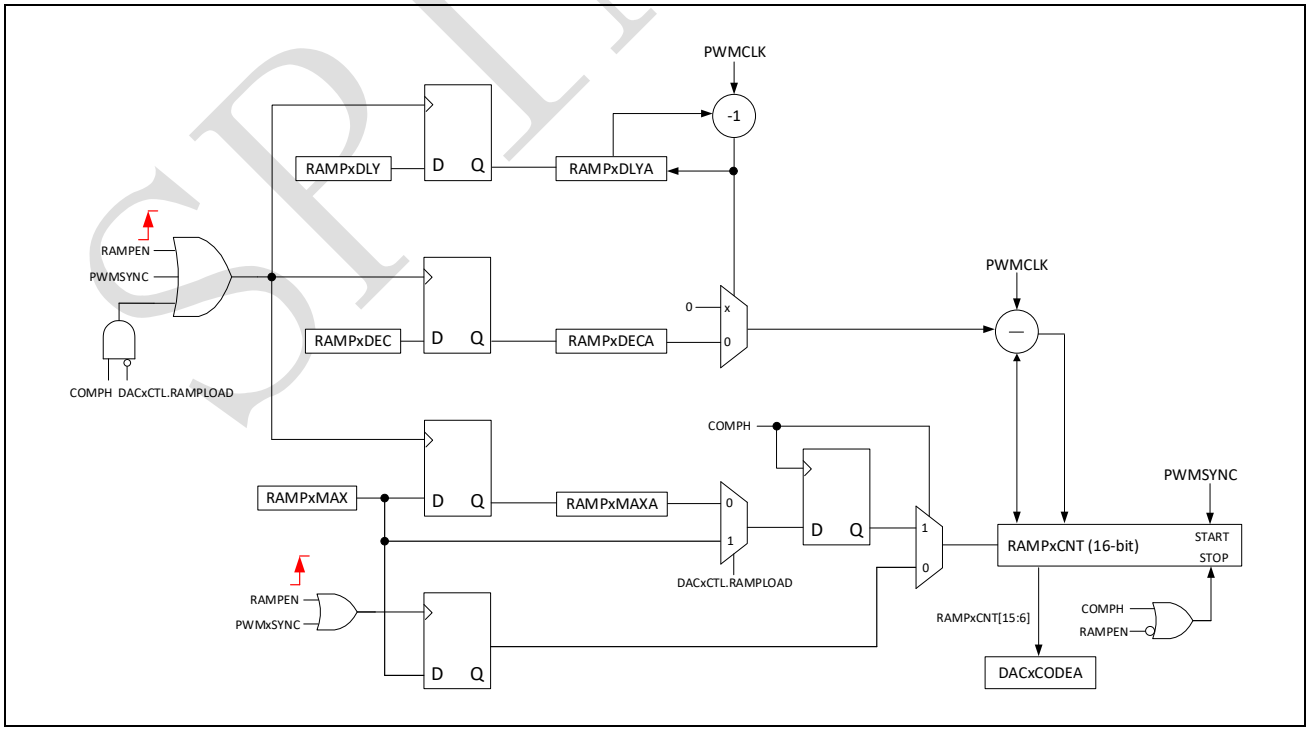
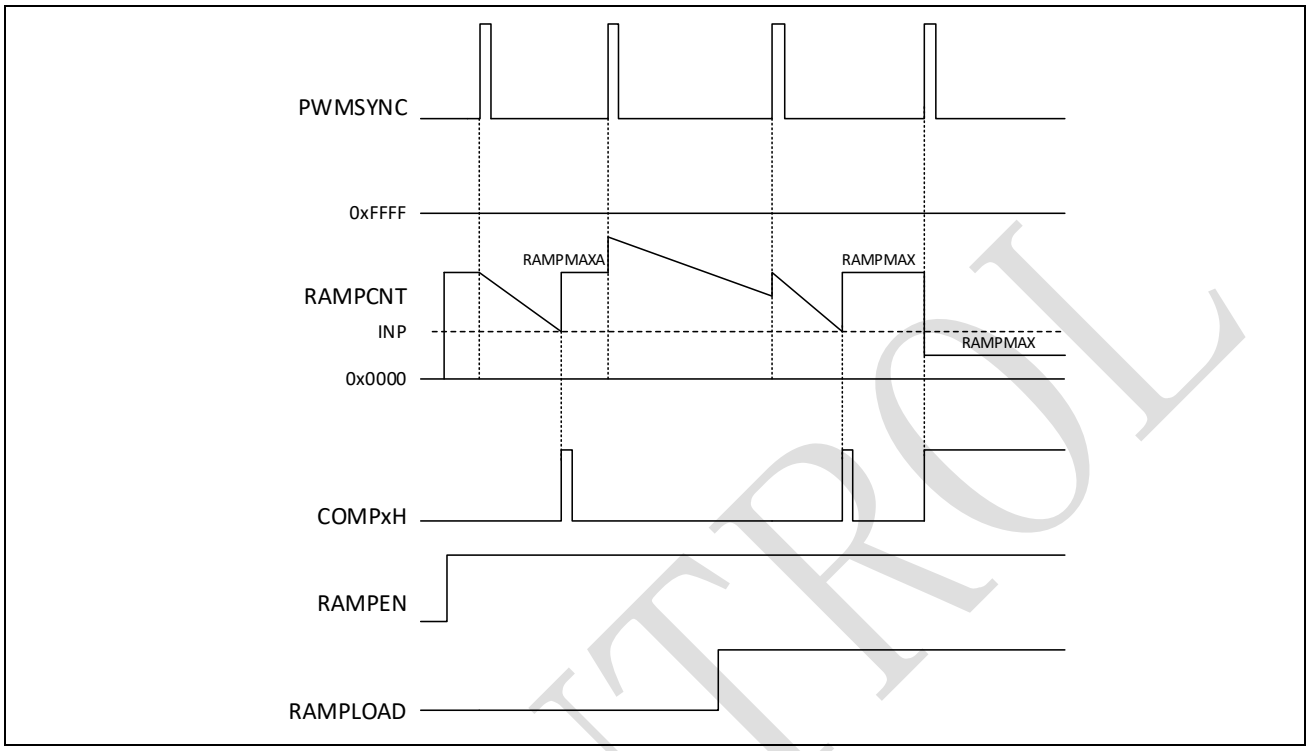


Figure 16-3 shows an example for DAC output (same as RAMPCNT) during PWMSYNC triggered and COMPxH asserted. Suppose the positive input to the too high comparator is fixed (INP in Figure 16-3).

Figure 16-3: Ramp generator waveform



16.4 Power-up sequence

When powering up the DAC or DAC buffer, the following sequence must be used:

Step 1: Power up ADC bandgap

Step 2: Power up the analog circuits of DAC or DAC buffer

16.5 Registers

Please see Section 15.6 for DAC related register map and definitions.

17 UART

17.1 UART overview

The SPD1148 implements one Universal Asynchronous Receiver/Transmitter (UART) port. The port contains a UART and a slow serial infrared Transmit encoder and Receive decoder that conform to the IrDA serial infrared specification.

The UART port performs serial-to-parallel conversion on data characters received from a peripheral device and parallel-to-serial conversion on data characters received from CPU.

The CPU can read a complete UART status for the Line Status Register. Status information includes the type and condition of transfer operations and error conditions (parity, overrun, framing or break interrupt) associated with UART.

The UART port operates in either FIFO or non-FIFO mode. In FIFO mode, a 64-byte Transmit FIFO holds data from the CPU until it is transmitted on the serial link; a 64-byte Receive FIFO buffers data from the serial link until it is read by the CPU. In non-FIFO mode, the Transmit and Receive FIFOs are bypassed, and the Transmit Holding Register and Receive buffer Register are used instead.

The UART includes a programmable baud-rate generator that can divide the input clock by 1 to (2^{16}-1), which produces a 16X clock that can be used to drive the internal transmit and receive logic. Software can program interrupts to meet its requirements, which minimizes the number of computations required to handle the communications link. UART operates in an environment that is either controlled by software and can be polled or is interrupt driven.

The UART port supports baud rates of 9600, 19.2K, 38.4K, 57.6K, 115.2K, 230K, 460K, 921K, 1.8M and 3.6M.

17.2 UART features

The UART has the following features:

- Ability to add or delete standard asynchronous communication bits (start, stop and parity) in the serial data
- Independently controlled transmit, receive, line status and data-set interrupts
- Programmable serial interface:
  - 5 – 8 data bits
  - Even, odd or no parity detection
  - One, one-and-a-half, or two stop bits generation
  - Baud-rate generation up to 12.5 Mbps
  - False start-bit detection
- 64-byte Transmit FIFO
- 64-byte Receive FIFO
- Complete status-reporting capability
- Ability to generate and detect line breaks
- Internal diagnostic capabilities that include:
  - Loopback controls for communications link fault isolation
  - Break, parity and framing-error simulation
- Fully prioritized interrupt system controls
- Serial Infrared asynchronous interface that conform to the Infrared Data Association(IrDA) specification

17.3 UART signal descriptions

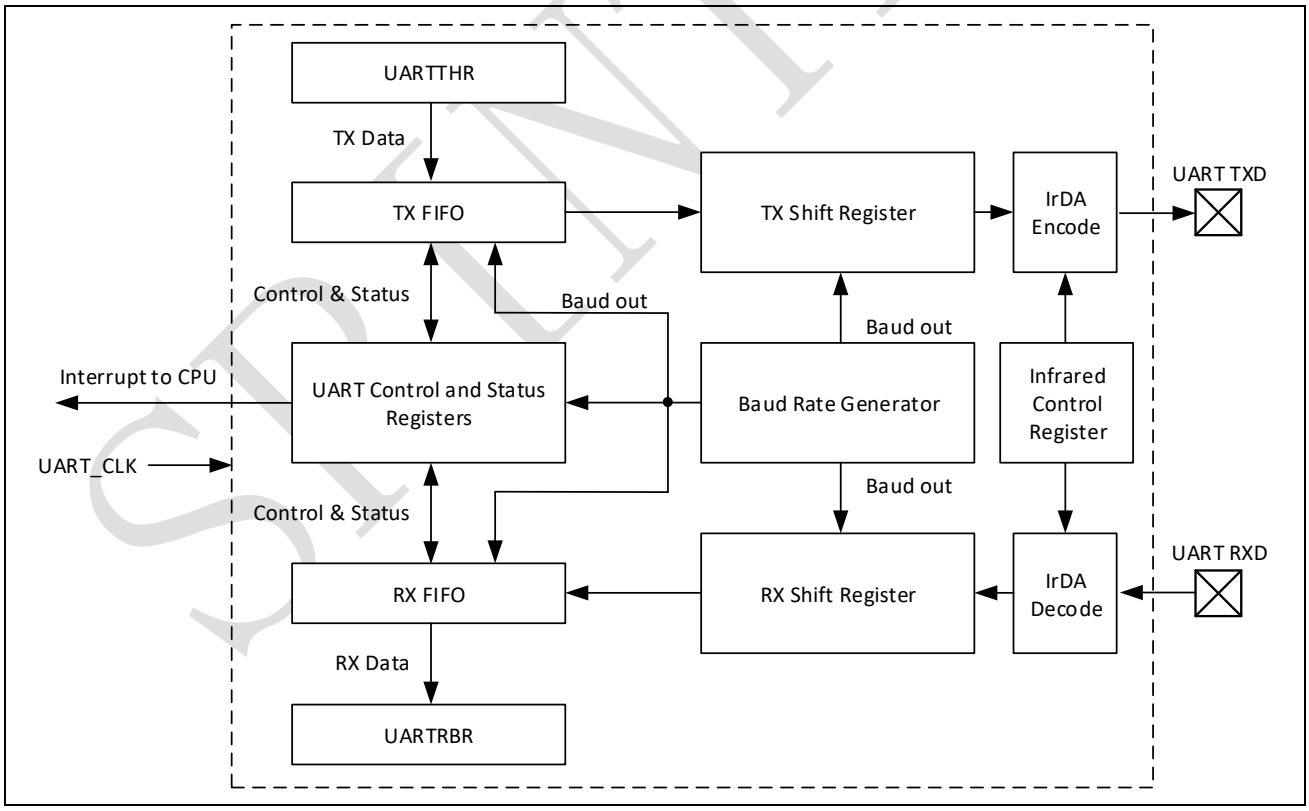
Table 17-1 lists and describes each external signal that is connected to a UART module. The pins transmit digital CMOS-level signals and are connected to through GPIOs. Please refer to the GPIO pin configuration registers for details.

Table 17-1: UART signal descriptions

Name	Type	Descriptions
RXD	Input	**Serial Input** Serial data input to the Receive Shift register; in Infrared mode, it is connected to the infrared receiver input
TXD	Output	**Serial Output** Serial data output to the communication-link peripheral or data set. The TXD signal is set to the logic 1 state upon a reset operation. It is connected to the output of the infrared transmitter in infrared mode.

The block diagram is shown in Figure 17-1.

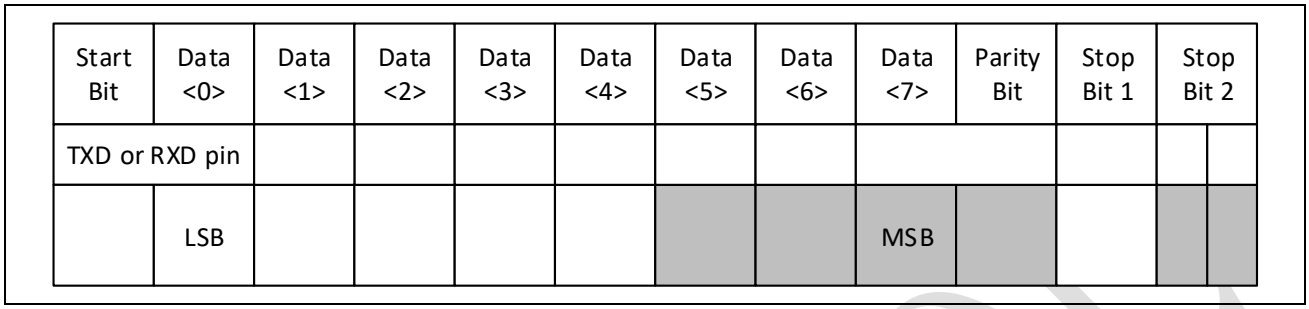
Figure 17-1: UART block diagram



17.4 UART operation

Figure 17-2 shows the format of a UART data frame.

Figure 17-2: Example UART data frame



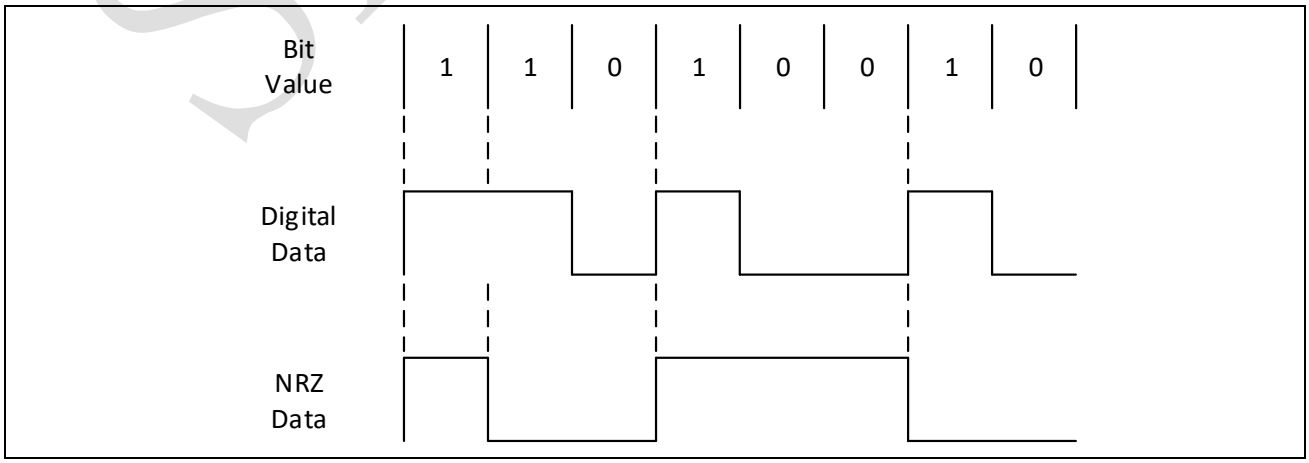
The receive-data sample-counter frequency is 16 times the value of the bit frequency. The 16X clock is created by the baud-rate generator. Each bit is sampled three times in the middle. Shaded bits in Figure 17-2 are optional and can be programmed by software.

Each data Frame is between 9 and 11 bits long, depending on the size of the data programmed, whether parity is enabled. A data frame begins by transmitting a start bit that is represented by a high to low transition. The start bit is followed by 8 bits of data that begin with the Least Significant bit (LSB). The data bits are followed by an optional parity bit. The parity bit is set if: even parity is enabled and the data byte has an odd number of ones or if odd parity is enabled and the data byte has an even number of ones. The data frame ends with 1 stop bit. The stop bit is represented by 1 successive bit period of logic one.

The UART module has two FIFOs: one Transmit and one Receive. The Transmit FIFO is 64 bytes deep and 8 bits wide. The Receive FIFO is 64 bytes deep and 11 bits wide. Three bits are used for tracking errors.

The UART can use NRZ coding to represent individual bit values. To enable NRZ coding, set the UARTIER.NRZME field in the Interrupt Enable Register. A bit value of 1 is represented by a line transition, and 0 is represented by no line transition. Figure 17-3 shows the data byte 0b0100_1011 in NRZ coding. The LSB in the byte is transmitted first.

Figure 17-3: Example NRZ bit Encoding – 0b0100_1011



17.4.1 Reset

The UART is disabled on reset. To enable it, software must program the GPIO Pin Configuration registers, and then set the UARTIER.UUE field in the Interrupt Enable Register. When the UART is enabled, the receiver waits for a frame start bit and the transmitter sends data if it is available in the Transmit Holding Register. Transmit data can be written to the Transmit Holding Register before the UART unit is enabled. In FIFO mode, data is transmitted from the FIFO to the Transmit Holding Register before it goes to the pin.

When the UART unit is disabled, the transmitter or receiver finishes the current byte and stops transmitting or receiving more data. Data in the FIFO is not cleared and transmission resumes when the UART is enabled.

17.4.2 FIFO operation

The UART has a Transmit FIFO and a Receive FIFO, with each FIFO holding 64 characters of data. There is one method for moving data into and out of the FIFOs: Program I/O. In Program I/O mode, polling is used.

17.4.2.1 FIFO interrupt mode operation

The UARTMCR.OUT2 field in the Modem Control Register is a global UART interrupt enable bit and must be set to enable UART interrupts.

17.4.2.2 Receive interrupt

For a receive interrupt to occur, the Receive FIFO and receive interrupts must be enabled. The UARTIIR.IID field in the Interrupt Identification Register changes to show that receive data is available when the FIFO reaches its trigger threshold. The UARTIIR.IID field changes to show the next waiting interrupt when the FIFO drops below the trigger threshold. A change in the UARTIIR.IID field triggers an interrupt to the core. Software reads the UARTIIR.IID field to determine the cause of the interrupt.

The receive-line-status interrupt has the highest priority; the received-data-available interrupt is lower. The line-status interrupt occurs only when the character at the front of the FIFO has errors.

The UARTLSR.DR field in the Line Status Register is set when a character is transferred from the Shift register to the Receive FIFO. UARTLSR.DR is cleared when the FIFO is empty.

Note: If the receive interrupt is enabled but the Receive FIFO is not enabled, a receive interrupt occurs upon a receive data is available in the Receive Buffer Register (UARTRBR).

17.4.2.3 Character timeout interrupt

A character timeout interrupt occurs when the receive FIFO and receive timeout interrupt are enabled and all of the following conditions exist:

- At least one character is in the FIFO.
- The most recently received character was received more than four continuous character times ago.
- The most recent FIFO read was performed more than four continuous character times ago.

After the reads one character from the Receive FIFO or a new start bit is received, the timeout interrupt is cleared, and the timeout is reset. If a timeout interrupt has not occurred, the timeout is reset when a new character is received or the reads the Receive FIFO.

17.4.2.4 Transmit interrupt

Send interrupts occur when FIFO is enabled and send interrupts are enabled. When the FIFO is off, a single write to the THR register creates a new send interrupt. The transmit data-request interrupt occurs when the Transmit FIFO is at least half empty. The interrupt is cleared when the Transmit Holding Register is written or the Interrupt Identification Register is read.

17.4.2.5 Removing trailing bytes

The CPU must remove trailing bytes. The presence of trailing bytes is signaled by the assertion of a character timeout interrupt. When servicing a character timeout interrupt, the CPU uses the following procedure:

1. Read the Line Status Register and check for errors.
2. Disable the receiver timeout interrupt via UARTIER.RTOIE field in the Interrupt Enable Register.
3. Read data from the UART FIFO.
4. Read the Line Status Register, check for errors, and loop back to step 3. If the UARTLSR.DR field is set, go to step 5.
5. No more data in FIFO: Re-enable RTO interrupt via the UARTIER.RTOIE field in the Interrupt Enable Register.
6. Done.

17.4.2.6 FIFO polled mode operation

When the FIFOs are enabled, clearing the bits [4:0] in the Interrupt Enable Register places the port in FIFO polled operating mode. The receiver and the transmitter are controlled separately. Either one or both can be in polled mode. In polled mode, software checks receiver and transmitter status via the Line Status Register. The CPU polls the following bits for receive and transmit data service:

- Receive Data Service – the CPU checks the UARTLSR.DR field, which is set when one or more bytes remain in the Receive FIFO or Receive Buffer Register.
- Transmit Data Service – the CPU checks the UARTLSR.TDRQ field in the Line Status Register, which is set when the transmitter needs data.

The CPU can also check the UARTLSR.TXEMPTY field in the Line Status Register, which is set when the Transmit FIFO is empty.

17.4.3 Auto-baud-rate detection

The UART supports auto-baud-rate detection. When enabled, the UART counts the numbers of clock cycles within the start-bit pulse. This number is then written into the Auto-Baud Count Register and is used to calculate the baud rate. When the Auto-Baud Count Register is written, an auto-baud-lock interrupt is generated (if enabled), and the UART automatically programs the Divisor Latch Registers with the appropriate baud rate. If preferred, the CPU can read the Auto-Baud Count Register and use this information to program the Divisor Latch Low Byte Register and Divisor Latch High Byte Register with a baud rate calculated by the CPU. After the baud rate has been programmed, the CPU verifies that the predetermined characters (usually AT or at) are being received correctly.

If the UART is to program the Divisor Latch Registers, software can use the formula-based method for auto-baud calculation. The baud rates that are seen in most commercial electronics, which are referred to as “common”. Any baud rate shown in Table 17-2 can be programmed by the UART. This method works well for higher baud rates, but it could fail below 28.8 kbps if the remote transmitter’s actual baud rate differs by more than one percent of its target.

When the baud rate is detected, the auto-baud circuitry disables itself by clearing the UARTACR.CNTVAL field in the Auto-Baud Count Register. To re-enable auto-baud detection, set the UARTABR.ABE field again.

Note:
1. Changing the baud rate is not permitted when actively transmitting or receiving data. Auto-baud-rate detection is not supported in IrDA (serial infrared) mode.
2. Only after the UART configuration is completed and enabled can the automatic baud rate function be configured.

17.4.4 Programmable baud-rate generator

The UART contains a programmable baud-rate generator that can take a fixed-input clock and divide it down to generate the preferred baud rate. The baud rate is calculated by taking the UART peripheral clock.

The baud-rate generator output frequency is 16 times the baud rate. Two 8-bit Divisor Latch Registers (Divisor Latch Low Register and Divisor Latch High Register) store the divisor in a 16-bit binary format. Load these divisor latches during initialization to ensure that the baud-rate generator operates properly. The 16X clock stops if each Divisor Latch register is loaded with 0x0. The baud rate of the data shifted into or out of a UART is given by the formula:

$$\text{BaudRate} = \frac{UART\_CLK}{16 \times Divisor}$$

For example: if the Divisor is 24 and UART clock is 24 MHz, the baud rate is 62500 bps.

Note: Changing the baud rate is not permitted when actively transmitting or receiving data.

Refer to Table 17-2 for a list of recommended baud rates based on divisor values (Divisor Latch High Byte Register: Divisor Latch Low Byte Register).

The divisor reset value is 0x0002. Changing the baud rate (writing to registers Divisor Latch Low Byte Register and Divisor Latch High Byte Register) is not permitted while actively transmitting or receiving data.

Table 17-2: Recommended baud rates

Required Baud Rate (bps)
9600
19200
38400
57600
115200
230400
460800
921600
1842000
3686400

17.4.5 Serial infrared asynchronous interface

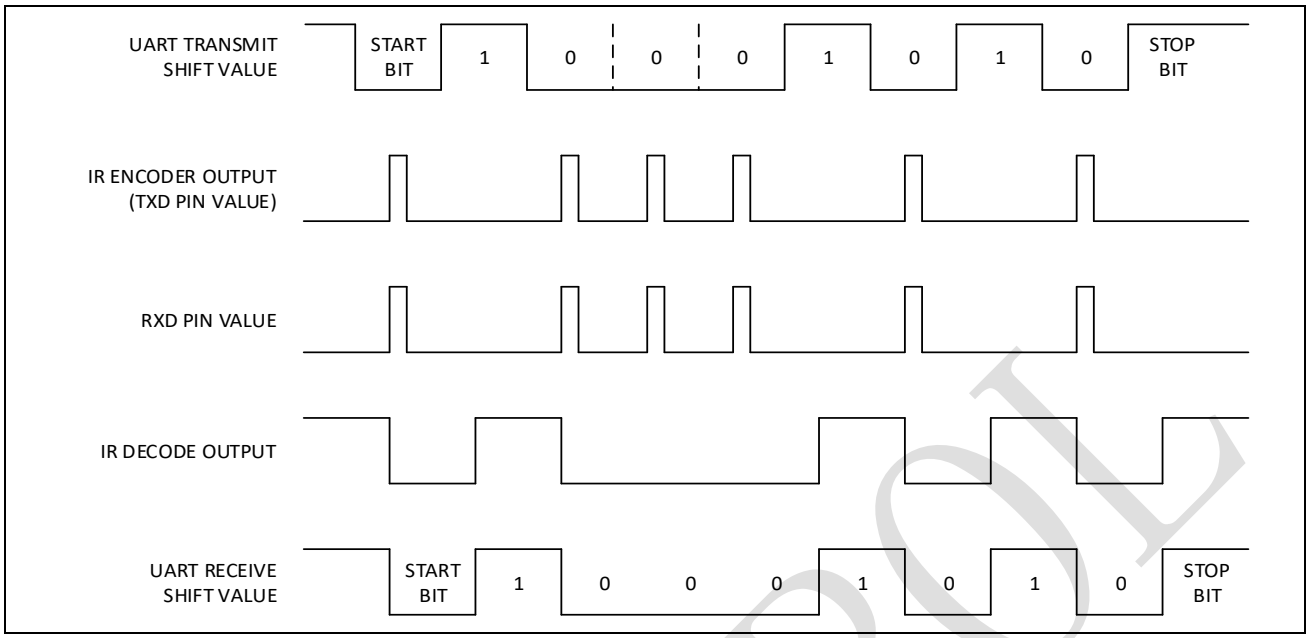
The Slow Serial Infrared (SIR) interface is used with the UART to support two-way wireless communication that uses infrared transmission. The SIR interface provides a Transmit encoder and Receive decoder to support a physical link that conforms to the lrDA Serial Infrared Specification.

The SIR interface does not contain the actual IR LED driver or the receive amplifier. The I/O pins attached to the SIR interface have only digital CMOS-level signals. SIR supports two-way communication, but full-duplex communication is not possible because reflections from the transmit LED enter the receiver. SIR supports frequencies up to 115.2 kbps.

Note: This SIR interface should be turned on and off only while the UART module is completely idle and disabled. On-the-fly changes are likely to produce problems and are not recommended.

The SIR modulation technique works with 7- or 8- bit characters with an optional parity bit. The data is preceded by a zero-value start bit and ends with one stop bit. The encoding scheme sends a pulse (3/16 of a bit wide) in the middle of every zero-value bit and sends no pulses for bits with a value of one. The pulse for each zero-value bit must occur, even for consecutive bits with no edge between them. Figure 17-4 shows an example of Transmit and Receive operation.

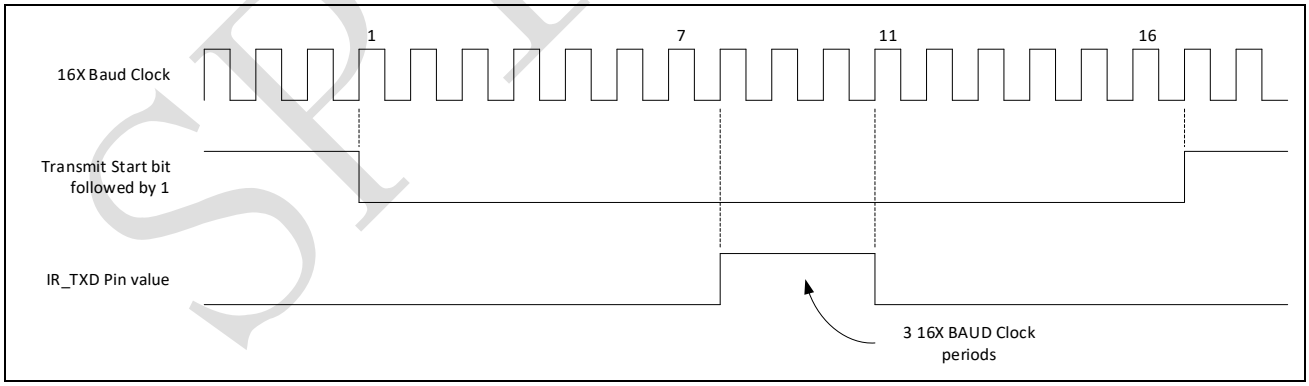
Figure 17-4: IR transmit and receive example



The top line in Figure 17-4 shows an asynchronous transmission as it is sent from the UART. The second line shows the pulses generated by the IR encoder at the TXD pin. A pulse (3/16 of a bit wide) is generated in the middle of the START bit and any data bit that is a zero. The third line shows the values received at the RXD input pin. The fourth line shows the receive-decoder output. The Receive decoder drives the receive data line low when it detects a pulse. The bottom line shows how the UART receiver interrupts the decoder action. This last line is the same as the first, but it is shifted half a bit period.

Each zero bit has a pulse width of 3/16 of a bit time, shown in Figure 17-5.

Figure 17-5: Pulse of a zero bit



To prevent transmitter LED reflection feedback to the receiver, disable the IR Receive decoder when the IR Transmit encoder transmits data, and disable the IR Transmit encoder Receive decoder when the IR Receive decoder receives data. RCVEIR and XMITIR must not be set at the same time.

Note: This restriction specifically is to ensure no feedback; no filtering is performed to remove this feedback data.

17.5 Registers

17.5.1 UART register map

Table 17-3: UART Module Base Address

Peripheral Module	Base Address
UART	0x4000 4000

Table 17-4: UART Register Map

Register	Offset	Description	Reset Value
UARTDLL	0x0	UART Divisor Latch Low Byte Register	0x00000002
UARTRBR	0x0	UART Receive Buffer Register	0x00000000
UARTTHR	0x0	UART Transmit Holding Register	0x00000000
UARTDLH	0x4	UART Divisor Latch High Byte Register	0x00000000
UARTIER	0x4	UART Interrupt Enable Register	0x00000000
UARTFCR	0x8	UART FIFO Control Register	0x00000000
UARTIIR	0x8	UART Interrupt Identification Register	0x00000001
UARTLCR	0xC	UART Line Control Register	0x00000000
UARTMCR	0x10	UART Modem Control Register	0x00000000
UARTLSR	0x14	UART Line Status Register	0x00000060
UARTISR	0x20	UART Infrared Selection Register	0x00000000
UARTFOR	0x24	UART Receive FIFO Occupancy Register	0x00000000
UARTABR	0x28	UART Auto-Baud Control Register	0x00000000
UARTACR	0x2C	UART Auto-Baud Count Register	0x00000000

17.5.2 UART registers

Table 17-5: UART Divisor Latch Low Byte Register (UARTDLL) Layout

UARTDLL (UART Divisor Latch Low Byte Register) Offset: 0x0 Default: 0x00000002 Access: UART -> UARTDLL.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
VAL							

Table 17-6: UART Divisor Latch Low Byte Register (UARTDLL) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:0	VAL	RW	0x2	Low-byte compare value to generate baud rate

Table 17-7: UART Receive Buffer Register (UARTRBR) Layout

UARTRBR (UART Receive Buffer Register) Offset: 0x0 Default: 0x00000000 Access: UART -> UARTRBR.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 17-8: UART Receive Buffer Register (UARTRBR) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x0	In non-FIFO mode, this register holds the characters received by the UART Receive Shift Register. If this register is configured to use fewer than eight bits, the bits are right-justified and the most significant bits (MSB) are zeroed. Reading the register empties the register and clears the <Data Ready> field in the Line Status Register. UARTRBR latches the value of the data byte at the front of the FIFO in FIFO mode.

Table 17-9: UART Transmit Holding Register (UARTTHR) Layout

UARTTHR (UART Transmit Holding Register) Offset: 0x0 Default: 0x00000000 Access: UART -> UARTTHR.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 17-10: UART Transmit Holding Register (UARTTHR) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	WO	0x0	This register holds the data byte to be transmitted next in non-FIFO mode. When the Transmit Shift Register (TSR) is emptied, the contents of this register are loaded into the Transmit Shift Register and the <Transmit Data Request> field in the Line Status Register is set. A write to Transmit Holding Register puts data into the top of the FIFO in FIFO mode. The data at the front of the FIFO is loaded into the TSR when the TSR is empty.

Table 17-11: UART Divisor Latch High Byte Register (UARTDLH) Layout

UARTDLH (UART Divisor Latch High Byte Register) Offset: 0x4 Default: 0x00000000 Access: UART -> UARTDLH.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
VAL							

Table 17-12: UART Divisor Latch High Byte Register (UARTDLH) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:0	VAL	RW	0x0	High-byte compare value to generate baud rate

Table 17-13: UART Interrupt Enable Register (UARTIER) Layout

UARTIER (UART Interrupt Enable Register) Offset: 0x4 Default: 0x00000000 Access: UART -> UARTIER.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7 RESERVED_7	6 UUE	5 NRZME	4 RTOIE	3 RESERVED_3	2 RLSE	1 TIE	0 RAVIE

Table 17-14: UART Interrupt Enable Register (UARTIER) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	RESERVED_7	RW	0x0	Reserved.
6	UUE	RW	0x0	UART Unit Enable 0: The unit is disabled 1: The unit is enabled
5	NRZME	RW	0x0	NRZM coding enable For infrared mode, this bit is ignored and NRM coding is disabled. 0: NRZM coding disabled. General NRZ level coding is used so signal is high at 1 and low at 0. 1: NRZM coding enabled. Signal toggles at 1 and keeps unchanged at 0
4	RTOIE	RW	0x0	Receiver Time-out Interrupt Enable (Source UARTIIR.TOD) 0: Receiver data time-out interrupt disabled 1: Receiver data time-out interrupt enabled
3	RESERVED_3	RW	0x0	Reserved.
2	RLSE	RW	0x0	Receiver Line Status Interrupt Enable (Source UARTIIR.IID) 0: Receiver line status interrupt disabled 1: Receiver line status interrupt enabled
1	TIE	RW	0x0	Transmit Data Request Interrupt Enable (Source UARTIIR.IID) 0: Transmit FIFO data request interrupt disabled 1: Transmit FIFO data request interrupt enabled
0	RAVIE	RW	0x0	Receiver Data Available Interrupt Enable (Source UARTIIR.IID) 0: Receiver data available (trigger threshold reached) interrupt disable 1: Receiver data available (trigger threshold reached) interrupt enabled

Table 17-15: UART FIFO Control Register (UARTFCR) Layout

UARTFCR (UART FIFO Control Register) Offset: 0x8 Default: 0x00000000 Access: UART -> UARTFCR.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
RXTH		BUS32	RESERVED_4	TXTH	CLRTF	CLRRF	TRFIFOE

Table 17-16: UART FIFO Control Register (UARTFCR) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:6	RXTH	WO	0x0	Receive Interrupt Trigger Level (threshold) When the number of bytes in the Receive FIFO equals the interrupt trigger threshold programmed into this field and the received-data-available interrupt is enabled via the UARTIER, an interrupt is generated and appropriate bits are set in the UARTIIR. 00: 1 byte or more in FIFO causes interrupt 01: 8 bytes or more in FIFO causes interrupt 10: 16 bytes or more in FIFO causes interrupt 11: 32 bytes or more in FIFO causes interrupt
5	BUS32	WO	0x0	32-Bit Peripheral Bus 0: 8-bit peripheral bus 1: 32-bit peripheral bus
4	RESERVED_4	WO	0x0	Reserved.
3	TXTH	WO	0x0	Transmitter Interrupt Trigger Level (threshold) 0: Interrupt request when FIFO is half empty 1: Interrupt request when FIFO is empty
2	CLRTF	W1C	0x0	Clear Transmit FIFO When this one is set, all the bytes in the Transmit FIFO are cleared. The TDRQ bit in the LSR is set and the UARTIIR shows a transmitter requests data interrupt, if the TIE bit in the UARTIER register is set. The transmit Shift register is not cleared, and it completes the current transmission. 0: Write a 0 has no effect and always reads back

Bits	Field Name	Type	Reset	Description
				0 1: Write a 1 clears the transmit FIFO This bit is self-cleared
1	CLRRF	W1C	0x0	Clear Receive FIFO When RESETRF is set, all the bytes in the Receive FIFO are cleared. The DR bit in the UARTLSR is cleared to 0. All the error bits in the FIFO and the FIFOE bit in the UARTLSR are cleared. Any error bits, OE, PE, FE or BI, that has been set in UARTLSR are still set. The Receive Shift Register is not cleared. If the UARTIIR has been set to "received data available", it is cleared. 0: Write a 0 has no effect and always reads back 0 1: Write a 1 clears the receive FIFO This bit is self-cleared
0	TRFIFOE	WO	0x0	Transmit and Receive FIFO Enable TRFIFOE enable or disables the Transmit and Receive FIFOs. When TRFIFOE is set, both FIFOs are enabled (FIFO mode). When TRFIFOE is clear, the FIFOs are both disabled (non-FIFO mode). Writing 0b0 to this bit clears all bytes in both FIFOs. When changing from FIFO mode to non-FIFO mode and vice versa, data is cleared automatically from the FIFOs. This bit must be set when other bits in this register are written or the othe r bits are not programmed. 0: FIFOs are disabled 1: FIFOs are enabled

Table 17-17: UART Interrupt Identification Register (UARTIIR) Layout

UARTIIR (UART Interrupt Identification Register) Offset: 0x8 Default: 0x00000001 Access: UART -> UARTIIR.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
FIFOSTS		RESERVED_5	ABL	TOD	IID		NIP

Table 17-18: UART Interrupt Identification Register (UARTIIR) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:6	FIFOSTS	RO	0x0	FIFO Mode Enable Status 00: Non-FIFO mode is selected 01: Reserved 10: Reserved 11: FIFO mode is selected (<Transmit and Receive FIFO Enable> field in the FIFO Control Register = 1)
5	RESERVED_5	RO	0x0	Reserved.
4	ABL	RO	0x0	Auto-baud Lock 0: Auto-baud circuitry has not programmed Divisor Latch registers (DLR) 1: Divisor Latch registers (DLR) programmed by auto-baud circuitry
3	TOD	RO	0x0	Time Out Detected 0: No time out interrupt is pending 1: Time out interrupt is pending (FIFO mode only)
2:1	IID	RO	0x0	Interrupt Source Encoded 00: Invalid option (Modem status) 01: Transmit requests data 10: Received data available 11: Receive error (Overrun, parity, framing, break, FIFO error)
0	NIP	RO	0x1	No Interrupt Pending 0: Interrupt is pending (active low) 1: No interrupt is pending

Table 17-19: UART Line Control Register (UARTLCR) Layout

UARTLCR (UART Line Control Register) Offset: 0xC Default: 0x00000000 Access: UART -> UARTLCR.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7 DLAB	6 SB	5 STKYP	4 EPS	3 PEN	2 STB	1 WLS	0

Table 17-20: UART Line Control Register (UARTLCR) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	DLAB	RW	0x0	Divisor Latch Access Bit Must be set to access the Divisor Latch registers of the baud-rate generator during a Read or Write operation. Must be clear to access the Receive buffer, the Transmit Holding Register, or the UARTIER. 0: Access Transmit Holding Register, Receive Buffer Register, and Interrupt Enable Register 1: Access Divisor Latch registers (UARTDLL and UARTDLH)
6	SB	RW	0x0	Set Break Causes a break condition to be transmitted to the receiving UART. Acts only on the TXD pin and has no effect on the Transmit logic. In FIFO mode, wait until the transmitter is idle (UARTLSR.TXDONE=1) to set and clear SB. 0: No effect on TXD output 1: Forces TXD output to 0 (space)
5	STKYP	RW	0x0	Sticky Parity. This bit is ignored if PEN=0 0: No effect on parity bit 1: Forces parity bit to be opposite of EPS bit
4	EPS	RW	0x0	Even Parity Select. This bit is ignored if PEN=0 0: Sends or checks for odd parity 1: Sends or checks for even parity
3	PEN	RW	0x0	Parity Enable 0: No parity bit 1: Enable parity bit

Bits	Field Name	Type	Reset	Description
2	STB	RW	0x0	Stop bits 0: 1 stop bit 1: 1.5 stop bits for 5-bit character 2 stop bits for 6/7/8 bit character
1:0	WLS	RW	0x0	Word length for each transferred character 00: 5-bit character 01: 6-bit character 10: 7-bit character 11: 8-bit character

Table 17-21: UART Modem Control Register (UARTMCR) Layout

UARTMCR (UART Modem Control Register) Offset: 0x10 Default: 0x00000000 Access: UART -> UARTMCR.all							
31	30	29	28	27	26	25	24
RESERVED_31_6							
23	22	21	20	19	18	17	16
RESERVED_31_6							
15	14	13	12	11	10	9	8
RESERVED_31_6							
7	6	5	4	3	2	1	0
RESERVED_31_6		RESERVED_5	RESERVED_4	GIE	RESERVED_2	RESERVED_1	RESERVED_0

Table 17-22: UART Modem Control Register (UARTMCR) Description

Bits	Field Name	Type	Reset	Description
31:6	RESERVED_31_6	RO	0x0	Reserved.
5	RESERVED_5	RW	0x0	Reserved.
4	RESERVED_4	RW	0x0	Reserved.
3	GIE	RW	0x0	Global interrupt enable 0: Disable UART interrupt to CPU 1: Enable UART interrupt to CPU
2	RESERVED_2	RW	0x0	Reserved.
1	RESERVED_1	RW	0x0	Reserved.
0	RESERVED_0	RW	0x0	Reserved.

Table 17-23: UART Line Status Register (UARTLSR) Layout

UARTLSR (UART Line Status Register) Offset: 0x14 Default: 0x00000060 Access: UART -> UARTLSR.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
FIFOERR	TXDONE	TDRQ	BI	FE	PE	OE	DR

Table 17-24: UART Line Status Register (UARTLSR) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	FIFOERR	RO	0x0	FIFO Error Status This bit is clear in non-FIFO mode. In FIFO mode, FIFOERR is set when there is at least one parity error, framing error, or break indication for any of the characters in the FIFO. The CPU read of the UARTLSR does not reset this bit. FIFOERR is reset when all errorneous characters have been read from the FIFO. 0: No FIFO or no errors in receive FIFO 1: At least one character in receive FIFO has errors
6	TXDONE	RO	0x1	Transmit done Set when the Transmit Holding Register and the Transmit Shift Register are both empty. It is cleared when either the Transmit Holding Register or the Transmit Shift Register contains a data character. In FIFO mode, TXDONE is set when the Transmit FIFO and the Transmit Shift Register are both empty. 0: There is data in the transmit shift register, the Transmit Holding Register, or the FIFO. 1: All the data in the transmitter has been shifted out
5	TDRQ	RO	0x1	Transmit Data Request Indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the CPU when the transmit-data request interrupt enable is set. The TDRQ bit is set when a character is transferred from the Transmit Holding Register into the Transmit Shift Register. The bit is cleared with the loading of the Transmit Holding Register. In FIFO mode, TDRQ is set when half of the characters in the FIFO have been loaded into the Shift Register or the RESETTF bit in UARTFCR has been set. It is cleared when the FIFO has more than half data. If more than 64 characters are loaded into the FIFO, the excess characters are lost. 0: There is data in the holding register or FIFO

Bits	Field Name	Type	Reset	Description
				waiting to be shifted out 1: The transmit FIFO has half or less than half data
4	BI	RO	0x0	Break Interrupt BI is set when the received data input is held low for longer than a full-word transmission time (the total time of start bit + data bits + parity bit + stop bit). BI is cleared when the CPU reads the UARTLSR. In FIFO mode, only one character equal to 0x00, is loaded into the FIFO regardless of the length of the break condition. BI shows the break condition for the character at the front of the FIFO, not the most recently received character. 0: No break signal has been received 1: Break signal received
3	FE	RO	0x0	Framing Error FE indicates that the received character did not have a valid stop bit. FE is set when the bit following the last data bit or parity bit is detected to be 0. FE is cleared when the CPU reads the LSR. The UART resynchronizes after a framing error. To do this, it assumes that the framing error was due to the next start bit, so it samples this start bit twice and then reads in the data. In FIFO mode, FE shows a framing error for the character at the front of the FIFO, not for the most recently received character. 0: No Framing error 1: Invalid stop bit has been detected
2	PE	RO	0x0	Parity Error Indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. PE is set upon detection of a parity error and is cleared when the CPU reads the UARTLSR. In FIFO mode, PE shows a parity error for the character at the front of the FIFO, not the most recently received character. 0: No Parity Error 1: Parity error has occurred

Bits	Field Name	Type	Reset	Description
1	OE	RO	0x0	Overrun Error In non-FIFO mode, indicates that data in Receive Buffer register was not read by the CPU before the next character was received. The new character is lost. In FIFO mode, OE indicates that all 64 bytes of the FIFO are full and the most recently received byte has been discarded. OE is set upon detection of an overrun condition and cleared when the CPU reads the UARTLSR. 0: No data has been overrun 1: Receive data has been overrun
0	DR	RO	0x0	Data Ready Set when a complete incoming character has been received and transferred into the Receive Buffer Register or the FIFO. In non-FIFO mode, DR is cleared when the Receive buffer is read. In FIFO mode, DR is cleared if the FIFO is empty (last character has been read from UARTRBR) or the FIFO is reset with UARTFCR.RESETRF. 0: No data has been received 1: Data is available in UARTRBR or the FIFO

Table 17-25: UART Infrared Selection Register (UARTISR) Layout

UARTISR (UART Infrared Selection Register) Offset: 0x20 Default: 0x00000000 Access: UART -> UARTISR.all							
31	30	29	28	27	26	25	24
RESERVED_31_5							
23	22	21	20	19	18	17	16
RESERVED_31_5							
15	14	13	12	11	10	9	8
RESERVED_31_5							
7	6	5	4	3	2	1	0
RESERVED_31_5			RXPL	TXPL	RESERVED_2	RCVEIR	XMITIR

Table 17-26: UART Infrared Selection Register (UARTISR) Description

Bits	Field Name	Type	Reset	Description
31:5	RESERVED_31_5	RO	0x0	Reserved.
4	RXPL	RW	0x0	Receive Data Polarity 0: SIR decoder takes positive pulses as zeros 1: SIR decoder takes negative pulses as zeros
3	TXPL	RW	0x0	Transmit Data Polarity 0: SIR encoder generates a positive pulse for a data bit of zero 1: SIR encoder generates a negative pulse for a data bit of zero
2	RESERVED_2	RW	0x0	Reserved.
1	RCVEIR	RW	0x0	Receiver SIR Enable When RCVEIR is set, the signal from the RXD pin is processed by lrDA decoder before it is fed to the UART. IF RCVEIR is clear, than all clocking to the lrDA decoder is blocked and the RXD pin is fed directly to the UART. 0: Receiver is in UART mode 1: Receiver is in infrared mode
0	XMITIR	RW	0x0	Transmitter SIR Enable When XMITIR is set, the normal TXD output from the UART is processed by the lrDA encoder before it is fed to the device pin. If XMITIR is clear, all clocking to the lrDA encoder is blocked and the UART's TXD signal is connected directly to the device pin. 0: TXD signal is connected directly 1: TXD output from the UART is processed by the lrDA encoder

Table 17-27: UART Receive FIFO Occupancy Register (UARTFOR) Layout

UARTFOR (UART Receive FIFO Occupancy Register) Offset: 0x24 Default: 0x00000000 Access: UART -> UARTFOR.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
BYTECNT							

Table 17-28: UART Receive FIFO Occupancy Register (UARTFOR) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6:0	BYTECNT	RO	0x0	Number of bytes (0-64) remaining in the Receive FIFO

Table 17-29: UART Auto-Baud Control Register (UARTABR) Layout

UARTABR (UART Auto-Baud Control Register) Offset: 0x28 Default: 0x00000000
Access: UART -> UARTABR.all

31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					ABUP	ABLIE	ABE

Table 17-30: UART Auto-Baud Control Register (UARTABR) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	ABUP	RW	0x0	Method to program Divisor Latch registers 0: Software Programs Divisor Latch registers 1: UART Programs Divisor Latch registers
1	ABLIE	RW	0x0	Auto-baud-lock interrupt enable 0: Auto-baud-lock interrupt disabled (Source UARTIIR.ABL) 1: Auto-baud-lock interrupt enabled (Source UARTIIR.ABL)
0	ABE	RW	0x0	Auto-baud detection enable 0: Auto-baud disabled 1: Auto-baud enabled

Table 17-31: UART Auto-Baud Count Register (UARTACR) Layout

UARTACR (UART Auto-Baud Count Register) Offset: 0x2C Default: 0x00000000 Access: UART -> UARTACR.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
CNTVAL							
7	6	5	4	3	2	1	0
CNTVAL							

Table 17-32: UART Auto-Baud Count Register (UARTACR) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	CNTVAL	RO	0x0	Number of UART clock cycles within a start-bit pulse

18 I2C

18.1 I2C overview

The SPD1148 implements a hardware I2C module, which provides an interface between a MCU and any I2C-bus-compatible device that connects via the I2C serial bus. The external components attached to the I2C bus can serially transmit/receive data to/from the MCU device through the two-wire I2C interface.

The I2C bus is multiple master bus and the I2C module supports the multi-master mode that allows more than one device capable of controlling the bus to be connected to it. Each I2C device is recognized by a unique address and can operate as either transmitter or receiver, according to the function of the device. In addition to being a transmitter or receiver, a device connected to the I2C bus can also be considered as master or slave when performing data transfers. Note that a master device is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. During this transfer, any device addressed by this master is considered a slave.

18.2 I2C features

The SPD1148 I2C module has the following features:

- Compliant with I2C specification version 2.1
- Supports three speeds: Standard mode (100 Kb/s), Fast mode (400 Kb/s) and High-Speed mode (2 Mb/s)
- Clock synchronization
- 7- or 10-bit addressing
- 7- or 10-bit combined format transfers
- Two independent transmit and receive FIFOs with 16 sample in depth and 32 bit in width

18.3 I2C signal description

Table 18-1: I2C signal description

Signal Name	Type	Descriptions
I2C_SCL	Input/Output	I2C serial clock line (open drain)
I2C_SDA	Input/Output	I2C serial data line (open drain)

18.4 I2C operation

The I2C module can be controlled via software to be either:

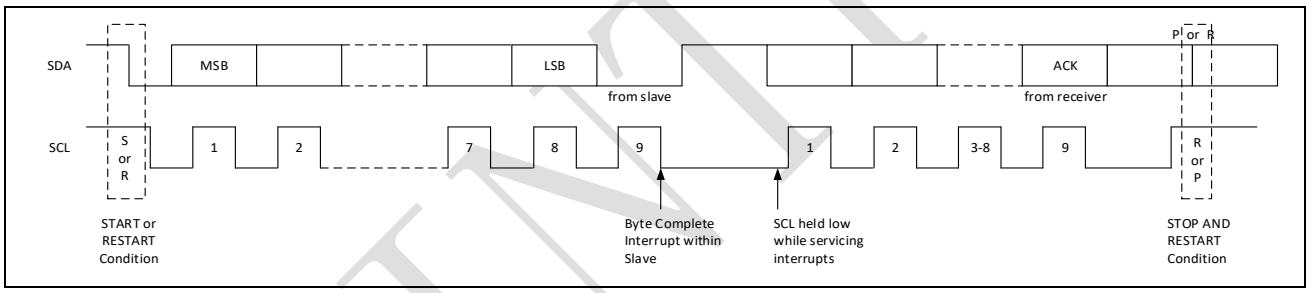
- An I2C master only, communicating with other I2C slaves; OR
- An I2C slave only, communicating with one more I2C masters.

The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. The acknowledgement of data is sent by the device that is receiving data, which can be either a master or a slave. As mentioned previously, the I2C protocol also allows multiple masters to reside on the I2C bus and uses an arbitration procedure to determine bus ownership.

Each slave has a unique address that is determined by the system designer. When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave's address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends an acknowledge (ACK) pulse after the address.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition. This behavior is illustrated in Figure 18-1.

Figure 18-1: Data transfer on the I2C bus



The I2C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only while the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain or open-collector to perform wire-AND functions on the bus. The maximum number of devices on the bus is limited by only the maximum capacitance specification of 400 pF. Data is transmitted in byte packages.

Note: Putting data into the FIFO generates a START, and emptying the FIFO generates a STOP. For more information, refer to Section 18.4.1.

18.4.1 START and STOP generation

When operating as an I2C master, putting data into the Transmit FIFO causes the I2C to generate a START condition on the I2C bus. Allowing the Transmit FIFO to empty causes the I2C to generate a STOP condition on the I2C bus.

When operating as a slave, the I2C does not generate START and STOP conditions, as per the protocol. However, if a read request is made to the I2C, it holds the SCL line low until read data has been

supplied to it. This action stalls the I2C bus until read data is provided by the slave I2C, or the I2C slave is disabled by writing a 0 to EN in register I2CENABLE.

18.4.2 Combined formats

The SPD1148 I2C module supports mixed read and write combined format transactions in both 7-bit and 10-bit addressing modes.

The I2C does not support mixed address combined format transactions, that is, a 7-bit address transaction followed by a 10-bit address transaction or vice versa.

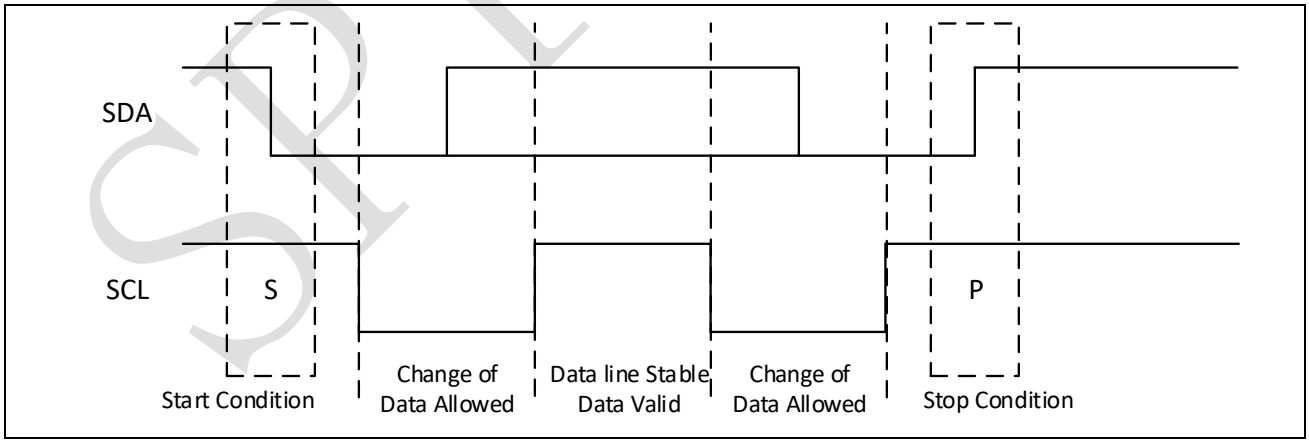
To initiate combined format transfers, set the register I2CCTL.RESTARTEN to 1. With this value set and operating as a master, when the I2C completes an I2C transfer, it checks the Transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer. If the Transmit FIFO is empty when the current I2C transfer completes, a STOP is issued and the next transfer is issued following a START condition.

18.5 I2C protocols

18.5.1 START and STOP conditions

When the bus is idle, both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When the master wants to start a transmission on the bus, the master issues a START condition. This is defined to be a high-to-low transition of the SDA signal while SCL is 1. When the master must terminate the transmission, it issues a STOP condition, which is defined to be a low-to-high transition of the SDA line while SCL is 1. Figure 18-2 shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when SCL is 1.

Figure 18-2: START and STOP condition



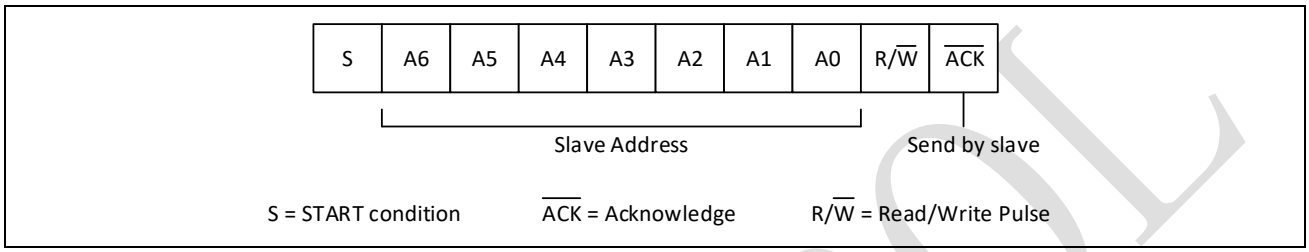
18.5.2 Address formats

The I2C module supports two address formats: the 7-bit and 10-bit address formats.

7-Bit Address Format

During the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the LSB bit (bit 0) is the R/W bit as shown in Figure 18-3. When bit 0 (R/W) is set to 0, the master writes to the slave. When bit 0 (R/W) is set to 1, the master reads from the slave.

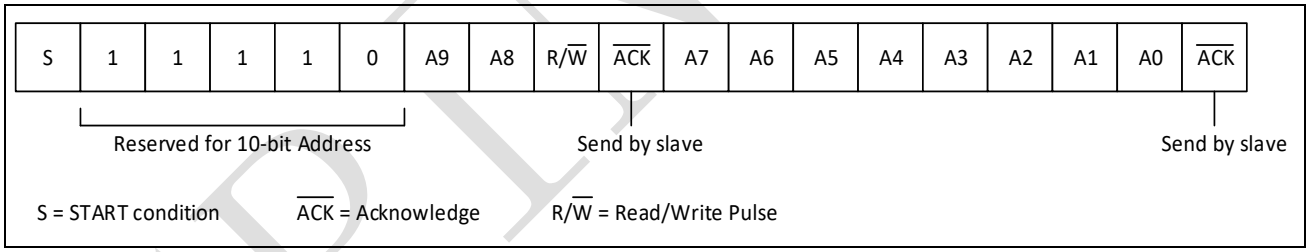
Figure 18-3: 7-bit address format



10-Bit Address Format

During 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slaves that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slaves address bits 9:8, and the LSB bit (bit 0) is the R/W bit. The second byte transferred sets bits 7:0 of the slave address. Figure 18-4 shows the 10-bit address format.

Figure 18-4: 10-bit address format



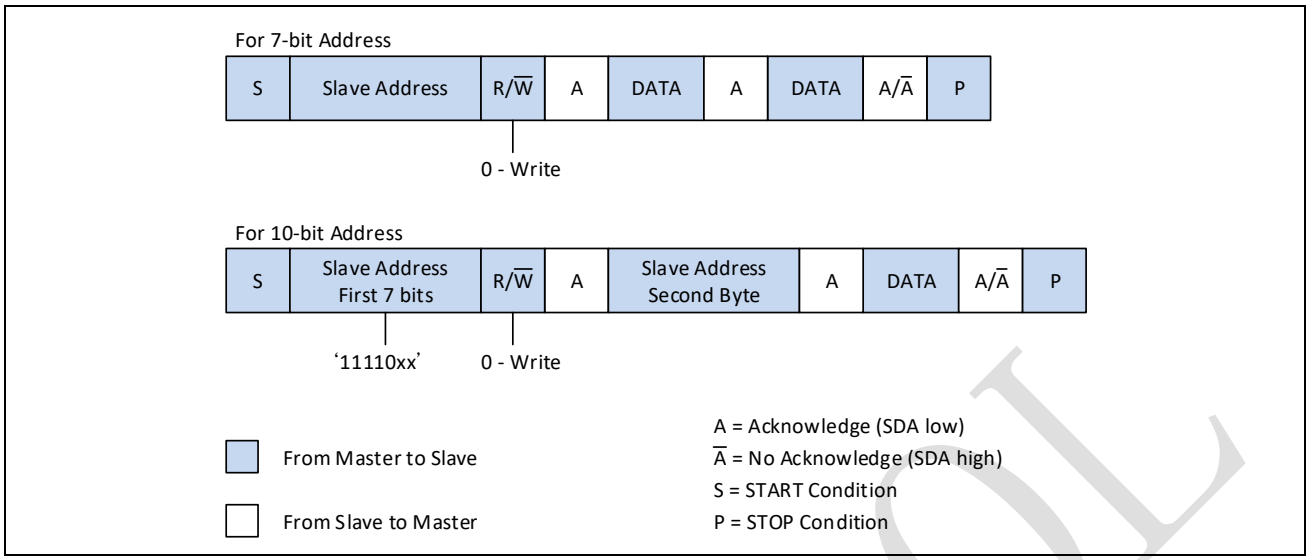
18.5.3 Transmitting and receiving protocol

The master can initiate data transmission and reception to/from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master to either transmit data or receive data to/from the bus, acting as either a slave-transmitter or slave-receiver, respectively.

Master-Transmitter and Slave-Receiver

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transaction. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal (ACK). When a slave-receiver does not respond with an ACK pulse, the master aborts the transfer by issuing a STOP condition. The slave must leave the SDA line high so that the master can abort the transfer.

Figure 18-5: Master-transmitter protocol



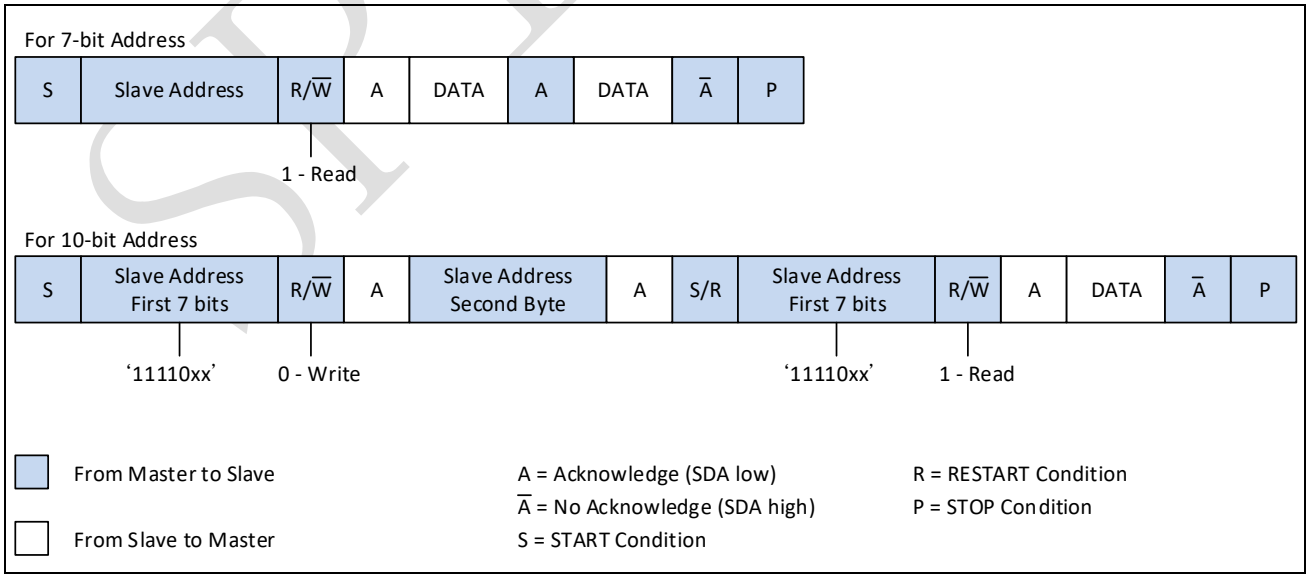
Master-Receiver and Slave-Transmitter

If the master is receiving data as shown in Figure 18-6, then the master responds to the slave-transmitter with an acknowledge pulse after a byte of data has been received, except for the last byte. This method is how the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

When a master refuses to relinquish the bus with a STOP condition, the master can issue a

RESTART condition. This is identical to a START condition except it occurs after the ACK pulse. Operating in master mode, the I2C can then communicate with the same slave using a transfer of a different direction.

Figure 18-6: Master-receiver protocol



18.6 Registers

18.6.1 I2C register map

Table 18-2: I2C Module Base Address

Peripheral Module	Base Address
I2C	0x4000 6000

Table 18-3: I2C Register Map

Register	Offset	Description	Reset Value
I2CCTL	0x0	I2C Control Register	0x0000007F
I2CMASTERADDR	0x4	I2C Master Address Register	0x00001055
I2CSLVADDR	0x8	I2C Slave Address Register	0x00000055
I2CHSMADDR	0xC	I2C High Speed Master Mode Code Address Register	0x00000001
I2CDATACMD	0x10	I2C Data Buffer and Command Register	0x00000000
I2CSSHCNT	0x14	Standard Speed I2C Clock SCL High Count Register	0x000001F4
I2CSSLCNT	0x18	Standard Speed I2C Clock SCL Low Count Register	0x0000024C
I2CFSHCNT	0x1C	Fast Speed I2C Clock SCL High Count Register	0x0000004B
I2CFSLCNT	0x20	Fast Speed I2C Clock SCL Low Count Register	0x000000A3
I2CHSHCNT	0x24	High Speed I2C Clock SCL High Count Register	0x00000008
I2CHSLCNT	0x28	High Speed I2C Clock SCL Low Count Register	0x00000014
I2CIF	0x2C	I2C Interrupt Flag Register	0x00000000
I2CIE	0x30	I2C Interrupt Enable Register	0x000008FF
I2CRAWIF	0x34	I2C Raw Interrupt Flag Register	0x00000000
I2CRXTH	0x38	I2C Receive FIFO Threshold Register	0x00000000
I2CTXTH	0x3C	I2C Transmit FIFO Threshold Register	0x00000000
I2CINTCLR	0x40	Clear Combined and Individual Interrupt Register	0x00000000
I2CRXUDFCLR	0x44	Clear RXUDF Interrupt Register	0x00000000
I2CRXOVFCLR	0x48	Clear RXOVF Interrupt Register	0x00000000
I2CTXOVFCLR	0x4C	Clear TXOVF Interrupt Register	0x00000000
I2CRDREQCLR	0x50	Clear RDREQ Interrupt Register	0x00000000
I2CTXABRTCLR	0x54	Clear TXABRT Interrupt Register	0x00000000

Register	Offset	Description	Reset Value
I2CRXDONECLR	0x58	Clear RXDONE Interrupt Register	0x00000000
I2CACTCLR	0x5C	Clear ACT Interrupt Register	0x00000000
I2CSTOPDETCLR	0x60	Clear STOPDET Interrupt Register	0x00000000
I2CSTARTDETCLR	0x64	Clear STARTDET Interrupt Register	0x00000000
I2CGENCALLCLR	0x68	Clear GENCALL Interrupt Register	0x00000000
I2CENABLE	0x6C	I2C Enable Register	0x00000000
I2CSTS	0x70	I2C Status Register	0x00000006
I2CTFLVL	0x74	I2C Transmit FIFO Level Register	0x00000000
I2CRFLVL	0x78	I2C Receive FIFO Level Register	0x00000000
I2CSDAHOLD	0x7C	I2C SDA Hold-Time Register	0x00000001
I2CTXABRTSRC	0x80	I2C Transmit Abort Source Register	0x00000000
I2CSDASETUP	0x94	I2C SDA Setup Register	0x00000064
I2CACKGC	0x98	I2C ACK General Call Register	0x00000001
I2CENSTS	0x9C	I2C Enable Status Register	0x00000000
I2CFSSPKLEN	0xA0	Fast Speed I2C Spike Suppresion Limit Register	0x00000006
I2CHSSPKLEN	0xA4	High Speed I2C Spike Suppresion Limit Register	0x00000002

18.6.2 I2C registers

Table 18-4: I2C Control Register (I2CCTL) Layout

I2CCTL (I2C Control Register) Offset: 0x0 Default: 0x0000007F Access: I2C -> I2CCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	SLVDIS	RESTARTEN	ADDRSTS	SLVADDR10B	SPEED		MASTER

Table 18-5: I2C Control Register (I2CCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	SLVDIS	RW	0x1	Disable I2C slave This bit should be the same as MASTER

Bits	Field Name	Type	Reset	Description
				0: Enable I2C slave 1: Dsiable I2C slave
5	RESTARTEN	RW	0x1	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several I2C operations. When RESTART is disabled, the master is prohibited from performing the following functions: - Change direction within a transfer (split) - Send a START BYTE - High-speed mode operation - Combined format transfers in 7-bit addressing modes - Read operation with a 10-bit address - Send multiple bytes per transfer By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple I2C transfers. If the above operations are performed, it will result in setting bit 6 (TXABRT) of the I2CRAWIF register. 0: Disable 1: Enable
4	ADDRSTS	RO	0x1	Read-only copy of I2CMASTERADDR[12] 0: 7-bit addressing 1: 10-bit addressing
3	SLVADDR10B	RW	0x1	Addressing mode in slave mode 0: 7-bit addressing The I2C ignores transactions that involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the I2CSLVADDR register are compared. 1: 10-bit addressing The I2C responds to only 10-bit addressing transfers that match the full 10 bits of the I2CSLVADDR register.
2:1	SPEED	RW	0x3	Speed in master mode 00: 01: Standard mode (100 kbit/s) 10: Fast mode (400 kbit/s) 11: High speed mode (3.4 Mbit/s)
0	MASTER	RW	0x1	Enable I2C master The bit should be the same as SLVDIS

Bits	Field Name	Type	Reset	Description
				0: Disable I2C master 1: Enable I2C master

Table 18-6: I2C Master Address Register (I2CMASTERADDR) Layout

I2CMASTERADDR (I2C Master Address Register) Offset: 0x4 Default: 0x00001055 Access: I2C -> I2CMASTERADDR.all							
31	30	29	28	27	26	25	24
RESERVED_31_13							
23	22	21	20	19	18	17	16
RESERVED_31_13							
15	14	13	12	11	10	9	8
RESERVED_31_13			MASTERADDR10B	SPECIAL	GCORSTART	TARADDR	
7	6	5	4	3	2	1	0
TARADDR							

Table 18-7: I2C Master Address Register (I2CMASTERADDR) Description

Bits	Field Name	Type	Reset	Description
31:13	RESERVED_31_13	RO	0x0	Reserved.
12	MASTERADDR10B	RW	0x1	Addressing mode in master mode 0: 7-bit addressing 1: 10-bit addressing
11	SPECIAL	RW	0x0	Special command enable 0: Disable special command and ignore GCORSTART bit 1: Enable special command as specified in GCORSTART bit
10	GCORSTART	RW	0x0	Special command select 0: General Call After issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TXABRT) of the I2CRAWIF register. The I2C remains in General Call mode until the SPECIAL bit is cleared 1: Start Byte
9:0	TARADDR	RW	0x55	Target address for any master transaction. These bits are ignored for issuing a General Call. To generate a START BYTE, the CPU needs to write only once into these bits. It should not be the same as I2CSLVADDR.

Table 18-8: I2C Slave Address Register (I2CSLVADDR) Layout

I2CSLVADDR (I2C Slave Address Register) Offset: 0x8 Default: 0x00000055 Access: I2C -> I2CSLVADDR.all							
31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						VAL	
7	6	5	4	3	2	1	0
VAL							

Table 18-9: I2C Slave Address Register (I2CSLVADDR) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9:0	VAL	RW	0x55	Slave address when I2C acts as a slave This register can be written only when the I2C interface is disabled (I2CENABLE=0). Writes at other times have no effect. Note: The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed if you program the I2CSLVADDR or I2CMASTERADDR to a reserved value.

Table 18-10: I2C High Speed Master Mode Code Address Register (I2CHSMADDR) Layout

I2CHSMADDR (I2C High Speed Master Mode Code Address Register) Offset: 0xC Default: 0x00000001 Access: I2C -> I2CHSMADDR.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					VAL		

Table 18-11: I2C High Speed Master Mode Code Address Register (I2CHSMADDR) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2:0	VAL	RW	0x1	The 3LSBs of the 8-bit high-speed mode master code in format of (0x00001xxx) This register can be written only when the I2C interface is disabled (I2CENABLE=0) Writes at other times have no effect.

Table 18-12: I2C Data Buffer and Command Register (I2CDATACMD) Layout

I2CDATACMD (I2C Data Buffer and Command Register) Offset: 0x10 Default: 0x00000000 Access: I2C -> I2CDATACMD.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							CMD
7	6	5	4	3	2	1	0
DATA							

Table 18-13: I2C Data Buffer and Command Register (I2CDATACMD) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8	CMD	RW	0x0	Read or write control 0: Write 1: Read
7:0	DATA	RW	0x0	Data transmitted or received on the I2C bus. A write to the register sets the data to be transmitted. A read to the register gets the received data.

Table 18-14: Standard Speed I2C Clock SCL High Count Register (I2CSSHCNT) Layout

I2CSSHCNT (Standard Speed I2C Clock SCL High Count Register) Offset: 0x14 Default: 0x000001F4 Access: I2C -> I2CSSHCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 18-15: Standard Speed I2C Clock SCL High Count Register (I2CSSHCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x1F4	Duration for SCL high in standard speed mode. i.e. SCL is high for I2CSSHCNT clock cycles The minimum valid value is 6. Write a value below 6 will be reset as 6 by the hardware. Register can be written only when I2CENABLE=0. Writes at other times have no effect.

Table 18-16: Standard Speed I2C Clock SCL Low Count Register (I2CSSLCNT) Layout

I2CSSLCNT (Standard Speed I2C Clock SCL Low Count Register) Offset: 0x18 Default: 0x0000024C Access: I2C -> I2CSSLCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 18-17: Standard Speed I2C Clock SCL Low Count Register (I2CSSLCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x24C	Duration for SCL low in standard speed mode. i.e. SCL is low for I2CSSLCNT clock cycles The minimum valid value is 8. Write a value below 8 will be reset as 8 by the hardware. Register can be written only when I2CENABLE=0. Writes at other times have no effect.

Table 18-18: Fast Speed I2C Clock SCL High Count Register (I2CFSHCNT) Layout

I2CFSHCNT (Fast Speed I2C Clock SCL High Count Register) Offset: 0x1C Default: 0x0000004B Access: I2C -> I2CFSHCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 18-19: Fast Speed I2C Clock SCL High Count Register (I2CFSHCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x4B	Duration for SCL high in fast speed mode. i.e. SCL is high for I2CFSHCNT clock cycles The minimum valid value is 6. Write a value below 6 will be reset as 6 by the hardware. Register can be written only when I2CENABLE=0. Writes at other times have no effect.

Table 18-20: Fast Speed I2C Clock SCL Low Count Register (I2CFSLCNT) Layout

I2CFSLCNT (Fast Speed I2C Clock SCL Low Count Register) Offset: 0x20 Default: 0x000000A3 Access: I2C -> I2CFSLCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 18-21: Fast Speed I2C Clock SCL Low Count Register (I2CFSLCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0xA3	Duration for SCL low in fast speed mode. i.e. SCL is low for I2CFSLCNT clock cycles The minimum valid value is 8. Write a value below 8 will be reset as 8 by the hardware. Register can be written only when

Bits	Field Name	Type	Reset	Description
				I2CENABLE=0. Writes at other times have no effect.

Table 18-22: High Speed I2C Clock SCL High Count Register (I2CHSHCNT) Layout

I2CHSHCNT (High Speed I2C Clock SCL High Count Register) Offset: 0x24 Default: 0x00000008 Access: I2C -> I2CHSHCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 18-23: High Speed I2C Clock SCL High Count Register (I2CHSHCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x8	Duration for SCL high in high speed mode. i.e. SCL is high for I2CHSHCNT clock cycles The minimum valid value is 6. Write a value below 6 will be reset as 6 by the hardware. Register can be written only when I2CENABLE=0. Writes at other times have no effect.

Table 18-24: High Speed I2C Clock SCL Low Count Register (I2CHSLCNT) Layout

I2CHSLCNT (High Speed I2C Clock SCL Low Count Register) Offset: 0x28 Default: 0x00000014 Access: I2C -> I2CHSLCNT.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 18-25: High Speed I2C Clock SCL Low Count Register (I2CHSLCNT) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x14	Duration for SCL low in high speed mode. i.e. SCL is low for I2CHSLCNT clock cycles The minimum valid value is 8. Write a value below 8 will be reset as 8 by the hardware. Register can be written only when I2CENABLE=0. Writes at other times have no effect.

Table 18-26: I2C Interrupt Flag Register (I2CIF) Layout

I2CIF (I2C Interrupt Flag Register) Offset: 0x2C Default: 0x00000000 Access: I2C -> I2CIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				GENCALL	STARTDET	STOPDET	ACT
7	6	5	4	3	2	1	0
RXDONE	TXABRT	RDREQ	TXDREQ	TXOVF	RXDAV	RXOVF	RXUDF

Table 18-27: I2C Interrupt Flag Register (I2CIF) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11	GENCALL	RO	0x0	General Call address received and acknowledged flag 0: Not occur 1: Occurred
10	STARTDET	RO	0x0	START or RESTART condition flag 0: Condition is not detected 1: Condition is detected
9	STOPDET	RO	0x0	STOP condition flag 0: Condition is not detected 1: Condition is detected
8	ACT	RO	0x0	Latched I2C activity status flag 0: There is no activity detected on I2C interface 1: Activity has been detected on I2C interface
7	RXDONE	RO	0x0	Flag to indicate external master receive is done when I2C acts as a slave transmitter 0: Slave transmission is acknowledged by the other master, which continues data receiving.

Bits	Field Name	Type	Reset	Description
				1: Slave transmission is not acknowledge by the other master, which finishes the data receiving
6	TXABRT	RO	0x0	Latched transmission abort flag 0: No transmission abortion occurred 1: Transimission aborted and TX FIFO remains in flushed state until the register I2CTXABRTCLR is read
5	RDREQ	RO	0x0	Latched external master read request flag when the I2C is acting as a slave 0: No read request is received from another master 1: Read request was received from another master. The processor must respond to this interrupt and then write the requested data to the I2CDATACMD register.
4	TXDREQ	RO	0x0	Flag to indicate the transmit buffer is at or below the threshold value set in the I2CTXTH register so requets more data 0: Transmit buffer is above threshold 1: Transmit buffer is at or below the threshold Self-cleared when level goes above threshold
3	TXOVF	RO	0x0	Flag to indicate an attempt to write I2CDATACMD register when transmit buffer is fill to 15 0: Transmit buffer does not overflow 1: Transmit buffer overflows
2	RXDAV	RO	0x0	Flag to indicate the receive buffer reaches or goes above the threshold in the I2CRXTH register so data is available to read 0: Receive buffer is below threshold 1: Receive buffer is at or above the threshold Self-cleared when level goes below threshold
1	RXOVF	RO	0x0	Flag to indicate the receive buffer is completely filled to 16 and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. 0: Receive buffer does not overflow 1: Receive buffer overflows
0	RXUDF	RO	0x0	Flag to indicate the processor attempts to read the receive buffer via I2CDATACMD register when it is empty

Bits	Field Name	Type	Reset	Description
				0: Receive buffer does not underflow 1: Receive buffer underflows

Table 18-28: I2C Interrupt Enable Register (I2CIE) Layout

I2CIE (I2C Interrupt Enable Register) Offset: 0x30 Default: 0x000008FF Access: I2C -> I2CIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				GENCALL	STARTDET	STOPDET	ACT
7	6	5	4	3	2	1	0
RXDONE	TXABRT	RDREQ	TXDREQ	TXOVF	RXDAV	RXOVF	RXUDF

Table 18-29: I2C Interrupt Enable Register (I2CIE) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11	GENCALL	RW	0x1	Enable/disable GENCALL interrupt 0: Disable GENCALL interrupt 1: Enable GENCALL interrupt
10	STARTDET	RW	0x0	Enable/disable STARTDET interrupt 0: Disable STARTDET interrupt 1: Enable STARTDET interrupt
9	STOPDET	RW	0x0	Enable/disable STOPDET interrupt 0: Disable STOPDET interrupt 1: Enable STOPDET interrupt
8	ACT	RW	0x0	Enable/disable ACTIVITY interrupt 0: Disable ACTIVITY interrupt 1: Enable ACTIVITY interrupt
7	RXDONE	RW	0x1	Enable/disable RXDONE interrupt 0: Disable RXDONE interrupt 1: Enable RXDONE interrupt
6	TXABRT	RW	0x1	Enable/disable TXABRT interrupt 0: Disable TXABRT interrupt 1: Enable TXABRT interrupt
5	RDREQ	RW	0x1	Enable/disable RDREQ interrupt 0: Disable RDREQ interrupt 1: Enable RDREQ interrupt

Bits	Field Name	Type	Reset	Description
4	TXDREQ	RW	0x1	Enable/disable TXDREQ interrupt 0: Disable TXDREQ interrupt 1: Enable TXDREQ interrupt
3	TXOVF	RW	0x1	Enable/disable TXOVF interrupt 0: Disable TXOVF interrupt 1: Enable TXOVF interrupt
2	RXDAV	RW	0x1	Enable/disable RXDAV interrupt 0: Disable RXDAV interrupt 1: Enable RXDAV interrupt
1	RXOVF	RW	0x1	Enable/disable RXOVF interrupt 0: Disable RXOVF interrupt 1: Enable RXOVF interrupt
0	RXUDF	RW	0x1	Enable/disable RXUDF interrupt 0: Disable RXUDF interrupt 1: Enable RXUDF interrupt

Table 18-30: I2C Raw Interrupt Flag Register (I2CRAWIF) Layout

I2CRAWIF (I2C Raw Interrupt Flag Register) Offset: 0x34 Default: 0x00000000 Access: I2C -> I2CRAWIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				GENCALL	STARTDET	STOPDET	ACT
7	6	5	4	3	2	1	0
RXDONE	TXABRT	RDREQ	TXDREQ	TXOVF	RXDAV	RXOVF	RXUDF

Table 18-31: I2C Raw Interrupt Flag Register (I2CRAWIF) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11	GENCALL	RO	0x0	General Call address received and acknowledged flag 0: Not occur 1: Occurred
10	STARTDET	RO	0x0	START or RESTART condition flag 0: Condition is not detected 1: Condition is detected
9	STOPDET	RO	0x0	STOP condition flag 0: Condition is not detected 1: Condition is detected

Bits	Field Name	Type	Reset	Description
8	ACT	RO	0x0	Latched I2C activity status flag 0: There is no activity detected on I2C interface 1: Activity has been detected on I2C interface
7	RXDONE	RO	0x0	Flag to indicate external master receive is done when I2C acts as a slave transmitter 0: Slave transmission is acknowledged by the other master, which continues data receiving. 1: Slave transmission is not acknowledge by the other master, which finishes the data receiving
6	TXABRT	RO	0x0	Latched transmission abort flag 0: No transmission abortion occurred 1: Transimission aborted and TX FIFO remains in flushed state until the register I2CTXABRTCLR is read
5	RDREQ	RO	0x0	Latched external master read request flag when the I2C is acting as a slave 0: No read request is received from another master 1: Read request was received from another master. The processor must respond to this interrupt and then write the requested data to the I2CDATACMD register.
4	TXDREQ	RO	0x0	Flag to indicate the transmit buffer is at or below the threshold value set in the I2CTXTH register so requets more data 0: Transmit buffer is above threshold 1: Transmit buffer is at or below the threshold Self-cleared when level goes above threshold
3	TXOVF	RO	0x0	Flag to indicate an attempt to write I2CDATACMD register when transmit buffer is fill to 15 0: Transmit buffer does not overflow 1: Transmit buffer overflows
2	RXDAV	RO	0x0	Flag to indicate the receive buffer reaches or goes above the threshold in the I2CRXTH register so data is available to read 0: Receive buffer is below threshold 1: Receive buffer is at or above the threshold Self-cleared when level goes below threshold
1	RXOVF	RO	0x0	Flag to indicate the receive buffer is completely filled to 16 and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received

Bits	Field Name	Type	Reset	Description
				after the FIFO is full are lost. 0: Receive buffer does not overflow 1: Receive buffer overflows
0	RXUDF	RO	0x0	Flag to indicate the processor attempts to read the receive buffer via I2CDATACMD register when it is empty 0: Receive buffer does not underflow 1: Receive buffer underflows

Table 18-32: I2C Receive FIFO Threshold Register (I2CRXTH) Layout

I2CRXTH (I2C Receive FIFO Threshold Register) Offset: 0x38 Default: 0x00000000 Access: I2C -> I2CRXTH.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
KEY				VAL			

Table 18-33: I2C Receive FIFO Threshold Register (I2CRXTH) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:4	KEY	WO	0x0	Must write 0x0 to this field when write to low LSBs, else LSBs is always 0xF. Always readback 0
3:0	VAL	RW	0x0	Receive FIFO threshold level to trigger the RXDAV interrupt

Table 18-34: I2C Transmit FIFO Threshold Register (I2CTXTH) Layout

I2CTXTH (I2C Transmit FIFO Threshold Register) Offset: 0x3C Default: 0x00000000 Access: I2C -> I2CTXTH.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
KEY				VAL			

Table 18-35: I2C Transmit FIFO Threshold Register (I2CTXTH) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:4	KEY	WO	0x0	Must write 0x0 to this field when write to low LSBs, else LSBs is always 0xF. Always readback 0
3:0	VAL	RW	0x0	Transmit FIFO threshold level to trigger the TXDREQ interrupt

Table 18-36: Clear Combined and Individual Interrupt Register (I2CINTCLR) Layout

I2CINTCLR (Clear Combined and Individual Interrupt Register) Offset: 0x40 Default: 0x00000000 Access: I2C -> I2CINTCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-37: Clear Combined and Individual Interrupt Register (I2CINTCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the combined interrupt, all individual interrupts, and the I2CTXABRTSRC register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to I2CTXABRTSRC.STARTNORESTART for an exception to clear I2CTXABRTSRC register.

Bits	Field Name	Type	Reset	Description
				0: Read to clear 1:

Table 18-38: Clear RXUDF Interrupt Register (I2CRXUDFCLR) Layout

I2CRXUDFCLR (Clear RXUDF Interrupt Register) Offset: 0x44 Default: 0x00000000 Access: I2C -> I2CRXUDFCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-39: Clear RXUDF Interrupt Register (I2CRXUDFCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the RXUDF flag 0: Read to clear 1:

Table 18-40: Clear RXOVF Interrupt Register (I2CRXOVFCLR) Layout

I2CRXOVFCLR (Clear RXOVF Interrupt Register) Offset: 0x48 Default: 0x00000000 Access: I2C -> I2CRXOVFCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-41: Clear RXOVF Interrupt Register (I2CRXOVFCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the RXOVF flag 0: Read to clear 1:

Table 18-42: Clear TXOVF Interrupt Register (I2CTXOVFCLR) Layout

I2CTXOVFCLR (Clear TXOVF Interrupt Register) Offset: 0x4C Default: 0x00000000 Access: I2C -> I2CTXOVFCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-43: Clear TXOVF Interrupt Register (I2CTXOVFCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the TXOVF flag 0: Read to clear 1:

Table 18-44: Clear RDREQ Interrupt Register (I2CRDREQCLR) Layout

I2CRDREQCLR (Clear RDREQ Interrupt Register) Offset: 0x50 Default: 0x00000000 Access: I2C -> I2CRDREQCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-45: Clear RDREQ Interrupt Register (I2CRDREQCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the RDREQ flag 0: Read to clear 1:

Table 18-46: Clear TXABRT Interrupt Register (I2CTXABRTCLR) Layout

I2CTXABRTCLR (Clear TXABRT Interrupt Register) Offset: 0x54 Default: 0x00000000 Access: I2C -> I2CTXABRTCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-47: Clear TXABRT Interrupt Register (I2CTXABRTCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the TXABRT flag and the I2CTXABRTSRC register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to I2CTXABRTSRC.STARTNORESTART for an exception to clear I2CTXABRTSRC register. 0: Read to clear 1:

Table 18-48: Clear RXDONE Interrupt Register (I2CRXDONECLR) Layout

I2CRXDONECLR (Clear RXDONE Interrupt Register) Offset: 0x58 Default: 0x00000000 Access: I2C -> I2CRXDONECLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-49: Clear RXDONE Interrupt Register (I2CRXDONECLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the RXDONE flag 0: Read to clear 1:

Table 18-50: Clear ACT Interrupt Register (I2CACTCLR) Layout

I2CACTCLR (Clear ACT Interrupt Register) Offset: 0x5C Default: 0x00000000 Access: I2C -> I2CACTCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-51: Clear ACT Interrupt Register (I2CACTCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Reading this register clears the ACT flag if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY flag continues to be set. It is automatically cleared by hardware if the module is disabled and there is no further activity on the bus. 0: Read to clear 1:

Table 18-52: Clear STOPDET Interrupt Register (I2CSTOPDETCLR) Layout

I2CSTOPDETCLR (Clear STOPDET Interrupt Register) Offset: 0x60 Default: 0x00000000 Access: I2C -> I2CSTOPDETCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-53: Clear STOPDET Interrupt Register (I2CSTOPDETCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the STOPDET flag 0: Read to clear 1:

Table 18-54: Clear STARTDET Interrupt Register (I2CSTARTDETCLR) Layout

I2CSTARTDETCLR (Clear STARTDET Interrupt Register) Offset: 0x64 Default: 0x00000000 Access: I2C -> I2CSTARTDETCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-55: Clear STARTDET Interrupt Register (I2CSTARTDETCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the STARTDET flag 0: Read to clear 1:

Table 18-56: Clear GENCALL Interrupt Register (I2CGENCALLCLR) Layout

I2CGENCALLCLR (Clear GENCALL Interrupt Register) Offset: 0x68 Default: 0x00000000 Access: I2C -> I2CGENCALLCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							CLR

Table 18-57: Clear GENCALL Interrupt Register (I2CGENCALLCLR) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	CLR	RO	0x0	Read this register to clear the GENCALL flag 0: Read to clear 1:

Table 18-58: I2C Enable Register (I2CENABLE) Layout

I2CENABLE (I2C Enable Register) Offset: 0x6C Default: 0x00000000 Access: I2C -> I2CENABLE.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							EN

Table 18-59: I2C Enable Register (I2CENABLE) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	EN	RW	0x0	Controls whether the I2C is enabled. There are two CLK_I2C delay when enable or disable the I2C. When I2C is disabled, the following occurs: - The TX FIFO and RX FIFO get flushed. - Status bits in the I2CIF register are still active until I2C goes into IDLE state. - If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. - If the module is receiving, the I2C stops the current transfer at the end of the current byte and does not acknowledge the transfer. 0: Disable I2C (Hold FIFOs in an erased state). Software can disable I2C while it is active. However, it is important that care be taken to ensure that i2c is disabled properly. 1: Enable I2C

Table 18-60: I2C Status Register (I2CSTS) Layout

I2CSTS (I2C Status Register) Offset: 0x70 Default: 0x00000006 Access: I2C -> I2CSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7 ESERVED_31_	6 SACT	5 MACT	4 RFF	3 RFNE	2 TFE	1 TFNF	0 ACT

Table 18-61: I2C Status Register (I2CSTS) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	SACT	RO	0x0	Slave FSM (Finite State Machine) activity status 0: Slave FSM is in IDLE state (Inactive) 1: Slave FSM is not in IDLE state (Active)
5	MACT	RO	0x0	Master FSM activity status 0: Master FSM is in IDLE state (Inactive) 1: Master FSM is not in IDLE state (Active)

Bits	Field Name	Type	Reset	Description
4	RFF	RO	0x0	Receive FIFO completely full 0: Receive FIFO contains empty location 1: Receive FIFO is completely full
3	RFNE	RO	0x0	Receive FIFO not empty 0: Receive FIFO is empty 1: Receive FIFO contains one or more entries
2	TFE	RO	0x1	Transmit FIFO completely empty 0: Transmit FIFO contains one or more entries 1: Transmit FIFO is empty
1	TFNF	RO	0x1	Transmit FIFO not full 0: Transmit FIFO is full 1: Transmit FIFO contains empty locations
0	ACT	RO	0x0	I2C activity status (Logic OR of SACT and MACT) 0: Both master and slave FSM are in idle state 1: Either master or slave FSM is active

Table 18-62: I2C Transmit FIFO Level Register (I2CTFLVL) Layout

I2CTFLVL (I2C Transmit FIFO Level Register) Offset: 0x74 Default: 0x00000000 Access: I2C -> I2CTFLVL.all							
31	30	29	28	27	26	25	24
RESERVED_31_5							
23	22	21	20	19	18	17	16
RESERVED_31_5							
15	14	13	12	11	10	9	8
RESERVED_31_5							
7	6	5	4	3	2	1	0
RESERVED_31_5			VAL				

Table 18-63: I2C Transmit FIFO Level Register (I2CTFLVL) Description

Bits	Field Name	Type	Reset	Description
31:5	RESERVED_31_5	RO	0x0	Reserved.
4:0	VAL	RO	0x0	Number of valid data entries in the transmit FIFO.

Table 18-64: I2C Receive FIFO Level Register (I2CRFLVL) Layout

I2CRFLVL (I2C Receive FIFO Level Register) Offset: 0x78 Default: 0x00000000 Access: I2C -> I2CRFLVL.all							
31	30	29	28	27	26	25	24
RESERVED_31_5							
23	22	21	20	19	18	17	16
RESERVED_31_5							
15	14	13	12	11	10	9	8
RESERVED_31_5							
7	6	5	4	3	2	1	0
RESERVED_31_5			VAL				

Table 18-65: I2C Receive FIFO Level Register (I2CRFLVL) Description

Bits	Field Name	Type	Reset	Description
31:5	RESERVED_31_5	RO	0x0	Reserved.
4:0	VAL	RO	0x0	Number of valid data entries in the receive FIFO.

Table 18-66: I2C SDA Hold-Time Register (I2CSDAHOLD) Layout

I2CSDAHOLD (I2C SDA Hold-Time Register) Offset: 0x7C Default: 0x00000001 Access: I2C -> I2CSDAHOLD.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 18-67: I2C SDA Hold-Time Register (I2CSDAHOLD) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15:0	VAL	RW	0x1	Configure SDA hold time as VAL CLK_I2C cycles i.e. If the required hold delay is 1000ns, then for an CLK_I2C frequency of 10 MHz, this register is recommended to be programmed as 11. This register can be written only when I2CENABLE=0. Writes at other times have no effect.

Table 18-68: I2C Transmit Abort Source Register (I2CTXABRTSRC) Layout

I2CTXABRTSRC (I2C Transmit Abort Source Register) Offset: 0x80 Default: 0x00000000 Access: I2C -> I2CTXABRTSRC.all							
31	30	29	28	27	26	25	24
RESERVED_31_16							
23	22	21	20	19	18	17	16
RESERVED_31_16							
15 SLVRDINTX	14 SARBLOST	13 SLVFLUSHTF	12 MARBLOST	11 MASTERDIS	10 RD10BNORESTART	9 STARTNORESTART	8 HSNORESTART
7 STARTACKDET	6 HSACKDET	5 GCREAD	4 GCNACK	3 TXDATANACK	2 ADDR10B2NACK	1 ADDR10B1NACK	0 ADDR7BNACK

Table 18-69: I2C Transmit Abort Source Register (I2CTXABRTSRC) Description

Bits	Field Name	Type	Reset	Description
31:16	RESERVED_31_16	RO	0x0	Reserved.
15	SLVRDINTX	RO	0x0	Role of I2C: Slave transmitter This bit is set when the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of I2CDATACMD register. 0: 1:
14	SARBLOST	RO	0x0	Role of I2C: Slave transmitter This bit is set when slave lost the bus while transmitting data to a remote master. I2CTXABRTSRC.MARBLOST is set at the same time. Note: Even though the slave never 'owns' the bus, something could go wrong on the bus. This is a failsafe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then I2C no longer own the bus. 0: 1:
13	SLVFLUSHTF	RO	0x0	Role of I2C: Slave transmitter This bis is set when slave has received a read command and some data exists in the TX FIFO so the slave issues a TXABRT interrupt to flush old data in TX FIFO. 0: 1:
12	MARBLOST	RO	0x0	Role of I2C: Master/slave transmitter This bit is set when master has lost arbitration,

Bits	Field Name	Type	Reset	Description
				or if I2CTXABRTSRC.SARBLOST is also set, then the slave transmitter has lost arbitration. 0: 1:
11	MASTERDIS	RO	0x0	Role of I2C: Master transmitter/receiver This bit is set when user tries to initiate a master operation with the master mode is disabled. 0: 1:
10	RD10BNORESTART	RO	0x0	Role of I2C: Master receiver This bit is set when the restart is disabled (I2CCTL.RESTARTEN=0) and the master sends a read command in 10-bit addressing mode. 0: 1:
9	STARTNORESTART	RO	0x0	Role of I2C: Master transmitter/receiver This bit is set when the restart is disabled (I2CCTL.RESTARTEN=0) and the user is trying to send a START Byte. Follow configurations below to avoid this error (1) I2CCTL.RESTARTEN=1 (2) I2CMASTERADDR.SPECIAL=0 or I2CMASTERADDR.GCORSTART=0 Otherwise, this bit is set again at next clock after it is cleared by reading from I2CTXABRTCLR register. 0: 1:
8	HSNORESTART	RO	0x0	Role of I2C: Master transmitter/receiver This bit is set when the restart is disabled (I2CCTL.RESTARTEN=0) and the user is trying to use the master to transfer data in high speed mode. 0: 1:
7	STARTACKDET	RO	0x0	Role of I2C: Master transmitter/receiver This bit is set when master has sent a START Byte and the START Byte was acknowledged (wrong behavior). 0: 1:
6	HSACKDET	RO	0x0	Role of I2C: Master transmitter/receiver This bit is set when master is in high speed

Bits	Field Name	Type	Reset	Description
				mode and the high speed master code was acknowledged (wrong behavior). 0: 1:
5	GCREAD	RO	0x0	Role of I2C: Master transmitter This bit is set when master sent a General Call but the user programmed the byte following the General Call to be a read from the bus (I2CDATACMD.CMD=1). 0: 1:
4	GCNACK	RO	0x0	Role of I2C: Master transmitter This bit is set when master sent a General Call and no slave on the bus acknowledged the General Call. 0: 1:
3	TXDATANACK	RO	0x0	Role of I2C: Master transmitter This bit is set when master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). 0: 1:
2	ADDR10B2NACK	RO	0x0	Role of I2C: Master transmitter/receiver This bit is set when master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave. 0: 1:
1	ADDR10B1NACK	RO	0x0	Role of I2C: Master transmitter/receiver This bit is set when master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. 0: 1:
0	ADDR7BNACK	RO	0x0	Role of I2C: Master transmitter/receiver This bit is set when master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. 0: 1:

Table 18-70: I2C SDA Setup Register (I2CSDASETUP) Layout

I2CSDASETUP (I2C SDA Setup Register) Offset: 0x94 Default: 0x00000064 Access: I2C -> I2CSDASETUP.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
VAL							

Table 18-71: I2C SDA Setup Register (I2CSDASETUP) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:0	VAL	RW	0x64	Configure SDA setup time as VAL CLK_I2C cycles i.e. If the required setup delay is 1000ns, then for an CLK_I2C frequency of 10 MHz, this register is recommended to be programmed as 11.

Table 18-72: I2C ACK General Call Register (I2CACKGC) Layout

I2CACKGC (I2C ACK General Call Register) Offset: 0x98 Default: 0x00000001 Access: I2C -> I2CACKGC.all							
31	30	29	28	27	26	25	24
RESERVED_31_1							
23	22	21	20	19	18	17	16
RESERVED_31_1							
15	14	13	12	11	10	9	8
RESERVED_31_1							
7	6	5	4	3	2	1	0
RESERVED_31_1							ACKGC

Table 18-73: I2C ACK General Call Register (I2CACKGC) Description

Bits	Field Name	Type	Reset	Description
31:1	RESERVED_31_1	RO	0x0	Reserved.
0	ACKGC	RW	0x1	Respond upon General Call 0: Responds General Call with an NACK 1: Responds General Call with an ACK

Table 18-74: I2C Enable Status Register (I2CENSTS) Layout

I2CENSTS (I2C Enable Status Register) Offset: 0x9C Default: 0x00000000 Access: I2C -> I2CENSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_3							
23	22	21	20	19	18	17	16
RESERVED_31_3							
15	14	13	12	11	10	9	8
RESERVED_31_3							
7	6	5	4	3	2	1	0
RESERVED_31_3					SLVRDLOST	SLVDISONBUSY	EN

Table 18-75: I2C Enable Status Register (I2CENSTS) Description

Bits	Field Name	Type	Reset	Description
31:3	RESERVED_31_3	RO	0x0	Reserved.
2	SLVRDLOST	RO	0x0	Slave received data lost upon disable I2C 0: Slave receiver operation is done before I2CENABLE is changed from 1 to 0 1: Slave receiver operation is aborted with at least one data byte received from an I2C transfer when I2CENABLE is changed from 1 to 0
1	SLVDISONBUSY	RO	0x0	Slave disabled while it is busy 0: I2CENABLE is stable during slave operation 1: I2CENABLE is changed from 1 to 0 when I2C slave is receiving from a remote mater on (a) The address byte of the slave transmitter (b) The address and data bytes of the slave receiver
0	EN	RO	0x0	I2C enabled status 0: I2C is deemed completely inactive SLVRDLOST and SLVDISONBUSY bits can be read safely 1: I2C is deemed to be in an enabled state

Table 18-76: Fast Speed I2C Spike Suppresion Limit Register (I2CFSSPKLEN) Layout

I2CFSSPKLEN (Fast Speed I2C Spike Suppresion Limit Register) Offset: 0xA0 Default: 0x00000006 Access: I2C -> I2CFSSPKLEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
VAL							

Table 18-77: Fast Speed I2C Spike Suppresion Limit Register (I2CFSSPKLEN) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:0	VAL	RW	0x6	Duration in fast speed mode of the longest spike in the SCL or SDA lines that are filtered out by the spike suppression logic. The minimum value is 1 and write a 0 will be reset as 1 by the hardware. This register can be written only when I2CENABLE=0. Writes at other times have no effect.

Table 18-78: High Speed I2C Spike Suppresion Limit Register (I2CHSSPKLEN) Layout

I2CHSSPKLEN (High Speed I2C Spike Suppresion Limit Register) Offset: 0xA4 Default: 0x00000002 Access: I2C -> I2CHSSPKLEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
VAL							

Table 18-79: High Speed I2C Spike Suppresion Limit Register (I2CHSSPKLEN) Description

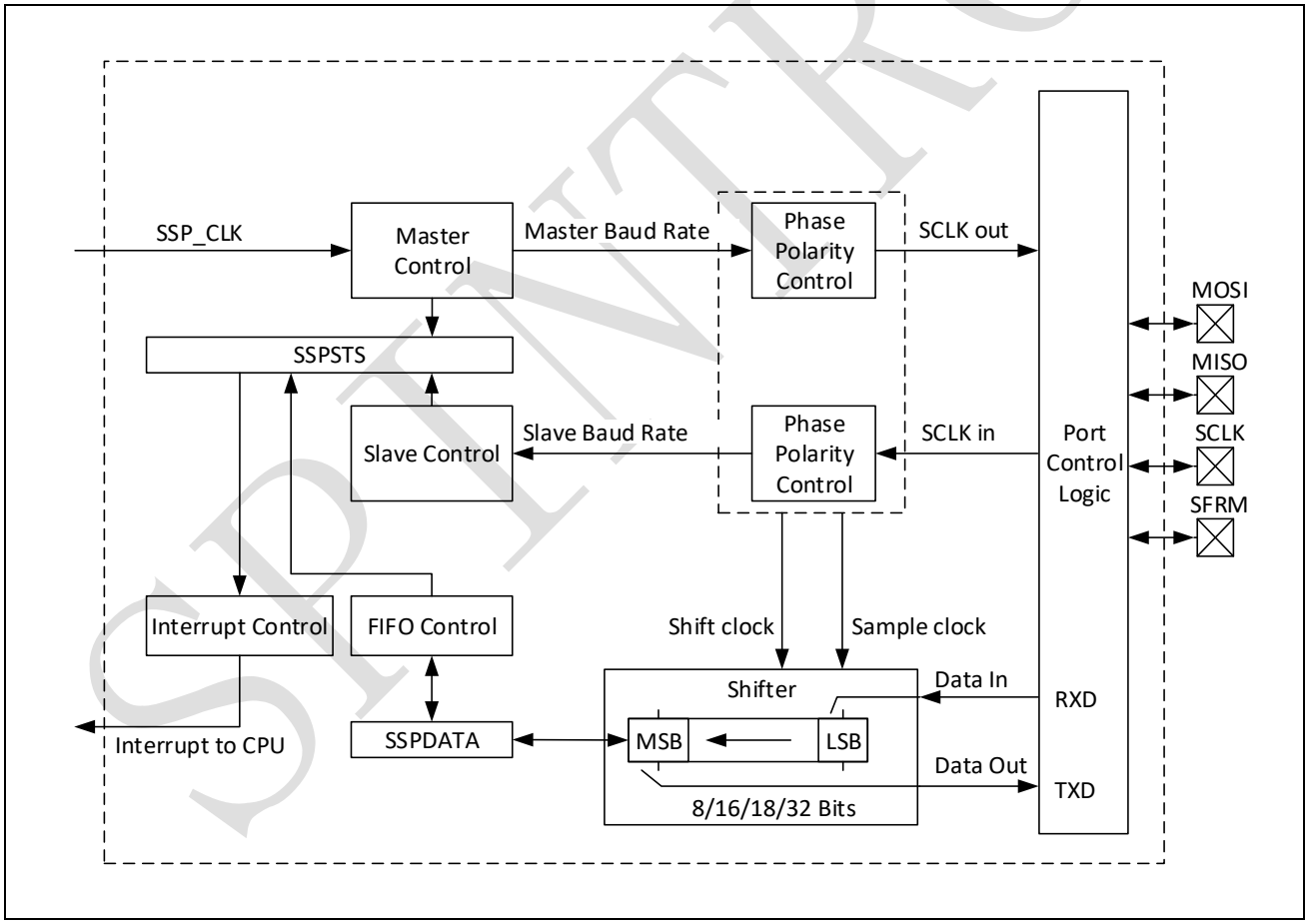
Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7:0	VAL	RW	0x2	Duration in high speed mode of the longest spike in the SCL or SDA lines that are filtered out by the spike suppression logic. The minimum value is 1 and write a 0 will be reset as 1 by the hardware. This register can be written only when I2CENABLE=0. Writes at other times have no effect.

19 SSP

19.1 SSP overview

The SPD1148 SSP interface is a synchronous, full duplex, serial data transfer port that can be used as a connection to a variety of external interfaces, such as flash, ADC, CODEC, etc. The SSP interface can be programmed to operate in Master Mode (the attached peripheral devices as slaves) or Slave Mode (the attached peripheral devices as masters). The SSP port supports different serial bit rates through the configurable interface clocks. Serial data sample size can be configured to 8, 16, 18, or 32 bits in length. A FIFO is provided for Transmit data and a second independent FIFO is provided for Receive data. Both FIFOs are 16 sample in depth and 32 bit in width. The FIFOs can be loaded or emptied by CPU. The SSP can be driven by polling mode or interrupt request mode. SSP module is mainly composed of control, status and data registers, shifter logic, FIFO control logic, master and slave control logic and port control logic, as shown in Figure 19-1.

Figure 19-1: SSP block diagram



19.2 SSP features

The SPD1148 SSP has following enhanced features:

- Directly supports Motorola Serial Peripheral interface (SPI)
- Data transfer rate up to 50 Mbps
- Master or Slave mode operation
- Full duplex
- Receive only operation
- Programmable data frame size: 1 to 32 bits
- Programmable polarity for select port (SFRM)
- MSB-first data order
- Programmable clock polarity and phase
- Two independent transmit and receive FIFO with 16 sample in depth and 32 bit in width, and all support packed mode

19.3 SSP signal description

Table 19-1 describes the SPD1148 SSP interface bus signals.

Table 19-1: SSP signal description

Signal Name	Type	Descriptions
MOSI	Input/ Output	This port is used to transmit data out of the SPI module when it is configured as a Master and receive data when it is configured as Slave.
MISO	Input/ Output	This port is used to transmit data out of the SPI module when it is configured as a Slave and receive data when it is configured as Master.
SCLK	Input/ Output	This port is used to output the clock with respect to which the SPI transfers data or receive clock in case of Slave.
SFRM	Input/ Output	This port is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when its configured as a Master and its used as an input to receive the slave select signal when the SPI is configured as Slave.

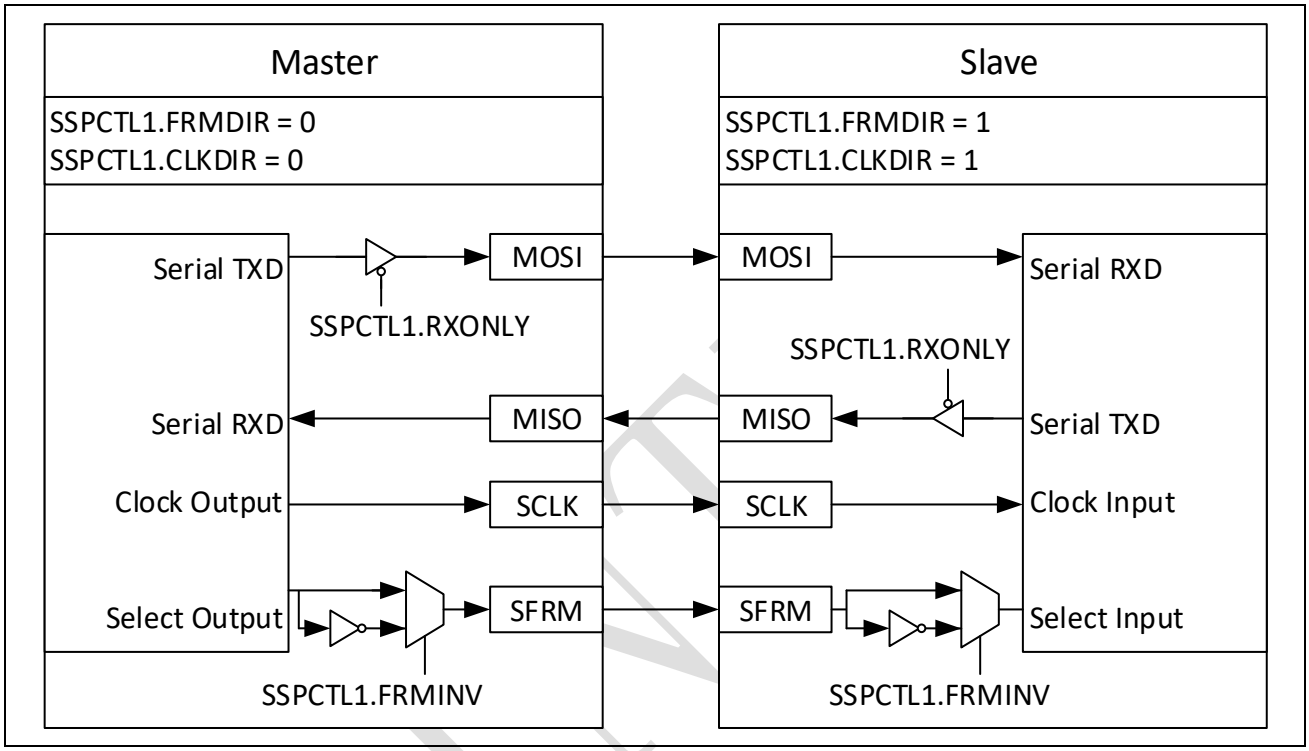
19.4 SSP function description

This section describes the various modes of operation of the SSP. Include connection, baud rate generation, FIFO operation, data frame formats and interrupts.

19.4.1 Introduction to operation

SSP can be configured as master or slave. The master send clock SCLK and signal SFRM to slave, and slave begin transfer data based on SCLK and SFRM. The master data port (MOSI/MISO) connect with slave data port with the same name. Master send data to MOSI, receive data from MISO. Slave receive data from MOSI, and send data to MISO. As shown in Figure 19-2, a typical connections of the SSP for communications between two controllers: a master and a slave.

Figure 19-2: SSP master-slave connection



There are three possible methods for data transmission:

- Master sends data; slave sends dummy data.
- Master sends data; slave sends data.
- Master sends dummy data; slave sends data.

When sending dummy data, SSPCTL1.RXONLY can be set high if do not want to send data from serial TXD to external MOSI port.

19.4.2 Baud rate generation

SSP baud rate clock is generated by SSPCLK, which is divided from SYSCLK1 as shown in Figure 3-11. The SSP clock divider is 16 bits.

19.4.3 FIFO operation

The SPD1148 SSP has two independent FIFOs for transmitting and receiving. The CPU will load and empty those FIFOs. The CPU will transfer one FIFO entry per access and the accesses must always be

32-bits wide. The CPU writes to TXFIFO are 32-bits wide, but the serializing logic will ignore all bits beyond the programmed FIFO data size (SSPCTL0.ESIZESEL and SSPCTL0.SIZESEL). The CPU reads from the RXFIFO are also 32-bits wide with zeros inserted in the MSBs down to the programmed data size.

The CPU only see one 32-bit location for both TXFIFO and RXFIFO. For data transmission, the SSP interface takes the data from TXFIFO, serialized the data and transmits the data through TXD interface signal to the external peripherals. For data receiving, the serial data through RXD serial interface signal is converted to parallel data and written into the RXFIFO. Depends on whether the access is a READ or WRITE transfer from CPU, the FIFO Data Register automatically target the RXFIFO or TXFIFO. From the memory-map perspective, the TXFIFO and RXFIFO share the same address.

An interrupt service request is generated if a programmable FIFO trigger threshold exceeded which signals the CPU to empty the RXFIFO or refill the TXFIFO.

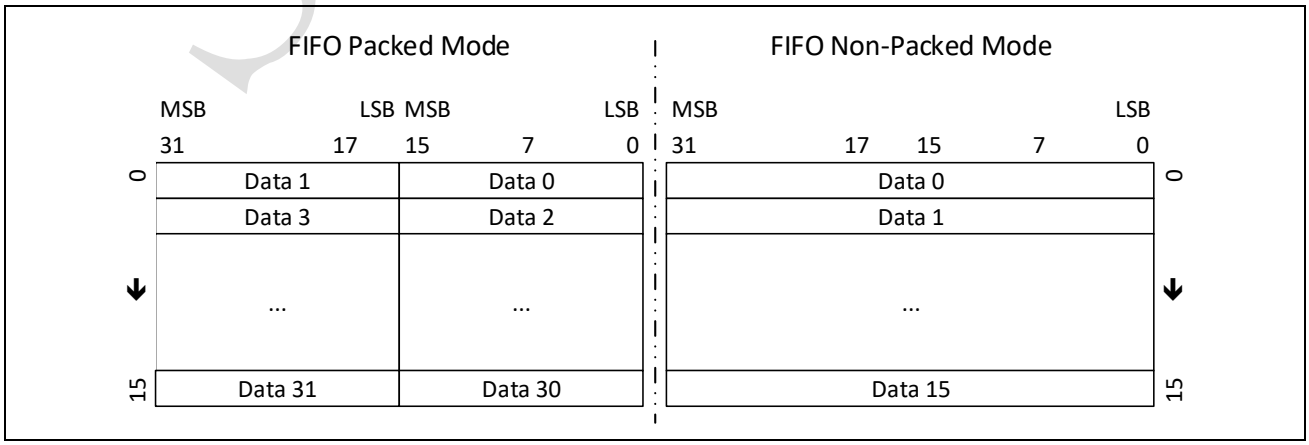
Note: If SSP was configured as slave and worked in receive only mode (SSPCTL1.RXONLY=1), there is no need to write Transmit FIFO when receiving data (half-duplex). However, writing Transmit FIFO can still sending data.

19.4.3.1 Parallel data formats for FIFO storage

The sample data size can be 8, 16, 18 or 32 bits. The data in the FIFOs is either stored as 32-bits wide per data sample or as 16-bits wide per sample (packed mode). Within 32-bits or 16-bits sample mode, the stored data sample is right-justified, with LSB of the word in bit 0. The unused bits are packed as zero above the MSB for read mode and ‘don’t care’ bit for write mode. The logic in SSP will automatically formats data in the Transmit FIFO so that the sample data will be properly transmitted on TXD signal with the selected frame format.

When the FIFOs are operating in packed mode, as shown in Figure 19-3, each FIFO is 32 rows in depth by 16-bits in width for a total of 32 data samples. Each sample can be 8 or 16 bits in length. When the data is serialized and transmitted, bits15-0 are transmitted first, followed by bits31-16. The best practice for packed mode is to treat FIFO as a single entry of two samples. Thus, the CPU should write and read 32 bits of data at a time where each Write or Read transfers two samples. The entire FIFO width (32 bits) must be read or written in this mode. Also the FIFO trigger threshold need to be calculated in 32-bits read and write, not based on 16-bits.

Figure 19-3: FIFO packed and non-packed mode



Read SSPSTS.RFODDSTS and SSPSTS.TFODDSTS can show even sampled data in FIFO or not. These state bits will help to read data in packed FIFO correctly.

19.4.3.2 Trailing bytes in RXFIFO

When the number of samples in the RXFIFO is less than its trigger threshold level and no additional data is received, the remaining bytes are called RXFIFO trailing bytes. RXFIFO trailing bytes can be handled by the CPU. RXFIFO trailing bytes are identified by means of a time-out mechanism and the existence of data within the RXFIFO after timeout.

The trailing byte is triggered by timeout logic in implementation. A timeout happens when the RXFIFO has been idled for a period time defined by SSPTO.VAL field. When a timeout occurs, the receiver timeout interrupt bit SSPSTS.RXTOINT is set to 1 and if the receiver timeout interrupt enable bit SSPCTL1.RXTOIE is set, a timeout interrupt signals the CPU that a timeout condition has occurred. The timeout timer is reset after a new data sample is received into the RXFIFO. Once the SSPSTS.RXTOINT bit is set, it must be cleared by writing 0x1 to the SSPSTS.RXTOINT bit.

The trailing bytes left in the RXFIFO are handled by the CPU polling method by default. If a timeout occurs, the CPU is only interrupted by a timeout interrupt if it has been enabled by setting the Receiver Time-out Interrupt Enable field (SSPCTL1.RXTOIE). To read out the trailing bytes from the RXFIFO, software should wait for the timeout interrupt and then read all trailing bytes as indicated by the RXFIFO Odd Sample Status (SSPSTS.RFODDSTS), Receive FIFO Level (SSPSTS.RFLVL), and Receive FIFO Not Empty (SSPSTS.RNE) fields in the SSP Status Register.

Note: If FIFO Packed mode is enabled (SSPCTL0.FPCKEN=1), trailing bytes must be removed using programmed polling method. If the SSPSTS.RFODDSTS field in SSP Status Register is set, then the last FIFO line only contains one sample.

19.4.3.3 FIFO threshold

Each FIFO has a configurable trigger threshold that can be used to trigger an interrupt. When the number of entries in RXFIFO exceeds the RXFIFO trigger threshold (SSPCTL1.RFTH field in SSP Control Register 1), an interrupt can be generated (if enabled) to signal the CPU to empty the RXFIFO. When the number of entries in the TXFIFO is less than or equal to the TXFIFO trigger threshold (SSPCTL1.TFTH field in SSP Control Register 1) plus 1, an interrupt can be generated (if enabled) to signal the CPU to refill the TXFIFO.

The CPU can poll the SSP Status Register to determine how many samples are in a FIFO or whether the FIFO is full or empty. Software is responsible for ensuring that the proper RXFIFO/TXFIFO trigger threshold values are chosen to prevent RXFIFO overflow or TXFIFO underflow errors.

19.4.4 Frame formats

19.4.4.1 Clock phase and polarity controls

The SSPCTL1.CLKPOL control clock polarity and specifies an active high or low clock and has no significant effect on the transmission format. The SSPCTL1.CLKPHS control clock phase and selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

SCLK only toggles during active transfers (does not run continuously).

19.4.4.2 SFRM polarity control

The SSPCTL1.FRMINV control select signal polarity, normally used as high when idle and low when transfer data.

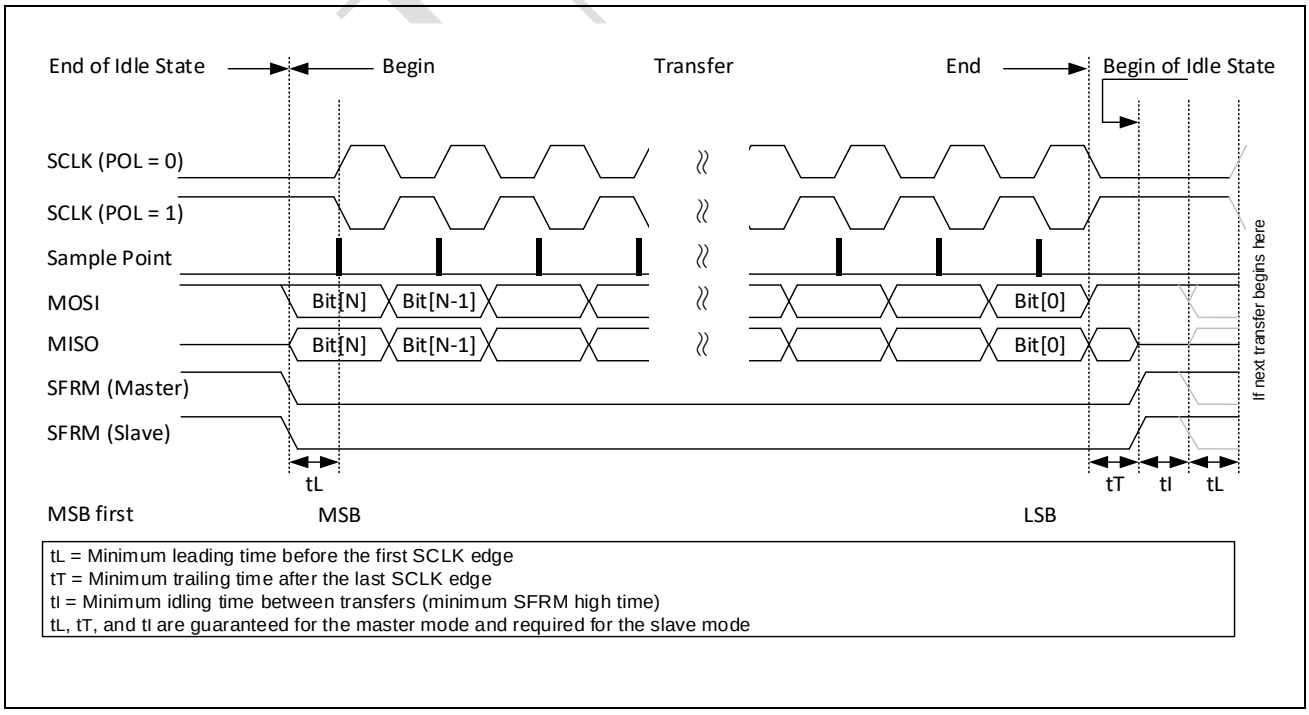
19.4.4.3 PHS = 0 frame format

When the SSP port is disabled or in idle mode, SFRM is high, MOSI and MISO is Hi-impedance, SCLK is low (SSPCTL1.CLKPOL = 0) or high (SSPCTL1.CLKPOL = 1). As shown in Figure 19-4.

When transmit data is ready to be sent and SSPCTL1.CLKPHS is low, the first SCLK edge is used to sample the first data bit for both master and slave. SFRM goes low (half clock period before the first edge of SCLK) and stay low during the transfer period. SFRM goes high again after half clock period of last SCLK edge.

The MSB of the transmit data is driven onto MOSI/MISO before half clock period of the first SCLK edge. The remainder bits of transmit data will be driven onto MOSI/MISO when the even edge of SCLK occurred. And odd edge of SCLK occurred will sample MOSI/MISO data bit and shift it into the LSB, MSB will be shift out and be latched for next edge driven onto MOSI/MISO. 1-32 bits can be transferred per frame.

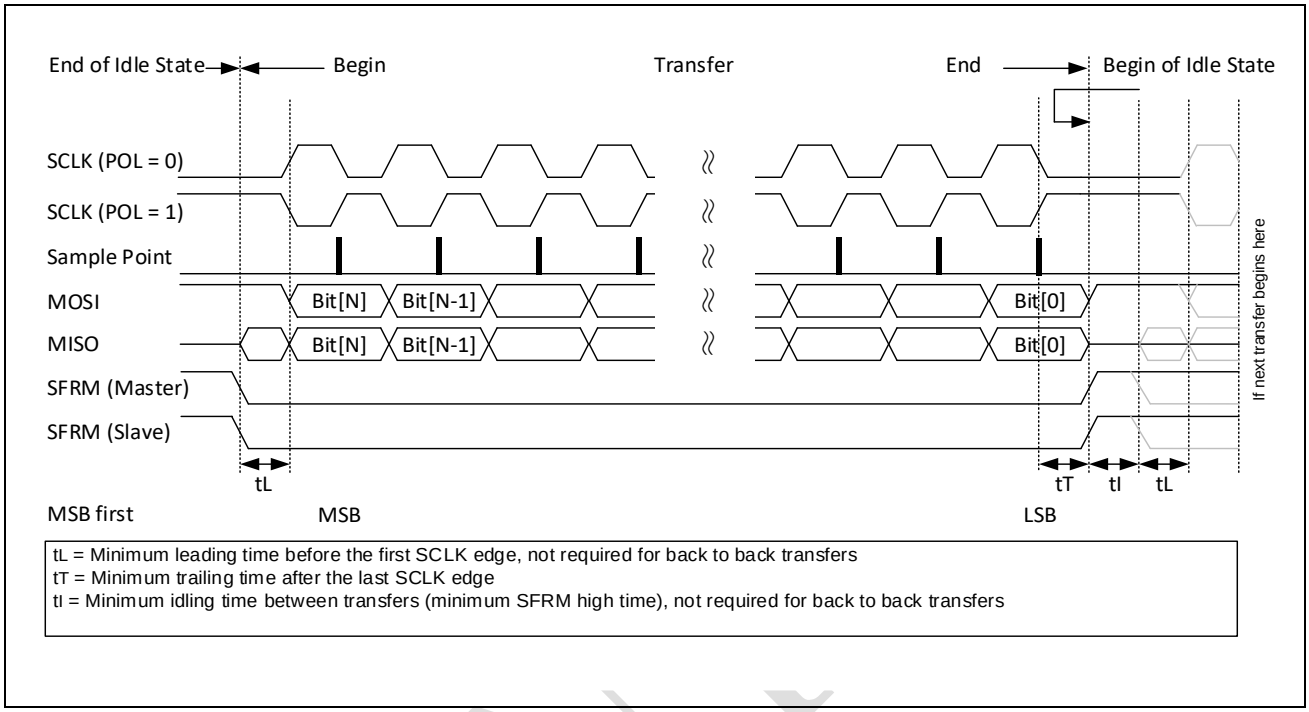
Figure 19-4: SSP clock format 0 (PHS = 0)



19.4.4.4 PHS = 1 frame format

When the SSP port is disabled or in idle mode, SFRM is high, MOSI and MISO is Hi-impedance, SCLK is low (SSPCTL1.CLKPOL = 0) or high (SSPCTL1.CLKPOL = 1). As shown in Figure 19-5.

Figure 19-5: SSP clock format 1 (PHS = 1)



When transmit data is ready to be sent and SSPCTL1.CLKPHS is low, the first SCLK edge is used to drive the first data bit for both master and slave. SFRM goes low (half clock period before the first edge of SCLK) and stay low during the transfer period. SFRM goes high again after last SCLK edge.

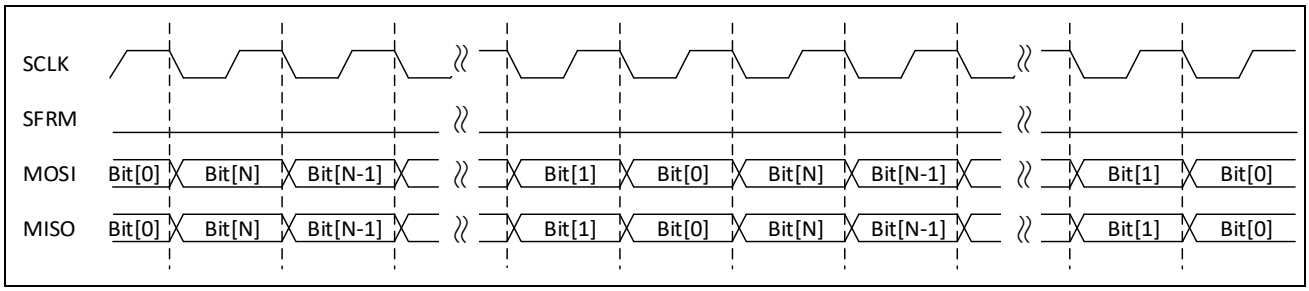
The MSB of the transmit data is driven onto MOSI/MISO on the first SCLK edge. The remainder bits of transmit data will be driven onto MOSI/MISO when the odd edge of SCLK occurred. And even edge of SCLK occurred will sample MOSI/MISO data bit and shift it into the LSB, MSB will be shift out and be latched for next edge driven onto MOSI/MISO. 1-32 bits can be transferred per frame.

19.4.4.5 Back to back mode

For back-to-back transfers, start and completion are like those of single transfer, but SFRM does not de-assert between words. Both transmitter and receiver are configured for the word length and internally track the start and end of frames. There is no “dead” bits and the LSB of one frame is followed immediately by the MSB of the next.

For enable back to back mode, be ensure that more than one data left in the TXFIFO when transfer data. If don’t want to use back to back mode, FIFO packed mode should be disabled, and TXFIFO should be empty before current data transfer finished.

Figure 19-6: Motorola SPI frame protocol (multiple transfers)



The SSP port can be either a master or a slave device, but the clock and frame direction must be the same. So the SSPCTL1.CLKDIR and SSPCTL1.FRMDIR fields must either both be set or cleared.

Note: The input clock to SSP port must not be active when SFRM is de-asserted. When the SSP port is slave to clock and frame, SSPCTL1.SLVCLKSEL must be set.

19.4.5 SSP interrupts

The SSP Status registers (SSPSTS) contain bits that signal overrun errors as well as the TXFIFO and RXFIFO service requests. Each of these hardware-detected events signals an interrupt request to the interrupt controller. SSPSTS also contains flags that indicate if the SSP port is actively transmitting data, if the TXFIFO is not full, and if the RXFIFO is not empty. A signal-interrupt signal is sent to the interrupt controller. These events can cause an interrupt request: receiver time out, RXFIFO overrun, RXFIFO service request, and TXFIFO service request.

Bits that cause an interrupt request remain set until they are cleared by writing a 0b1 to each bit. Once a status bit is cleared, the interrupt is cleared. Read-write bits are called status bits (status bits are referred to as ‘sticky’; and once set by hardware, they can only be cleared by writing a 0b1 to each bit), read-only bits are called flags. Writing a 0b1 to a sticky status bit clears it, writing a 0b0 has no effect. Read-only flags are set to 0b1 and are cleared automatically to 0b0 by hardware, and writes have no effect. Some bits that cause interrupt requests have corresponding mask bits in the Control registers (SSPCTL0/ SSPCTL1).

All bits in SSPSTS are read-only except the RFOVF, TFUDF and RXTO, which are all read-write.

The reset state of read-write bits is 0b0 and all bits return to their reset state when SSPCTL0.bit.EN is cleared.

19.5 Registers

19.5.1 SSP register map

Table 19-2: SSP Module Base Address

Peripheral Module	Base Address
SSP	0x4000 5000

Table 19-3: SSP Register Map

Register	Offset	Description	Reset Value
SSPCTL0	0x0	SSP Control Register 0	0x00000000
SSPCTL1	0x4	SSP Control Register 1	0x00000000
SSPSTS	0x8	SSP Status Register	0x0000F004
SSPFRC	0xC	SSP Interrupt Force Register	0x00000000
SSPDATA	0x10	SSP Data Register	0x00000000
SSPTO	0x28	SSP Time Out Register	0x00000000

19.5.2 SSP registers

Table 19-4: SSP Control Register 0 (SSPCTL0) Layout

SSPCTL0 (SSP Control Register 0) Offset: 0x0 Default: 0x00000000
Access: SSP -> SSPCTL0.all

31	30	29	28	27	26	25	24
RESERVED_31	RESERVED_30	FPACKEN	RESERVED_28_27		RESERVED_26_24		
23	22	21	20	19	18	17	16
TFINTMSK	RFINTMSK	RESERVED_21	ESIZESEL	RESERVED_19_8			
15	14	13	12	11	10	9	8
RESERVED_19_8							
7	6	5	4	3	2	1	0
EN	RESERVED_6	FRMSEL		SIZESEL			

Table 19-5: SSP Control Register 0 (SSPCTL0) Description

Bits	Field Name	Type	Reset	Description
31	RESERVED_31	RW	0x0	Reserved.
30	RESERVED_30	RO	0x0	Reserved.
29	FPACKEN	RW	0x0	FIFO packing enable 0: Disable FIFO packing 1: Enable FIFO packing

Bits	Field Name	Type	Reset	Description
28:27	RESERVED_28_27	RO	0x0	Reserved.
26:24	RESERVED_26_24	RW	0x0	Reserved.
23	TFINTMSK	RW	0x0	Transmit FIFO underflow interrupt mask 0: TFUDF events generate an SSP interrupt 1: TFUDF events do NOT generate an SSP interrupt
22	RFINTMSK	RW	0x0	Receive FIFO overflow interrupt mask 0: RFOVF events generate an SSP interrupt 1: RFOVF events do NOT generate an SSP interrupt
21	RESERVED_21	RO	0x0	Reserved.
20	ESIZESEL	RW	0x0	Extended data size select 0: A 0 is pre-appended to the SIZESEL value to set the SIZESEL range from 1 to 16 bits 1: A 1 is pre-appended to the SIZESEL value to set the SIZESEL range from 17 to 32 bits
19:8	RESERVED_19_8	RO	0x0	Reserved.
7	EN	RW	0x0	Synchronous serial port Enable 0: Disable SSP 1: Enable SSP
6	RESERVED_6	RO	0x0	Reserved.
5:4	FRMSEL	RW	0x0	Frame Format 00: Motorola Serial Peripheral Interface (SPI) 01: Texas Instruments Synchronous Serial Protocol (SSP) 10: Reserved 11: Programmable Serial Protocol (PSP)
3:0	SIZESEL	RW	0x0	Data size select 0000: 1-bit if ESIZESEL=0 and 17-bit if ESIZESEL=1 0001: 2-bit if ESIZESEL=0 and 18-bit if ESIZESEL=1 0010: 3-bit if ESIZESEL=0 and 19-bit if ESIZESEL=1 0011: 4-bit if ESIZESEL=0 and 20-bit if ESIZESEL=1 0100: 5-bit if ESIZESEL=0 and 21-bit if ESIZESEL=1 0101: 6-bit if ESIZESEL=0 and 22-bit if ESIZESEL=1 0110: 7-bit if ESIZESEL=0 and 23-bit if ESIZESEL=1

Bits	Field Name	Type	Reset	Description
				0111: 8-bit if ESIZESEL=0 and 24-bit if ESIZESEL=1 1000: 9-bit if ESIZESEL=0 and 25-bit if ESIZESEL=1 1001: 10-bit if ESIZESEL=0 and 26-bit if ESIZESEL=1 1010: 11-bit if ESIZESEL=0 and 27-bit if ESIZESEL=1 1011: 12-bit if ESIZESEL=0 and 28-bit if ESIZESEL=1 1100: 13-bit if ESIZESEL=0 and 29-bit if ESIZESEL=1 1101: 14-bit if ESIZESEL=0 and 30-bit if ESIZESEL=1 1110: 15-bit if ESIZESEL=0 and 31-bit if ESIZESEL=1 1111: 16-bit if ESIZESEL=0 and 32-bit if ESIZESEL=1

Table 19-6: SSP Control Register 1 (SSPCTL1) Layout

SSPCTL1 (SSP Control Register 1) Offset: 0x4 Default: 0x00000000
Access: SSP -> SSPCTL1.all

31	30	29	28	27	26	25	24
TXTRITIME	TXTRIEN	RESERVED_29	SLVCLKSEL	RESERVED_27_26		CLKDIR	FRMDIR
23	22	21	20	19	18	17	16
RXONLY	TRAIL	RESERVED_21	RESERVED_20	RXTOIE	RESERVED_18_17		FRMINV
15	14	13	12	11	10	9	8
RESERVED_15	RESERVED_14	RFTH				TFTH	
7	6	5	4	3	2	1	0
TFTH		RESERVED_5	CLKPHS	CLKPOL	RESERVED_2	TFIE	RFIE

Table 19-7: SSP Control Register 1 (SSPCTL1) Description

Bits	Field Name	Type	Reset	Description
31	TXTRITIME	RW	0x0	TXD tri-stated enable on last phase 0: SSP_TXD is tri-stated 1/2 clock cycle after the beginning of the LSB 1: SSP_TXD output signal is tri-stated on the clock edge that ends the LSB
30	TXTRIEN	RW	0x0	TXD tri-stated enable 0: SSP_TXD output signal is not tri-stated 1: SSP_TXD is tri-stated when not transmitting data
29	RESERVED_29	RW	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
28	SLVCLKSEL	RW	0x0	Slave clock free running control 0: Clock input to SSP_SCLK is not active when SSPCTL1.CLKDIR=1 (slave mode) 1: Clock input to SSP_SCLK is active when SSPCTL1.CLKDIR=1 (slave mode)
27:26	RESERVED_27_26	RO	0x0	Reserved.
25	CLKDIR	RW	0x0	SSP clock (SSP_SCLK) direction 0: Master mode, SSP port drives SSP_SCLK 1: Slave mode, SSP port receives SSP_SCLK
24	FRMDIR	RW	0x0	SSP frame (SSP_SFRM) direction 0: Master mode, SSP port drives SSP_SFRM 1: Slave mode, SSP port receives SSP_SFRM
23	RXONLY	RW	0x0	Receive-only mode 0: Transmit/receive mode 1: Receive without transmit mode
22	TRAIL	RW	0x0	Trailing byte control 0: Trailing bytes are handled by CPU 1: Trailing bytes are handled by DMA
21	RESERVED_21	RW	0x0	Reserved.
20	RESERVED_20	RW	0x0	Reserved.
19	RXTOIE	RW	0x0	Receiver time-out interrupt enable 0: Disable receiver time-out interrupt 1: Enable receiver time-out interrupt
18:17	RESERVED_18_17	RO	0x0	Reserved.
16	FRMINV	RW	0x0	Invert frame signal 0: SSP_SFRM polarity is determined by the PSP polarity bits 1: SSP_SFRM polarity is inverted to the definition by PSP polarity bit
15	RESERVED_15	RW	0x0	Reserved.
14	RESERVED_14	RW	0x0	Reserved.
13:10	RFTH	RW	0x0	Threshold level at which RXFIFO asserts interrupt. Level should be set to the preferred threshold value minus 1.
9:6	TFTH	RW	0x0	Threshold level at which TXFIFO asserts interrupt. Level should be set to the preferred threshold value.
5	RESERVED_5	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
4	CLKPHS	RW	0x0	Motorola SPI SSP_SCLK phase setting 0: Start sampling data on the 1st SSP_SCLK edge after the start of a frame 1: Start sampling data on the 2nd SSP_SCLK edge after the start of a frame
3	CLKPOL	RW	0x0	Motorola SPI SSP_SCLK polarity setting 0: SSP_SCLK of inactive or idle state is low 1: SSP_SCLK of inactive or idle state is high
2	RESERVED_2	RW	0x0	Reserved.
1	TFIE	RW	0x0	Transmit FIFO interrupt enable 0: Disable TXFIFO interrupt 1: Enable TXFIFO interrupt
0	RFIE	RW	0x0	Receive FIFO interrupt enable 0: Disable RXFIFO interrupt 1: Enable RXFIFO interrupt

Table 19-8: SSP Status Register (SSPSTS) Layout

SSPSTS (SSP Status Register) Offset: 0x8 Default: 0x0000F004 Access: SSP -> SSPSTS.all							
31	30	29	28	27	26	25	24
RFODDSTS	TFODDSTS	RESERVED_29_24					
23	22	21	20	19	18	17	16
RESERVED_23	SLVCLKSTS	TFUDF	RESERVED_20	RXTO	RESERVED_18_16		
15	14	13	12	11	10	9	8
RFLVL				TFLVL			
7	6	5	4	3	2	1	0
RFOVF	RFS	TFS	BUSY	RNE	TNF	RESERVED_1_0	

Table 19-9: SSP Status Register (SSPSTS) Description

Bits	Field Name	Type	Reset	Description
31	RFODDSTS	RO	0x0	RX FIFO odd sample status This bit is used only when FIFO packing is enabled 0: RXFIFO entry has 2 samples 1: RXFIFO entry has 1 sample
30	TFODDSTS	RO	0x0	TX FIFO odd sample status This bit is used only when FIFO packing is enabled 0: TXFIFO entry has a even number of samples 1: TXFIFO entry has an odd number of samples
29:24	RESERVED_29_24	RO	0x0	Reserved.

Bits	Field Name	Type	Reset	Description
23	RESERVED_23	RW	0x0	Reserved.
22	SLVCLKSTS	RO	0x0	Slave clock status 0: The SSP port is ready for slave clock operations 1: The SSP port is currently busy synchronizing slave mode signals
21	TFUDF	RW	0x0	Transmit FIFO underflow 0: Read a 0 indicates no TXFIFO overflow Write a 0 has no effect 1: Read a 1 indicates data fetch from an empty TXFIFO was attempted Write a 1 clears the latched flag
20	RESERVED_20	RO	0x0	Reserved.
19	RXTO	RW	0x0	Receiver time-out 0: Read a 0 indicates no time-out on receiving. Write a 0 has no effect. 1: Read a 1 indicates time-out on receiving. Write a 1 clears the latched flag.
18:16	RESERVED_18_16	RO	0x0	Reserved.
15:12	RFLVL	RO	0xF	Receive FIFO level The value is number of entries minus one in RXFIFO.
11:8	TFLVL	RO	0x0	Transmit FIFO level The value is number of entries in TXFIFO.
7	RFOVF	RW	0x0	Receive FIFO overflow 0: Read a 0 indicates no receive FIFO overflow. Write a 0 has no effect. 1: Read a 1 indicates data put to a full RXFIFO was attempted. Write a 1 clears the latched flag.
6	RFS	RO	0x0	Receive FIFO service request 0: RXFIFO level is at or below RXFIFO threshold (RFTH + 1), or SSP port is disabled 1: RXFIFO level exceeds RXFIFO threshold (RFTH + 1)
5	TFS	RO	0x0	Transmit FIFO service request 0: TXFIFO level exceeds the threshold (TFTH), or SSP port disabled 1: TXFIFO level is at or below threshold (TFTH)
4	BUSY	RO	0x0	SSP busy 0: SSP port is idle or disabled

Bits	Field Name	Type	Reset	Description
				1: SSP port is currently transmitting or receiving framed data
3	RNE	RO	0x0	Receive FIFO not empty 0: RXFIFO is empty 1: RXFIFO is not empty
2	TNF	RO	0x1	Transmit FIFO not full 0: TXFIFO is full 1: TXFIFO is not full
1:0	RESERVED_1_0	RO	0x0	Reserved.

Table 19-10: SSP Interrupt Force Register (SSPFRC) Layout

SSPFRC (SSP Interrupt Force Register) Offset: 0xC Default: 0x00000000
Access: SSP -> SSPFRC.all

31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
RFOVF	RFREQ	TFREQ	RESERVED_4_0				

Table 19-11: SSP Interrupt Force Register (SSPFRC) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	RFOVF	RW	0x0	Force RXFIFO overrun (Non-maskable) 0: Write a 0 release the force 1: Write a 1 force a RXFIFO overrun interrupt This bit needs to be cleared manually
6	RFREQ	RW	0x0	Force RXFIFO service request (Non-maskable) 0: Write a 0 release the force 1: Write a 1 force a RXFIFO service request interrupt This bit needs to be cleared manually
5	TFREQ	RW	0x0	Force TXFIFO service request (Non-maskable) 0: Write a 0 release the force 1: Write a 1 force a TXFIFO service request interrupt This bit needs to be cleared manually
4:0	RESERVED_4_0	RO	0x0	Reserved.

Table 19-12: SSP Data Register (SSPDATA) Layout

SSPDATA (SSP Data Register) Offset: 0x10 Default: 0x00000000 Access: SSP -> SSPDATA.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 19-13: SSP Data Register (SSPDATA) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Data to be written to the TXFIFO or read from the RXFIFO

Table 19-14: SSP Time Out Register (SSPTO) Layout

SSPTO (SSP Time Out Register) Offset: 0x28 Default: 0x00000000 Access: SSP -> SSPTO.all							
31	30	29	28	27	26	25	24
RESERVED_31_24							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 19-15: SSP Time Out Register (SSPTO) Description

Bits	Field Name	Type	Reset	Description
31:24	RESERVED_31_24	RO	0x0	Reserved.
23:0	VAL	RW	0x0	Timeout interval defined as CLK*(VAL+1).

20 Flash memory

20.1 Features

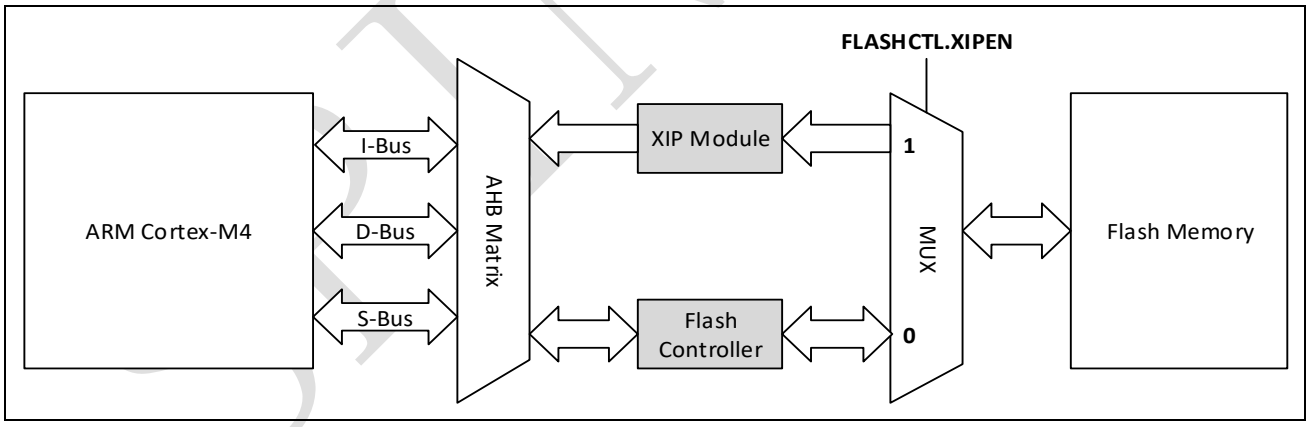
The SPD1148 Flash memory module has the following key features:

- Up to 128KB on-chip Flash memory
- Memory organization:
  - Main Memory Block: with total 256 sectors; 128 x 32 bits per sector
  - NVR Memory Block: with total 2 sectors; 128 x 32 bits per sector
- Program/Sector Erase cycles: minimum 100000 cycles @ T_J=85 °C
- Data retention period: over 10 years @ T_J=85 °C
- ECC (Error Checking and Correction) protection

Flash memory interface features:

- Read interface with XIP (Execute-In-Place) module
- Flash controller for Read/Program/Erase operation
- Read/Write protection
- Multi-zone protection

Figure 20-1: Flash memory access interfaces



20.2 Flash module organization

The flash memory organization is based on a main memory block containing 256 sectors of 128 word and a NVR block containing 2 sectors of 128 word as shown in Figure 20-2 and Table 20-1.

The flash memory is organized as 32-bit wide memory cells that can be used for storing both code and data constants.

The NVR memory block is divided into two parts:

- OTP memory – The area can be configured as OTP memory by programming the first word cell with data 0x4C4F434B. Otherwise, the area is just used as normal data memory.
- Configuration Words – The area contains the word data for multi-zone security setting.

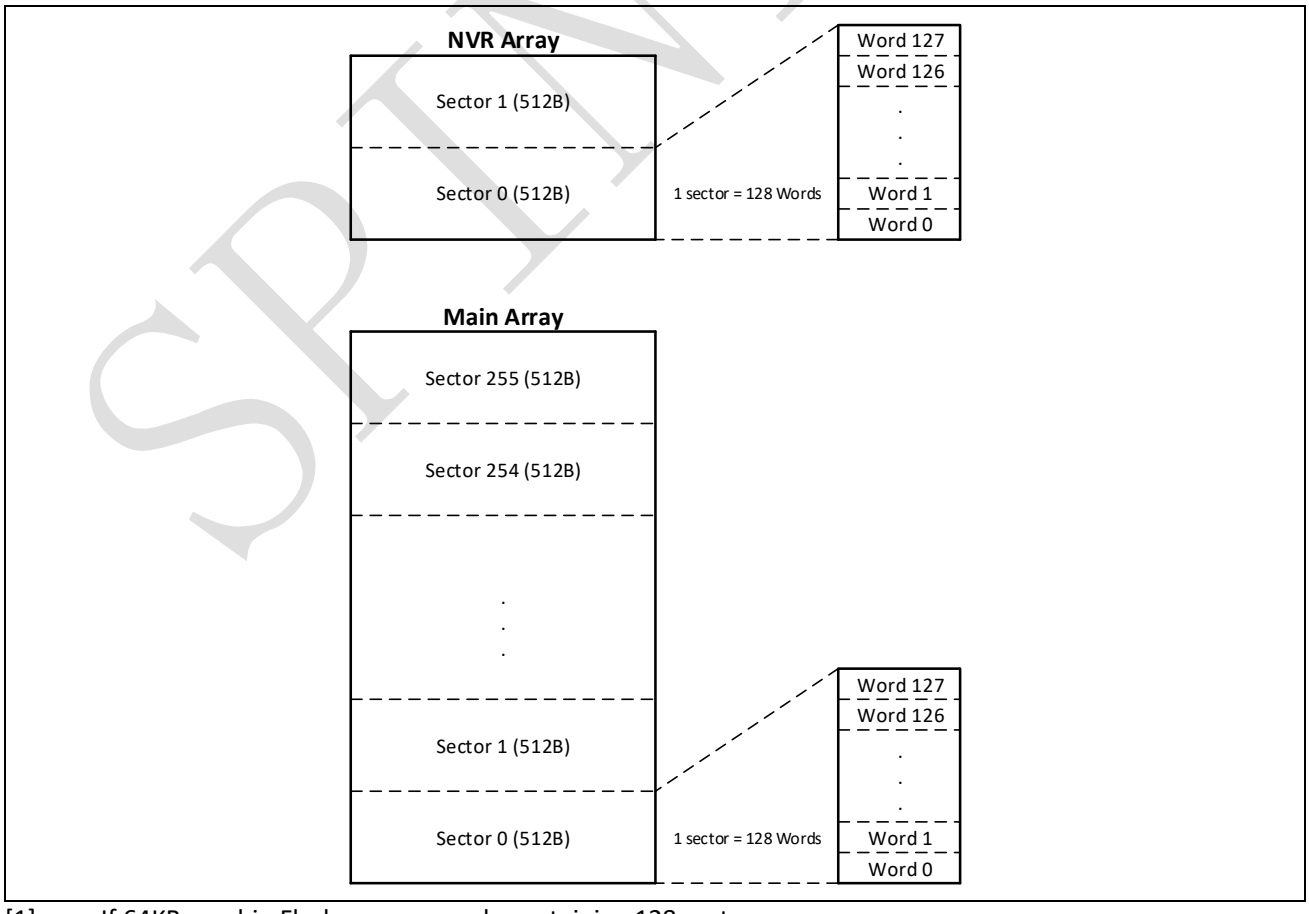
Write operation to the main memory block and the NVR memory block are managed by the Flash Controller as shown in Figure 20-1. The high voltage needed for Program/Erase operations is internally generated.

The main Flash memory can be protected against different types of unwanted access (read/write/erase). There are two types of protection:

- Sector Write Protection
- Read Protection

For operations on the Flash memory through Flash Controller, the XIP module should be disabled first (FLASHCTL.XIPEN = 0).

Figure 20-2: Logical structure of the Flash module



[1] If 64KB on-chip Flash memory, only containing 128 sectors.

Table 20-1: Flash module organization

Block	Name	Base address	Size
Main Memory	Sector 0	0x1000 0000	512 bytes
	Sector 1	0x1000 0200	512 bytes
	Sector 2	0x1000 0400	512 bytes
	Sector 3	0x1000 0600	512 bytes
	…	…	…
	Sector 255	0x1001 FE00	512 bytes
NVR Memory	OTP memory	0x11000400	128 Words
	Configuration Words	0x11000600	128 Words
Flash memory interface registers	FLASHCTL	0x4000 8800	4 bytes
	FLASHADDR	0x4000 8804	4 bytes
	FLASHDIN	0x4000 8808	4 bytes
	FLASHDOUT	0x4000 880C	4 bytes
	Reserved	0x4000 8810	28 bytes
	FLASHWP0	0x4000 882C	4 bytes
	FLASHWP1	0x4000 8830	4 bytes
	FLASHWP2	0x4000 8834	4 bytes
	FLASHWP3	0x40008838	4 bytes
	FLASHREGKEY	0x4000 883C	4 bytes

[1] If 64KB on-chip Flash memory, only containing 128 sectors.

20.3 Read operation

The CPU can read data from the embedded Flash module through the XIP module or the Flash Controller.

20.3.1 Read through XIP module

Through the XIP module, the flash memory can be addressed directly, as a common memory space. The main task of the XIP module is to generate the control signals to read from the flash memory. The Cortex-M4 can use the XIP module to fetch the instruction over the I-Code bus and the literal pool (constant/data) over the D-Code bus.

Read accesses may be performed with latency. The latency is the number of wait states for a read operation on-the-fly. The wait states represent the ratio of the SYSCLK (system clock) period to the Flash memory access time. Please see Table 20-2 for details.

Table 20-2: Flash memory read access latency

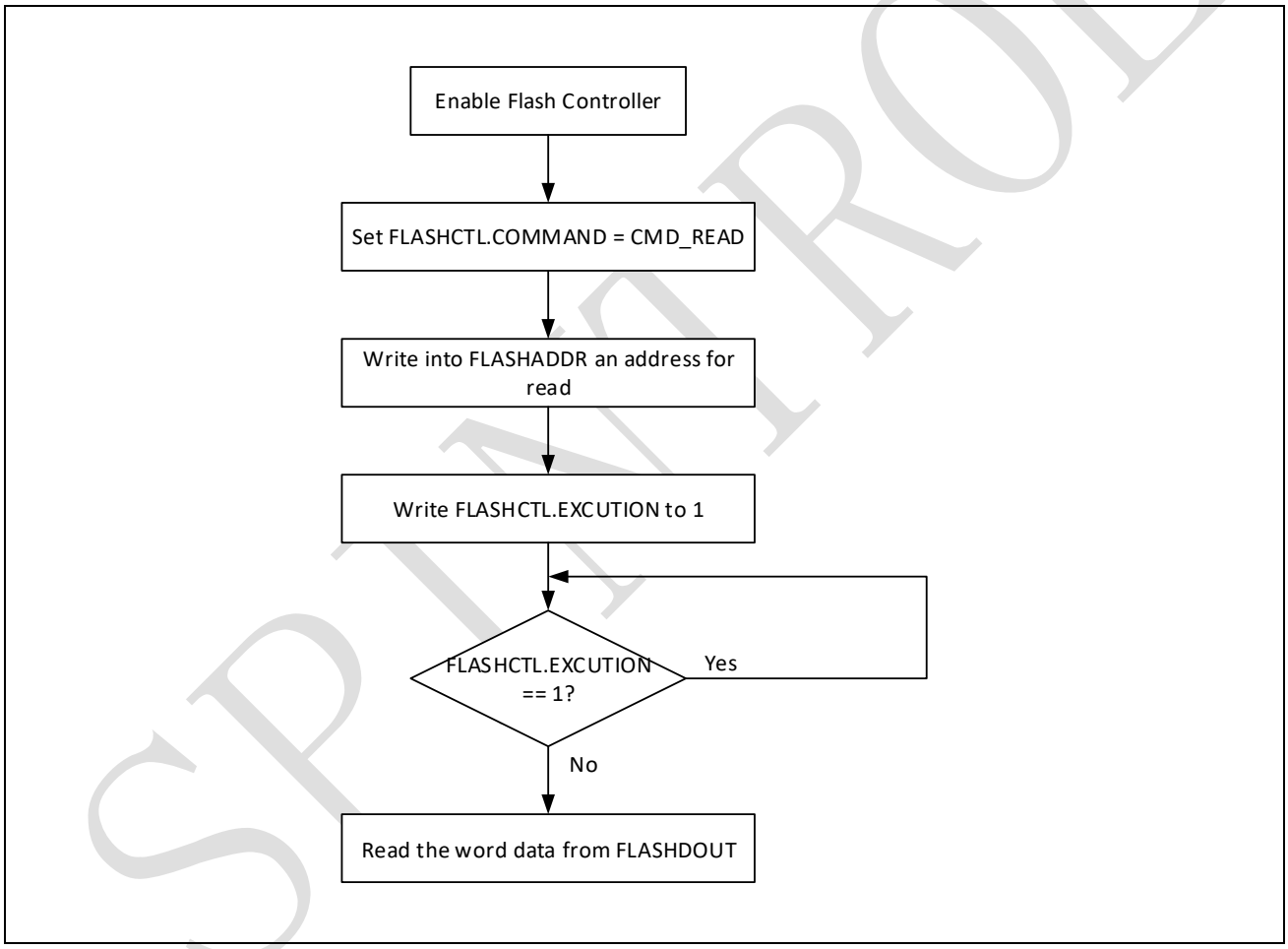
System Clock Frequency	Wait States
0 < SYSCLK ≤ 25 MHz	0
25 MHz < SYSCLK ≤ 50 MHz	1
50 MHz < SYSCLK ≤ 75 MHz	2
75 MHz < SYSCLK ≤ 100 MHz	3
100 MHz < SYSCLK ≤ 125 MHz	4
125 MHz < SYSCLK ≤ 150 MHz	5
150 MHz < SYSCLK ≤ 175 MHz	6
175 MHz < SYSCLK ≤ 200 MHz	7

20.3.2 Read through Flash Controller

The CPU can also read a word data at a time from the Flash memory through the Flash Controller. The reading sequence is as follows:

- Set the COMMAND bits in the FLASHCTL register with CMD_READ
- Set the FLASHADDR register to select a memory cell to read
- Set the EXCUTION bit in the FLASHCTL register
- Wait for the EXCUTION bit to be reset
- Read the word data from the FLASHDOUT register

Figure 20-3: Flash memory reading procedure



20.4 Program and erase operation

The Flash Controller handles the program and erase operations of the flash memory. The Flash Controller consists of nine 32-bit registers.

- Flash control register (FLASHCTL)
- Flash address register (FLASHADDR)
- Flash input data register (FLASHDIN)
- Flash output data register (FLASHDOUT)

- Flash write protection registers (FLASHWP0 ~ FLASHWP3)
- Flash unlock key register (FLASHREGKEY)

20.4.1 Enable Flash controller

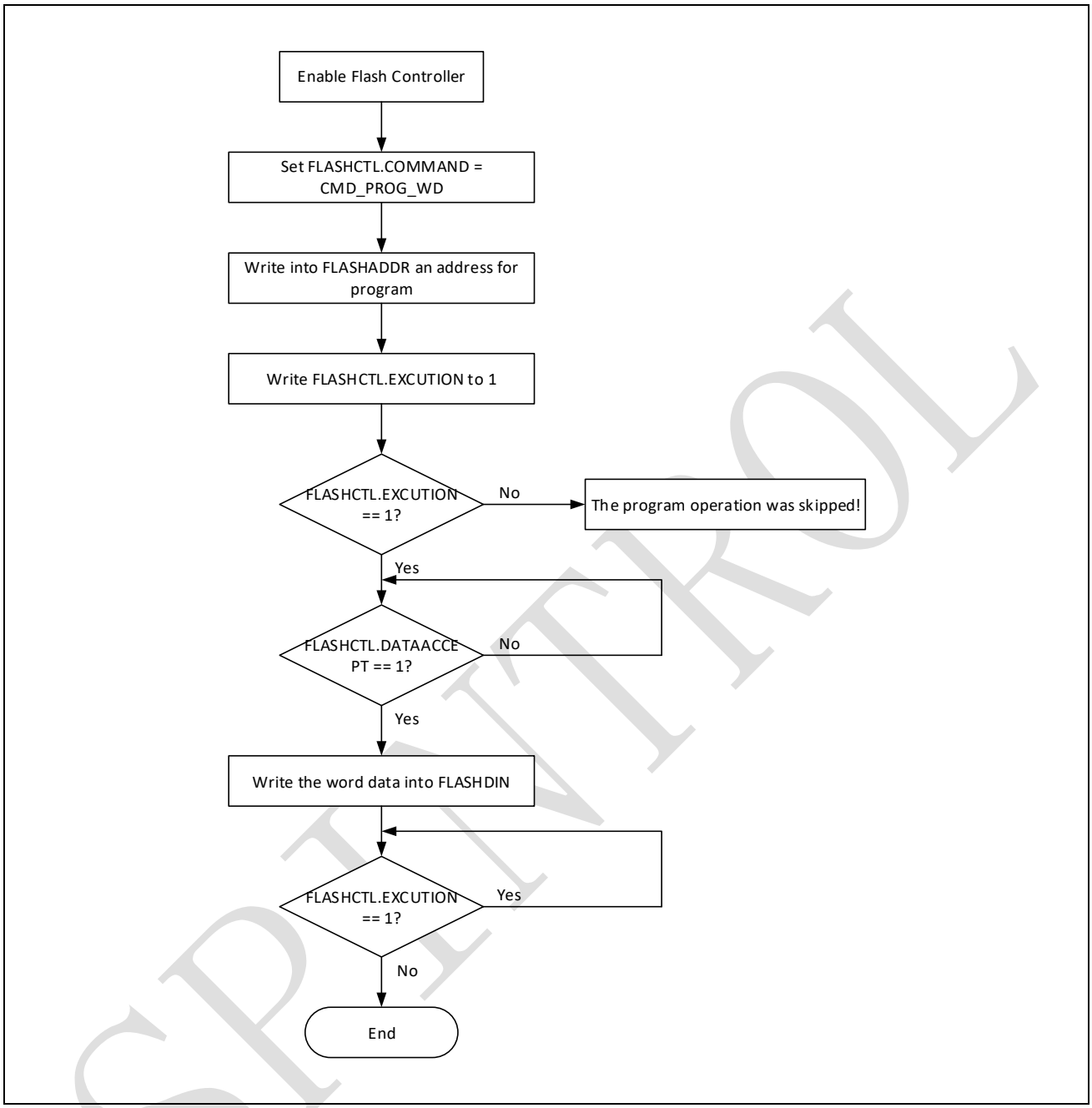
After reset, the Flash controller is disabled. The Flash controller can be enabled by setting FLASHCTL.XIPEN to 0. However, the FLASHCTL register is write-protected. An unlock key value (0x1ACCE551) should be written to the FLASHREGKEY register to enable the write access to the protected registers.

20.4.2 Flash memory programming

The Flash memory can be programmed 32 bits (a word) at a time. The Flash memory programming sequence is as follows:

- Set the COMMAND bits in the FLASHCTL register with CMD_PROG_WD
- Set the FLASHADDR register to select a memory cell to program
- Set the EXCUTION bit in the FLASHCTL register and then check whether the bit value was set
- Wait for the DATAACCEPT bit in the FLASHCTL register to be set
- Write the word data to the FLASHDIN register
- Wait for the EXCUTION bit to be reset

Figure 20-4: Flash memory programming procedure



If the addressed main Flash memory location is write-protected by the FLASHWPx register, the EXCUTION bit could not be set and the program operation is skipped.

20.4.3 Flash memory erase

The main Flash memory can be erased by sector or completely (Chip Erase). And the NVR memory can only be erased by sector.

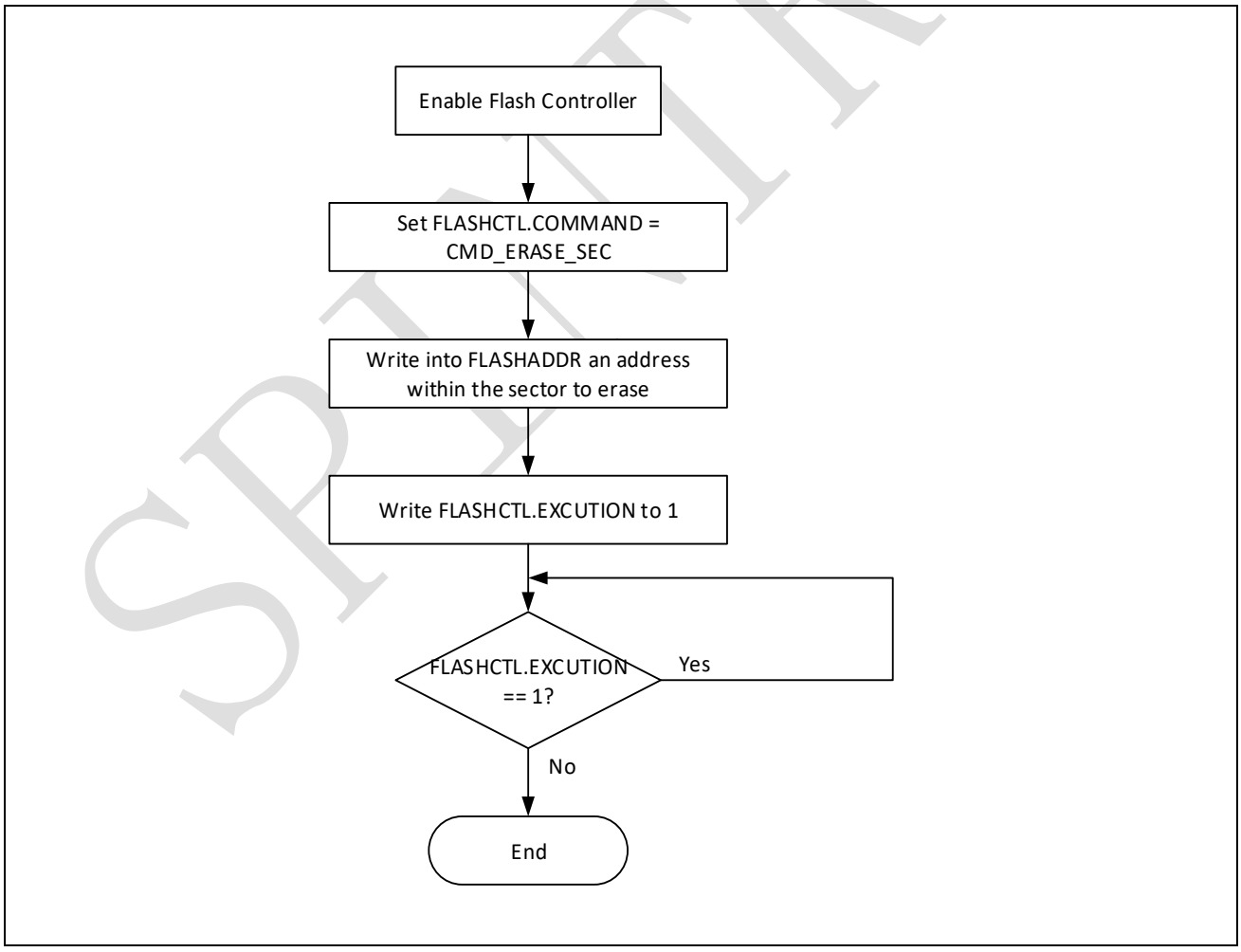
Sector Erase

A sector of the Flash memory can be erased using the Sector Erase feature of the Flash Controller. To erase a sector, the procedure below should be followed:

- Set the COMMAND bits in the FLASHCTL register with CMD_ERASE_SEC
- Set the FLASHADDR register to select a sector to erase
- Set the EXCUTION bit in the FLASHCTL register
- Wait for the EXCUTION bit to be reset

If the addressed main Flash memory location is write-protected by the FLASHWPx register, the EXCUTION bit could not be set and the sector erase operation is skipped.

Figure 20-5: Flash memory Sector Erase procedure

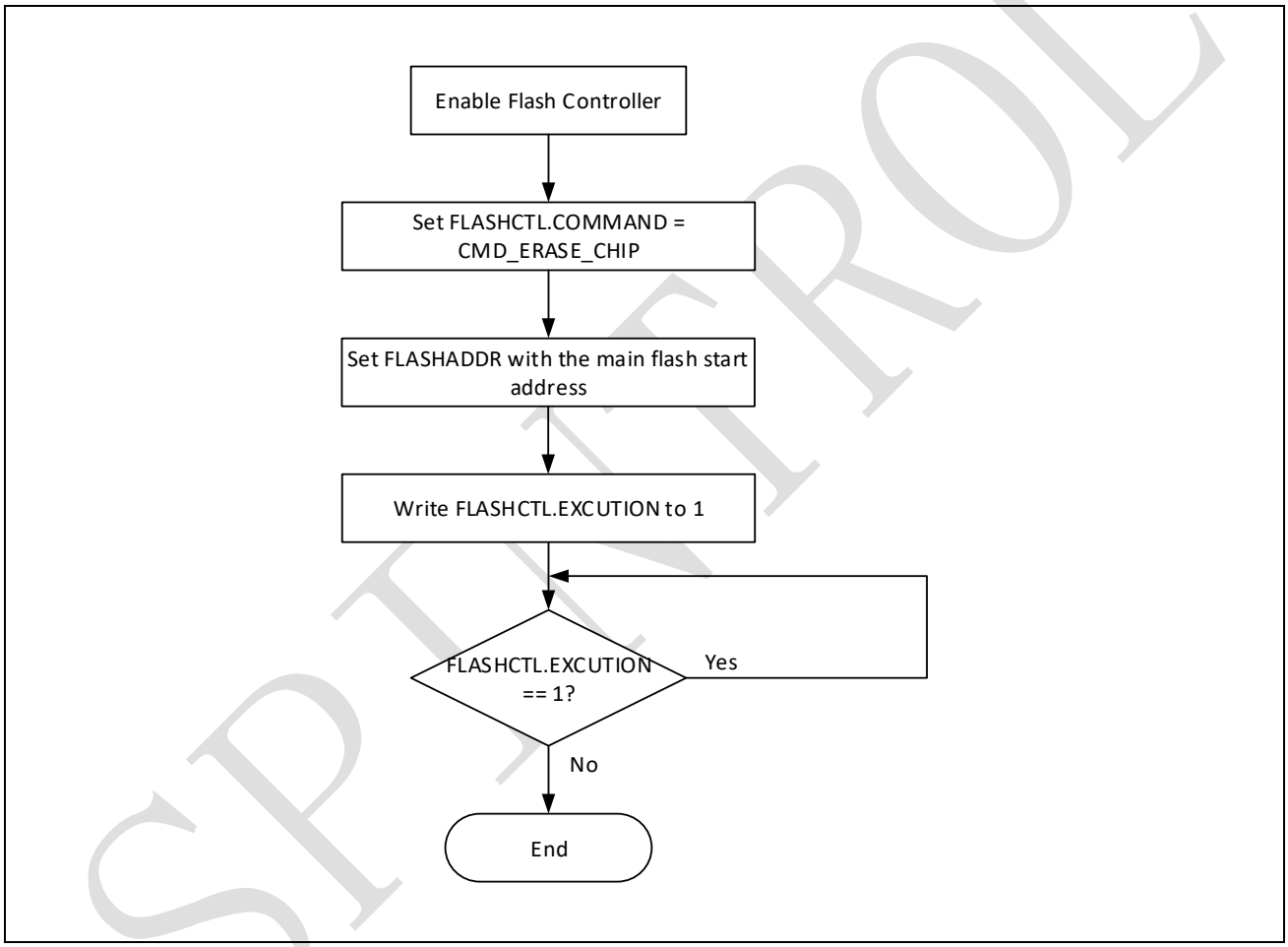


Chip Erase

The Chip Erase command can be used to completely erase all the sectors of the main Flash memory. The NVR memory block is not affected by this procedure. The following sequence is recommended:

- Set the COMMAND bits in the FLASHCTL register with CMD_ERASE_CHIP
- Set the FLASHADDR register with the main flash start address
- Set the EXCUTION bit in the FLASHCTL register
- Wait for the EXCUTION bit to be reset

Figure 20-6: Flash memory Chip Erase procedure



20.5 Protections

20.5.1 Multi-zone protection

The main Flash memory can be organized into four code space segments (Flash zones). The multi-zone protection technology provides an advanced code protection scheme to help multiple users securely share resources on a single chip. Once a Flash zone protection has been enabled:

Access to the Flash zone from other Flash zones is limited severely. The code running in other Flash zones can only jump to or call a routine in the protected Flash zone and the read/program/erase operations on the protected Flash zone are completely disabled.

- Access to the Flash zone through the debug interface is not allowed.

The multi-zone protection is activated by setting the Configuration Words in the NVR memory block and then, by applying a system reset to reload the new Configuration Words.

To disable the multi-zone protections:

- Erase the entire Configuration Words area (sector). As a result, the Configuration Words will be 0xFFFFFFFF. At this stage, the multi-zone protections are still enable.
- Reset the device to reload the Configuration Words and, to disable the multi-zone protection.

Note: The multi-zone protection feature doesn't include DRAM.
Erasing the Configuration Words area will automatically trigger a Chip Erase of the main Flash memory.

20.5.2 Write protection

The sectors of the main Flash memory can be protected against unwanted write due to loss of program counter contexts. The write protection is implemented with a granularity of one sector at a time. If a program or an erase operation is performed on a protected sector, the operation will be skipped.

The write protection is activated by set the FLASHWPx register bits and reset the register bits will disable the write protection.

20.6 Setup Flash timing

There are physically timing requirements for the Flash to avoid malfunction. It is a list of absolute values for the timing relationship of a series of events. An finite state machine (FSM) is designed in both the XIP module and the Flash controller, which guarantee the timing based on couting clock cycles with a list of timeout counts. For the ease of usage, a function is provide in the SDK named FLASH_SetTiming. It takes the frequency of CPU clock (HCLK) as input parameter and automatically sets all the timeout counts.

Whenever user plans to increase the HCLK frequency, FLASH_SetTiming should be called before the actual change of the HCLK due to the increased timeout counts requirement.

Whenever user plans to decrease the HCLK frequency, FLASH_SetTiming should be called after the actual change of the HCLK due to the decreased timeout counts requirement.

20.7 Configuration Words description

There are twenty-eight Configuration Words for multi-zone security, seven for each zone. They are configured by the end user depending on the application requirements.

The organization and description of these Configuration Words are as shown in Table 20-3. The Configuration Words can be read from the memory locations listed in Table 20-3 through the Flash Controller. The new programmed Configuration Words are loaded after a system reset.

Table 20-3: Organization and description of the Configuration Words

Address	Name	Description
0x11000600	ZONE0_INFO_LOCK	Lock the Configuration Words of zone 0 (0x11000600 – 0x1100063F) 0xFFFFFFFF: The Configuration Words of zone 0 can be programmed Others: The Configuration Words of zone 0 can not be programmed
0x11000604	ZONE0_ENCRYPT	Encrypt status of code in Flash zone 0 0xDEC0DE: The code in Flash zone 0 has been encrypted Others: The code in Flash zone 0 is raw code
0x11000608	ZONE0_DECRYPT	Decrypt status of code in Flash zone 0 0xAA621623: Success to decrypt code in Flash zone 0 0x1E55051: Fail to decrypt code in Flash zone 0 Others: Not valid
0x1100060C	ZONE0_FLASH_PROT	Flash zone 0 protection word 0xFFFFFFFF: Disable Flash zone 0 protection Others: Enable Flash zone 0 protection
0x11000610	ZONE0_FLASH_ADDR	Flash zone 0 start address
0x11000614	ZONE0_RAM_PROT	RAM zone 0 protection word 0xFFFFFFFF: Disable RAM zone 0 protection Others: Enable RAM zone 0 protection
0x11000618	ZONE0_RAM_ADDR	RAM zone 0 start address
0x1100061C ~ 0x1100063F	Reserved for ZONE0	Reserved Configuration Words for ZONE 0
0x11000640	ZONE1_INFO_LOCK	Lock the Configuration Words of zone 1 (0x11000640 – 0x1100067F) 0xFFFFFFFF: The Configuration Words of zone 1 can be programmed Others: The Configuration Words of zone 1 can not be programmed
0x11000644	ZONE1_ENCRYPT	Encrypt status of code in Flash zone 1 0xDEC0DE: The code in Flash zone 1 has been encrypted

Address	Name	Description
		Others: The code in Flash zone 1 is raw code
0x11000648	ZONE1_DECRYPT	Decrypt status of code in Flash zone 1 0xAA621623: Success to decrypt code in Flash zone 1 0x1E55051: Fail to decrypt code in Flash zone 1 Others: Not valid
0x1100064C	ZONE1_FLASH_PROT	Flash zone 1 protection word 0xFFFFFFFF: Disable Flash zone 1 protection Others: Enable Flash zone 1 protection
0x11000650	ZONE1_FLASH_ADDR	Flash zone 1 start address
0x11000654	ZONE1_RAM_PROT	RAM zone 1 protection word 0xFFFFFFFF: Disable RAM zone 1 protection Others: Enable RAM zone 1 protection
0x11000658	ZONE1_RAM_ADDR	RAM zone 1 start address
0x1100065C ~ 0x1100067F	Reserved for ZONE1	Reserved Configuration Words for ZONE 1
0x11000680	ZONE2_INFO_LOCK	Lock the Configuration Words of zone 2 (0x11000680 – 0x110006BF) 0xFFFFFFFF: The Configuration Words of zone 2 can be programmed Others: The Configuration Words of zone 2 can not be programmed
0x11000684	ZONE2_ENCRYPT	Encrypt status of code in Flash zone 2 0xDEC0DE: The code in Flash zone 2 has been encrypted Others: The code in Flash zone 2 is raw code
0x1100688	ZONE2_DECRYPT	Decrypt status of code in Flash zone 2 0xAA621623: Success to decrypt code in Flash zone 2 0x1E55051: Fail to decrypt code in Flash zone 2 Others: Not valid
0x1100068C	ZONE2_FLASH_PROT	Flash zone 2 protection word 0xFFFFFFFF: Disable Flash zone 2 protection Others: Enable Flash zone 2 protection
0x11000690	ZONE2_FLASH_ADDR	Flash zone 2 start address
0x11000694	ZONE2_RAM_PROT	RAM zone 2 protection word 0xFFFFFFFF: Disable RAM zone 2 protection Others: Enable RAM zone 2 protection
0x11000698	ZONE2_RAM_ADDR	RAM zone 2 start address
0x1100069C ~ 0x110006BF	Reserved for ZONE2	Reserved Configuration Words for ZONE 2

Address	Name	Description
0x110006C0	ZONE3_INFO_LOCK	Lock the Configuration Words of zone 3 (0x110006C0 – 0x110006FF) 0xFFFFFFFF: The Configuration Words of zone 3 can be programmed Others: The Configuration Words of zone 3 can not be programmed
0x110006C4	ZONE3_ENCRYPT	Encrypt status of code in Flash zone 3 0xDEC0DE: The code in Flash zone 3 has been encrypted Others: The code in Flash zone 3 is raw code
0x110006C8	ZONE3_DECRYPT	Decrypt status of code in Flash zone 3 0xAA621623: Success to decrypt code in Flash zone 3 0x1E55051: Fail to decrypt code in Flash zone 3 Others: Not valid
0x110006CC	ZONE3_FLASH_PROT	Flash zone 3 protection word 0xFFFFFFFF: Disable Flash zone 3 protection Others: Enable Flash zone 3 protection
0x110006D0	ZONE3_FLASH_ADDR	Flash zone 3 start address
0x110006D4	ZONE3_RAM_PROT	RAM zone 3 protection word 0xFFFFFFFF: Disable RAM zone 3 protection Others: Enable RAM zone 3 protection
0x110006D8	ZONE3_RAM_ADDR	RAM zone 3 start address
0x110006DC ~ 0x110006FF	Reserved for ZONE3	Reserved Configuration Words for ZONE 3
0x11000700	WDT_ENABLE	Watchdog enable word 0xFFFFFFFF: Disable watchdog when chip start-up Others: Enable watchdog when chip start-up
0x11000704 ~ 0x110007FF	Reserved	Not used

20.8 Registers

20.8.1 Flash register map

Table 20-4: Flash Module Base Address

Peripheral Module	Base Address
FLASH	0x4000 8800

Table 20-5: FLASH Register Map

Register	Offset	Description	Reset Value
FLASHCTL*	0x0	Flash Control Register	0x00000000
FLASHADDR	0x4	Flash Address Register	0x00000000
FLASHDIN	0x8	Flash Data Input Register	0x00000000
FLASHDOUT	0xC	Flash Data Output Register	0x00000000
FLASHWP0*	0x2C	Flash Write Protect Register 0	0x00000000
FLASHWP1*	0x30	Flash Write Protect Register 1	0x00000000
FLASHWP2*	0x34	Flash Write Protect Register 2	0x00000000
FLASHWP3*	0x38	Flash Write Protect Register 3	0x00000000
FLASHREGKEY	0x3C	Flash Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the FLASHREGKEY=0x1ACCE551.

20.8.2 Flash registers

Table 20-6: Flash Control Register (FLASHCTL) Layout

FLASHCTL (Flash Control Register) Offset: 0x0 Default: 0x00000000 Access: FLASH -> FLASHCTL.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	XIPEN	EXECUTION	DATAACCEPT	COMMAND			

Table 20-7: Flash Control Register (FLASHCTL) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	XIPEN	RW	0x0	XIP module enable 0: Disable XIP module 1: Enable XIP module
5	EXECUTION	RW	0x0	Command is in execution
4	DATAACCEPT	RO	0x0	FLASHDIN cannot be changed when this bit is 0
3:0	COMMAND	RW	0x0	Flash command 0000: Power up 0001: Read 0010: Verify read 0 0011: Verify read 1 0100: Prepare Program word 0101: Program word 0110: Prepare Program page words 0111: Program page words 1000: Erase sector 1001: Erase block 1010: Erase chip 1011: 1100: 1101: Recall read 1110: Set configuration 1111: Powe down

Table 20-8: Flash Address Register (FLASHADDR) Layout

FLASHADDR (Flash Address Register) Offset: 0x4 Default: 0x00000000 Access: FLASH -> FLASHADDR.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 20-9: Flash Address Register (FLASHADDR) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	Flash address

Table 20-10: Flash Data Input Register (FLASHDIN) Layout

FLASHDIN (Flash Data Input Register) Offset: 0x8 Default: 0x00000000 Access: FLASH -> FLASHDIN.all							
31	30	29	28	27	26	25	24
FLASHDIN							
23	22	21	20	19	18	17	16
FLASHDIN							
15	14	13	12	11	10	9	8
FLASHDIN							
7	6	5	4	3	2	1	0
FLASHDIN							

Table 20-11: Flash Data Input Register (FLASHDIN) Description

Bits	Field Name	Type	Reset	Description
31:0	FLASHDIN	RW	0x0	Flash input data

Table 20-12: Flash Data Output Register (FLASHDOUT) Layout

FLASHDOUT (Flash Data Output Register) Offset: 0xC Default: 0x00000000 Access: FLASH -> FLASHDOUT.all							
31	30	29	28	27	26	25	24
FLASHDOUT							
23	22	21	20	19	18	17	16
FLASHDOUT							
15	14	13	12	11	10	9	8
FLASHDOUT							
7	6	5	4	3	2	1	0
FLASHDOUT							

Table 20-13: Flash Data Output Register (FLASHDOUT) Description

Bits	Field Name	Type	Reset	Description
31:0	FLASHDOUT	RO	0x0	Flash output data

Table 20-14: Flash Write Protect Register 0 (FLASHWP0) Layout

FLASHWP0 (Flash Write Protect Register 0) Offset: 0x2C Default: 0x00000000 Access: FLASH -> FLASHWP0.all							
31	30	29	28	27	26	25	24
UNIT31	UNIT30	UNIT29	UNIT28	UNIT27	UNIT26	UNIT25	UNIT24
23	22	21	20	19	18	17	16
UNIT23	UNIT22	UNIT21	UNIT20	UNIT19	UNIT18	UNIT17	UNIT16
15	14	13	12	11	10	9	8
UNIT15	UNIT14	UNIT13	UNIT12	UNIT11	UNIT10	UNIT9	UNIT8
7	6	5	4	3	2	1	0
UNIT7	UNIT6	UNIT5	UNIT4	UNIT3	UNIT2	UNIT1	UNIT0

Table 20-15: Flash Write Protect Register 0 (FLASHWP0) Description

Bits	Field Name	Type	Reset	Description
31	UNIT31	RW	0x0	Write protection for flash unit 31 0: Can be programed or erased 1: Cannot be programed or sector erased
30	UNIT30	RW	0x0	Write protection for flash unit 30 0: Can be programed or erased 1: Cannot be programed or sector erased
29	UNIT29	RW	0x0	Write protection for flash unit 29 0: Can be programed or erased 1: Cannot be programed or sector erased
28	UNIT28	RW	0x0	Write protection for flash unit 28 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
27	UNIT27	RW	0x0	Write protection for flash unit 27 0: Can be programed or erased 1: Cannot be programed or sector erased
26	UNIT26	RW	0x0	Write protection for flash unit 26 0: Can be programed or erased 1: Cannot be programed or sector erased
25	UNIT25	RW	0x0	Write protection for flash unit 25 0: Can be programed or erased 1: Cannot be programed or sector erased
24	UNIT24	RW	0x0	Write protection for flash unit 24 0: Can be programed or erased 1: Cannot be programed or sector erased
23	UNIT23	RW	0x0	Write protection for flash unit 23 0: Can be programed or erased 1: Cannot be programed or sector erased
22	UNIT22	RW	0x0	Write protection for flash unit 22 0: Can be programed or erased 1: Cannot be programed or sector erased
21	UNIT21	RW	0x0	Write protection for flash unit 21 0: Can be programed or erased 1: Cannot be programed or sector erased
20	UNIT20	RW	0x0	Write protection for flash unit 20 0: Can be programed or erased 1: Cannot be programed or sector erased
19	UNIT19	RW	0x0	Write protection for flash unit 19 0: Can be programed or erased 1: Cannot be programed or sector erased
18	UNIT18	RW	0x0	Write protection for flash unit 18 0: Can be programed or erased 1: Cannot be programed or sector erased
17	UNIT17	RW	0x0	Write protection for flash unit 17 0: Can be programed or erased 1: Cannot be programed or sector erased
16	UNIT16	RW	0x0	Write protection for flash unit 16 0: Can be programed or erased 1: Cannot be programed or sector erased
15	UNIT15	RW	0x0	Write protection for flash unit 15 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
14	UNIT14	RW	0x0	Write protection for flash unit 14 0: Can be programed or erased 1: Cannot be programed or sector erased
13	UNIT13	RW	0x0	Write protection for flash unit 13 0: Can be programed or erased 1: Cannot be programed or sector erased
12	UNIT12	RW	0x0	Write protection for flash unit 12 0: Can be programed or erased 1: Cannot be programed or sector erased
11	UNIT11	RW	0x0	Write protection for flash unit 11 0: Can be programed or erased 1: Cannot be programed or sector erased
10	UNIT10	RW	0x0	Write protection for flash unit 10 0: Can be programed or erased 1: Cannot be programed or sector erased
9	UNIT9	RW	0x0	Write protection for flash unit 9 0: Can be programed or erased 1: Cannot be programed or sector erased
8	UNIT8	RW	0x0	Write protection for flash unit 8 0: Can be programed or erased 1: Cannot be programed or sector erased
7	UNIT7	RW	0x0	Write protection for flash unit 7 0: Can be programed or erased 1: Cannot be programed or sector erased
6	UNIT6	RW	0x0	Write protection for flash unit 6 0: Can be programed or erased 1: Cannot be programed or sector erased
5	UNIT5	RW	0x0	Write protection for flash unit 5 0: Can be programed or erased 1: Cannot be programed or sector erased
4	UNIT4	RW	0x0	Write protection for flash unit 4 0: Can be programed or erased 1: Cannot be programed or sector erased
3	UNIT3	RW	0x0	Write protection for flash unit 3 0: Can be programed or erased 1: Cannot be programed or sector erased
2	UNIT2	RW	0x0	Write protection for flash unit 2 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
1	UNIT1	RW	0x0	Write protection for flash unit 1 0: Can be programed or erased 1: Cannot be programed or sector erased
0	UNIT0	RW	0x0	Write protection for flash unit 0 0: Can be programed or erased 1: Cannot be programed or sector erased

Table 20-16: Flash Write Protect Register 1 (FLASHWP1) Layout

FLASHWP1 (Flash Write Protect Register 1) Offset: 0x30 Default: 0x00000000
Access: FLASH -> FLASHWP1.all

31	30	29	28	27	26	25	24
UNIT63	UNIT62	UNIT61	UNIT60	UNIT59	UNIT58	UNIT57	UNIT56
23	22	21	20	19	18	17	16
UNIT55	UNIT54	UNIT53	UNIT52	UNIT51	UNIT50	UNIT49	UNIT48
15	14	13	12	11	10	9	8
UNIT47	UNIT46	UNIT45	UNIT44	UNIT43	UNIT42	UNIT41	UNIT40
7	6	5	4	3	2	1	0
UNIT39	UNIT38	UNIT37	UNIT36	UNIT35	UNIT34	UNIT33	UNIT32

Table 20-17: Flash Write Protect Register 1 (FLASHWP1) Description

Bits	Field Name	Type	Reset	Description
31	UNIT63	RW	0x0	Write protection for flash unit 63 0: Can be programed or erased 1: Cannot be programed or sector erased
30	UNIT62	RW	0x0	Write protection for flash unit 62 0: Can be programed or erased 1: Cannot be programed or sector erased
29	UNIT61	RW	0x0	Write protection for flash unit 61 0: Can be programed or erased 1: Cannot be programed or sector erased
28	UNIT60	RW	0x0	Write protection for flash unit 60 0: Can be programed or erased 1: Cannot be programed or sector erased
27	UNIT59	RW	0x0	Write protection for flash unit 59 0: Can be programed or erased 1: Cannot be programed or sector erased
26	UNIT58	RW	0x0	Write protection for flash unit 58 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
25	UNIT57	RW	0x0	Write protection for flash unit 57 0: Can be programed or erased 1: Cannot be programed or sector erased
24	UNIT56	RW	0x0	Write protection for flash unit 56 0: Can be programed or erased 1: Cannot be programed or sector erased
23	UNIT55	RW	0x0	Write protection for flash unit 55 0: Can be programed or erased 1: Cannot be programed or sector erased
22	UNIT54	RW	0x0	Write protection for flash unit 54 0: Can be programed or erased 1: Cannot be programed or sector erased
21	UNIT53	RW	0x0	Write protection for flash unit 53 0: Can be programed or erased 1: Cannot be programed or sector erased
20	UNIT52	RW	0x0	Write protection for flash unit 52 0: Can be programed or erased 1: Cannot be programed or sector erased
19	UNIT51	RW	0x0	Write protection for flash unit 51 0: Can be programed or erased 1: Cannot be programed or sector erased
18	UNIT50	RW	0x0	Write protection for flash unit 50 0: Can be programed or erased 1: Cannot be programed or sector erased
17	UNIT49	RW	0x0	Write protection for flash unit 49 0: Can be programed or erased 1: Cannot be programed or sector erased
16	UNIT48	RW	0x0	Write protection for flash unit 48 0: Can be programed or erased 1: Cannot be programed or sector erased
15	UNIT47	RW	0x0	Write protection for flash unit 47 0: Can be programed or erased 1: Cannot be programed or sector erased
14	UNIT46	RW	0x0	Write protection for flash unit 46 0: Can be programed or erased 1: Cannot be programed or sector erased
13	UNIT45	RW	0x0	Write protection for flash unit 45 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
12	UNIT44	RW	0x0	Write protection for flash unit 44 0: Can be programed or erased 1: Cannot be programed or sector erased
11	UNIT43	RW	0x0	Write protection for flash unit 43 0: Can be programed or erased 1: Cannot be programed or sector erased
10	UNIT42	RW	0x0	Write protection for flash unit 42 0: Can be programed or erased 1: Cannot be programed or sector erased
9	UNIT41	RW	0x0	Write protection for flash unit 41 0: Can be programed or erased 1: Cannot be programed or sector erased
8	UNIT40	RW	0x0	Write protection for flash unit 40 0: Can be programed or erased 1: Cannot be programed or sector erased
7	UNIT39	RW	0x0	Write protection for flash unit 39 0: Can be programed or erased 1: Cannot be programed or sector erased
6	UNIT38	RW	0x0	Write protection for flash unit 38 0: Can be programed or erased 1: Cannot be programed or sector erased
5	UNIT37	RW	0x0	Write protection for flash unit 37 0: Can be programed or erased 1: Cannot be programed or sector erased
4	UNIT36	RW	0x0	Write protection for flash unit 36 0: Can be programed or erased 1: Cannot be programed or sector erased
3	UNIT35	RW	0x0	Write protection for flash unit 35 0: Can be programed or erased 1: Cannot be programed or sector erased
2	UNIT34	RW	0x0	Write protection for flash unit 34 0: Can be programed or erased 1: Cannot be programed or sector erased
1	UNIT33	RW	0x0	Write protection for flash unit 33 0: Can be programed or erased 1: Cannot be programed or sector erased
0	UNIT32	RW	0x0	Write protection for flash unit 32 0: Can be programed or erased 1: Cannot be programed or sector erased

Table 20-18: Flash Write Protect Register 2 (FLASHWP2) Layout

FLASHWP2 (Flash Write Protect Register 2) Offset: 0x34 Default: 0x00000000 Access: FLASH -> FLASHWP2.all							
31	30	29	28	27	26	25	24
UNIT95	UNIT94	UNIT93	UNIT92	UNIT91	UNIT90	UNIT89	UNIT88
23	22	21	20	19	18	17	16
UNIT87	UNIT86	UNIT85	UNIT84	UNIT83	UNIT82	UNIT81	UNIT80
15	14	13	12	11	10	9	8
UNIT79	UNIT78	UNIT77	UNIT76	UNIT75	UNIT74	UNIT73	UNIT72
7	6	5	4	3	2	1	0
UNIT71	UNIT70	UNIT69	UNIT68	UNIT67	UNIT66	UNIT65	UNIT64

Table 20-19: Flash Write Protect Register 2 (FLASHWP2) Description

Bits	Field Name	Type	Reset	Description
31	UNIT95	RW	0x0	Write protection for flash unit 95 0: Can be programed or erased 1: Cannot be programed or sector erased
30	UNIT94	RW	0x0	Write protection for flash unit 94 0: Can be programed or erased 1: Cannot be programed or sector erased
29	UNIT93	RW	0x0	Write protection for flash unit 93 0: Can be programed or erased 1: Cannot be programed or sector erased
28	UNIT92	RW	0x0	Write protection for flash unit 92 0: Can be programed or erased 1: Cannot be programed or sector erased
27	UNIT91	RW	0x0	Write protection for flash unit 91 0: Can be programed or erased 1: Cannot be programed or sector erased
26	UNIT90	RW	0x0	Write protection for flash unit 90 0: Can be programed or erased 1: Cannot be programed or sector erased
25	UNIT89	RW	0x0	Write protection for flash unit 89 0: Can be programed or erased 1: Cannot be programed or sector erased
24	UNIT88	RW	0x0	Write protection for flash unit 88 0: Can be programed or erased 1: Cannot be programed or sector erased
23	UNIT87	RW	0x0	Write protection for flash unit 87 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
22	UNIT86	RW	0x0	Write protection for flash unit 86 0: Can be programed or erased 1: Cannot be programed or sector erased
21	UNIT85	RW	0x0	Write protection for flash unit 85 0: Can be programed or erased 1: Cannot be programed or sector erased
20	UNIT84	RW	0x0	Write protection for flash unit 84 0: Can be programed or erased 1: Cannot be programed or sector erased
19	UNIT83	RW	0x0	Write protection for flash unit 83 0: Can be programed or erased 1: Cannot be programed or sector erased
18	UNIT82	RW	0x0	Write protection for flash unit 82 0: Can be programed or erased 1: Cannot be programed or sector erased
17	UNIT81	RW	0x0	Write protection for flash unit 81 0: Can be programed or erased 1: Cannot be programed or sector erased
16	UNIT80	RW	0x0	Write protection for flash unit 80 0: Can be programed or erased 1: Cannot be programed or sector erased
15	UNIT79	RW	0x0	Write protection for flash unit 79 0: Can be programed or erased 1: Cannot be programed or sector erased
14	UNIT78	RW	0x0	Write protection for flash unit 78 0: Can be programed or erased 1: Cannot be programed or sector erased
13	UNIT77	RW	0x0	Write protection for flash unit 77 0: Can be programed or erased 1: Cannot be programed or sector erased
12	UNIT76	RW	0x0	Write protection for flash unit 76 0: Can be programed or erased 1: Cannot be programed or sector erased
11	UNIT75	RW	0x0	Write protection for flash unit 75 0: Can be programed or erased 1: Cannot be programed or sector erased
10	UNIT74	RW	0x0	Write protection for flash unit 74 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
9	UNIT73	RW	0x0	Write protection for flash unit 73 0: Can be programed or erased 1: Cannot be programed or sector erased
8	UNIT72	RW	0x0	Write protection for flash unit 72 0: Can be programed or erased 1: Cannot be programed or sector erased
7	UNIT71	RW	0x0	Write protection for flash unit 71 0: Can be programed or erased 1: Cannot be programed or sector erased
6	UNIT70	RW	0x0	Write protection for flash unit 70 0: Can be programed or erased 1: Cannot be programed or sector erased
5	UNIT69	RW	0x0	Write protection for flash unit 69 0: Can be programed or erased 1: Cannot be programed or sector erased
4	UNIT68	RW	0x0	Write protection for flash unit 68 0: Can be programed or erased 1: Cannot be programed or sector erased
3	UNIT67	RW	0x0	Write protection for flash unit 67 0: Can be programed or erased 1: Cannot be programed or sector erased
2	UNIT66	RW	0x0	Write protection for flash unit 66 0: Can be programed or erased 1: Cannot be programed or sector erased
1	UNIT65	RW	0x0	Write protection for flash unit 65 0: Can be programed or erased 1: Cannot be programed or sector erased
0	UNIT64	RW	0x0	Write protection for flash unit 64 0: Can be programed or erased 1: Cannot be programed or sector erased

Table 20-20: Flash Write Protect Register 3 (FLASHWP3) Layout

FLASHWP3 (Flash Write Protect Register 3) Offset: 0x38 Default: 0x00000000 Access: FLASH -> FLASHWP3.all							
31 UNIT127	30 UNIT126	29 UNIT125	28 UNIT124	27 UNIT123	26 UNIT122	25 UNIT121	24 UNIT120
23 UNIT119	22 UNIT118	21 UNIT117	20 UNIT116	19 UNIT115	18 UNIT114	17 UNIT113	16 UNIT112
15 UNIT111	14 UNIT110	13 UNIT109	12 UNIT108	11 UNIT107	10 UNIT106	9 UNIT105	8 UNIT104
7 UNIT103	6 UNIT102	5 UNIT101	4 UNIT100	3 UNIT99	2 UNIT98	1 UNIT97	0 UNIT96

Table 20-21: Flash Write Protect Register 3 (FLASHWP3) Description

Bits	Field Name	Type	Reset	Description
31	UNIT127	RW	0x0	Write protection for flash unit 127 0: Can be programed or erased 1: Cannot be programed or sector erased
30	UNIT126	RW	0x0	Write protection for flash unit 126 0: Can be programed or erased 1: Cannot be programed or sector erased
29	UNIT125	RW	0x0	Write protection for flash unit 125 0: Can be programed or erased 1: Cannot be programed or sector erased
28	UNIT124	RW	0x0	Write protection for flash unit 124 0: Can be programed or erased 1: Cannot be programed or sector erased
27	UNIT123	RW	0x0	Write protection for flash unit 123 0: Can be programed or erased 1: Cannot be programed or sector erased
26	UNIT122	RW	0x0	Write protection for flash unit 122 0: Can be programed or erased 1: Cannot be programed or sector erased
25	UNIT121	RW	0x0	Write protection for flash unit 121 0: Can be programed or erased 1: Cannot be programed or sector erased
24	UNIT120	RW	0x0	Write protection for flash unit 120 0: Can be programed or erased 1: Cannot be programed or sector erased
23	UNIT119	RW	0x0	Write protection for flash unit 119 0: Can be programed or erased 1: Cannot be programed or sector erased
22	UNIT118	RW	0x0	Write protection for flash unit 118 0: Can be programed or erased 1: Cannot be programed or sector erased
21	UNIT117	RW	0x0	Write protection for flash unit 117 0: Can be programed or erased 1: Cannot be programed or sector erased
20	UNIT116	RW	0x0	Write protection for flash unit 116 0: Can be programed or erased 1: Cannot be programed or sector erased
19	UNIT115	RW	0x0	Write protection for flash unit 115 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
18	UNIT114	RW	0x0	Write protection for flash unit 114 0: Can be programed or erased 1: Cannot be programed or sector erased
17	UNIT113	RW	0x0	Write protection for flash unit 113 0: Can be programed or erased 1: Cannot be programed or sector erased
16	UNIT112	RW	0x0	Write protection for flash unit 112 0: Can be programed or erased 1: Cannot be programed or sector erased
15	UNIT111	RW	0x0	Write protection for flash unit 111 0: Can be programed or erased 1: Cannot be programed or sector erased
14	UNIT110	RW	0x0	Write protection for flash unit 110 0: Can be programed or erased 1: Cannot be programed or sector erased
13	UNIT109	RW	0x0	Write protection for flash unit 109 0: Can be programed or erased 1: Cannot be programed or sector erased
12	UNIT108	RW	0x0	Write protection for flash unit 108 0: Can be programed or erased 1: Cannot be programed or sector erased
11	UNIT107	RW	0x0	Write protection for flash unit 107 0: Can be programed or erased 1: Cannot be programed or sector erased
10	UNIT106	RW	0x0	Write protection for flash unit 106 0: Can be programed or erased 1: Cannot be programed or sector erased
9	UNIT105	RW	0x0	Write protection for flash unit 105 0: Can be programed or erased 1: Cannot be programed or sector erased
8	UNIT104	RW	0x0	Write protection for flash unit 104 0: Can be programed or erased 1: Cannot be programed or sector erased
7	UNIT103	RW	0x0	Write protection for flash unit 103 0: Can be programed or erased 1: Cannot be programed or sector erased
6	UNIT102	RW	0x0	Write protection for flash unit 102 0: Can be programed or erased 1: Cannot be programed or sector erased

Bits	Field Name	Type	Reset	Description
5	UNIT101	RW	0x0	Write protection for flash unit 101 0: Can be programed or erased 1: Cannot be programed or sector erased
4	UNIT100	RW	0x0	Write protection for flash unit 100 0: Can be programed or erased 1: Cannot be programed or sector erased
3	UNIT99	RW	0x0	Write protection for flash unit 99 0: Can be programed or erased 1: Cannot be programed or sector erased
2	UNIT98	RW	0x0	Write protection for flash unit 98 0: Can be programed or erased 1: Cannot be programed or sector erased
1	UNIT97	RW	0x0	Write protection for flash unit 97 0: Can be programed or erased 1: Cannot be programed or sector erased
0	UNIT96	RW	0x0	Write protection for flash unit 96 0: Can be programed or erased 1: Cannot be programed or sector erased

Table 20-22: Flash Register Write-Allow Key Register (FLASHREGKEY) Layout

FLASHREGKEY (Flash Register Write-Allow Key Register) Offset: 0x3C Default: 0x1ACCE551 Access: FLASH -> FLASHREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 20-23: Flash Register Write-Allow Key Register (FLASHREGKEY) Description

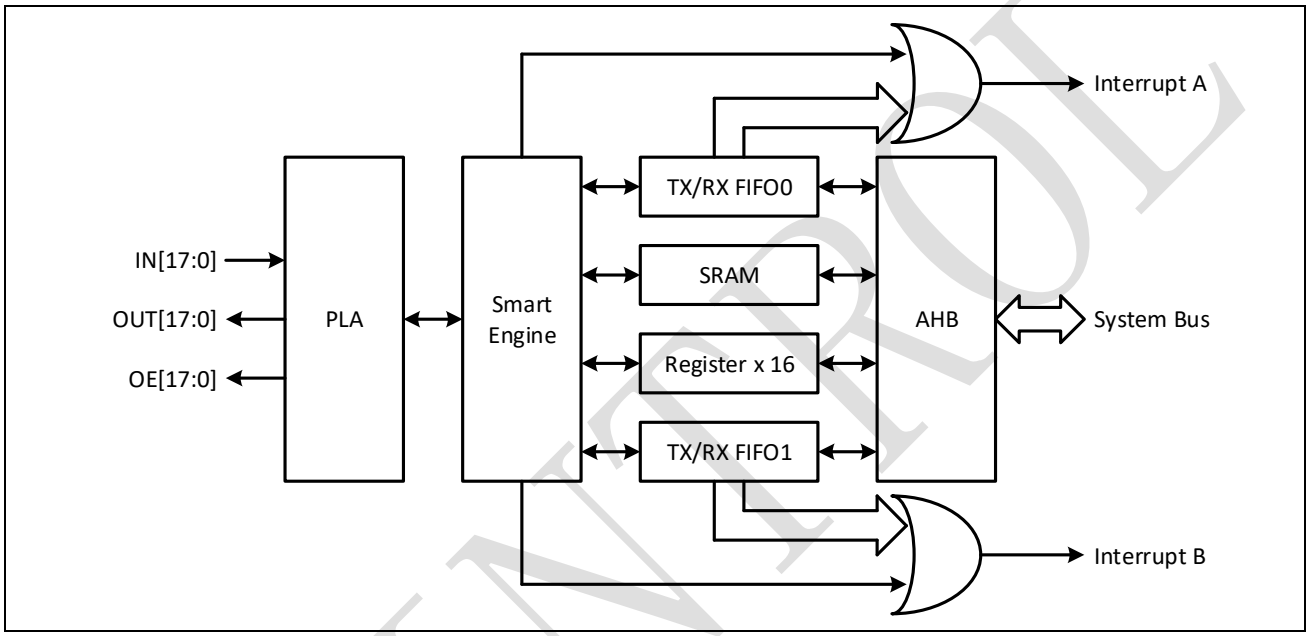
Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected flash registers

21 SIO

21.1 Overview

SIO is a Spintrol patented technology. It has programmable capability which can convert the SIO block into pre-defined communication module. One SIO is implemented in SPD1148 and supports up to 18 IO channels as listed in IO_Channel_Defination.

Figure 21-1: SIO architecture diagram



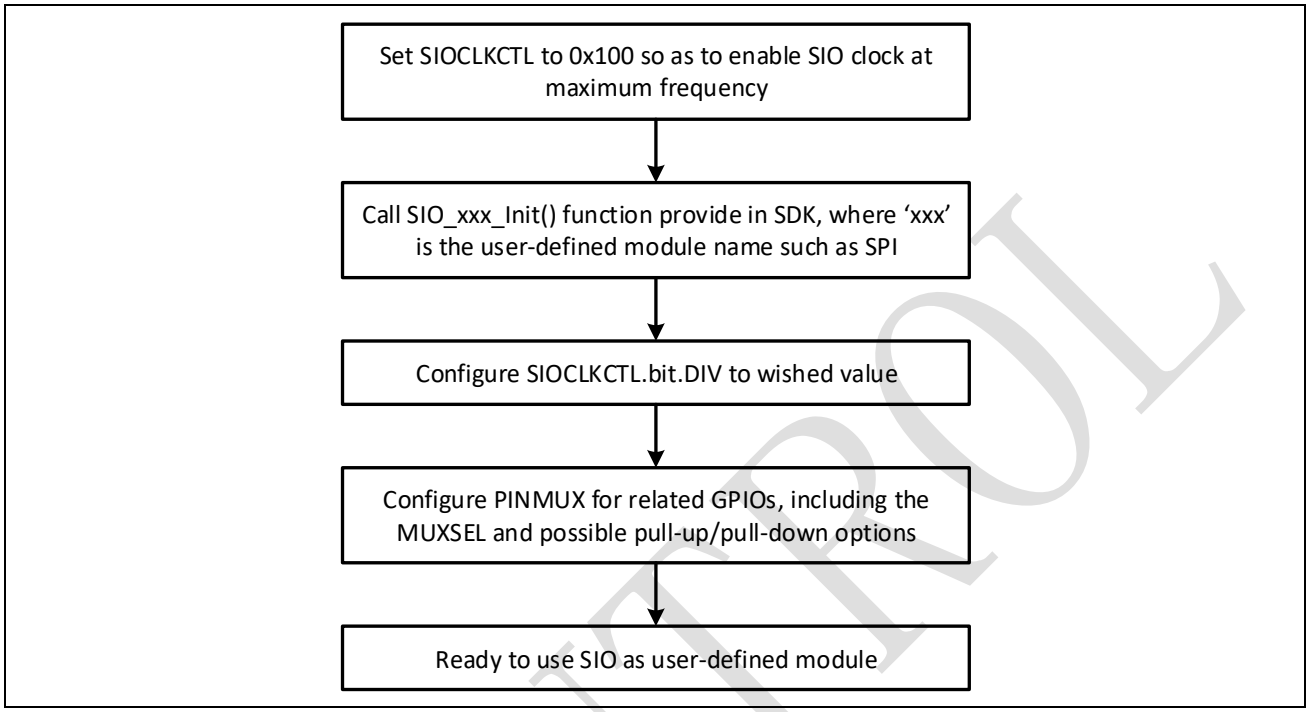
The architecture of the SIO is shown in Figure 21-1. It basically consists of two TX FIFOs, two RX FIFOs, 16 registers and an SRAM on AHB bus, a programmable logic array, and a smart-engine. The SIO provides up to two interrupts to the main CPU. The PLA provides fast parallel logic processing, while the smart engine is capable to deal with state control and transfer for complicated protocols. By ultilizing the befenefits of the two, the SIO in SPD1148 enables following usage cases:

- Re-define the pins for a given interface on a given SPD1148 device.
  i.e. An UART interface implemented on SIO with GPIO20/GPIO21 as TXD/RXD can be changed to GPIO28/GPIO29 as TXD/RXD via firmware update.
- Re-define the whole interface function on a given SPD1148 device.
  i.e. The SIO is originally defined as I2C. It can be changed via firmware update to UART per user's system requirement.
- Add-on feature for a standard interface on a given SPD1148 device.
- i.e. Embed user defined data format and communication 716rotocol based on standard UART into a single new interface via firmware update.

21.2 SIO Initialization

Figure 21-2 shows the flow to initialize the SIO as a user-defined module.

Figure 21-2: SIO initialization flow



Example 21.2.1

```
void  SIO_Example21_2_1(void)
{
  /* Enable SIO clock at maximum frequency to minimize initialization time */
  CLOCK->SIOCLKCTL.all = 0x100;

  /* Initialize the SIO PLA and smart engine to make the SIO as a UART */
  SIO_UART_Init();

  /* Configure SIO clock frequency as SYSCLK1/4 */
  CLOCK->SIOCLKCTL.bit.DIV = 0x3;

  /* Configure GPIO36 and GPIO37 to SIO channel with pull-up enabled for UART */
  PINMUX->GPIO36.bit.MUXSEL  = GPIO36_BIT_MUXSEL_SIO0_14;
  PINMUX->GPIO36.bit.PU      |= 1 << GPIO36_BIT_MUXSEL_SIO0_14;
  PINMUX->GPIO37.bit.MUXSEL  = GPIO37_BIT_MUXSEL_SIO0_15;
  PINMUX->GPIO37.bit.PU      |= 1 << GPIO37_BIT_MUXSEL_SIO0_15;

  /* Ready to use SIO as UART */
}
```

21.3 Registers

Please refer to application note for corresponding SIO based user-defined modules.

22 System control

22.1 Overview

The system control module provides:

- Unique device ID register (64 bits)
- Memory error interrupt control registers
- Memory ECC enable register
- Memory lock status register
- Reset event control registers
- System information register

22.2 Unique device ID register (64 bits)

The unique device identifier is ideally suited:

- for use as serial numbers
- for use as security keys in order to increase the security of code in Flash memory while using and combining this unique ID with software cryptographic primitives and protocols before programming the internal Flash memory
- to activate secure boot processes, etc.

The 64-bit unique device identifier provides a reference number which is unique for any device and in any context. These bits can never be altered by the user.

22.3 Memory error interrupt control registers

These registers are used to enable, indicate and clear memory error interrupts.

22.4 Memory ECC enable register

The register enables ROM and Flash memory ECC feature.

22.5 Memory lock status register

The register contains Flash memory and RAM multi-zone protection status, Configuration Words lock status.

22.6 Reset event control registers

These registers are used to enable, indicate and clear reset events.

22.7 System information register

The register provides information about Flash size, RAM size, ADC effective bits and clock frequency option.

22.8 Registers

22.8.1 System register map

Table 22-1: SYSTEM Module Base Address

Peripheral Module	Base Address
SYSTEM	0x4000 0000

Table 22-2: SYSTEM Register Map

Register	Offset	Description	Reset Value
CID0	0x0	Chip ID Register 0	0x00000000
CID1	0x4	Chip ID Register 1	0x00000000
UID0	0x8	Unique ID Register 0	0x00000000
UID1	0xC	Unique ID Register 1	0x00000000
RND0	0x10	Random Number Register 0	0x00000000
RND1	0x14	Random Number Register 1	0x00000000
REV0	0x18	Revision Information Register 0	0x676F1111
REV1	0x1C	Revision Information Register 1	0x201807A0
MEMIF	0x20	Memory Error Interrupt Flag Register	0x00000000
MEMIC	0x24	Memory Error Interrupt Clear Register	0x00000000
MEMIE*	0x28	Memory Error Interrupt Enable Register	0x00000000
MEMECCEN*	0x2C	Memory ECC Enable Register	0x00000003
MEMLOCKSTS	0x30	Memory Lock Status Register	0x00000000
RSTEVTSTS	0x34	Reset Event Status Register	0x00000000
RSTEVTCLR	0x38	Reset Event Status Clear Register	0x00000000
RSTEVTEN*	0x3C	Reset Event Enable Register	0x00000005
SYSINFO	0x40	System Information Register	0x000001FF
SYSREGKEY	0x44	System Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the SYSREGKEY=0x1ACCE551.

22.8.2 System registers

Table 22-3: Chip ID Register 0 (CID0) Layout

CID0 (Chip ID Register 0) Offset: 0x0 Default: 0x00000000 Access: SYSTEM -> CID0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 22-4: Chip ID Register 0 (CID0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	32 LSBs of the 64-bit chip ID

Table 22-5: Chip ID Register 1 (CID1) Layout

CID1 (Chip ID Register 1) Offset: 0x4 Default: 0x00000000 Access: SYSTEM -> CID1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 22-6: Chip ID Register 1 (CID1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	32 MSBs of the 64-bit chip ID

Table 22-7: Unique ID Register 0 (UID0) Layout

UID0 (Unique ID Register 0) Offset: 0x8 Default: 0x00000000 Access: SYSTEM -> UID0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 22-8: Unique ID Register 0 (UID0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	32 LSBs of the 64-bit unique ID

Table 22-9: Unique ID Register 1 (UID1) Layout

UID1 (Unique ID Register 1) Offset: 0xC Default: 0x00000000 Access: SYSTEM -> UID1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 22-10: Unique ID Register 1 (UID1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	32 MSBs of the 64-bit unique ID

Table 22-11: Random Number Register 0 (RND0) Layout

RND0 (Random Number Register 0) Offset: 0x10 Default: 0x00000000 Access: SYSTEM -> RND0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 22-12: Random Number Register 0 (RND0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	32 LSBs of the 64-bit random number

Table 22-13: Random Number Register 1 (RND1) Layout

RND1 (Random Number Register 1) Offset: 0x14 Default: 0x00000000 Access: SYSTEM -> RND1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 22-14: Random Number Register 1 (RND1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RW	0x0	32 MSBs of the 64-bit random number

Table 22-15: Revision Information Register 0 (REV0) Layout

REV0 (Revision Information Register 0) Offset: 0x18 Default: 0x676F1111 Access: SYSTEM -> REV0.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 22-16: Revision Information Register 0 (REV0) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x676F1111	32 LSBs of the 64-bit revision number

Table 22-17: Revision Information Register 1 (REV1) Layout

REV1 (Revision Information Register 1) Offset: 0x1C Default: 0x201807A0 Access: SYSTEM -> REV1.all							
31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

Table 22-18: Revision Information Register 1 (REV1) Description

Bits	Field Name	Type	Reset	Description
31:0	VAL	RO	0x201807A0	32 MSBs of the 64-bit revision number

Table 22-19: Memory Error Interrupt Flag Register (MEMIF) Layout

MEMIF (Memory Error Interrupt Flag Register) Offset: 0x20 Default: 0x00000000 Access: SYSTEM -> MEMIF.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
MEMINT	SIOERR	DRAMERR	IRAMERR	FLASH2ERR	FLASH1ERR	ROM2ERR	ROM1ERR

Table 22-20: Memory Error Interrupt Flag Register (MEMIF) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	MEMINT	RO	0x0	Memory error interrupt flag 0: Memory error interrupt not occur 1: Memory error interrupt occurred
6	SIOERR	RO	0x0	Latched SIO memory parity error flag 0: SIOERR interrupt not occur 1: SIOERR interrupt occurred
5	DRAMERR	RO	0x0	Latched data RAM parity error flag 0: DRAMERR interrupt not occur 1: DRAMERR interrupt occurred
4	IRAMERR	RO	0x0	Latched instruction RAM parity error flag 0: IRAMERR interrupt not occur 1: IRAMERR interrupt occurred
3	FLASH2ERR	RO	0x0	Latched flash ECC 2-bit error flag 0: FLASH2ERR interrupt not occur 1: FLASH2ERR interrupt occurred
2	FLASH1ERR	RO	0x0	Latched flash ECC 1-bit error flag 0: FLASH1ERR interrupt not occur 1: FLASH1ERR interrupt occurred
1	ROM2ERR	RO	0x0	Latched ROM ECC 2-bit error flag 0: ROM2ERR interrupt not occur 1: ROM2ERR interrupt occurred
0	ROM1ERR	RO	0x0	Latched ROM ECC 1-bit error flag 0: ROM1ERR interrupt not occur 1: ROM1ERR interrupt occurred

Table 22-21: Memory Error Interrupt Clear Register (MEMIC) Layout

MEMIC (Memory Error Interrupt Clear Register) Offset: 0x24 Default: 0x00000000 Access: SYSTEM -> MEMIC.all							
31	30	29	28	27	26	25	24
RESERVED_31_8							
23	22	21	20	19	18	17	16
RESERVED_31_8							
15	14	13	12	11	10	9	8
RESERVED_31_8							
7	6	5	4	3	2	1	0
MEMINT	SIOERR	DRAMERR	IRAMERR	FLASH2ERR	FLASH1ERR	ROM2ERR	ROM1ERR

Table 22-22: Memory Error Interrupt Clear Register (MEMIC) Description

Bits	Field Name	Type	Reset	Description
31:8	RESERVED_31_8	RO	0x0	Reserved.
7	MEMINT	W1C	0x0	Memory error interrupt clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
6	SIOERR	W1C	0x0	SIO access parity error occurred flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the flag. This bit is self-cleared to 0.
5	DRAMERR	W1C	0x0	Data RAM parity error occurred flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the flag. This bit is self-cleared to 0.
4	IRAMERR	W1C	0x0	Instruction RAM parity error occurred flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the flag. This bit is self-cleared to 0.
3	FLASH2ERR	W1C	0x0	Flash access two-bit error occurred flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the flag. This bit is self-cleared to 0.
2	FLASH1ERR	W1C	0x0	Flash access one-bit error occurred flag clear 0: Write a 0 has no effect. Always readback 0.

Bits	Field Name	Type	Reset	Description
				1: Write a 1 clear the flag. This bit is self-cleared to 0.
1	ROM2ERR	W1C	0x0	ROM access two-bit error occurred flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the flag. This bit is self-cleared to 0.
0	ROM1ERR	W1C	0x0	ROM access one-bit error occurred flag clear 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the flag. This bit is self-cleared to 0.

Table 22-23: Memory Error Interrupt Enable Register (MEMIE) Layout

MEMIE (Memory Error Interrupt Enable Register) Offset: 0x28 Default: 0x00000000 Access: SYSTEM -> MEMIE.all							
31	30	29	28	27	26	25	24
RESERVED_31_7							
23	22	21	20	19	18	17	16
RESERVED_31_7							
15	14	13	12	11	10	9	8
RESERVED_31_7							
7	6	5	4	3	2	1	0
ESERVED_31_	SIOERR	DRAMERR	IRAMERR	FLASH2ERR	FLASH1ERR	ROM2ERR	ROM1ERR

Table 22-24: Memory Error Interrupt Enable Register (MEMIE) Description

Bits	Field Name	Type	Reset	Description
31:7	RESERVED_31_7	RO	0x0	Reserved.
6	SIOERR	RW	0x0	SIO access parity error interrupt enable 0: Disable 1: Enable
5	DRAMERR	RW	0x0	Data RAM parity error interrupt enable 0: Disable 1: Enable
4	IRAMERR	RW	0x0	Instruction RAM parity error interrupt enable 0: Disable 1: Enable
3	FLASH2ERR	RW	0x0	Flash access two-bit error interrupt enable 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
2	FLASH1ERR	RW	0x0	Flash access one-bit error interrupt enable 0: Disable 1: Enable
1	ROM2ERR	RW	0x0	ROM access two-bit error interrupt enable 0: Disable 1: Enable
0	ROM1ERR	RW	0x0	ROM access one-bit error interrupt enable 0: Disable 1: Enable

Table 22-25: Memory ECC Enable Register (MEMECCEN) Layout

MEMECCEN (Memory ECC Enable Register) Offset: 0x2C Default: 0x00000003 Access: SYSTEM -> MEMECCEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						FLASHECC	ROMECC

Table 22-26: Memory ECC Enable Register (MEMECCEN) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	FLASHECC	RW	0x1	FLASH ECC enable 0: Disable 1: Enable
0	ROMECC	RW	0x1	ROM ECC enable 0: Disable 1: Enable

Table 22-27: Memory Lock Status Register (MEMLOCKSTS) Layout

MEMLOCKSTS (Memory Lock Status Register) Offset: 0x30 Default: 0x00000000 Access: SYSTEM -> MEMLOCKSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_13							
23	22	21	20	19	18	17	16
RESERVED_31_13							
15	14	13	12	11	10	9	8
RESERVED_31_13			RAM3	RAM2	RAM1	RAM0	FLASHOTP
7	6	5	4	3	2	1	0
USER3INFO	USER2INFO	USER1INFO	USER0INFO	FLASH3	FLASH2	FLASH1	FLASH0

Table 22-28: Memory Lock Status Register (MEMLOCKSTS) Description

Bits	Field Name	Type	Reset	Description
31:13	RESERVED_31_13	RO	0x0	Reserved.
12	RAM3	RO	0x0	RAM region 3 0: Can read/write content and execute 1: Can not read/write content, just execute
11	RAM2	RO	0x0	RAM region 2 0: Can read/write content and execute 1: Can not read/write content, just execute
10	RAM1	RO	0x0	RAM region 1 0: Can read/write content and execute 1: Can not read/write content, just execute
9	RAM0	RO	0x0	RAM region 0 0: Can read/write content and execute 1: Can not read/write content, just execute
8	FLASHOTP	RO	0x0	Lock Flash NVR3 0: Can read, write and erase FLASHNVR3 1: FLASHNVR3 is read-only
7	USER3INFO	RO	0x0	Lock Flash NVR4 0: Can read, write and erase 1: Read and erase only, can not write
6	USER2INFO	RO	0x0	Lock Flash NVR4 0: Can read, write and erase 1: Read and erase only, can not write
5	USER1INFO	RO	0x0	Lock Flash NVR4 0: Can read, write and erase 1: Read and erase only, can not write
4	USER0INFO	RO	0x0	Lock Flash NVR4 0: Can read, write and erase 1: Read and erase only, can not write

Bits	Field Name	Type	Reset	Description
3	FLASH3	RO	0x0	FLASH region 3 0: Can read/write content and execute 1: Can not read/write content, just execute
2	FLASH2	RO	0x0	FLASH region 2 0: Can read/write content and execute 1: Can not read/write content, just execute
1	FLASH1	RO	0x0	FLASH region 1 0: Can read/write content and execute 1: Can not read/write content, just execute
0	FLASH0	RO	0x0	FLASH region 0 0: Can read/write content and execute 1: Can not read/write content, just execute

Table 22-29: Reset Event Status Register (RSTEVTSTS) Layout

RSTEVTSTS (Reset Event Status Register) Offset: 0x34 Default: 0x00000000 Access: SYSTEM -> RSTEVTSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_15							
23	22	21	20	19	18	17	16
RESERVED_31_15							
15 ESERVED_31_1	14 SYSRESET	13 WDT1RST	12 WDT0RST	11 CLKDETERR	10 PLLUNLOCK	9 VDD33H	8 VDD33L
7 VDD12H	6 VDD12L1	5 VDD12L0	4 SIOERR	3 DRAMERR	2 IRAMERR	1 FLASHERR	0 ROMERR

Table 22-30: Reset Event Status Register (RSTEVTSTS) Description

Bits	Field Name	Type	Reset	Description
31:15	RESERVED_31_15	RO	0x0	Reserved.
14	SYSRESET	RO	0x0	Latched reset triggered by system reset event 0: Specified reset not happen 1: Specified reset happened
13	WDT1RST	RO	0x0	Latched reset triggered by WDT1 reset event 0: Specified reset not happen 1: Specified reset happened
12	WDT0RST	RO	0x0	Latched reset triggered by WDT0 reset event 0: Specified reset not happen 1: Specified reset happened
11	CLKDETERR	RO	0x0	Latched reset triggered by CLKDETERR event 0: Specified reset not happen 1: Specified reset happened

Bits	Field Name	Type	Reset	Description
10	PLLUNLOCK	RO	0x0	Latched reset triggered by PLL unlock event 0: Specified reset not happen 1: Specified reset happened
9	VDD33H	RO	0x0	Latched reset triggered by VDD33H event 0: Specified reset not happen 1: Specified reset happened
8	VDD33L	RO	0x0	Latched reset triggered by VDD33L event 0: Specified reset not happen 1: Specified reset happened
7	VDD12H	RO	0x0	Latched reset triggered by VDD12H event 0: Specified reset not happen 1: Specified reset happened
6	VDD12L1	RO	0x0	Latched reset triggered by VDD12L1 event 0: Specified reset not happen 1: Specified reset happened
5	VDD12L0	RO	0x0	Latched reset triggered by VDD12L0 event 0: Specified reset not happen 1: Specified reset happened
4	SIOERR	RO	0x0	Latched reset triggered by parity error during SIO access 0: Specified reset not happen 1: Specified reset happened
3	DRAMERR	RO	0x0	Latched reset triggered by parity error during data RAM access 0: Specified reset not happen 1: Specified reset happened
2	IRAMERR	RO	0x0	Latched reset triggered by parity error during instruction RAM access 0: Specified reset not happen 1: Specified reset happened
1	FLASHERR	RO	0x0	Latched reset triggered by 2-bit error during flash access 0: Specified reset not happen 1: Specified reset happened
0	ROMERR	RO	0x0	Latched reset triggered by 2-bit error during ROM access 0: Specified reset not happen 1: Specified reset happened

Table 22-31: Reset Event Status Clear Register (RSTEVTCLR) Layout

RSTEVTCLR (Reset Event Status Clear Register) Offset: 0x38 Default: 0x00000000 Access: SYSTEM -> RSTEVTCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_15							
23	22	21	20	19	18	17	16
RESERVED_31_15							
15 ESERVED_31_	14 SYSRESET	13 WDT1RST	12 WDT0RST	11 CLKDETERR	10 PLLUNLOCK	9 VDD33H	8 VDD33L
7 VDD12H	6 VDD12L1	5 VDD12L0	4 SIOERR	3 DRAMERR	2 IRAMERR	1 FLASHERR	0 ROMERR

Table 22-32: Reset Event Status Clear Register (RSTEVTCLR) Description

Bits	Field Name	Type	Reset	Description
31:15	RESERVED_31_15	RO	0x0	Reserved.
14	SYSRESET	W1C	0x0	Clear reset status happened upon system reset event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
13	WDT1RST	W1C	0x0	Clear reset status happened upon WDT1 reset event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
12	WDT0RST	W1C	0x0	Clear reset status happened upon WDT0 reset event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
11	CLKDETERR	W1C	0x0	Clear reset status happened upon CLKDETERR event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
10	PLLUNLOCK	W1C	0x0	Clear reset status happened upon PLL unlock event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
9	VDD33H	W1C	0x0	Clear reset status happened upon VDD33H event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
8	VDD33L	W1C	0x0	Clear reset status happened upon VDD33L event 0: Write a 0 has no effect. Always readback 0.

Bits	Field Name	Type	Reset	Description
				1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
7	VDD12H	W1C	0x0	Clear reset status happened upon VDD12H event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
6	VDD12L1	W1C	0x0	Clear reset status happened upon VDD12L1 event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
5	VDD12L0	W1C	0x0	Clear reset status happened upon VDD12L0 event 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
4	SIOERR	W1C	0x0	Clear reset status happened upon parity error during SIO access 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
3	DRAMERR	W1C	0x0	Clear reset status happened upon parity error during data RAM access 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
2	IRAMERR	W1C	0x0	Clear reset status happened upon parity error during instruction RAM access 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.
1	FLASHERR	W1C	0x0	Clear reset status happened upon 2-bit error during flash access 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.

Bits	Field Name	Type	Reset	Description
0	ROMERR	W1C	0x0	Clear reset status happened upon 2-bit error during ROM access 0: Write a 0 has no effect. Always readback 0. 1: Write a 1 clear the interrupt and the flag This bit is self-cleared to 0.

Table 22-33: Reset Event Enable Register (RSTEVTEN) Layout

RSTEVTEN (Reset Event Enable Register) Offset: 0x3C Default: 0x00000005 Access: SYSTEM -> RSTEVTEN.all							
31	30	29	28	27	26	25	24
RESERVED_31_12							
23	22	21	20	19	18	17	16
RESERVED_31_12							
15	14	13	12	11	10	9	8
RESERVED_31_12				CLKDETERR	PLLUNLOCK	VDD33H	VDD33L
7	6	5	4	3	2	1	0
VDD12H	VDD12L1	VDD12L0	SIOERR	DRAMERR	IRAMERR	FLASHERR	ROMERR

Table 22-34: Reset Event Enable Register (RSTEVTEN) Description

Bits	Field Name	Type	Reset	Description
31:12	RESERVED_31_12	RO	0x0	Reserved.
11	CLKDETERR	RW	0x0	Enable reset on CLKDETERR event 0: Disable 1: Enable
10	PLLUNLOCK	RW	0x0	Enable reset on PLL unlock event 0: Disable 1: Enable
9	VDD33H	RW	0x0	Enable reset on VDD33H event 0: Disable 1: Enable
8	VDD33L	RW	0x0	Enable reset on VDD33L event 0: Disable 1: Enable
7	VDD12H	RW	0x0	Enable reset on VDD12H event 0: Disable 1: Enable
6	VDD12L1	RW	0x0	Enable reset on VDD12L1 event 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
5	VDD12L0	RW	0x0	Enable reset on VDD12L0 event 0: Disable 1: Enable
4	SIOERR	RW	0x0	Enable reset on parity error during SIO access 0: Disable 1: Enable
3	DRAMERR	RW	0x0	Enable reset on parity error during data RAM access 0: Disable 1: Enable
2	IRAMERR	RW	0x1	Enable reset on parity error during instruction RAM access 0: Disable 1: Enable
1	FLASHERR	RW	0x0	Enable reset on 2-bit error during flash access 0: Disable 1: Enable
0	ROMERR	RW	0x1	Enable reset on 2-bit error during ROM access 0: Disable 1: Enable

Table 22-35: System Information Register (SYSINFO) Layout

SYSINFO (System Information Register) Offset: 0x40 Default: 0x000001FF Access: SYSTEM -> SYSINFO.all							
31	30	29	28	27	26	25	24
RESERVED_31_9							
23	22	21	20	19	18	17	16
RESERVED_31_9							
15	14	13	12	11	10	9	8
RESERVED_31_9							ADCBIT
7 ADCBIT	6	5	4	3	2	1	0
	CLKOPT			RAMSIZE		FLASHSIZE	

Table 22-36: System Information Register (SYSINFO) Description

Bits	Field Name	Type	Reset	Description
31:9	RESERVED_31_9	RO	0x0	Reserved.
8:7	ADCBIT	RO	0x3	ADC bit width 00: 10 bit 01: 11 bit 10: 12 bit 11: 14 bit
6:4	CLKOPT	RO	0x7	Clock option 000: Minimum PLL output dividing ratio is 12 and XO is unavailable 001: Minimum PLL output dividing ratio is 6 and XO is unavailable 010: Minimum PLL output dividing ratio is 4 and XO is unavailable 011: Minimum PLL output dividing ratio is 12 and XO is available 100: Minimum PLL output dividing ratio is 6 and XO is available 101: Minimum PLL output dividing ratio is 4 and XO available 110: Minimum PLL output dividing ratio is 3 and XO is available 111: No limitation
3:2	RAMSIZE	RO	0x3	RAM size 00: 16kB start at 0x20000000 01: 32kB start at 0x1FFFC000 10: 48kB start at 0x1FFF8000 11: 64kB start at 0x1FFF4000
1:0	FLASHSIZE	RO	0x3	FLASH size 00: 32kB Flash 01: 64kB Flash 10: 96kB Flash 11: 128kB Flash

Table 22-37: System Register Write-Allow Key Register (SYSREGKEY) Layout

SYSREGKEY (System Register Write-Allow Key Register) Offset: 0x44 Default: 0x1ACCE551 Access: SYSTEM -> SYSREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 22-38: System Register Write-Allow Key Register (SYSREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected system registers

23 Debug Behavior

This chapter describes the debug behavior of some peripharals.

23.1 Watchdog timers

The WDT function can be suspended or free-running when the CPU enters HALT mode. WDT debug function is controlled by HALTEDRUN bit-field in WDTCTL register, as listed in Table 23-1.

Table 23-1: WDT debug behavior

WDTCTL.HALTEDRUN	Description
0	Watchdog timer counter stops immediately when CPU enters HALT mode
1	Watchdog timer counter keeps running when CPU enters HALT mode

23.2 General-purpose timers

The general-purpose timers have no specific debug feature. The timer counter keeps running when the CPU enters HALT mode.

23.3 PWM

The PWM function can be suspended or free-running when the CPU enters HALT mode. PWM debug function is controlled by DBGRUN bit-field in TBCTL register, as shown in Table 23-2.

Table 23-2: PWM debug behavior

TBCTL.DBGRUN	Description
0	PWM counter stops after the next time-base counter increment or decrement when CPU enters HALT mode
1	PWM counter stops when TBCNT completes a whole cycle (TBCNT=TBPRD for up-count mode and TBCNT=0 for other modes)
2	PWM counter keeps running when CPU enters HALT mode

23.4 ECAP

The ECAP function can be suspended or free-running when the CPU enters HALT mode. ECAP debug function is controlled by DBGRUN bit-field in CAPCTL register, as shown in Table 23-3.

Table 23-3: ECAP debug behavior

CAPCTL.DBGRUN	Description
0	ECAP counter stops immediately when CPU enters HALT mode
1	ECAP counter runs until TSCNT=0 when CPU enters HALT mode
2	ECAP counter keeps running when CPU enters HALT mode

23.5 UART

The registers of UART module can be read by the external debugger via debugger interface. Note that debug read operations can produce the same effect as CPU read operations. Table 23-4 showns some register debug behavior.

Table 23-4: UART debug behavior

Register	Description
UARTRBR	In non-FIFO mode, debug read this register will empty UARTRBR register. In FIFO mode, debug read this register will get the data byte at the front of the Receive FIFO and the value of UARTFOR will decrease by 1 at the same time.
UARTIIR.NIP UARTIIR.IID UARTIIR.ABL	Debug read UARTIIR will clear these register bit-fields.

23.6 SSP

The registers of SSP module can be read by the external debugger via debugger interface. Note that debug read operations can produce the same effect as CPU read operations. Table 23-5 showns some register debug behavior.

Table 23-5: SSP debug behavior

Register	Description
SSPDATA	If RXFIFO is not empty, debug read this register will get the data at the front of the RXFIFO and the value of SSPSTS.RXLVL will decrease by 1 at the same time.

23.7 I2C

The registers of I2C module can be read by the external debugger via debugger interface. Note that debug read operations can produce the same effect as CPU read operations. Table 23-6 showns some register debug behavior.

Table 23-6: I2C debug behavior

Register	Description
I2CDATACMD	If Receive FIFO is not empty and I2CDATACMD.CMD = 1, debug read this register will get the data at the front of the Receive FIFO and the value of I2CRFLVL will decrease by 1 at the same time.
I2CINTCLR	Debug read this register will clear the combined interrupt, all individual interrupts, and the I2CTXABRTSRC register. Refer to I2CTXABRTSRC.STARTNORESTART for an exception to clear I2CTXABRTSRC register.
I2CRXUDFCLR	Debug read this register will clear RXUDF flag
I2CRXOVFCLR	Debug read this register will clear RXOVF flag
I2CTXOVFCLR	Debug read this register will clear TXOVF flag
I2CRDREQCLR	Debug read this register will clear RDREQ flag
I2CTXABRTCLR	Debug read this register will clear the TXABRT flag and the I2CTXABRTSRC register. Refer to I2CTXABRTSRC.STARTNORESTART for an exception to clear I2CTXABRTSRC register.
I2CRXDONECLR	Debug read this register will clear RXDONE flag
I2CACTCLR	Debug read this register will clear the ACT flag if the I2C is not active anymore
I2CSTOPDETCLR	Debug read this register will clear STOPDET flag
I2CSTARTDETCLR	Debug read this register will clear STARTDET flag
I2CGENCALLCLR	Debug read this register will clear GENCALL flag

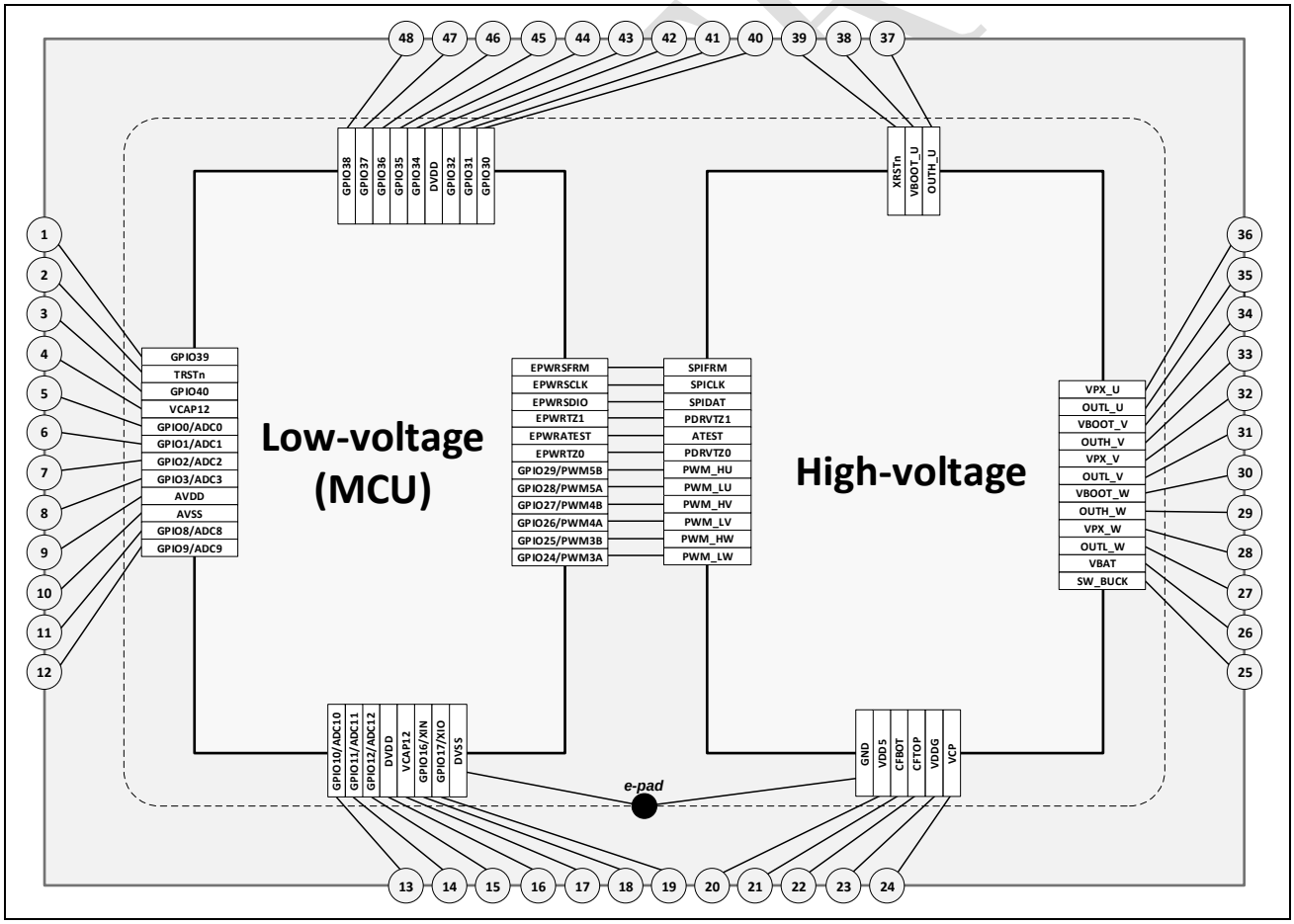
24 Low voltage and high voltage communication protocol

As shown in the Figure 1-1, there are two major parts inside the chip, low voltage module (including MCU core) and High voltage power/pre-driver module. The physical connection scheme between the two modules is shown in Figure 24-1.

Master power VBAT powerup enables key biasing circuits and SPI clock of the High voltage module. A Spintrol-standard ePower control interface is used between the MCU and the High voltage module. The power portion of High voltage module (VDD5 LDO and buck DC-DC producing DVDD power) are enabled by default at powerup. When setting EPWRCTL.ENSPI to 1, the SSP peripheral port of MCU will connect to EPWRSDIO (SPI data channel), EPWRSFRM (SPI frame channel) and EPWRSCLK (SPI clock) pins automatically. Then, MCU can communicate with High voltage module through a dedicated EPWR SPI interface.

Afterwards, MCU can write High voltage module internal registers through SPI interface in order to control the module behavior, including enabling and configuring Pre-Driver and it's charge pump, high voltage Brown-Out Detectors (BOD's), buck DC-DC and so on.

Figure 24-1: Physical communication scheme between MCU and HV module



The connection details of the low-voltage (MCU) block and the high-voltage module are shown in Figure 24-1. The corresponding mapping details of ePower interface as well as low voltage PWM channels for Pre-Driver block is shown in Table 24-1. Please refer to Table 4-1 for details of the MCU GPIO channel definition, to properly configure general MCU bonded-out GPIOs (such as

GPIO0~GPIO17 and GPIO30~GPIO40) and internally-connected GPIOs typically used as PWMs (GPIO24~GPIO29).

Table 24-1: MCU channels connected to High Voltage module

MCU channel description	Pre-Driver block description
EPWRSFRM	SPI frame
EPWRSCLK	SPI clock
EPWRSDIO	SPI data
EPWRATEST	Allow transfering analog signals from High voltage module, for subsequent ADC measurement
EPWRTZ1	PDRVTZ1 (trip-zone 1)
EPWRTZ0	PDRVTZ0 (trip-zone 0)
GPIO29/PWM5B	PWM_HU (high-side U-phase PWM input)
GPIO28/PWM5A	PWM_LU (low-side U-phase PWM input)
GPIO27/PWM4B	PWM_HV (high-side V-phase PWM input)
GPIO26/PWM4A	PWM_LV (high-side V-phase PWM input)
GPIO25/PWM3B	PWM_HW (high-side W-phase PWM input)
GPIO24/PWM3A	PWM_LW (high-side W-phase PWM input)

24.1 High-Voltage Module initialization

The High-Voltage Module initialization flow is as follows:

- Program MCU ePower registers to enable High-Voltage Module control
  - Program ePower Register Write-Allow Key Register EPWRREGKEY with 0x1ACCE551 to disable write protection and enable MCU writing ePower control registers.
  - Program ePower Trip-Zone Event 0 Control Register EPWRTZ0CTL, ePower Trip-Zone Event 1 Control Register EPWRTZ1CTL and ePower Control Register EPWRCTL. Especially, set EPWRCTL.ENSPI to enable SPI communication interface between MCU and High-Voltage Module.
  - Program ePower Register Write-Allow Key Register EPWRREGKEY with any value that not equal to 0x1ACCE551 to enable write protection and disable MCU writing ePower control registers.
- Program MCU SSP peripheral registers

  Select SSP as normal master mode, frame size is 32bits width, SSP_SCLK polarity and phase are all zero, frame polarity is low. Remember to enable SSP clock before doing these action
- Query high voltage modue ID

  MCU transmits request code (0x514E4944) to High-Voltage Module, and High-Voltage Module will return code with High-Voltage Module ID. After getting ID successfully, the High-Voltage Module is ready for MCU control – Read or write High-Voltage Module registers.

- Program MCU PINMUX registers to configure the GPIO pins as PWM outputs, which are connected to Pre-Driver, in order to toggle it's inputs.

24.2 Communication format between MCU and High-Voltage Module

The communication format between MCU and High-Voltage Module is based on SPI format. Just one data pin EPWRSDIO is used to transfer data.

- Set EPWRCTL.SPIDIR: MCU transmit data to High-Voltage Module
- Clear EPWRCTL.SPIDIR: MCU receive data from High-Voltage Module

The communition SPI format is shown in Figure 24-2 and Figure 24-3.

Figure 24-2: MCU Transmit Data to High-Voltage Module

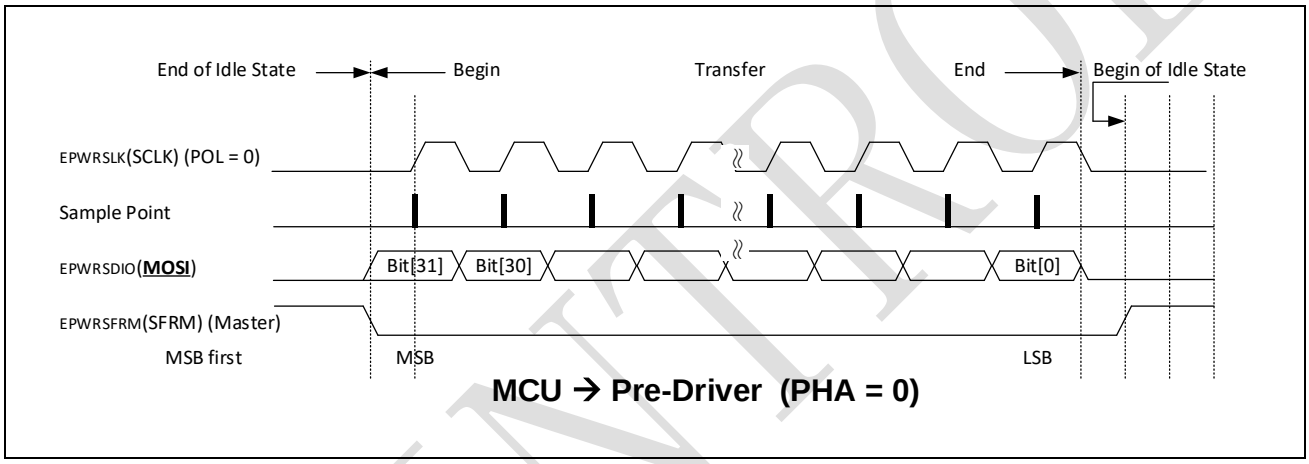
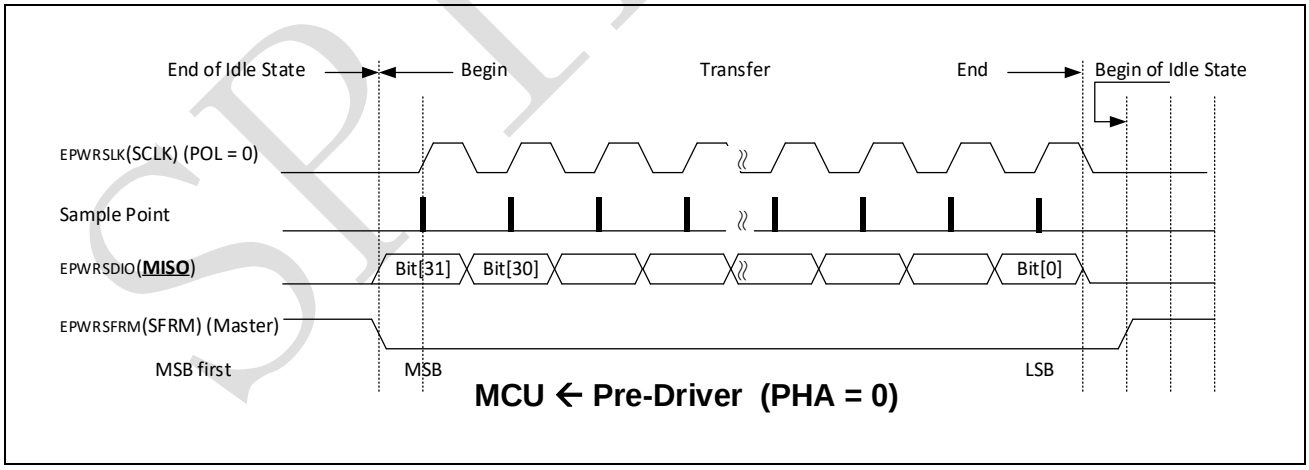


Figure 24-3: MCU Receive Data from High-Voltage Module



24.3 Communication protocol or command

There are four command types for SPI communication between MCU and High-Voltage Module. For each type, always MCU transmits data to High-Voltage Module first, and then High-Voltage Module answers by sending data back to MCU.

- Query High-Voltage Module ID

This command requests High-Voltage Module to send its ID to MCU.

MCU Transmit: 0x514E4944

High-Voltage Module Answer: (0x4944 << 16) + (0x4B << 8) + (ID)

High-Voltage Module ID is a 8-bit wide number.

- Write Register Request
  This command just writes TXDATA to High-Voltage Module shadow register.
  Note: After MCU transmitting **Write Register Execute Request** to High-Voltage Module successfully, the TXDATA would be written to the actual High-Voltage Module register.
  MCU Transmit: (0x57<<24) + (0x2<<21) + (ADDR<<16) + (TXDATA<<8) + 0x5A
  High-Voltage Module answer: (0x57<<24) + (0x5<<21) + (ADDR<<16) + (TXDATA<<8) + 0x5A
  ADDR is the High-Voltage Module register address, 5-bit wide; TXDATA is the data to be written to the High-Voltage Module register, 8-bit wide.
- Write Register Execute Request
  This command requests High-Voltage Module to write the actual register with the vaule in the shadow register.
  MCU Transmit: 0x45786563
  High-Voltage Module Answer: 0x446F6E65
- Read Register Request
  This command reads High-Voltage Module actual register value. After receiving MCU request successfully, High-Voltage Module will send the register value to MCU.
  MCU Transmit: (0x52<<24) + (0x2<<21) + (ADDR<<16) + 0x445A
  High-Voltage Module Answer: (0x52<<24) + (0x5<<21) + (ADDR<<16) + (RXDATA<<8) + RXDATA
  ADDR is the High-Voltage Module register address, 5-bit wide; RXDATA is the register value, 8-bit wide.

24.4 High-Voltage Module register write protection

Before writing High-Voltage Module registers, MCU should write High-Voltage Module register CTLKEY with a corresponding value described in Table 24-54 to disable register write protection first.

24.5 Fault handling

All the falut signals like VBAT/VDDG over-voltage/under-voltage, Pre-Driver over-current events reported by the VDS mornitors, over-temperature, can be selected and configured as trip zone through EPWRTZ0 and EPWRTZ1. VBAT/VDDG UV, OV, VDS monitors, and over-temperature's protection threshold can be programed by setting registers. For over temperature protection, 2 different protection level can be set and recommended to use one as warning and the other one as shutdown immediately. Once the selected fault triggered, PWM signal from MCU can be immediately stopped. All the fault statius can be read through registers. Once the fault is triggered, the status registers will not be cleared automatically when the fault situation is gone. User has to manually write clear registers. This feature helps to track which fault happened after trip zone had been triggered.

24.6 Registers

24.6.1 EPWR register map

Table 24-2: EPWR Module Base Address

Peripheral Module	Base Address
EPWR	0x4000 0400

Table 24-3: EPWR Register Map

Register	Offset	Description	Reset Value
EPWRTZFLT	0x0	ePower Module Filtered Trip-Zone Event Register	0x00000000
EPWRTZSTS	0x4	ePower Module Latched Trip-Zone Status Register	0x00000000
EPWRTZCLR	0x8	ePower Module Latched Trip-Zone Status Clear Register	0x00000000
EPWRTZ0CTL*	0xC	ePower Module Trip-Zone Event 0 Control Register	0x00000288
EPWRTZ1CTL*	0x10	ePower Module Trip-Zone Event 1 Control Register	0x00000288
EPWRCTL*	0x14	ePower Module Control Register	0x00000000
EPWRREGKEY	0x18	ePower Module Register Write-Allow Key Register	0x1ACCE551

Note: Registers marked with * are write-allowed only when the EPWRREGKEY=0x1ACCE551.

24.6.2 EPWR registers

Table 24-4: ePower Module Filtered Trip-Zone Event Register (EPWRTZFLT) Layout

EPWRTZFLT (ePower Module Filtered Trip-Zone Event Register) Offset: 0x0 Default: 0x00000000 Access: EPWR -> EPWRTZFLT.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						EPWRTZ1	EPWRTZ0

Table 24-5: ePower Module Filtered Trip-Zone Event Register (EPWRTZFLT) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	EPWRTZ1	RO	0x0	Filtered EPWRTZ1 event without latch 0: EPWRTZ1 event does not happen 1: EPWRTZ1 event happened
0	EPWRTZ0	RO	0x0	Filtered EPWRTZ0 event without latch 0: EPWRTZ0 event does not happen 1: EPWRTZ0 event happened

Table 24-6: ePower Module Latched Trip-Zone Status Register (EPWRTZSTS) Layout

EPWRTZSTS (ePower Module Latched Trip-Zone Status Register) Offset: 0x4 Default: 0x00000000 Access: EPWR -> EPWRTZSTS.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						EPWRTZ1	EPWRTZ0

Table 24-7: ePower Module Latched Trip-Zone Status Register (EPWRTZSTS) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	EPWRTZ1	RO	0x0	Latched EPWRTZ1 event after filtering 0: EPWRTZ1 event does not happen 1: EPWRTZ1 event happened

Bits	Field Name	Type	Reset	Description
0	EPWRTZ0	RO	0x0	Latched EPWRTZ0 event after filtering 0: EPWRTZ0 event does not happen 1: EPWRTZ0 event happened

Table 24-8: ePower Module Latched Trip-Zone Status Clear Register (EPWRTZCLR) Layout

EPWRTZCLR (ePower Module Latched Trip-Zone Status Clear Register) Offset: 0x8 Default: 0x00000000 Access: EPWR -> EPWRTZCLR.all							
31	30	29	28	27	26	25	24
RESERVED_31_2							
23	22	21	20	19	18	17	16
RESERVED_31_2							
15	14	13	12	11	10	9	8
RESERVED_31_2							
7	6	5	4	3	2	1	0
RESERVED_31_2						EPWRTZ1	EPWRTZ0

Table 24-9: ePower Module Latched Trip-Zone Status Clear Register (EPWRTZCLR) Description

Bits	Field Name	Type	Reset	Description
31:2	RESERVED_31_2	RO	0x0	Reserved.
1	EPWRTZ1	W1C	0x0	Latched EPWRTZ1 status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears EPWRTZSTS[EPWRTZ1]. This bit is self-cleared.
0	EPWRTZ0	W1C	0x0	Latched EPWRTZ0 status clear 0: Write a 0 has no effect. Always readback 0 1: Write a 1 clears EPWRTZSTS[EPWRTZ0]. This bit is self-cleared.

Table 24-10: ePower Module Trip-Zone Event 0 Control Register (EPWRTZ0CTL) Layout

EPWRTZ0CTL (ePower Module Trip-Zone Event 0 Control Register) Offset: 0xC Default: 0x00000288 Access: EPWR -> EPWRTZ0CTL.all							
31	30	29	28	27	26	25	24
FLTRST	FLTTH					FLTWIN	
23	22	21	20	19	18	17	16
FLTWIN			FLTDIV				
15	14	13	12	11	10	9	8
FLTDIV					SYNCCLREN	INPOL	IE
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 24-11: ePower Module Trip-Zone Event 0 Control Register (EPWRTZ0CTL) Description

Bits	Field Name	Type	Reset	Description
31	FLTRST	W1C	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
30:26	FLTTH	RW	0x0	Filter's threhold is (FLTTH+1) 1's/0's for output to toggle
25:21	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
20:11	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
10	SYNCCLREN	RW	0x0	Enable latched EPWRTZ0 status clear by PWMSYNC 0: PWMSYNC does not affect the latched EPWRTZ0 status 1: PWMSYNC will clear the latched EPWRTZ0 status
9	INPOL	RW	0x1	Trip-zone input polarity 0: Trip-zone event happens when input is low 1: Trip-zone event happens when input is high
8	IE	RW	0x0	Trip-zone event input enable 0: Disable trip-zone input 1: Enable trip-zone input
7	POL4GPIO	RW	0x1	Trip-zone output polarity for GPIO 0: Output 0 upon EPWRTZ0 event 1: Output 1 upon EPWRTZ0 event
6:4	SEL4GPIO	RW	0x0	Trip-zone output select for GPIO 000: Original output 001: Synchronous output with PWM clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter

Bits	Field Name	Type	Reset	Description
				output 101: Synchronous output with PWM clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with PWM clock or Latched digital filter output
3	POL4PWM	RW	0x1	Trip-zone output polarity for PWM 0: Output 0 upon EPWRTZ0 event 1: Output 1 upon EPWRTZ0 event
2:0	SEL4PWM	RW	0x0	Trip-zone output select for PWM 000: Original output 001: Synchronous output with PWM clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with PWM clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with PWM clock or Latched digital filter output

Table 24-12: ePower Module Trip-Zone Event 1 Control Register (EPWRTZ1CTL) Layout

EPWRTZ1CTL (ePower Module Trip-Zone Event 1 Control Register) Offset: 0x10 Default: 0x00000288
Access: EPWR -> EPWRTZ1CTL.all

31	30	29	28	27	26	25	24
FLTRST	FLTTH					FLTWIN	
23	22	21	20	19	18	17	16
FLTWIN			FLTDIV				
15	14	13	12	11	10	9	8
FLTDIV					SYNCCLREN	INPOL	IE
7	6	5	4	3	2	1	0
POL4GPIO	SEL4GPIO			POL4PWM	SEL4PWM		

Table 24-13: ePower Module Trip-Zone Event 1 Control Register (EPWRTZ1CTL) Description

Bits	Field Name	Type	Reset	Description
31	FLTRST	W1C	0x0	Reset all registers in the filter with the input 0: Write 0 has no effect. Always read back 0. 1: Write 1 will fill all registers in the filter with the input
30:26	FLTTH	RW	0x0	Filter's threhold is (FLTTH+1) 1's/0's for output to toggle
25:21	FLTWIN	RW	0x0	Filter's window size is (FLTWIN+1) samples
20:11	FLTDIV	RW	0x0	Filter's clock dividing ratio is (FLTDIV+1)
10	SYNCCLREN	RW	0x0	Enable latched EPWRTZ1 status clear by PWMSYNC 0: PWMSYNC does not affect the latched EPWRTZ1 status 1: PWMSYNC will clear the latched EPWRTZ1 status
9	INPOL	RW	0x1	Trip-zone input polarity 0: Trip-zone event happens when input is low 1: Trip-zone event happens when input is high
8	IE	RW	0x0	Trip-zone event input enable 0: Disable trip-zone input 1: Enable trip-zone input
7	POL4GPIO	RW	0x1	Trip-zone output polarity for GPIO 0: Output 0 upon EPWRTZ1 event 1: Output 1 upon EPWRTZ1 event
6:4	SEL4GPIO	RW	0x0	Trip-zone output select for GPIO 000: Original output 001: Synchronous output with PWM clock 010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with PWM clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with PWM clock or Latched digital filter output
3	POL4PWM	RW	0x1	Trip-zone output polarity for PWM 0: Output 0 upon EPWRTZ1 event 1: Output 1 upon EPWRTZ1 event
2:0	SEL4PWM	RW	0x0	Trip-zone output select for PWM 000: Original output 001: Synchronous output with PWM clock

Bits	Field Name	Type	Reset	Description
				010: Digital filter output 011: Latched digital filter output 100: Original output or Latched digital filter output 101: Synchronous output with PWM clock or Latched digital filter output 110: Digital filter output or the latched one 111: Original output or Synchronous output with PWM clock or Latched digital filter output

Table 24-14: ePower Module Control Register (EPWRCTL) Layout

EPWRCTL (ePower Module Control Register) Offset: 0x14 Default: 0x00000000
Access: EPWR -> EPWRCTL.all

31	30	29	28	27	26	25	24
RESERVED_31_10							
23	22	21	20	19	18	17	16
RESERVED_31_10							
15	14	13	12	11	10	9	8
RESERVED_31_10						TZSEL4GPIO11	SYNCSEL1
7	6	5	4	3	2	1	0
SYNCSEL1		SYNCSEL0			SPIDIR	ENSPI	EN

Table 24-15: ePower Module Control Register (EPWRCTL) Description

Bits	Field Name	Type	Reset	Description
31:10	RESERVED_31_10	RO	0x0	Reserved.
9	TZSEL4GPIO11	RW	0x0	Select which TZ signal is to output to GPIO11 0: EPWRTZ0 1: EPWRTZ1
8:6	SYNCSEL1	RW	0x0	Select SYNC from PWM[SYNCSEL1] to clear latched EPWRTZ1 status
5:3	SYNCSEL0	RW	0x0	Select SYNC from PWM[SYNCSEL0] to clear latched EPWRTZ0 status
2	SPIDIR	RW	0x0	SPI TX/RX direction 0: Receive data from pre-driver module 1: Transmit data to pre-dirver module
1	ENSPI	RW	0x0	Enable SPI interface with the ePower module 0: Disable 1: Enable
0	EN	RW	0x0	Enable the ePower module 0: Disable 1: Enable

Table 24-16: ePower Module Register Write-Allow Key Register (EPWRREGKEY) Layout

EPWRREGKEY (ePower Module Register Write-Allow Key Register) Offset: 0x18 Default: 0x1ACCE551 Access: EPWR -> EPWRREGKEY.all							
31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
KEY							

Table 24-17: ePower Module Register Write-Allow Key Register (EPWRREGKEY) Description

Bits	Field Name	Type	Reset	Description
31:0	KEY	RW	0x1ACCE551	Write 0x1ACCE551 to unlock protected registers

24.6.3 HV module register map

Table 24-18: HV Register Map

Register	Offset	Description	Reset Value
GLBSTS0	0x0	Global Status Register 0	0x00
GLBSTS1	0x1	Global Status Register 1	0x00
GLBSTSCLR0	0x2	Global Status Clear Register 0	0x00
GLBSTSCLR1	0x3	Global Status Clear Register 1	0x00
PDRVTZ0SEL0	0x4	PDRVTZ0 Select Register 0	0x00
PDRVTZ0SEL1	0x5	PDRVTZ0 Select Register 1	0x00
PDRVTZ1SEL0	0x6	PDRVTZ1 Select Register 0	0x00
PDRVTZ1SEL1	0x7	PDRVTZ1 Select Register 1	0x00
BODVBATVTH	0x8	VBAT BOD Threshold Register	0x70
BODVDDGVTH	0x9	VDDG BOD Threshold Register	0x70
OTTH	0xA	Over-Temperature Threshold Register	0x77
SYSCTL	0xB	System Control Register	0x1A
BUCKCTL	0xC	BUCK Control Register	0x81
SSCTL	0xD	Spread-Spectrum Control Register	0x00
PDRVCTL	0xE	Pre-Driver Control Register	0x90
PDRVOCCTL	0xF	Pre-Driver Over-Current Control Register	0xFF
VMONCTL	0x10	Supply Voltage Monitor Control Register	0x00
CTLKEY	0x11	Control Key Register	0x00
CHPENGR	0x1C	Charge-Pump Engineering Register	0x04

24.6.4 HV module registers

Table 24-19: Global Status Register 0 (GLBSTS0) Layout

GLBSTS0 (Global Status Register 0) Offset: 0x0 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED_7	CPRDY	OT1	OT0	VDDGH	VDDGL	VBATH	VBATL

Table 24-20: Global Status Register 0 (GLBSTS0) Description

Bits	Field Name	Type	Reset	Description
7	RESERVED_7	RO	0x0	Reserved.
6	CPRDY	RO	0x0	Indicate charge-pump is ready 0: Not ready 1: Ready
5	OT1	RO	0x0	Indicate an over-temperature event 1 0: Not occurred 1: Occurred
4	OT0	RO	0x0	Indicate an over-temperature event 0 0: Not occurred 1: Occurred
3	VDDGH	RO	0x0	Indicate VDDG is too high 0: Not occurred 1: Occurred
2	VDDGL	RO	0x0	Indicate VDDG is too low 0: Not occurred 1: Occurred
1	VBATH	RO	0x0	Indicate VBAT is too high 0: Not occurred 1: Occurred
0	VBATL	RO	0x0	Indicate VBAT is too low 0: Not occurred 1: Occurred

Table 24-21: Global Status Register 1 (GLBSTS1) Layout

GLBSTS1 (Global Status Register 1) Offset: 0x1 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED_7	RESERVED_6	OCHW	OCHV	OCHU	OCLW	OCLV	OCLU

Table 24-22: Global Status Register 1 (GLBSTS1) Description

Bits	Field Name	Type	Reset	Description
7	RESERVED_7	RO	0x0	Reserved.
6	RESERVED_6	RO	0x0	Reserved.
5	OCHW	RO	0x0	Indicate over-current on high-side of W phase 0: Not occurred 1: Occurred
4	OCHV	RO	0x0	Indicate over-current on high-side of V phase 0: Not occurred 1: Occurred
3	OCHU	RO	0x0	Indicate over-current on high-side of U phase 0: Not occurred 1: Occurred
2	OCLW	RO	0x0	Indicate over-current on low-side of W phase 0: Not occurred 1: Occurred
1	OCLV	RO	0x0	Indicate over-current on low-side of V phase 0: Not occurred 1: Occurred
0	OCLU	RO	0x0	Indicate over-current on low-side of U phase 0: Not occurred 1: Occurred

Table 24-23: Global Status Clear Register 0 (GLBSTSCLR0) Layout

GLBSTSCLR0 (Global Status Clear Register 0) Offset: 0x2 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED_7_6		OT1	OT0	VDDGH	VDDGL	VBATH	VBATL

Table 24-24: Global Status Clear Register 0 (GLBSTSCLR0) Description

Bits	Field Name	Type	Reset	Description
7:6	RESERVED_7_6	RO	0x0	Reserved.
5	OT1	W1C	0x0	Clear GLBSTS0[OT1] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
4	OT0	W1C	0x0	Clear GLBSTS0[OT0] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit

Bits	Field Name	Type	Reset	Description
3	VDDGH	W1C	0x0	Clear GLBSTS0[VDDGH] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
2	VDDGL	W1C	0x0	Clear GLBSTS0[VDDGL] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
1	VBATH	W1C	0x0	Clear GLBSTS0[VBATH] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
0	VBATL	W1C	0x0	Clear GLBSTS0[VBATL] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit

Table 24-25: Global Status Clear Register 1 (GLBSTSCLR1) Layout

GLBSTSCLR1 (Global Status Clear Register 1) Offset: 0x3 Default: 0x00

7	6	5	4	3	2	1	0
RESERVED_7_6		OCHW	OCHV	OCHU	OCLW	OCLV	OCLU

Table 24-26: Global Status Clear Register 1 (GLBSTSCLR1) Description

Bits	Field Name	Type	Reset	Description
7:6	RESERVED_7_6	RO	0x0	Reserved.
5	OCHW	W1C	0x0	Clear GLBSTS1[OCHW] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
4	OCHV	W1C	0x0	Clear GLBSTS1[OCHV] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
3	OCHU	W1C	0x0	Clear GLBSTS1[OCHU] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
2	OCLW	W1C	0x0	Clear GLBSTS1[OCLW] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
1	OCLV	W1C	0x0	Clear GLBSTS1[OCLV] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit
0	OCLU	W1C	0x0	Clear GLBSTS1[OCLU] 0: Write a 0 has no effect. Always read back 0 1: Write a 1 clears the bit

Table 24-27: PDRVTZ0 Select Register 0 (PDRVTZ0SEL0) Layout

PDRVTZ0SEL0 (PDRVTZ0 Select Register 0) Offset: 0x4 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED_7_6		OT1	OT0	VDDGH	VDDGL	VBATH	VBATL

Table 24-28: PDRVTZ0 Select Register 0 (PDRVTZ0SEL0) Description

Bits	Field Name	Type	Reset	Description
7:6	RESERVED_7_6	RO	0x0	Reserved.
5	OT1	RW	0x0	Select over-temperature event 1 as PDRVTZ0 event 0: Disable 1: Enable
4	OT0	RW	0x0	Select over-temperature event 0 as PDRVTZ0 event 0: Disable 1: Enable
3	VDDGH	RW	0x0	Select VDDG-too-high as PDRVTZ0 event 0: Disable 1: Enable
2	VDDGL	RW	0x0	Select VDDG-too-low as PDRVTZ0 event 0: Disable 1: Enable
1	VBATH	RW	0x0	Select VBAT-too-high as PDRVTZ0 event 0: Disable 1: Enable
0	VBATL	RW	0x0	Select VBAT-too-low as PDRVTZ0 event 0: Disable 1: Enable

Table 24-29: PDRVTZ0 Select Register 1 (PDRVTZ0SEL1) Layout

PDRVTZ0SEL1 (PDRVTZ0 Select Register 1) Offset: 0x5 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED_7_6		OCHW	OCHV	OCHU	OCLW	OCLV	OCLU

Table 24-30: PDRVTZ0 Select Register 1 (PDRVTZ0SEL1) Description

Bits	Field Name	Type	Reset	Description
7:6	RESERVED_7_6	RO	0x0	Reserved.
5	OCHW	RW	0x0	Select over-current on high-side of W phase as PDRVTZ0 event 0: Disable 1: Enable
4	OCHV	RW	0x0	Select over-current on high-side of V phase as PDRVTZ0 event 0: Disable 1: Enable
3	OCHU	RW	0x0	Select over-current on high-side of U phase as PDRVTZ0 event 0: Disable 1: Enable
2	OCLW	RW	0x0	Select over-current on low-side of W phase as PDRVTZ0 event 0: Disable 1: Enable
1	OCLV	RW	0x0	Select over-current on low-side of V phase as PDRVTZ0 event 0: Disable 1: Enable
0	OCLU	RW	0x0	Select over-current on low-side of U phase as PDRVTZ0 event 0: Disable 1: Enable

Table 24-31: PDRVTZ1 Select Register 0 (PDRVTZ1SEL0) Layout

PDRVTZ1SEL0 (PDRVTZ1 Select Register 0) Offset: 0x6 Default: 0x00

7	6	5	4	3	2	1	0
RESERVED_7_6		OT1	OT0	VDDGH	VDDGL	VBATH	VBATL

Table 24-32: PDRVTZ1 Select Register 0 (PDRVTZ1SEL0) Description

Bits	Field Name	Type	Reset	Description
7:6	RESERVED_7_6	RO	0x0	Reserved.
5	OT1	RW	0x0	Select over-temperature event 1 as PDRVTZ1 event 0: Disable 1: Enable

Bits	Field Name	Type	Reset	Description
4	OT0	RW	0x0	Select over-temperature event 0 as PDRVTZ1 event 0: Disable 1: Enable
3	VDDGH	RW	0x0	Select VDDG-too-high as PDRVTZ1 event 0: Disable 1: Enable
2	VDDGL	RW	0x0	Select VDDG-too-low as PDRVTZ1 event 0: Disable 1: Enable
1	VBATH	RW	0x0	Select VBAT-too-high as PDRVTZ1 event 0: Disable 1: Enable
0	VBATL	RW	0x0	Select VBAT-too-low as PDRVTZ1 event 0: Disable 1: Enable

Table 24-33: PDRVTZ1 Select Register 1 (PDRVTZ1SEL1) Layout

PDRVTZ1SEL1 (PDRVTZ1 Select Register 1) Offset: 0x7 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED_7_6		OCHW	OCHV	OCHU	OCLW	OCLV	OCLU

Table 24-34: PDRVTZ1 Select Register 1 (PDRVTZ1SEL1) Description

Bits	Field Name	Type	Reset	Description
7:6	RESERVED_7_6	RO	0x0	Reserved.
5	OCHW	RW	0x0	Select over-current on high-side of W phase as PDRVTZ1 event 0: Disable 1: Enable
4	OCHV	RW	0x0	Select over-current on high-side of V phase as PDRVTZ1 event 0: Disable 1: Enable
3	OCHU	RW	0x0	Select over-current on high-side of U phase as PDRVTZ1 event 0: Disable 1: Enable
2	OCLW	RW	0x0	Select over-current on low-side of W phase as PDRVTZ1 event 0: Disable 1: Enable
1	OCLV	RW	0x0	Select over-current on low-side of V phase as PDRVTZ1 event 0: Disable 1: Enable
0	OCLU	RW	0x0	Select over-current on low-side of U phase as PDRVTZ1 event 0: Disable 1: Enable

Table 24-35: VBAT BOD Threshold Register (BODVBATVTH) Layout

BODVBATVTH (VBAT BOD Threshold Register) Offset: 0x8 Default: 0x70							
7	6	5	4	3	2	1	0
RESERVED_7	VMAX			RESERVED_3	VMIN		

Table 24-36: VBAT BOD Threshold Register (BODVBATVTH) Description

Bits	Field Name	Type	Reset	Description
7	RESERVED_7	RO	0x0	Reserved.
6:4	VMAX	RW	0x7	VBAT-too-high threshold 000: Assert if Vbat>14.3V, de-assert if Vbat<13.6V 001: Assert if Vbat>16.8V, de-assert if Vbat<15.8V 010: Assert if Vbat>20.2V, de-assert if Vbat<18.8V 011: Assert if Vbat>23.4V, de-assert if Vbat<21.6V 100: Assert if Vbat>27.8V, de-assert if Vbat<25.3V 101: Assert if Vbat>30.6V, de-assert if Vbat<27.6V 110: Assert if Vbat>34.2V, de-assert if Vbat<30.3V 111: Assert if Vbat>44.4V, de-assert if Vbat<38.5V
3	RESERVED_3	RO	0x0	Reserved.
2:0	VMIN	RW	0x0	VBAT-too-low threshold select 000: Assert if Vbat<5.0V, de-assert if Vbat>5.2V 001: Assert if Vbat<6.0V, de-assert if Vbat>6.3V 010: Assert if Vbat<7.3V, de-assert if Vbat>7.6V 011: Assert if Vbat<9.5V, de-assert if Vbat>9.8V 100: Assert if Vbat<12.8V, de-assert if Vbat>13.5V 101: Assert if Vbat<16.5V, de-assert if Vbat>17.5V 110: Assert if Vbat<17.5V, de-assert if Vbat>19.9V 111: Assert if Vbat<23.0V, de-assert if Vbat>25.0V

Table 24-37: VDDG BOD Threshold Register (BODVDDGVTH) Layout

BODVDDGVTH (VDDG BOD Threshold Register) Offset: 0x9 Default: 0x70							
7	6	5	4	3	2	1	0
RESERVED_7	VMAX			RESERVED_3	VMIN		

Table 24-38: VDDG BOD Threshold Register (BODVDDGVTH) Description

Bits	Field Name	Type	Reset	Description
7	RESERVED_7	RO	0x0	Reserved.
6:4	VMAX	RW	0x7	VDDG-too-high threshold select 000: Assert if VDDG>14.2V, de-assert if VDDG <12.6V 001: Assert if VDDG>16.2V, de-assert if VDDG <14.2V 010: Assert if VDDG>17.3V, de-assert if VDDG <16.0V 011: Assert if VDDG>17.6V, de-assert if VDDG <16.3V 100: Assert if VDDG>18.6V, de-assert if VDDG <17.2V 101: Assert if VDDG>19.0V, de-assert if VDDG <17.5V 110: Assert if VDDG>19.7V, de-assert if VDDG<18.2V 111: Assert if VDDG>20.2V, de-assert if VDDG <18.5V
3	RESERVED_3	RO	0x0	Reserved.
2:0	VMIN	RW	0x0	VDDG-too-low threshold select 000: Assert if VDDG<7.7V, de-assert if VDDG>8.8V 001: Assert if VDDG<8.0V, de-assert if VDDG>9.2V 010: Assert if VDDG<8.2V, de-assert if VDDG>9.5V 011: Assert if VDDG<8.5V, de-assert if VDDG>9.9V 100: Assert if VDDG<8.8V, de-assert if VDDG>10.4V 101: Assert if VDDG<9.1V, de-assert if VDDG>10.8V 110: Assert if VDDG<9.5V, de-assert if VDDG>11.4V 111: Assert if VDDG<9.9V, de-assert if VDDG>11.9V

Table 24-39: Over-Temperature Threshold Register (OTTH) Layout

OTTH (Over-Temperature Threshold Register) Offset: 0xA Default: 0x77							
7	6	5	4	3	2	1	0
RESERVED_7	OT1			RESERVED_3	OT0		

Table 24-40: Over-Temperature Threshold Register (OTTH) Description

Bits	Field Name	Type	Reset	Description
7	RESERVED_7	RO	0x0	Reserved.
6:4	OT1	RW	0x7	Threshold for over-temperature event 1 000: Assert if 102 degrees, deassert if 92 degrees 001: Assert if 114 degrees, deassert if 101 degrees 010: Assert if 123 degrees, deassert if 114 degrees 011: Assert if 132 degrees, deassert if 122 degrees 100: Assert if 140 degrees, deassert if 131 degrees 101: Assert if 149 degrees, deassert if 139 degrees 110: Assert if 157 degrees, deassert if 148 degrees 111: Assert if 166 degrees, deassert if 157 degrees
3	RESERVED_3	RO	0x0	Reserved.
2:0	OT0	RW	0x7	Threshold for over-temperature event 0 000: Assert if 102 degrees, deassert if 92 degrees 001: Assert if 114 degrees, deassert if 101 degrees 010: Assert if 123 degrees, deassert if 114 degrees 011: Assert if 132 degrees, deassert if 122 degrees 100: Assert if 140 degrees, deassert if 131 degrees 101: Assert if 149 degrees, deassert if 139 degrees 110: Assert if 157 degrees, deassert if 148 degrees 111: Assert if 166 degrees, deassert if 157 degrees

Table 24-41: System Control Register (SYSCTL) Layout

SYSCTL (System Control Register) Offset: 0xB Default: 0x1A							
7	6	5	4	3	2	1	0
VBATDIVEN	VDDGRDYBYP	VDDGSEL			RSTDGWIN		

Table 24-42: System Control Register (SYSCTL) Description

Bits	Field Name	Type	Reset	Description
7	VBATDIVEN	RW	0x0	High-speed VBAT divider enabled for fast ADC sampling (Vbat/20) 0: Disable 1: Enable Vbat/20
6	VDDGRDYBYP	RW	0x0	Bypass LDO ready 0: Not bypass 1: Bypass
5:3	VDDGSEL	RW	0x3	VDDG selection 000: 5V 001: 6V 010: 9V 011: 10V 100: 12V 101: 15V 110: 18V 111: 20V (do not use)
2:0	RSTDGWIN	RW	0x2	Pin reset deglitch for VDD5 domian 000: 4.2us 001: 97us 010: 394us 011: 1.58ms 100: 3.17ms 101: 6.34ms 110: 26.68ms 111: 102.7ms

Table 24-43: BUCK Control Register (BUCKCTL) Layout

BUCKCTL (BUCK Control Register) Offset: 0xC Default: 0x81							
7	6	5	4	3	2	1	0
FREQSEL		FRCPWM	LOWPWREN	OVPROTEN	TZ1EN	TZ0EN	EN

Table 24-44: BUCK Control Register (BUCKCTL) Description

Bits	Field Name	Type	Reset	Description
7:6	FREQSEL	RW	0x2	BUCKG switching frequency select 00: 400kHz 01: 600kHz 10: 1.2MHz 11: 2.4MHz
5	FRCPWM	RW	0x0	Force BUCK operates in PWM mode 0: BUCK's mode is controlled by the circuits 1: BUCK is forced to operate in PWM mode
4	LOWPWREN	RW	0x0	Low-power enable in PFM mode 0: Normal operation 1: Shut-down over-current monitor
3	OVPROTEN	RW	0x0	BUCK output (VDD33) over-voltage protection enable 0: Disable over-voltage protection 1: Enable over-voltage protection
2	TZ1EN	RW	0x0	Enable TZ1 mask on BUCK enable so that the BUCK is self-disabled upon PDRVTZ1 event This bit is writable only when CTLKEY=0x88 0: PDRVTZ1 event will not affect the BUCK operation 1: BUCK is immediatedly disabled upon PDRVTZ1 event
1	TZ0EN	RW	0x0	Enable TZ0 mask on BUCK enable so that the BUCK is self-disabled upon PDRVTZ0 event. This bit is writable only when CTLKEY=0x88 0: PDRVTZ0 event will not affect the BUCK operation 1: BUCK is immediatedly disabled upon PDRVTZ0 event
0	EN	RW	0x1	BUCK enable. This bit is writable only when CTLKEY=0x88 0: Disable the BUCK 1: Enable the BUCK

Table 24-45: Spread-Spectrum Control Register (SSCTL) Layout

SSCTL (Spread-Spectrum Control Register) Offset: 0xD Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED_7_4				PRD	AMP		EN

Table 24-46: Spread-Spectrum Control Register (SSCTL) Description

Bits	Field Name	Type	Reset	Description
7:4	RESERVED_7_4	RO	0x0	Reserved.
3	PRD	RW	0x0	BUCK spread-spectrum period 0: Normal period, which is 2 x BUCKGSSAMP clocks 1: Twice the period, which is 4x BUCKGSSAMP clocks
2:1	AMP	RW	0x0	BUCK spread-spectrum amplitude 00: Cap code is from 192 to 360 01: Cap code is from 171 to 341 10: Cap code is from 128 to 384 11: Cap code is from 0 to 511
0	EN	RW	0x0	BUCK spread-spectrum enable 0: Disable 1: Enable

Table 24-47: Pre-Driver Control Register (PDRVCTL) Layout

PDRVCTL (Pre-Driver Control Register) Offset: 0xE Default: 0x90							
7	6	5	4	3	2	1	0
CPCONEN	FRCCFM	RESERVED_5	NOVWIN	CPPROTEN	TZ1EN	TZ0EN	EN

Table 24-48: Pre-Driver Control Register (PDRVCTL) Description

Bits	Field Name	Type	Reset	Description
7	CPCONEN	RW	0x1	Pre-driver charge-pump connection enable This bit is writable only when CTLKEY=0xAC 0: Disable Always readback 0 when CTLKEY is not 0xAC 1: Enable
6	FRCCFM	RW	0x0	Pre-driver force confirmation This bit is writable only when CTLKEY=0xAC 0: Normal handshake Always readback 0 when CTLKEY is not 0xAC 1: Force confirmation, disabling built-in handshake

Bits	Field Name	Type	Reset	Description
5	RESERVED_5	RO	0x0	Reserved.
4	NOVWIN	RW	0x1	Pre-driver non-overlap time window size 0: 48ns 1: 96ns
3	CPPROTEN	RW	0x0	Pre-driver charge-pump protection enable 0: Disable 1: Enable so that the charge pump can be disconnected when a high-side V_{DS} Monitor fault occurs
2	TZ1EN	RW	0x0	Enable TZ1 mask on pre-driver enable so that the pre-driver is self-disabled upon PDRVTZ1 event This bit is writable only when CTLKEY=0x99 0: PDRVTZ1 event will not affect the pre-driver operation 1: Pre-driver is immediatedly disabled upon PDRVTZ1 event
1	TZ0EN	RW	0x0	Enable TZ0 mask on pre-driver enable so that the pre-driver is self-disabled upon PDRVTZ0 event. This bit is writable only when CTLKEY=0x99 0: PDRVTZ0 event will not affect the pre-driver operation 1: Pre-driver is immediatedly disabled upon PDRVTZ0 event
0	EN	RW	0x0	Pre-driver enable This bit is writable only when CTLKEY=0x99 0: Disable 1: Enable

Table 24-49: Pre-Driver Over-Current Control Register (PDRVOCCTL) Layout

PDRVOCCTL (Pre-Driver Over-Current Control Register) Offset: 0xF Default: 0xFF							
7	6	5	4	3	2	1	0
BLANKWIN		DGWIN		OCTH			

Table 24-50: Pre-Driver Over-Current Control Register (PDRVOCCTL) Description

Bits	Field Name	Type	Reset	Description
7:6	BLANKWIN	RW	0x3	Pre-driver over-current monitor blanking window size & charge pump connection blanking time 00: 0.4us 01: 0.8us 10: 1.6us 11: 3.2us
5:4	DGWIN	RW	0x3	Pre-driver over-current monitor filtering window size & charge pump connection filtering time 00: 0.4us 01: 0.8us 10: 1.6us 11: 3.2us
3:0	OCTH	RW	0xF	Pre-driver over-current monitor threshold 0000: Vds threshold=0.15V 0001: Vds threshold=0.3V 0010: Vds threshold=0.45V 0011: Vds threshold=0.6V 0100: Vds threshold=0.75V 0101: Vds threshold=0.9V 0110: Vds threshold=1.05V 0111: Vds threshold=1.2V 1000: Vds threshold=1.35V 1001: Vds threshold=1.5V 1010: Vds threshold=1.65V 1011: Vds threshold=1.8V 1100: Vds threshold=1.95V 1101: Vds threshold=2.1V 1110: Vds threshold=2.25V 1111: Vds threshold=2.4V

Table 24-51: Supply Voltage Monitor Control Register (VMONCTL) Layout

VMONCTL (Supply Voltage Monitor Control Register) Offset: 0x10 Default: 0x00							
7	6	5	4	3	2	1	0
RESERVED_7_2						TPSEL	EN

Table 24-52: Supply Voltage Monitor Control Register (VMONCTL) Description

Bits	Field Name	Type	Reset	Description
7:2	RESERVED_7_2	RO	0x0	Reserved.
1	TPSEL	RW	0x0	Test point select; cannot use at the same time with SYSCTL.VBATDIVEN 0: VDDG/20 1: VBAT/50
0	EN	RW	0x0	Supply voltage monitor enable 0: Disable 1: Enable

Table 24-53: Control Key Register (CTLKEY) Layout

CTLKEY (Control Key Register) Offset: 0x11 Default: 0x00							
7	6	5	4	3	2	1	0
KEY							

Table 24-54: Control Key Register (CTLKEY) Description

Bits	Field Name	Type	Reset	Description
7:0	KEY	RW	0x0	Key for changing critical settings 00000000: Lock all critical settings 01110111: Allow to soft-reset all registers including those xxxTRIM registers 10001000: Allow to write BUCKCTL[2:0] 10011001: Allow to write PDRVCTL[2:0] 10101100: Allow to write engineering registers 10111011: Allow to response to deep-sleep command

Table 24-55: Charge-Pump Engineering Register (CHPENGR) Layout

CHPENGR(Charge-Pump Engineering Register) Offset: 0x1C Default: 0x04							
7	6	5	4	3	2	1	0
RESERVED_7_4				CLKDIV		FRCRDY	FRCEN

Table 24-56: Charge-Pump Engineering Register (CHPENGR) Description

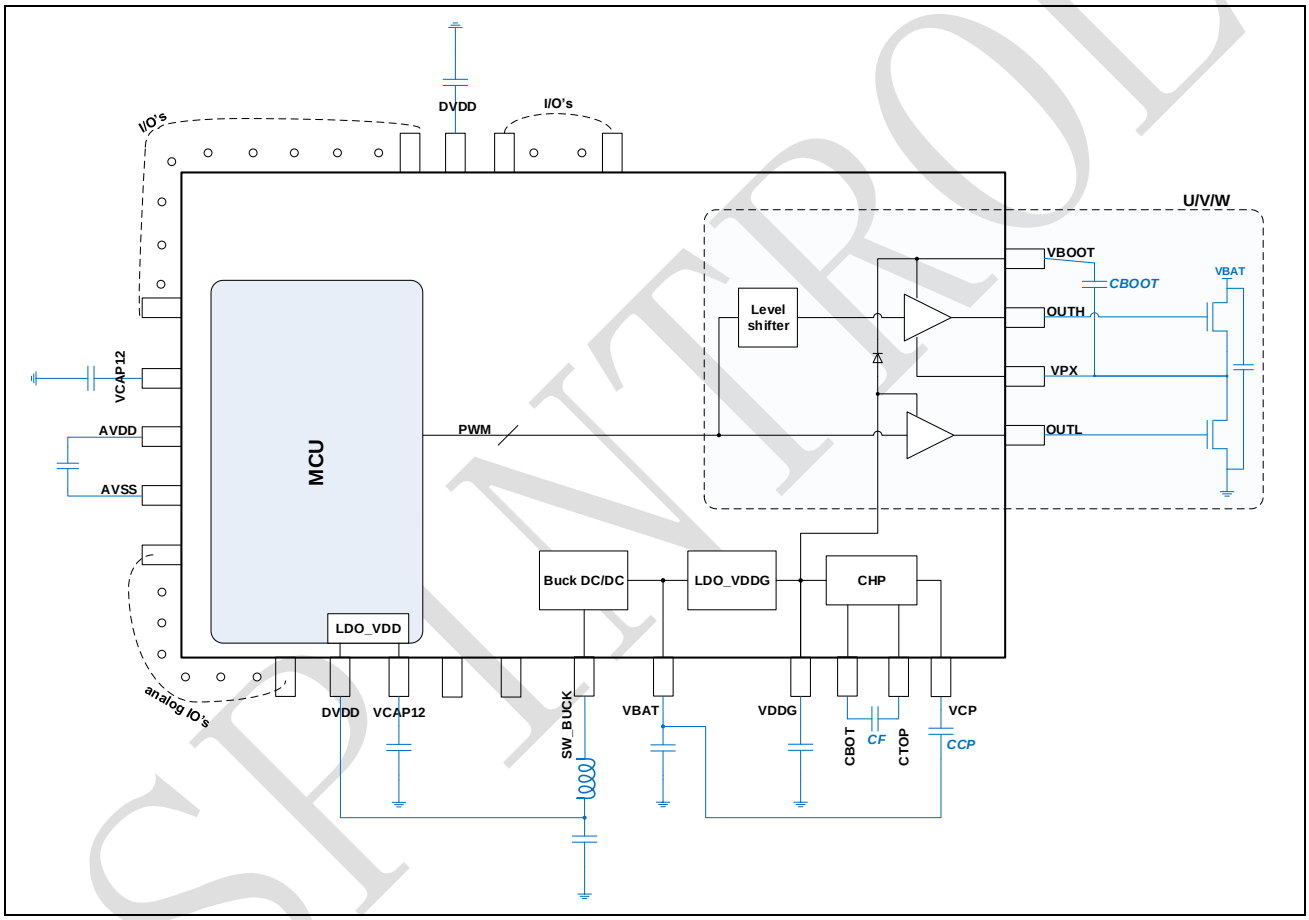
Bits	Field Name	Type	Reset	Description
7:4	RESERVED_7_4	RO	0x0	Reserved.
3:2	CLKDIV	RW	0x1	Chage-pump clock dividing ratio 00: Divide by 8 01: Divide by 16 10: Divide by 32 11: Divide by 64
1	FRCRDY	RW	0x0	Charge-pump force ready 0: Do not force so as to controlled by circuits 1: Force to be ready
0	FRCEN	RW	0x0	Charge-pump force enable 0: Do not force so as to controlled by circuits 1: Force to enable

25 Power management system

25.1 Overview

The power management system in High voltage module enables single power input in the motor driver application by providing VDDG voltage via LDO and DVDD voltage via buck DC-DC regulator. Also, deep sleep mode in VBAT domain will shutdown all power blocks to realize shutdown current as low as 6uA. The wakeup is achieved by toggling pin reset XRSTn low. Figure 25-1 shows the typical application diagram with some power blocks and Pre-Driver inside.

Figure 25-1: SPD1148 typical application diagram



25.2 Buck DC-DC regulator

The buck DC-DC regulator converts VBAT to DVDD rail, which is typically 3.3V for powering MCU. There is no external power FET or diode required, as well as no external compensation resistors or capacitors. The inductor value is 10uH and output cap is 10uF. The buck DC-DC regulator will start operating when VBAT voltage is higher than 4.56V and shut down when VBAT voltage drops below 4.15V. Typical switching frequency is 1.2MHz and typical current limit value is 500mA.

25.2.1 Programmable switching frequency

Refer to Table 24-44 on configuring the switching frequency. The default value is 1.2MHz.

25.2.2 PWM and PFM mode transition

For PFM mode, the ripple over inductor current can be estimated as:

$$V_{Ripple\_PFM} = \frac{T_{sw}^{\ 2} \cdot Vout \cdot (Vin - Vout)}{2 \cdot L \cdot C_{out} \cdot Vin}$$

For PWM mode, the ripple over inductor current can be estimated as:

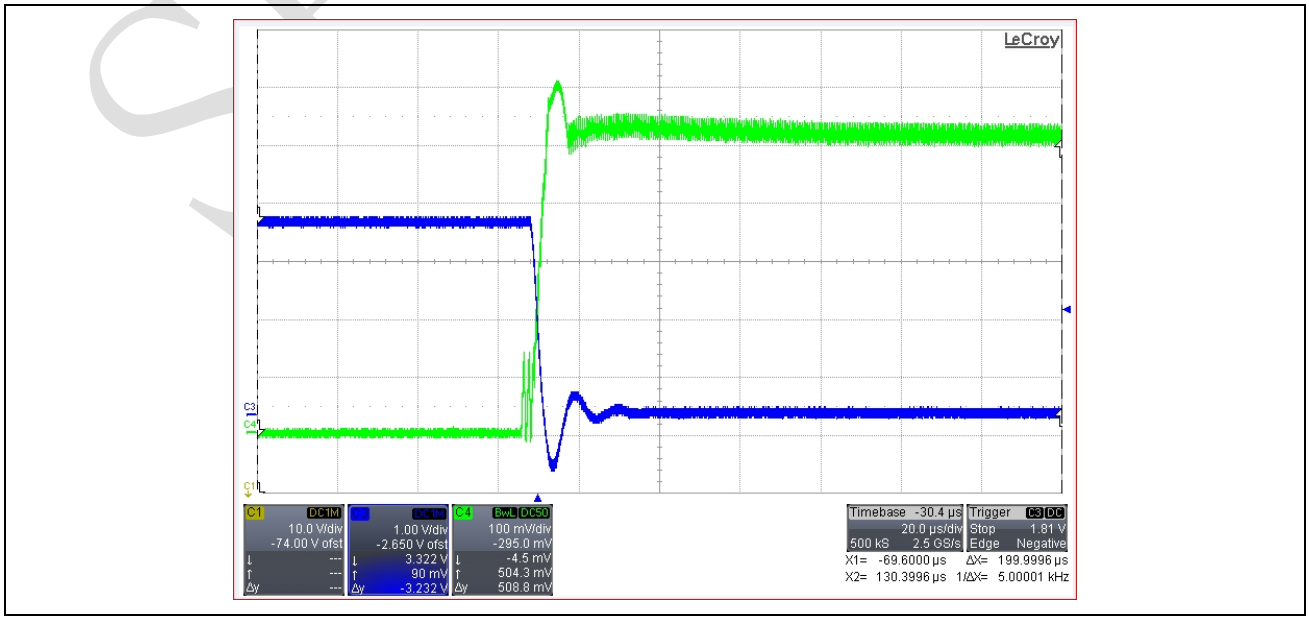
$$V_{Ripple\_PWM} = \frac{T_{sw}^{\ 2} \cdot Vout \cdot (Vin - Vout)}{8 \cdot L \cdot C_{out} \cdot Vin}$$

In the light load, Buck DC-DC will be operating at PFM mode to save power, at the expense of higher ripple on the 3.3V output. For the ripple-sensitive and regulation value-sensitive applications, it is recommended to configure the buck DC-DC at forced PWM mode (BUCKCTL0.FORCEPWM=1). During this mode at light load, the regulator will allow current flowing from SW_DC-DC to ground which degrades efficiency but provides better output ripple performance.

25.2.3 Output short protection

The Buck DC-DC will limit its output current to be less than 500mA. This feature is added for safety, to protect against output short to ground and problematic current increases which will cause chip and board reliability problems. If load current greater than 500mA is applied to the DC-DC output, the Buck DC-DC will lower down its 3.3V output and will operate as a 500mA current source. Figure 25-2 shows the inductor current (green curve) and DVDD voltage (blue curve) behavior when 3.3V output is suddenly shorted to ground.

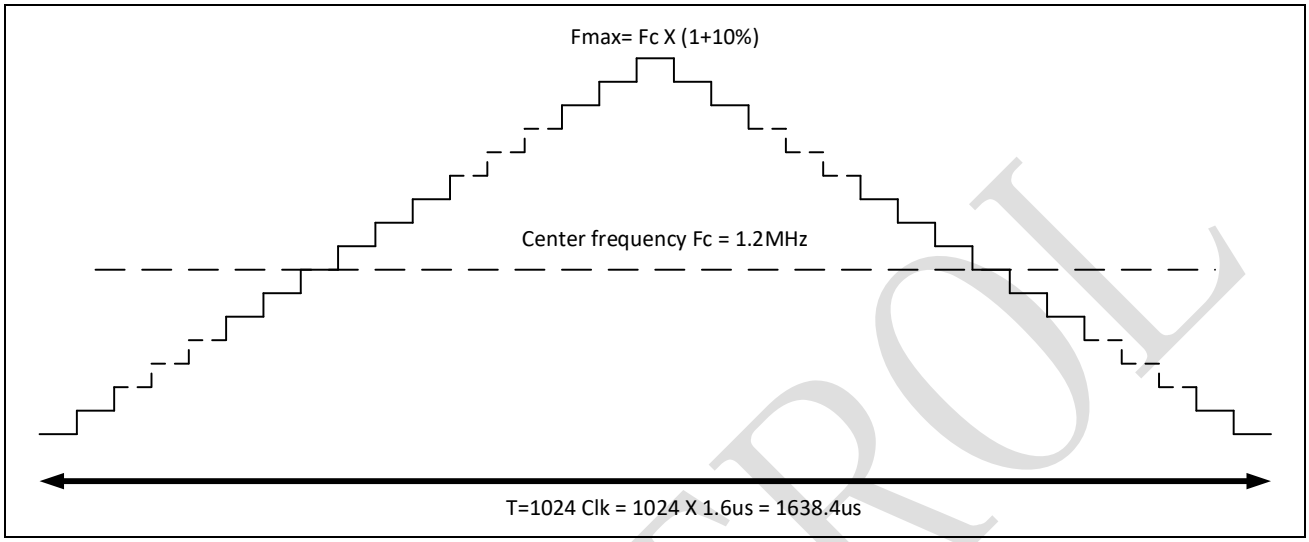
Figure 25-2: Current limit



25.2.4 Frequency dithering option for EMI consideration

In order to reduce EMI, switching frequency dithering option is added for Buck DC-DC. Both the amplitude and period of the frequency change can be programed according to Table 24-46.

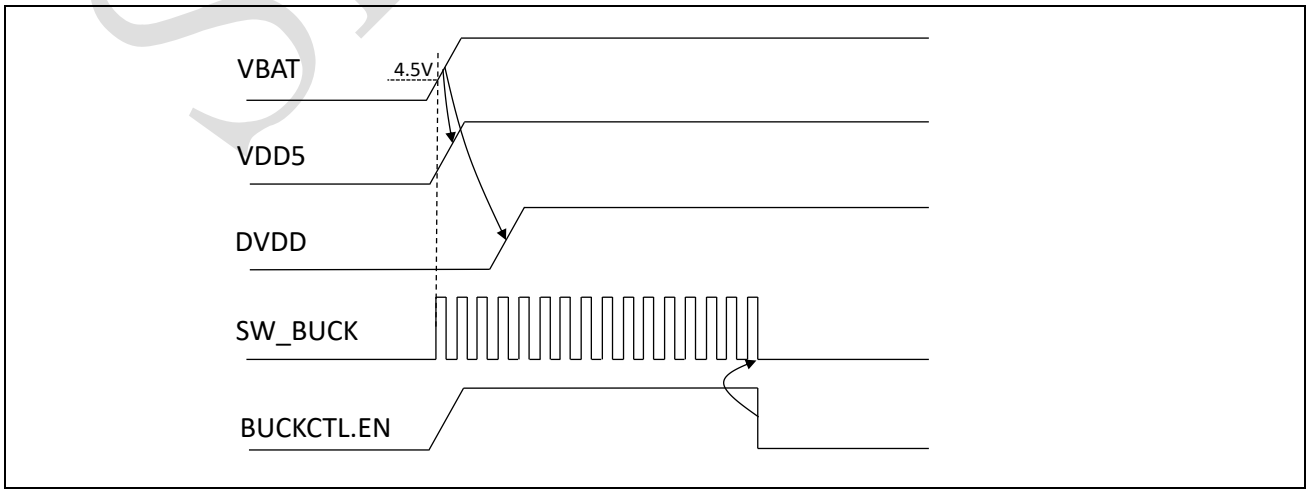
Figure 25-3: Frequency dithering behavior in BUCK DC-DC



25.2.5 Using SPD1148 with external 3.3V Buck regulator

In some rare cases, the system has pre-existing 3.3V supply and the on-chip integrated Buck regulator is not needed. SPD1148 can support using external 3.3V supply. First, on the board the Buck inductor shown in Figure 25-1, which is connecting SW_BUCK and DVDD must be removed. The powerup process is outlined in Figure 25-4. First, VBAT comes out, which causes internal power VDD5 to also come out. After VBAT came out, internal Buck operation will cause SW_BUCK pin to toggle between 0 and VBAT level. Nevertheless, SW_BUCK must be disconnected from DVDD so it has no impact on that power. Afterwards, DVDD comes up from external power source. Finally, as MCU powers up, the user must disable Buck by writing 0 to high-voltage register BUCKCTL.EN, as defined in Section 24.6.3 and Section 24.6.4. Disabling Buck will eliminate toggling of SW_BUCK pin.

Figure 25-4: Power up flow in case of external DVDD supply



25.3 Deep sleep mode in VBAT domain

VBAT domain deep sleep mode can be entered by writing the register CTLKEY=10111011. In this mode, all the control blocks will be disabled except for the logic waiting for wake up command. The Buck DC-DC, VDDG LDO, pre-driver of High voltage module will be disabled. The total current consumption from VBAT can be as low as 6uA. Deep sleep mode can be exited by pulling XRSTn pin low. A detailed waveform showing each power rail with deep sleep mode is shown in Figure 26-1.

25.4 High voltage BOD

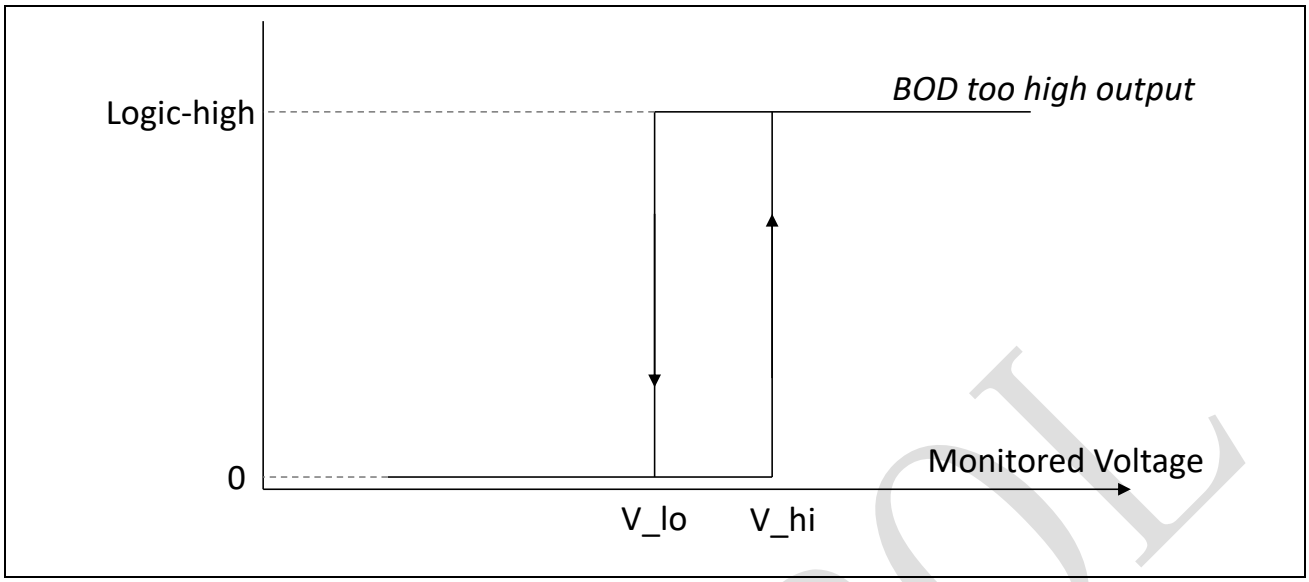
Both VBAT and VDDG voltage is observed by high voltage brown-out-detection circuit. If their value is higher or lower than pre-set value, fault will be decleared. The referece for too high or too low is listed in Table 24-36 and Table 24-38. Local high-voltage over-temperature detection reference levels are described in Table 24-40.

Note that to avoid false triggering due to excessive noise, BOD relies on hysteresis. The concept of hysteresis is demonstrated in Figure 25-5. The BOD logic output reacts to the level of the monitored voltage (in our case, either VBAT or VDDG). If the monitored voltage is larger than the threshold V_hi, then the BOD logic output is logic-high. If the monitored voltage is smaller than V_lo, then the BOD logic output is logic-low. In the transition region, between V_lo and V_hi, the logic output of BOD depends on prior history. If monitored voltage progressed from below V_lo upwards, the output is low transitioning to high at V_hi threshold. If monitored voltage progressed from above V_hi downwards, the output is high transitioning to low at V_lo threshold. The too-high or too-low event of BOD output is latched into GLBSTS0 register described in Table 24-20. Note that during powerup or adjustment of BOD settings, a fake too-high or too-low pulse can be created. Therefore, after the system is powered up and BOD settings are set, the user should use GLBSTSCLR0 register of Table 24-24 to clear GLBSTS0 register before starting to monitor BOD status conveyed in GLBSTS0.

Going back to Figure 25-5, because of hysteresis the BOD output depends on prior history. When BOD is initialized (i.e. at the moment of applying GLBSTSCLR0), if the monitored voltage is either below V_lo or above V_hi, there is no ambiguity in BOD output. However, if BOD is initialized when the monitored voltage is in the hysteresis region (i.e. between V_lo and V_hi), then the BOD output can be either high or low, i.e. it does not represent known state of the monitored voltage. Therefore, if the BOD is used, the user must make sure that the BOD is initialized when the monitored voltage is outside of the hysteresis region.

The signal displayed in Figure 25-5 corresponds to BOD too high output. For BOD too low output, the output curve will look similar, reflected around y-axis, with below V_lo BOD output being logic-high, and above V_hi BOD output equal to 0V.

Figure 25-5: BOD hysteresis illustration



25.5 VBAT and VDDG monitoring by the ADC

Divided versions of VBAT and VDDG can be sent to the MCU via analog test (ATEST) channel for subsequent ADC memasurement, as shown in Figure 24-1. The high-speed VBAT divider is enabled by asserting SYSCTL.VBATDIVEN high (Table 24-42). This will send VBAT/20 voltage to the EPWRATEST channel of the MCU. Sampling time no less than 10us is recommended.

In addition, VMONCNTL register allows to send both VBAT/50 and VDDG/20 to EPWRATEST channel of the MCU, as shown in Table 24-52. Sampling time no less than 1ms is recommended.

Note: In order to avoid signal collision on the EPWRATEST channel, registers SYSCTL.VBATDIVEN and VMONCTL.EN cannot be both high at the same time.

Inside MCU, in order to connect EPWRATEST to main ATEST channel of ADC shown in Figure 12-1, users can call function HV_ConnectAnalogTestToADC().

25.6 Registers

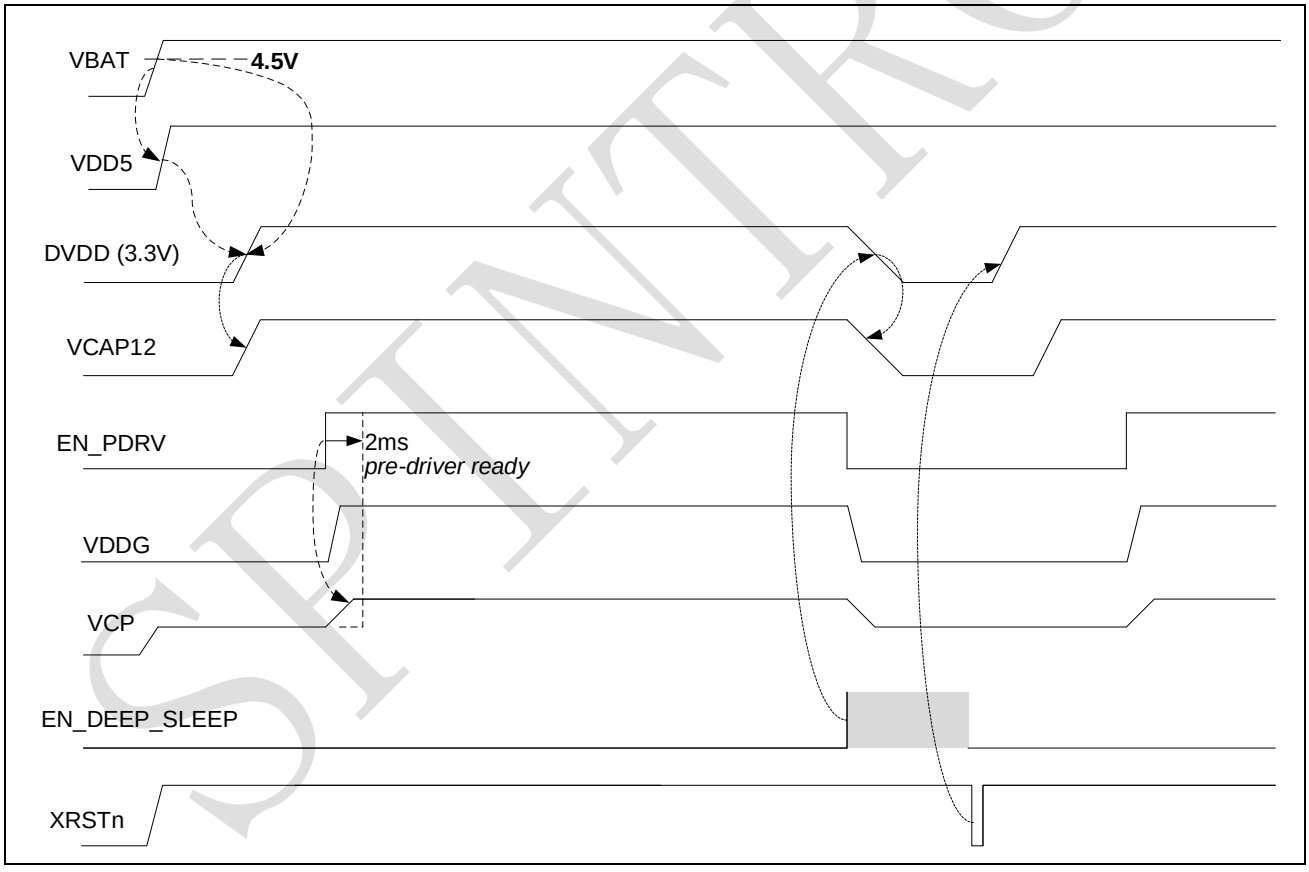
Please see Section 24.6.3 and Section 24.6.4 for register details.

26 Pre-Driver system

26.1 Full system basic power-up with Pre-Driver

The full SPD1148 system power-up scheme is shown in Figure 26-1. It is important to note that the High voltage module is both master and slave in the system: master because it provides power to the whole chip including MCU, slave because it's registers are controlled by MCU action described above. As the master power (VBAT) comes out, internal 5V power domain VDD5 also comes out. As VBAT passes 4.56V level, 3.3V supply of MCU (DVDD) comes out, MCU digital power VCAP12 also comes out after DVDD and MCU operation is enabled. Then pre-driver is enabled by asserting register PDRVCTL.EN. The charge pump producing VCP power is turned on by writing register bit field PDRVCTL.EN. After VCP is ready, which happens within 2ms after enabling, MCU PWM waveform will propagate to respective Pre-Driver outputs as depicted. For example, as shown in Figure 24-1, toggling GPIO28 channel of the MCU will cause toggling of U-phase low-side FET drive signal.

Figure 26-1: Pre-Driver signal power-up scheme



Note: Although the system will block the internal PWM (GPIO) signal from propagating to the Pre-Driver until it is ready for operation, it is recommended for the software program to first enable the special Pre-Driver powers (VCP), and only afterwards proceed to toggle PWMs.

26.2 Pre-Driver features

26.2.1 3-phase low and high side Pre-Drivers

SPD1148 provides three phase low- and high-side power FET drive outputs. Each phase U, V, and W has a corresponding low-side FET gate output OUTL, high-side FET gate output OUTH, high-side return ground and external power FET switching node VPX, and bootstrapped high-side Pre-Driver power supply VBOOT.

Figure 26-2 shows a simplified diagram of one Pre-Driver phase. The input PWM signals are asserted by MCU and enter the non-overlap control (NOV) block. The scheme itself has a built-in self-timed non-overlap at several different levels. First, main Pre-Driver inverter FET's of both high- and low-sides are driven in non-overlap fashion. For example, in Figure 26-2 if Pre-Driver PMOS is not shut down, then Pre-Driver NMOS will not turn on, and vice versa. Second, main power FET's gate-source voltage is measured by a special V_{GS} sensor. If high-side FET is not shut down, main control will not allow low-side FET to turn on regardless of PWM input, and vice versa. In addition to self-timed safety handshaking done by two VGS sensors, the NOV block adds a pre-set intrinsic non-overlap time set by PDRVCTL.NOVWIN register. If the non-overlap built-in by the MCU PWM signals is smaller than the pre-set intrinsic non-overlap, then the outputs of NOV block attain the NOV pre-set non-overlap. Otherwise, outputs of NOV block keeps non-overlap of input PWM control signals. The typical pre-set NOV values are displayed in Table 26-1. The output signals of NOV block must have non-overlap such that external power FETs are not turned on at the same time, in order to avoid large crowbar current from VBAT to ground.

Figure 26-2: Simplified diagram of a phase of the Pre-Driver

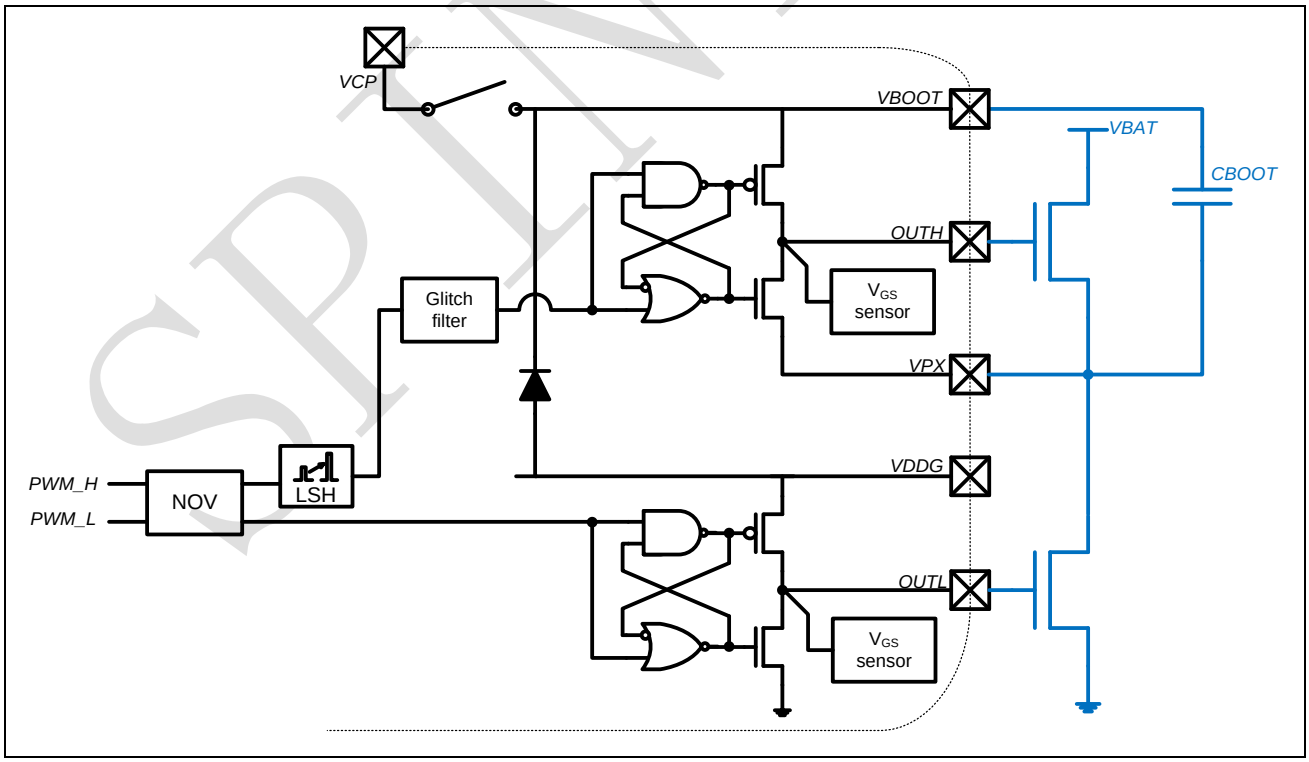
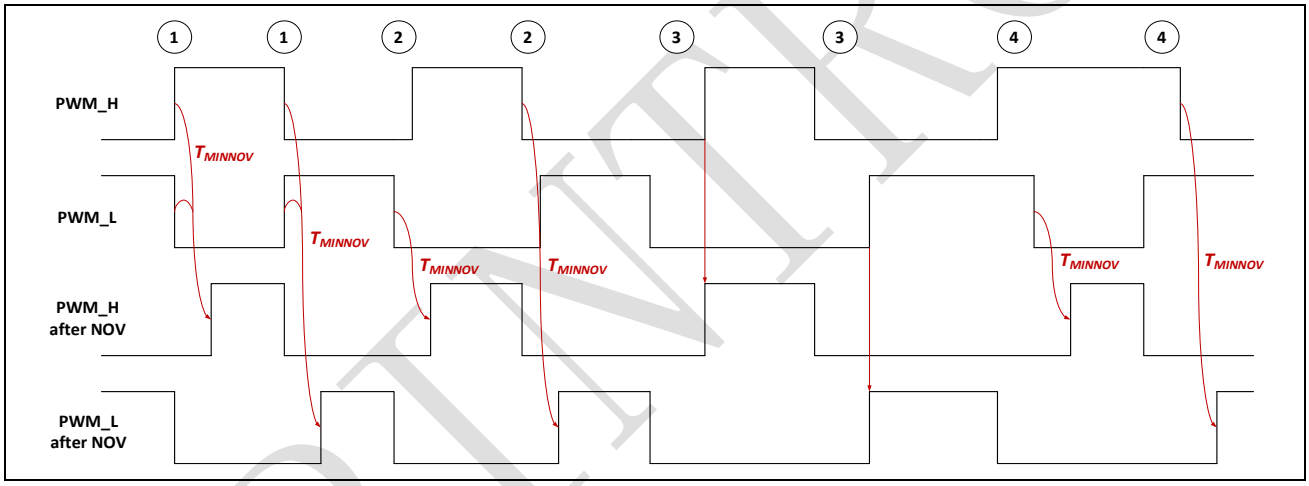


Table 26-1: Intrinsic NOV delay vs. PDRVCTL.NOVWIN register setting

PDRVCTL.NOVWIN	Delay (T_{MINNOV})	Unit
0	48	ns
1	96	ns

The high-side output of NOV block goes through level shifter and glitch filter to drive high-side Pre-Driver, while low-side output of NOV block directly drives low-side Pre-Driver. The operation of NOV block is demonstrated in Figure 26-3. Four cases are shown in connection with PWM_H and PWM_L: case 1 is zero non-overlap, case 2 is non-overlap smaller than minimum non-overlap time T_{MINNOV}, case 3 is non-overlap larger than T_{MINNOV}, case 4 has overlap of input PWM's. As shown in Figure 26-3, in cases 1 and 2, T_{MINNOV} non-overlap is added to outputs of NOV block. In case 3, NOV block does not do anything to the signal. In case 4, NOV block changes PWM's overlap to non-overlap and also adds T_{MINNOV} to the non-overlap time segment.

Figure 26-3: Illustration of NOV block operation



Going back to Figure 26-2, on-chip components are on the left and on-board external components are on the right, while the dotted line shows chip/board boundary. The high-voltage pins of each phase are denoted as OUTL (low-side Pre-Driver output), VPX (high-side floating ground, which is also motor node), OUTH (high-side Pre-Driver output), and VBOOT (high-side floating bootstrapped power rail). Only one phase is shown for brevity, and it is understood that there are three phases (denoted as U/V/W). The internal power rail VDDG is shared between all the phases of the device and can be generated by on-chip buck DC/DC. When the low-side is turned on, PWM_L is high and corresponding low-side on-board power FET (whose gate is driven by OUTL) is turned on. At the same time, PWM_H is low and high-side on-board power FET (whose gate is driven by OUTH) is off. This means that VPX is zero, or close to zero in case if there is a current sensing resistor for brevity not shown in the figure. The voltage across on-board bootstrap capacitor CBOOT charges up close to VDDG level via a bootstrap diode pointing from VDDG to VBOOT. Afterwards, low-side turns off and high-side turns on. VBOOT loaded by CBOOT serves as a power rail for the high-side Pre-Driver, which charges up HO to the level close to VPX+VDDG. As the high-side on-board power FET starts turning on, VPX is charged to VBAT, and VBOOT capacitively reaches the level close to VBAT+VDDG. When the high-side is on, the voltage on VBOOT (voltage across CBOOT) will typically leak out. However,

SPD1148 incorporates charge pump which allows keeping voltage on VBOOT close to VBAT+VDDG indefinitely.

When high-side turns off PWM_H goes low, OUTH is discharged back to VPX level and high-side on-board power FET turns off. The low-side turns on and OUTL reaches level of VDDG, VPX is discharged close to ground again.

After power-up, there is no charge across bootstrap capacitor. To avoid the misoperation, bootstrap capacitor must be charged before high side turns on. The user must first apply low-side PWM pulse to discharge VPX node and charge bootstrap capacitor via bootstrap diode depicted in Figure 26-2. Note that this signal must be applied after VCP comes out in ready state, which is typically within 1ms after enabling of Pre-Driver system. The pulse must be at least 6us long for 100nF bootstrap capacitor, and chosen in proportion to this number for larger values of bootstrap capacitor. Note that as a rule of thumb, VDDG rail bypass capacitor must be much larger than all bootstrap capacitors. This is because during initialization, bootstrap capacitors are charged from the VDDG capacitor, and a large dip in VDDG voltage must be avoided. In general, charging three 100nF bootstrap capacitors with 2.2uF VDDG capacitor is OK. If bootstrap capacitors are increased, then it is recommended to increase VDDG capacitor as well. If VDDG capacitor increase is not desired, the user can assert low-side on signals one-phase-at-a-time, with 50us delay for each assertion. This will allow to initialize each bootstrap capacitor at a time and give sufficient time for VDDG capacitor voltage to recover from a small dip, i.e. one large charge transfer from VDDG capacitor is split into three smaller charge transfers.

Finally, for best operation the choice for CBOOT has to be such that it is at least 10X larger than the capacitance of the on-board power FET. Typically, the value for CBOOT is around 100nF, with the range being from 22nF to 1uF. For proper CBOOT please check parameters of used on-board power FETs.

26.2.2 VDDG LDO and Charge Pump

After Pre-Driver is enabled by writing 0x99 to High voltage module CTLKEY and writing 1 to High voltage module register PDRVCTL.EN, as described in Section 26.1, VDDG LDO and charge pump will be enabled.

VDDG LDO supplies current from input power node VBAT to VDDG, which is pre-driver low-side power rail. If VDDG is used by other on-board circuitry, the allowable current out of VDDG power is 60mA (including pre-driver on-chip self-loading and external load). Care should be taken to protect the chip from damage due to of over-heating because of exceeding VDDG current in case of power FET malfunction or in case of anomalous shorting of VDDG to ground. This is done by choosing one of two events OT0 and OT1 set up in register OTTH, configuring registers PDRVTZ0SEL0.OT1, PDRVTZ0SEL0.OT0, PDRVTZ1SEL0.OT1, PDRVTZ1SEL0.OT0 as corresponding trip-zone events, and configuring registers PDRVCTL.TZ0EN and PDRVCTL.TZ1EN to be able to shut down pre-driver upon over-temperature trip-zone.

The charge pump uses 22nF flying capacitor CF (see Figure 25-1) connected between pins CFBOT and CFTOP, to electrically add VDDG to VBAT. It takes about 2ms for charge pump to come to ready state, as shown in Figure 26-1. The output of charge pump is the power rail VCP. The charge pump 2.2uF bypass capacitor CCP (see Figure 25-1) is connected between VCP and VBAT. It is preferred to connect CCP to VBAT rather than ground, because VCP is actually referred to VBAT. In case of accidental loss of power or exceedingly high power noise, this capacitor will prevent damage of the chip elements connected to VCP.

As can be seen in Figure 26-2, a switch connects charge pump output VCP to VBOOT. Each phase has its charge pump connection switch. The typical switching frequency is 625kHz. The switch is activated a fixed time delay after high side of the phase turns on. The delay should be large enough to ensure that the switching noise dies out, and is controlled by registers PDRVOCCTL.DGWIN and PDRVOCCTL.BLANKWIN. The actual delay time is blanking time plus filtering time (t_blank+t_filter). Those registers are described more in detail in Section 24.6.4.

26.2.3 High voltage supply monitoring

Figure 26-4 shows the power level detector for VBAT and VDDG. The trigger levels are controlled by BODVBATVTH and BODVDDGVTH registers as shown in Figure 26-4. The output of BOD comparator interacts comparator operation in order to build a small hysteresis.

Table 26-2 shows the trigger voltage levels corresponding to particular register settings for VBAT. Table 26-3 shows the trigger voltage levels corresponding to particular 780egister settings for VDDG.

Figure 26-4: Power rail detector for VBAT and VDDG

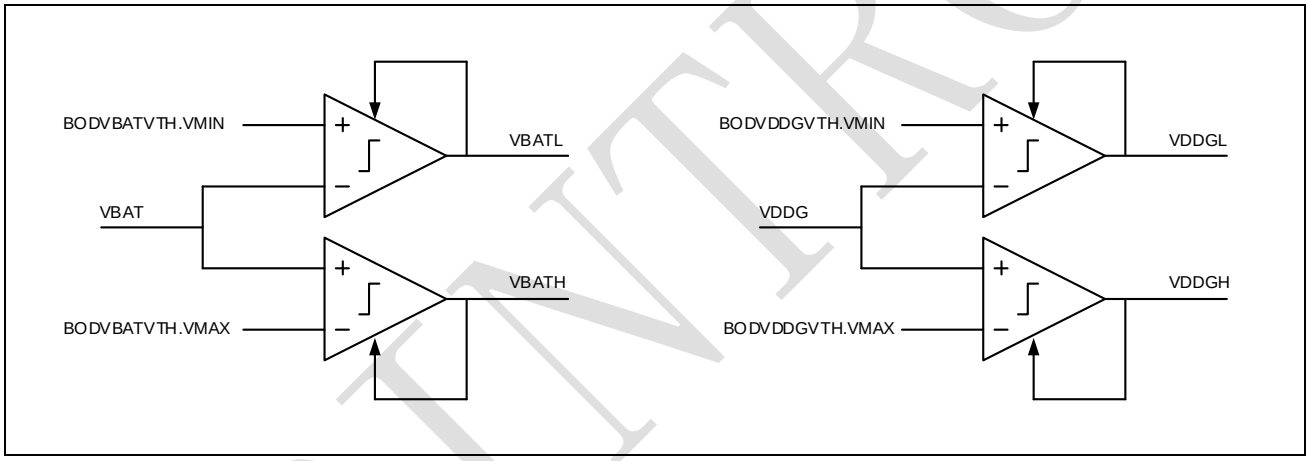


Table 26-2: VBAT monitor toggling levels vs. related register BODVBATVTH

VMAX/VMIN	VBAT_too_high rising	VBAT_too_high falling	VBAT_too_low rising	VBAT_too_low falling	UNIT
0	14.3	13.6	5.0	5.2	V
1	16.8	15.8	6.0	6.3	V
2	20.2	18.8	7.3	7.6	V
3	23.4	21.6	9.5	9.8	V
4	27.8	25.3	12.8	13.5	V
5	30.6	27.6	16.5	17.5	V
6	34.2	30.3	17.5	19.9	V
7	44.4	38.5	23.0	25.0	V

Table 26-3: VDDG monitor toggling levels vs. related register BODVDDGVTH

VMAX/VMIN	VDDG_too_high rising	VDDG_too_high falling	VDDG_too_low rising	VDDG_too_low falling	UNIT
0	14.2	12.6	7.7	8.8	V
1	16.2	14.2	8.0	9.2	V
2	17.3	16.0	8.2	9.5	V
3	17.6	16.3	8.5	9.9	V
4	18.6	17.2	8.8	10.4	V
5	19.0	17.5	9.1	10.8	V
6	19.7	18.2	9.5	11.4	V
7	20.2	18.5	9.9	11.9	V

26.2.4 External FETs Overcurrent Monitoring

SPD1148 has external high-side and low side power FETs' drain-to-source (V_{ds}) voltage monitors and asserts when Vds is higher than a pre-set reference voltage. Using the monitors, over-current condition can be deduced. For external, FET on-resistance R_{ds-on}, FET current is equal to:

$$I = \frac{V_{ds}}{R_{ds-on}}$$

For example, if R_{ds-on} = 100mΩ and V_{ds} exceeds 0.15V, then current is higher than 1.5A. Additionally, high-side V_{ds} sensing is very useful in detecting motor input node (VPX) short to ground. The low-side V_{ds} sensing is very useful in detecting VPX short to VBAT, although this condition can also be detected by MCU overcurrent comparators. The register controlling the V_{ds} sensing threshold is PDRVOCCTL.OCTH.

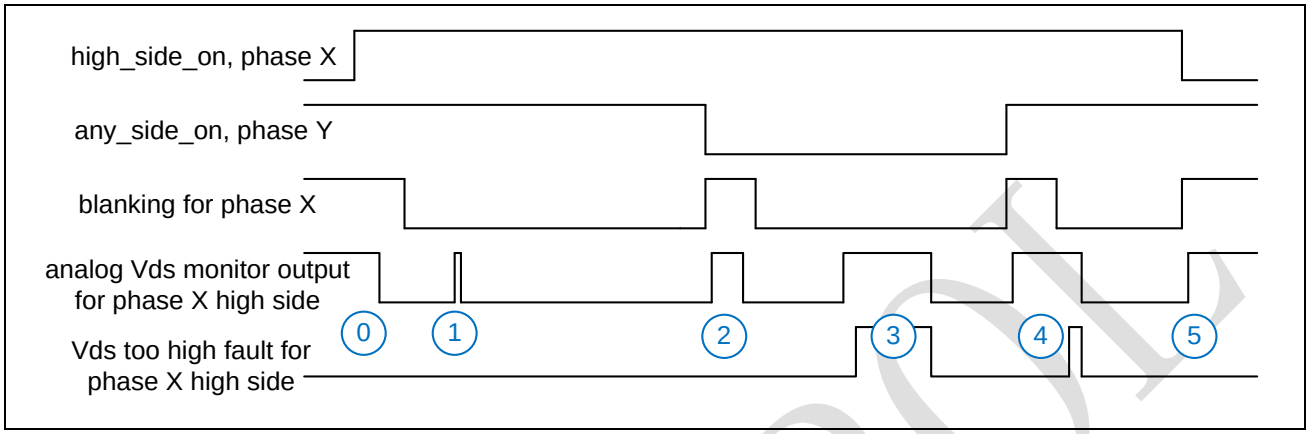
Note:
1. The detector can detect only positive V_{ds} voltage, which corresponds to current supplied to the load by high-side external FET and current supplied to the driver in case of low-side external FET. This also allows to detect VPX short to ground or VBAT.
2. The over-current condition in low-side FET, both positive and negative, can be observed in the current sensing resistor R_{sense} by the over-current monitoring comparators within MCU block.

To prevent erroneous $V_{ds\_too\_high}$ assertions, signals have to be properly blanked. Blanking here means properly masking off analog detector output during times of excessive noise or FET-off. In addition, signals are also filtered using digital glitch filters.

The concept is shown in Figure 26-5, which shows several scenarios numbered from 0 to 5. When internal blanking signal for phase X is asserted high, $V_{ds\_too\_high}$ fault for the phase is not asserted (i.e. it is blanked). The reason for this is because when high side of phase X (X=U/V/W of the motor under consideration) is off, the high side power FET is off and high V_{ds} across it can be a normal condition which does not represent over-current. After high-side PWM on signal, the system waits for the FET to switch and for noise to settle. After time t_blank, blanking signal is de-asserted (case 0). Due to

glitch filtering, if analog V_{ds} monitor detects V_{ds} over-voltage condition which is shorter than filter time t_filter, the fault is not asserted (case 1). On the other hand, if analog V_{ds} monitor detects V_{ds} over-voltage for period longer than t_filter, the fault is asserted (case 3).

Figure 26-5: Blanking diagram of V_{ds} monitors



As can be seen from Figure 26-5, blanking for phase X is also asserted when the other phases are toggling. Case 2 shows that when another phase toggles, it induces noise in phase X and causes analog V_{ds} monitor to erroneously toggle V_{ds} over-voltage. Nevertheless, it is masked by the blanking signal and no fault is issued. In case 4, however, switching noise of phase Y is so high, that analog V_{ds} monitor issues over-voltage for time longer than t_blank+t_filter, where t_filter is the glitch filter time setting. Then, as seen from Figure 26-5, a $V_{ds\_too\_high}$ fault for phase X high side FET is issued. Finally, after high side of phase X is disabled, it is immediately blanked (case 5).

After the V_{ds} overvoltage fault is asserted, it can be passed to MCU by proper configuration of registers PDRVTZ0/1SEL1.OCU/OCV/OCW. Using PWM trip zone, several corresponding or all PWM signals can be immediately shut down within a few MCU clock cycles.

Figure 26-6: Effects of blanking and filtering on fault assertion

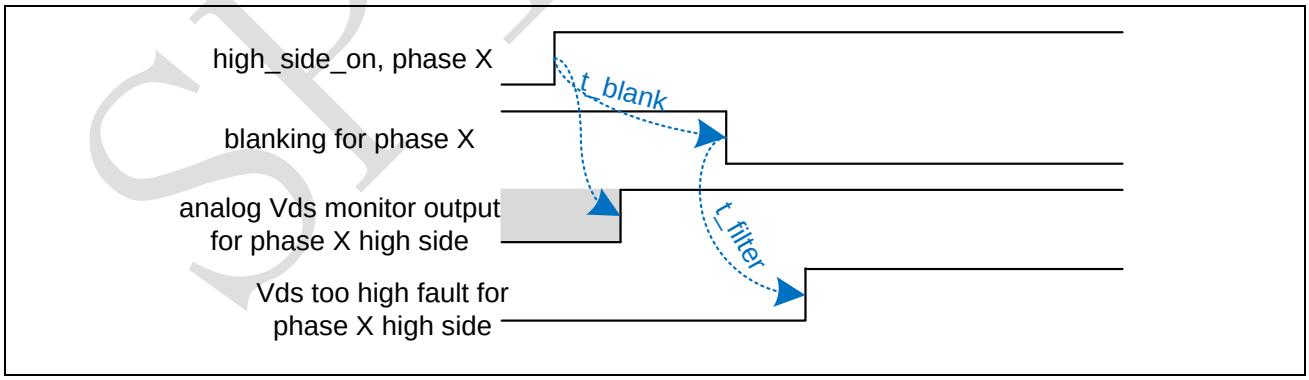


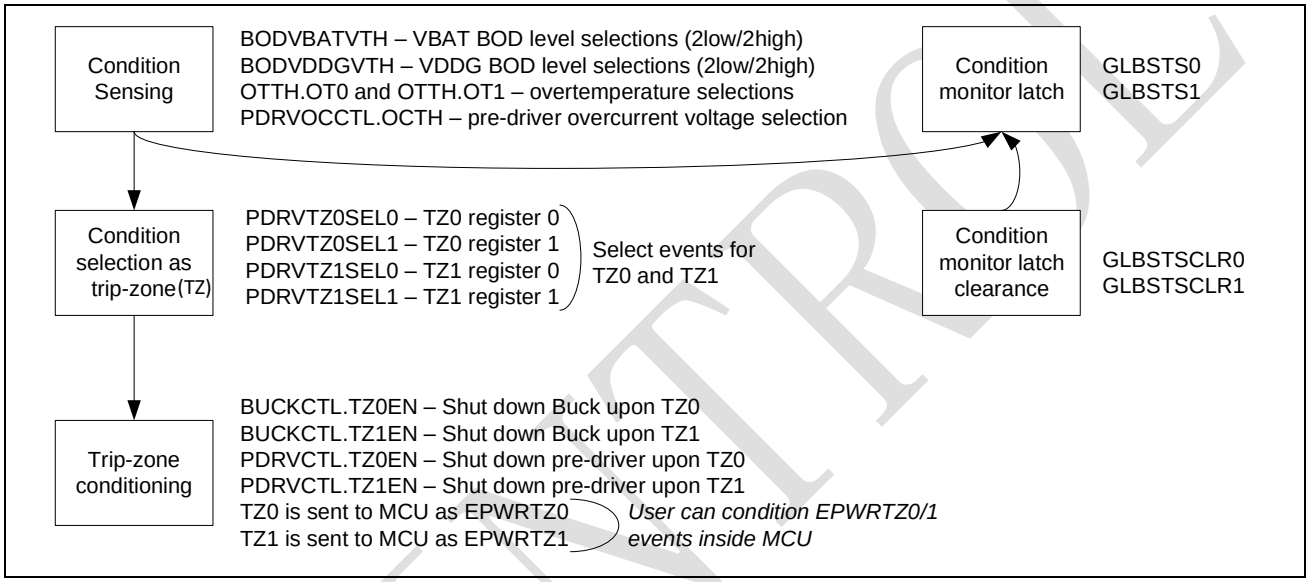
Figure 26-6 demonstrates that there is a fundamental delay from the high side phase-on to fault, equal to t_blank+t_filter. Same concept is applied to low side. Therefore, care should be taken to choose proper values of t_blank and t_filter. If it is too short, then an erroneous fault can be issued due to switching noise of the board components. Yet longer values add to delay of fault assertion. The settings for blanking time t_blank and and deglitch (filter) time t_filter are set by registers PDRVOCCTL.BLANKWIN and PDRVOCCTL.DGWIN.

26.2.5 Safety Conditioning of High-voltage module

The configuration of safety elements of High-voltage module are outlined in Figure 26-7. The setting flow consists of Condition sensing (which can be monitored and the monitor can be cleared), Condition selection as trip-zone, and Trip-zone conditioning.

Condition sensing basically senses parameter such as related voltage or temperature, and determines if there is over- or under- event. Condition selection as a trip zone configures related sensing event as related trip-zone event. Finally, trip-zone conditioning allows system action upon a particular trip-zone event. All the registers are listed in Section 25.6.

Figure 26-7: Safety setting registers for High-voltage module



26.3 Low-voltage Pre-Driver operation

Default Pre-Driver operation is guaranteed down to VBAT=8V approximately. If below-8V operation is necessary, the user has to assert register SYSCTL.VDDGRDYBYP high. The following flow is recommended:

- Enable pre-driver
- Assert SYSCTL.VDDGRDYBYP high
- Wait 200us
- Start toggling Pre-Driver

This way, pre-driver will remain functional down to VBAT=5.5V. **Before disable the Pre-Driver, SYSCTL.VDDGRDYBYP is recommended to be deassert to avoid Pre-Driver false toggling when VDDG is lower than specified operating range.**

In order to further reduce Pre-Driver functionality down to about VBAT=5V, register PDRVCTL.FRCCFM must be asserted high, in addition to asserting SYSCTL.VDDGRDYBYP high. Note that after doing so, the user must make sure that the inputs to High-voltage module from the PWM module have sufficient non-overlap, because automatic non-overlap insertion described in Table

26-1 will be disabled. **To avoid false triggering, it is reommended to set VBAT under voltage protection threshold at 5V above which toggling function is guaranteed**.

26.4 VBAT voltage sampling

SPD1148 provides internal voltage divider resistors for sampling VBAT voltage. The divider resistors are respectively 380Kohm and 20Kohm to realize 1/20 voltage divider of VBAT and send it to the ADC channel. It's enabled by SYSCTL.VBATDIVEN register.

26.5 Registers

Please see Section 24.6.3 and Section 24.6.4 for register details.