IS3XCS9202

Preliminary

32-bit MCU with 256KB ECC E-Flash and 16KB ECC SRAM

GENERAL DESCRIPTION

CS9202 is a general-purpose micro-controller with 256KB embedded flash memory and 8KB data SRAM plus 8KB DMA shared SRAM. All memories have built-in ECC capabilities. The CPU inside the CS9202 is an ARM Cortex-M0+ core. The M0+ includes a standard SWD debug/ISP interface. Both the e-Flash and SRAM have ECC capability for enhanced data integrity. In addition, CS9202 includes two WDT, clock monitoring, low-supply detection, and built-in self-test function for enhanced safety. CS9202 includes an accurate on-chip 16MHz/32MHz oscillator, optional crystal oscillator or external clock input, a constant 128KHz slow oscillator, and ultra-low power RTC (32KHz) crystal oscillator for system clock source. In addition, an on-chip PLL can be used for system clock generation up to 80MHz.

The CS9202 is optimized for use in LED applications especially for automotive lighting scenarios. These include automotive communication interfaces such as CAN FD and LIN bus controller. For LED control, CS9202 includes video serial controller (VSB), and also UART based Lumibus Master controller (LMC). For LED patterns, a dedicated external SPI flash controller (SPF) is also included for fast access of LED configuration and data.

APPLICATIONS

- General purpose automotive MCU
- LED control and lighting control

FEATURES

CPU

- ARM® Cortex™- M0+ core up to 50MHz
  - Single cycle MAC
  - Integrated break point controller and debug port through 2-pin SWI
  - NVIC interrupt controller /w GPIO interrupt

Memory

- 16KB RAM with ECC
  - 8KB for data memory
  - 8KB for DMA memory
- DMA Controller
  - 16 slot round-robin
  - 8-slot allocation for CPU
  - 8-slot allocation for peripherals
- 256KB Flash Memory with four 1KB IFB with ECC
  - Data Flash emulation
  - 1K/10K/100K cycles endurance @ 125/105 /85°C and 10 year retention @125°C

Clock Sources

- IOSC – 16MHz/32MHz +/- 2% /w spread spectrum
- SOSC – 128KHz/256KHz +/- 30%
- XRTC – 32KHz crystal < 1uA
- XOSC – 4MHz/8MHz/12MHz/16MHz
- PLL – 80MHz/64MHz/48MHz/32MHz

Timers

- Watchdog Timers
  - 16-bit Windowed WDT1 (SYSCLK)
  - 16-bit Independent WDT2 (SOSC32K)
- 24-bit SysTick Timer
- T0/T1 16-bit and T2 24-bit Timers

Digital Peripheral (DMA)

- CS/CRC Accelerator
- SPF (SPI Flash) Controller
  - Support single or QPI
  - ECC options
- SPV (Video Serial Bus) Controller
- Lumibus Bus LED Controller
- CAN FD Controller up to 10Mbps

Digital Peripherals

- Two 16-bit Timer/Capture controller
- Quadrature Encoder
- 8CH 16-bit PWM Controller
  - Center aligned
  - ADC trigger and Interrupt
- Buzzer/melody generator
- I^2C Master and I^2C Slave
- EUART1
- EUART2 with LIN 2.0/2.2 Extension
- SPI Master/Slave Controllers

Analog Peripherals

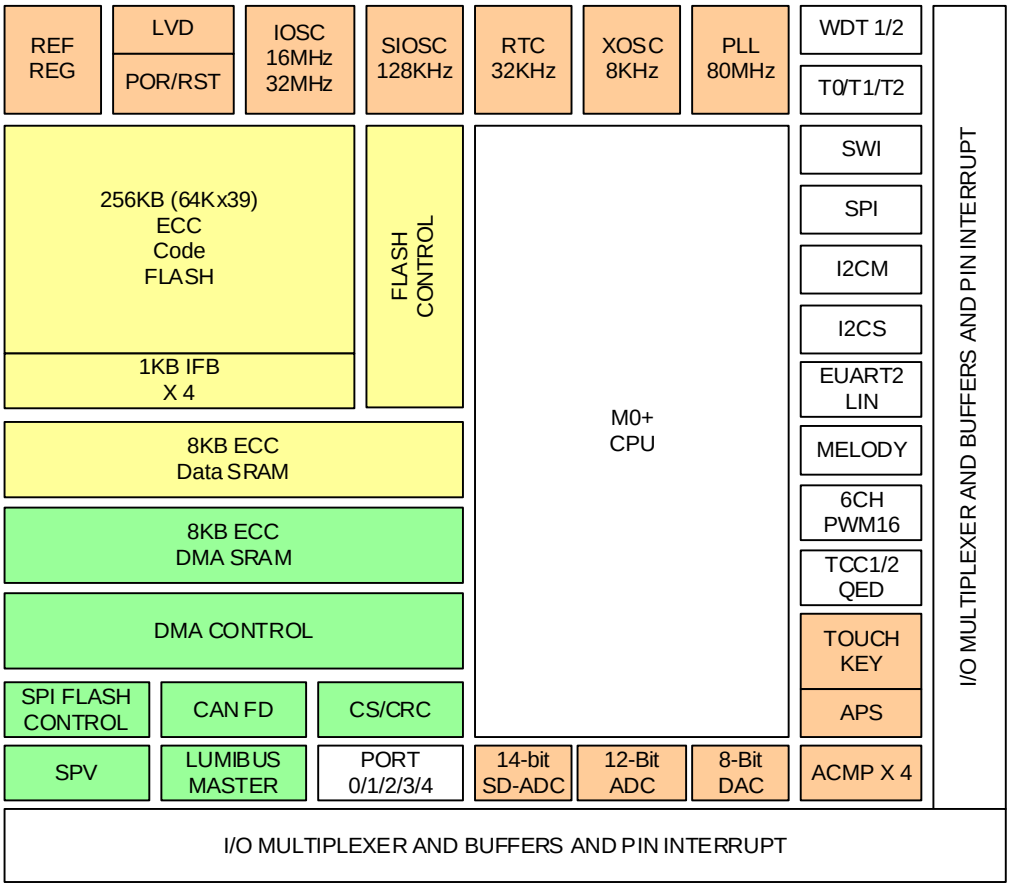
- 4 analog comparators with programmable threshold
- 8-bit DAC
- 14-bit SD Incremental ADC/12-bit SAR ADC
  - T/H front-end
  - ¼ supply and temperature sensor
  - High-Side PGA differential to single-end
- Touch Key Controller – 24-bit resolution
  - Dedicated SHIELD output
  - Pseudo-Random sequence for EMI
  - Up to 25Keys/39Keys (32/48 pins)
- Active Proximity Sensor
- Power on reset (1.5V) and LVD/LVR (2.0V-5.5V)

Miscellaneous

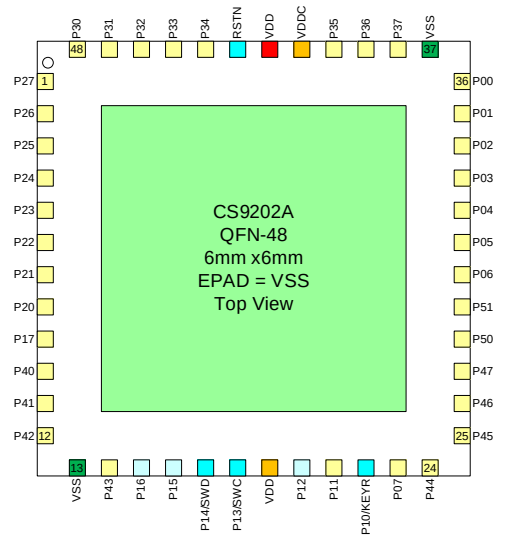
- Up to 42 GPIO pins
  - Configurable drive and Multifunction option
  - Interrupt and wake-up with debounce filter
- 2.5V to 5.5V single supply with on-chip 1.5V regulator.
- Active current < 100uA/MHz in NORMAL mode

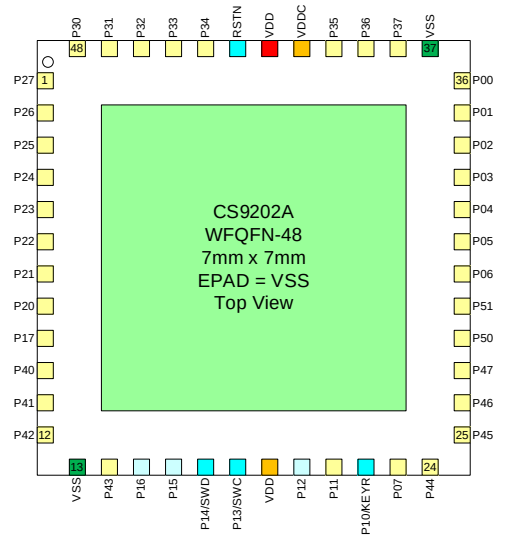
- Low power standby (< 2.5uA) in DEEP SLEEP mode
- IS31CS9202 -40°C to 85°C
- IS32CS9202 -40°C to 125°C
  - AEC-Q100
  - ASIL-B
- RoHS & Halogen-Free compliant package
- Wettable Flank QFN-32 (WFQFN-32), Wettable Flank QFN-48 (WFQFN-48), QFN-48 and LQFP-48

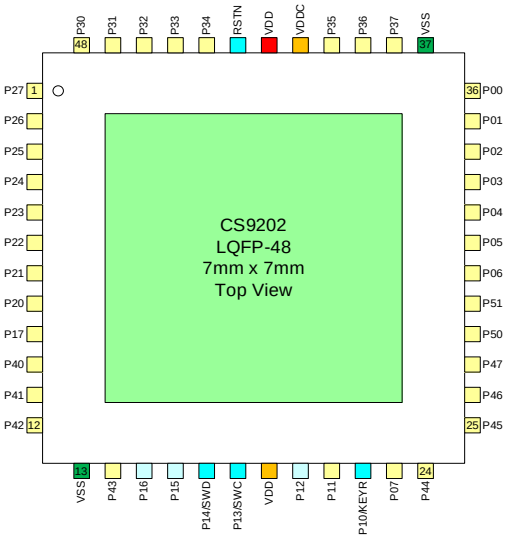
BLOCK DIAGRAM

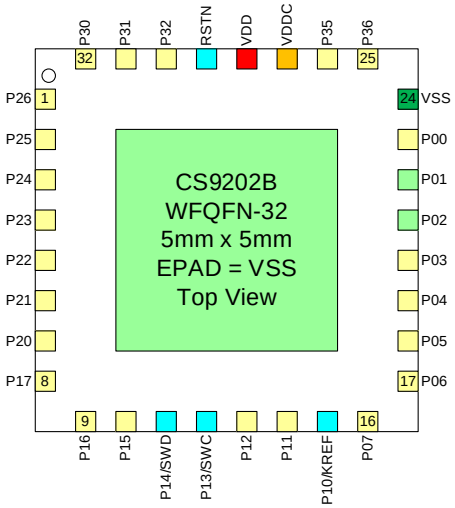


PIN CONNECTION









Note:

1. SWD/SWC are used for ISP/debug. These pins must be bonded out.
2. RSTN/VDD/VDDC/VSS must be bonded out.
3. Each GPIO pin can be multiplexed with multi-functions. The analog functions are fixed.

Table of Contents

List of Tables

List of Figures

List of Registers

1. PIN Description

48PIN	32 PIN	Name	Special Function	ANIO1	ANIO2	Comments
1	-	P27	SPVDI	SHIELD	ADCA	
2	1	P26	SPVCK	SHIELD	ADCB	
3	2	P25	SPVDO	KEY	ADCA	
4	3	P24	SPVLE	KEY	ADCB	
5	4	P23	CANTX	KEY	ADCREF	
6	5	P22	CANRX	KEY	ADCB	
7	6	P21	CANTX	KEY	PLLLPF	
8	7	P20	LMCTX	KEY	ADCB	XIN
9	8	P17	LMCRX	KEY*	DAC	XOUT Do not use for TK key input as XOUT interference.
10	-	P40	LMCRX	KEY	PGAINN	
11	-	P41	LMCTX	KEY	PGAINP	
12	-	P42	LMCRX	KEY	ADCA	
13	-	VSS	-	-	-	
14	-	P43	LMCRX	KEY	ADCB	
15	9	P16	LMCTX	KEY	CMPB	
16	10	P15	LMCRX	KEY	CMPA	
17	11	P14	CANTX	KEY	CMPB	SWD
18	12	P13	CANRX	KEY	CMPC	SWC, Input Only.
19	-	VDD	-	-	-	
20	13	P12	SPVDI	SHIELD	CMPTH	
21	14	P11	SPVLE	SHIELD	CMPD	
22	15	P10	SPVDO	KEYR	KEYR	
23	16	P07	SPVCK	KEY	CMPA	
24	-	P44	SPFCK	KEY	CMPB	
25	-	P45	SPFCS	KEY	CMPC	
26	-	P46	SPFIO0	KEY	RTCXI	
27	-	P47	SPFIO1	KEY	RTCXO	
28	-	P50	SPFIO2	KEY	ADCA	
29	-	P51	SPFIO3	KEY	ADCB	
30	17	P06	SPFIO3	KEY	SHIELD	
31	18	P05	SPFIO2	KEY	SHIELD	
32	19	P04	SPFIO1	KEY	ADCA	
33	20	P03	SPFIO0	KEY	ADCB	
34	21	P02	SPFCS	KEY	PGAINN	
35	22	P01	SPFCK	KEY	PGAINP	
36	23	P00	TESTOUT	KEY	CMPA	
37	24	VSS	-	-	-	

48PIN	32 PIN	Name	Special Function	ANIO1	ANIO2	Comments
38	-	P37	CANTX	KEY	CMPTH	
39	25	P36	CANRX	KEY	CMPB	
40	26	P35	CANTX	KEY	CMPC	
41	27	VDDC	-	-	-	
42	28	VDD	-	-	-	
43	29	RSTN	-	-	-	
44	-	P34	CANRX	KEY	DAC	
45	-	P33	SPVDI	SHIELD	DAC	
46	30	P32	SPVDO	SHIELD	CMPTH	
47	31	P31	SPVCK	KEY	CMPD	
48	32	P30	SPVLE	KEY	DAC	

Table 1-1 PIN Descriptions

2. Memory and Register Map

2.1 AHB Partitions

Module	Size	Base Address	Slave Port	Comment
256K e-Flash	256K	0x0000_0000	AHBS0	
8KB Data SRAM	8KB	0x2000_0000	AHBS1	
8KB DMA SRAM	8KB	0x3000_0000	AHBS2	
APB Bridge	64KB	0x4000_0000	AHBS3	Peripherals' Registers
System Control	4KB	0x5000_0000	AHBS4	
T0/T1/T2	4KB	0x5000_1000	AHBS5	
GPIO	4KB	0x5000_2000	AHBS6	
AHBC/APBC Register	4KB	0x5000_3000	AHBS7	
M0+ Debug	4KB	0x5000_4000	AHBS9	
System ROM	8KB	0xF000_0000	AHBS8	

Table 2-1 AHB Partitions

2.2 APB Partitions

Module	Size	Base Address	Slave Port	Comment
DMA Controller	1KB	0x4000_0000	APBS0	SYSCLK only
e-Flash Controller	1KB	0x4000_1000	APBS1	SYSCLK only
SPF Controller	1KB	0x4000_2000	APBS2	SYSCLK only
SPV Controller	1KB	0x4000_3000	APBS3	SYSCLK only
Lumibus Controller	1KB	0x4000_4000	APBS4	SYSCLK only
CAN FD Controller	1KB	0x4000_5000	APBS5	SYSCLK only
I2CM/I2CS	1KB	0x4000_6000	APBS6	
EUART1/EUART2	1KB	0x4000_7000	APBS7	
SPI	1KB	0x4000_8000	APBS8	
TCC1/TCC2/QEC	1KB	0x4000_9000	APBS9	
PWM	1KB	0x4000_A000	APBS10	
MELODY	1KB	0x4000_B000	APBS11	
TK Controller	1KB	0x4000_C000	APBS12	
DAC/ACMP	1KB	0x4000_D000	APBS13	
ADC	1KB	0x4000_E000	APBS14	
Essential Analog	4KB	0x4000_F000	APBS15	
Reserved	1KB	0x4001_0000	APBS16	Not implemented
Reserved	1KB	0x4001_1000	APBS17	Not implemented
Reserved	1KB	0x4001_2000	APBS18	Not implemented
Reserved	1KB	0x4001_3000	APBS19	Not implemented
Reserved	1KB	0x4001_4000	APBS20	Not implemented
Reserved	1KB	0x4001_5000	APBS21	Not implemented
Reserved	1KB	0x4001_6000	APBS22	Not implemented
Reserved	1KB	0x4001_7000	APBS23	Not implemented
Reserved	1KB	0x4001_8000	APBS24	Not implemented

Module	Size	Base Address	Slave Port	Comment
Reserved	1KB	0x4001_9000	APBS25	Not implemented
Reserved	1KB	0x4001_A000	APBS26	Not implemented
Reserved	1KB	0x4001_B000	APBS27	Not implemented
Reserved	1KB	0x4001_C000	APBS28	Not implemented
Reserved	1KB	0x4001_D000	APBS29	Not implemented
Reserved	1KB	0x4001_E000	APBS30	Not implemented
Reserved	1KB	0x4001_F000	APBS31	Not implemented

Table 2-2 APB Partitions

3. Interrupt Assignment

Exception #	INT #	Vector*	Type	P*	S*	Description
1	-15	0x04	RESET	-3	A	Reset (including software reset)
2	-14	0x08	NMI	-2	A	
3	-13	0x0C	HARDFAULT	-1	S	
4 – 10	-	-	-	C	S	Reserved
11	-5	0x2C	SVCALL	C	S	
12 – 13	-	-	-	C	A	Reserved
14	-2	0x38	PENDSV	C	A	
15	-1	0x3C	SYSTICK	C	A	
16	0	0x40	LVT	C	A	Supply Low Voltage Detect
17	1	0x44	WDT	C	A	Watchdog Timer
18	2	0x48	AHB	C	A	Memory Access Error
19	3	0x4C	ECC	C	A	ECC Error / APB Fault
20	4	0x50	DMA	C	A	DMA Move
21	5	0x54	CRC	C	A	DMA CRC
22	6	0x58	T0	C	A	Timer 0
23	7	0x5C	T1	C	A	Timer 1
24	8	0x60	T2	C	A	Timer 2
25	9	0x64	GDB0	C	A	GPIO Debug
26	10	0x68	GDB1	C	A	GPIO Debug
27	11	0x6C	GPIO0	C	A	GPIO
28	12	0x70	GPIO1	C	A	GPIO
29	13	0x74	SPF	C	A	SPI Flash Controller
30	14	0x78	SPV	C	A	SPV Controller
31	15	0x7C	LMC	C	A	Lumibus Controller
32	16	0x80	CANFD	C	A	CAN FD Controller
33	17	0x84	I2CS	C	A	I^2C Slave
34	18	0x88	I2CM	C	A	I^2C Master
35	19	0x8C	SPI	C	A	SPI Controller
36	20	0x90	PWM	C	A	PWM Controller
37	21	0x94	TCC1	C	A	Timer/Counter/Capture Unit 1
38	22	0x98	TCC2	C	A	Timer/Counter/Capture Unit 2
39	23	0x9C	QEC	C	A	Quadrature Encoder
40	24	0xA0	EUART1	C	A	EUART 1
41	25	0xA4	EUART2/LIN	C	A	EUART 2
42	26	0xA8	ACMP	C	A	Analog Comparator
43	27	0xAC	ADC	C	A	ADC
44	28	0xB0	TK	C	A	Touch Key Controller
45	29	0xB4	BZ	C	A	Buzzer Melody Controller
46	30	0xB8	RTC	C	A	Real Timer Clock

Exception #	INT #	Vector*	Type	P*	S*	Description
47	31	0xBC	-	C	A	Reserved

Table 3-1 Interrupt Assignment

Note:

1. Interrupt vector is not relocatable.
2. S/A indicates synchronous or asynchronous.
3. C indicates priority is configurable.

4. CPU and Bus Control

The CPU is ARM Cortex M0+, which implements the ARMv6-M architecture. M0+ is a two-stage pipeline implementation including fetch/decode/execute. 56 instructions are implemented based on Thumb ISA. The CPU includes a hardware single-cycle 32-bit multiplier, nest vector interrupt controller (NVIC) along with wakeup interrupt controller (WIC), a 32-bit SYSTICK timer, and serial-wire debug interface (SWD) using two wire interfaces (SWD/SWC) for debug and programming. For more detailed descriptions of M0+, please refer to ARM M0+ technical reference manual.

The CPU connects to external memory and peripherals through AHB-LITE bus. Memory and fast peripherals such as I/O control are connected to AHB-LITE directly. To reduce AHB-LITE bus loading, a AHB to APB bridge, which buffers the access to APB bus is used to connect other peripherals' control registers.

The AHB Controller also implements a time-out mechanism that prevent a master take control of the bus but keep idle. Lastly, the AHB Controller monitors the address of the access, if a non-allocated address is attempted, a bus error is detected. This will cause a hardware fault to the master and usually leads to a hardware reset. Alternatively, AHB Controller can generate an interrupt to the CPU for this fault condition. These configurations are done through the AHB Controller registers.

AHBIEN (0x5000_3000) AHB Controller Interrupt Configuration Register RW (0x00)

	31-5	4	3	2	1	0
RD	-	RSPEN	IEN	TOERR	MEMERR	INTF
WR	-	RSPEN	IEN	-	-	INTF

RSPEN — Response Control
RSPEN setting controls the error response to the bus master when non-allocated address access occurs.
RSPEN=0 always responds without error.
RSPEN=1 responds error to master.

IEN — Interrupt Enable
IEN=1 enables interrupt
IEN=0 disables the interrupt

TOERR

MEMERR

INTF — Interrupt Flag
INTF is set to 1 by non-allocated address access or idle time out. INTF is set independent of IEN setting.
INTF must be cleared by software by writing 1.

AHBIDL (0x5000_3004) AHB Time Out Configuration Register RW (0x01)

	31-16	15-6	5-0
RD	-	-	IDLTO[5-0]
WR	-	-	IDLTO[5-0]

IDLTO[5-0] — Idle Time Out
IDLTO[5-0] defines the number of AHB clock cycles allows for the master time out.
IDLTO[5-0] cannot be written to all 0. Writing all 0 to IDLTO[5-0] results IDLTO[5-0]=6b'000001.

The APB bridge separates the high speed AHB bus from the peripheral bus. The peripheral bus is allocated with memory address of 0x4000_0000 to 0x4000_FFFF. APB bus is used for register accesses of peripherals. APB controller can allow up to 32 memory partitions in this range but there are only 16 used in the CS9202. Please note for APB bus that if an un-allocated address access occurs, it does not generate any error. The control registers of APB bridge are located in 0x5000_3000 range.

APBSCS0 (0x5000_3100) APBS0 Clock Setting Register RW (0x01)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] — Clock Setting
00=SYSCLK/1

01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN — Clock Enable
0=Clock off
1=Clock on.

APBS0 is for DMA Controller. It is SYSCLK only.

APBSCS1 (0x5000_3101) APBS1 Clock Setting Register RW (0x01)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] — Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN — Clock Enable
0=Clock off
1=Clock on.

APBS1 is for e-Flash Controller. It is SYSCLK only.

APBSCS2 (0x5000_3102) APBS2 Clock Setting Register RW (0x01)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] — Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN — Clock Enable
0=Clock off
1=Clock on.

APBS2 is for SPF Controller. It is SYSCLK only.

APBSCS3 (0x5000_3103) APBS3 Clock Setting Register RW (0x01)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] — Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN — Clock Enable
0=Clock off
1=Clock on.

APBS3 is for SPV Controller. It is SYSCLK only.

APBSCS4 (0x5000_3104) APBS4 Clock Setting Register RW (0x01)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8
CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS4 is for LMC Controller. It is SYSCLK only.

APBSCS5 (0x5000_3105) APBS5 Clock Setting Register RW (0x01)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8
CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS5 is for CAN FD Controller. It is SYSCLK only.

APBSCS6 (0x5000_3106) APBS6 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8
CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS6 is for I2CM/I2CS Controller.

APBSCS7 (0x5000_3107) APBS7 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8
CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS7 is for EUART1 and EUART2/LIN Controller.

APBSCS8 (0x5000_3108) APBS8 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS8 is for SPI Controller.

APBSCS9 (0x5000_3109) APBS9 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS9 is for TCC1/TCC2/QEC Controller.

APBSCS10 (0x5000_310A) APBS10 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS10 is for PWM Controller.

APBSCS11 (0x5000_310B) APBS11 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN Clock Enable
0=Clock off

1=Clock on.

APBS11 is for Melody Controller.

APBSCS12 (0x5000_310C) APBS12 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS12 is for Touch Key Controller.

APBSCS13 (0x5000_310D) APBS13 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS13 is for DAC/ACMP.

APBSCS14 (0x5000_310E) APBS14 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4
11=SYSCLK/8

CLKEN Clock Enable
0=Clock off
1=Clock on.

APBS14 is for ADC Controller.

APBSCS15 (0x5000_310F) APBS15 Clock Setting Register RW (0x31)

	7-6	5-4	3-1	0
RD	-	CS[1-0]	-	CLKEN
WR	-	CS[1-0]	-	CLKEN

CS[1-0] Clock Setting
00=SYSCLK/1
01=SYSCLK/2
10=SYSCLK/4

	11=SYSCLK/8
CLKEN	Clock Enable 0=Clock off 1=Clock on.

APBS15 is for Essential Analog peripherals.

There are up to 32 APB peripheral partitions as shown in the APB partition section. Only portions are used in CS9202. If there is clock or bus command faults, the bus transaction may still complete however the transaction result is erroneous. In this can a fault interrupt (INT #3) can be generated for software corrections.

APBIEN (0x5000_3300) APB Fault Interrupt Enable Register RW (0x0000)

	31-16	15-0
RD	-	APBIEN[15-0]
WR	-	APBIEN[15-0]

APBIEN[15-0]	APB Fault Interrupt Enable APBIEN[i] controls the APB fault interrupt enable for the corresponding APB partition. When APB command fault occurs, it generates an interrupt (INT #3). For example, APBIEN[10]=1 will enable APBS10 (PWM Controller) fault interrupt. APBIEN[10]=0 will turns off the interrupt.

APBINTF (0x5000_3304) APB Fault Interrupt Flag Register RW (0x0000)

	31-16	15-0
RD	-	APBINTF[15-0]
WR	-	APBINTF[15-0]

APBINTF15-0]	APB Fault Interrupt Flag APBINTF[i] is set to 1 when a corresponding APB partition has fault and its APBIEN is enabled. APBINTF[i] is cleared by writing “1”. For example, if APBIEN[10]=1, and APBS10 has APB command or bus fault, APBINTF[10] is set to 1 by hardware. To clear APBINTF[10], write 1 into APBINTF[10].

5. System Control

5.1 Clock Control

CKSEL (0x5000_0000) System Clock Selection Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-					CKSEL[2-0]		
WR	-					CKSEL[2-0]		

CKSEL[2-0] SYSCLK Selection
000 = IOSC post divider output
001 = SOSC
010 = RTC
011 = External Clock
100 = PLL
101 = PLL/2
110 = IOSC
111 = IOSC

WTST (0x5000_0004) E-Flash Wait State Controlled Register R/W (0x07)

	7	6	5	4	3	2	1	0
RD	SRAMINS	-	-	IBUSTM	-	WTST[2-0]		
WR	SRAMINS	-	-	IBUSTM	-	WTST[2-0]		

SRAMINS SRAM Instruction Mode
SRAMINS=1 allows the instruction to be fetched from SRAM.
SRAMINS=0 only allows the instruction from e-Flash.
Please note default only allows instruction to be fetched from e-Flash. To run instruction from SRAM, SRAMINS must be set to 1. Since there are cases in debug mode, execution code are downloaded into SRAM then be executed, therefore it is recommended SRAMINS is set to 1 in debug mode. SRAMINS=1 condition will impact normal system performance.

IBUSTM Instruction Bus Timing
IBUSTM=0 uses address latch for e-Flash instruction access with no delay on AE. This is default.
IBUSTM=1 no address latch for e-Flash instruction access with AE 0.5T delay.
IBUSTM can only be modified while WTST=7.

WTST Main Flash Access time is defined (WTST+1) SYSCLK.
WTST=0 is ignored and treated the same as WTST=1, in other words, minimum wait state is 1.
WTST=1, the effective instruction fetch from e-Flash takes three cycles, one cycle for address latch and two cycles for e-Flash access. Therefore, WTST=1 allows up to 32MHz SYSCLK assuming 40ns e-Flash access time.

5.2 Sleep Mode

M0+CPU enters sleep mode after WFI (Wait for Interrupt) instruction. The sleep mode behavior is determined by bits 3, 2, 1 in SCR register of M0+. SYSCLK is always off in sleep mode, and SOSC128K is always on. SCR[1] is SLEEPONEXIT which indicates sleep-on-exit when returning from handler mode to thread mode as described by Arm. SCR[2] is SLEEPDEEP and SCR[3] is SLEEPLIGHT which control low power sleep mode behavior as defined in the following table. Compared with sleep mode, light sleep allows fast wakeup by maintaining internal regulator always on.

	SCR register		Behavior in Sleep mode (after WFI)		
MODE	**SCR[2]**	**SCR[3]**	**Regulator**	**IOSC**	**Wakeup Sources**
Normal	0	X	ON	ON	All except AHB bus IPs, such as CAN, SPV, SPF, CRC, EUART1, Memory-Move.
Deep Sleep	1	0	OFF	OFF	TK, LIN, I2CS, T2(SOSC128K), RTC, GPIO

	SCR register		Behavior in Sleep mode (after WFI)		
MODE	**SCR[2]**	**SCR[3]**	**Regulator**	**IOSC**	**Wakeup Sources**
Light Sleep	1	1	ON	OFF	TK, LIN, I2CS, T2(SOSC128K), RTC, GPIO

5.3 Reset

There are several reset sources including both software resets and hardware resets. Software resets include software command reset, WDT reset and reset issued through debug port. Hardware resets include power-on reset (low voltage detect on VDDC), LVD reset (low voltage detect on VDD), SYSCLK clock monitor reset, and external RSTN reset. Power on reset and LVD reset as well as clock monitor reset are also routed to RSTN pin to allow external RC extension. Both hardware and software reset are combined to generate e-Flash reset and system reset including CPU with restoring all registers to their default values.

Power-on reset clears all registers to default values. REGTRM, LVDCFG, LVDTHD and LVDHYS registers are cleared to default values only by power-on reset.

Extended RSTN reset (SP reset by RSTN=0 longer than 250msec) is also used for clearing special modes. Some of the registers are restored only by SP reset. These include RSTCMD, BSTCMD, , and RTCCFGA, JTAGCFG etc. and are described in their register description. In addition, the debug port logic, and its associated state and registers is reset only by SP reset and not by SYSTEM reset. This behavior is shown in the following Block Diagram.

The total delay in system startup at power-on is approximately 3msec.

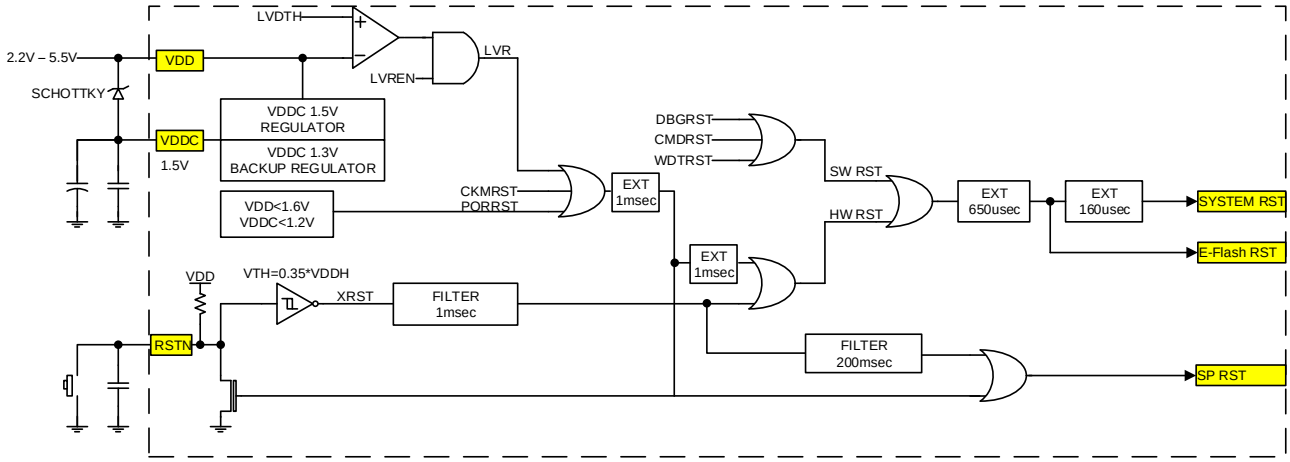


Figure 5-1 Reset Block Diagram

RSTCMD (0X5000_0008) Reset Command Register R/W (0x80)

	7	6	5	4	3	2	1	0
RD	RSTCKM	RSTNFLTEN	-	LVRF	CKMRF	WDT2RF	WDT1RF	CMDRF
WR	RSTCKM	RSTNFLTEN	CLRF	-	RSTCMD[3-0]			

RSTCKM — Reset Enable for Clock Monitor Fault
RENCKM=1 enables reset after clock fault detection. RSTCKM is cleared to 0 after any reset. Default RSTCKM is 0.

RSTNFLTEN — RSTN Filter Enable
RSTNFLTEN = 1 enables a noise filter on the RSTN circuits. The filter is set at around 30usec

CLRF — Clear Reset Flag
Writing 1 to CLRF will clear LVRF, CKMRF, WDT2RF, WDT1RFand CMDRF. It is self-cleared.

LVRF — Low voltage Reset Flag
LVRF is set to 1 by hardware when LVD caused a reset

CKMRF — Clock Monitor Fault Reset Flag
CKMRF is set to 1 by hardware when a clock fault reset has occurred. CKMRF is not cleared by reset except power-on reset.

WDT2RF — WDT2 Reset Flag

WDT2RF is set to 1 by hardware when WDT2 caused a reset.

WDT1RF — WDT1 Reset Flag
WDT1RF is set to 1 by hardware when WDT1 caused a reset.

CMDRF — RST Command Reset Flag
CMDRF is set to 1 by hardware when reset is caused by issuing software reset command.

RSTCMD[3-0] — Software Reset Command
Writing RSTCMD[3-0] with consecutive 4b'0101, 4b'1010 sequences will cause a software reset. Any other value will clear the sequence state. These bits are write-only and self-cleared.

Please note all flags in this register are cleared by either CLRF or Power-On-Reset (or its equivalent).

5.4 Built-in Self Test

BSTCMD (0x5000_000C) SRAM Built-In and Logic Self-Test R/W (0x00) AHB

	7	6	5	4	3	2	1	0
RD	MODE[3-0]				BST	-	FAIL	FINISH
WR	MODE[3-0]				BSTCMD[3-0]			

MODE[3-0] — BIST Mode Selection
0000 – Normal Mode
0001 – Data SRAM MBIST
0010 – DMA SRAM MBIST
0011 – Reserved
0100 – Register LBIST
0101 – Reserved
0110 – Reserved
0111 – Reserved
1000 – Normal Mode
1001 – SRAM MBIST and monitor on pins
1010 – Reserved
1011 – Reserved
1100 – Register LBIST and monitor on pins
1101 – Reserved
1110 – Reserved
1111 – Reserved
Please note MODE[3-0] is cleared only by POR and RSTN. Software can read this setting along with the Pass/Fail status to determine which BIST was performed and its result after a software reset.
If a "Reserved" mode command is issued, neither FINISH or FAIL will be set to 1. It is recommended that software should check BST=0 or FINISH=1 to determine if a command is finished or not.

BST — BIST Status
BST is set to 1 by hardware when BIST in ongoing.

FAIL — BIST Test Fail Flag
FAIL is set to 1 by hardware when BIST error has occurred. FAIL is cleared to 0 by hardware when a new BIST command is issued.

FINISH — BIST Completion Flag
FINISH is set to 1 by hardware when BIST controller finishes the test. FINISH is cleared to 0 by hardware when a new BIST command is issued.

BSTCMD[3-0] — Memory BIST Command
Writing BSTCMD[3-0] with value 4b'0101 causes the BIST controller to perform BIST. Writing BSTCMD[3-0] with value 4b'1010 causes the BIST controller to perform BIST, and after BIST is completed, it automatically generates a software reset.
Writing BSTCMD[3-0] with value 4b'0000 causes FAIL and FINISH bits to be cleared to 0.
Any other value will either have no effect or abort any ongoing BIST.

Please note after the BSTCMD is issued, CPU is paused until BIST is completed and any BIST operations will

cause the state of the CPU to be undefined, and the content of the SRAM undefined. Therefore, it is highly recommended that a software reset or initiation should be performed after any BIST operation. Please also note MODE[3-0], FINISH, FAIL bits are not cleared by software resets.

5.5 Watchdog Timers

WDT1 is a 16-bit windowed watchdog timer clocked by SYSCLK. Please note WDT1 is only stopped during sleep mode as SYSCLK is stopped. And WDT1REN default after powered on is disabled.

WDT1CF (0x5000_0010) WDT1 Configure Registers R/W (0x8E)

	7	6	5	4	3	2	1	0
RD	-	WDT1REN	-	WDT1IEN	WDT1CS[2-0]			WDT1IF
WR	WDT1CLR	WDT1REN	-	WDT1IEN	WDT1CS[2-0]			WDT1IF

WDT1CLR — WDT1 Counter Clear
Writing “1” to WDT1CLR clears the WDT1 count to 0. It is self-cleared by hardware.

WDT1REN — WDT1 Reset Enable
WDT1REN=1 configures WDT1 to perform software reset.

WDT1IEN — WDT1 Interrupt Enable
WDT1IEN=1 enables WDT1 interrupt.

WDT1CS[2-0] — WDT1 Clock Scaling

WDT1CS[2-0]	SYSCLK DIVIDER
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

WDT1IF — WDT1 Interrupt Flag
WDT1IF is set to “1” if WDT1IEN is set as 1 and WDT1 time out or WDT1CLR is issued while the WDT1CNT is less than WDT1MN. This must be cleared by software by writing “1”.

WDT1CNT (0x5000_0014) WDT1 Current Count RO (0x0000)

	31-16	15-0
RD	-	WDT1CNT[15-0]
WR	-	-

WDT1TOV (0x5000_0018) WDT1 Time Out Value Register RW (0xFFFF)

	31-16	15-0
RD	-	WDT1TOV[15-0]
WR	-	WDT1TOV[15-0]

WDT1TMN (0x5000_001C) WDT1 Minimum Value Register RW (0xFFFF)

	31-16	15-0
RD	-	WDT1TMN[15-0]
WR	-	WDT1TMN[15-0]

WDT2 is a 16-bit independent watchdog timer clocked by 32KHz from the non-stop SOSC. Please note WDT2 is always enabled and cannot be turned off. The default time out of WDT2 is 2 seconds and the maximum time out value is roughly 9 hours.

WDT2CF (0x5000_0020) WDT2 Configure Registers R/W (0xD0)

	7	6	5	4	3	2	1	0
RD	-	DBGDIS[2-0]			WDT2CS[2-0]			WDT2RF
WR	WDT2CLR	DBGDIS[2-0]			WDT2CS[2-0]			

WDT2CLR — WDT2 Counter Clear
Writing “1” to WDT2CLR clears the WDT2 count to 0. It is self-cleared by hardware.

DBGDIS[2-0] — Debug Mode Disable Enable
If DBGDIS[2-0] = 101, WDT2 counting is stopped in JTAG or CJTAG debug mode.
If DBGDIS[2-0] is not “101”, WDT2 function is not affected in debug mode.
Please note DBGDIS[2-0] defaults to 101. The software should take care of WDT2 counting when entering and exiting the debug mode. Also DBGDIS[2-0] does not affect WDT2 behavior in the non-debug modes, in other words, WDT2 is always enabled and continue counting during non-debug modes.

WDT2CS[2-0] — WDT2 Clock Scaling

WDT2CS[2-0]	SOSC DIVIDER	Default (Max) Duration
000	1	2 second
001	4	8 second
010	16	3 second
011	64	131 second
100	256	524 second
101	1024	2097 second
110	4096	8398 second
111	16384	33554 second

WDT2RF — WDT2 Reset Flag
WDT2RF is set to 1 by hardware when WDT2 caused a reset. This bit also shown on bit 2 of RSTCMD and clear by write CLRF as 1 of RSTCMD or power on reset

WDT2CNT (0x5000_0024) WDT2 Current Count RO (0x0000)

	31-16	15-0
RD	-	WDT2CNT[15-0]
WR	-	-

WDT2TOV (0x5000_0028) WDT2 Time Out Value Register RW (0xFFFF)

	31-16	15-0
RD	-	WDT2TOV[15-0]
WR	-	WDT2TOV[15-0]

5.6 Clock Fault Monitoring

SYSCLK in normal running mode is monitored by SOSC (128KHz). If SYSCLK is not present in normal mode for four SOSC cycles, a hardware reset is triggered.

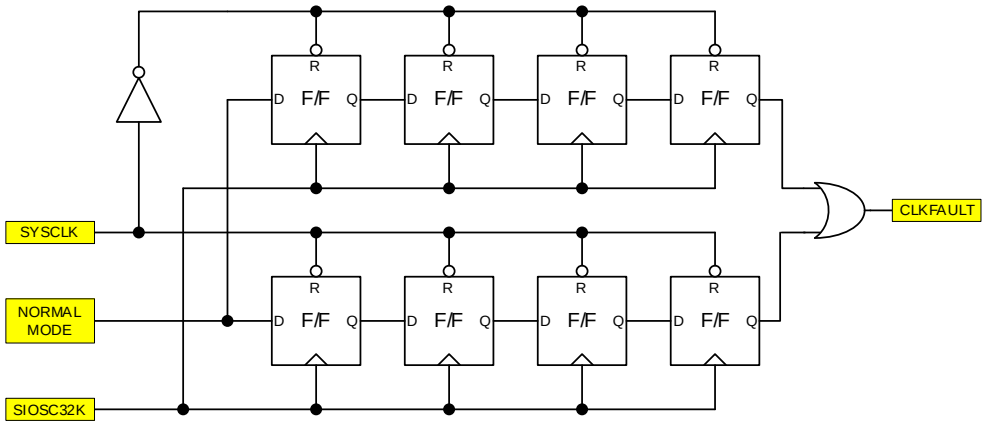


Figure 5-2 Clock Fault Monitor diagram

The clock monitoring is default turned off after reset. This is enabled by the RSTCKM bit in RSTCMD register.

5.7 ECC Controller

There are several memory macros, including data SRAM, DMA SRAM, e-Flash on chip as well as external SPI flash. All these memories have their own ECC checking due to different width, access path and speed. The handling of ECC error can also be different depending on the application. For this reason, ECC checking logic can be configured to generate an interrupt when it detects an ECC error. The ECC interrupt flags are grouped in ECCFLAG register, when the ECC interrupt occurs, software can check this register to determine which memory has ECC error and if it is correctable or un-correctable and software can take appropriate actions.

ECCFLAGA (0x5000_0100) ECC FLAG Register A RO (0x00)

	7	6	5	4	3	2	1	0
RD	FERRF1	FERRF0	SERRF1	SERRF0	MERRF1	MERRF0	DERRF1	DERRF0
WR	-	-	-	-	-	-	-	-

FERRF[1-0] e-Flash ECC Error Flags
FERRF1 is set to 1 if unrecoverable error detected when read Flash
FERRF0 is set to 1 if recoverable error detected when read Flash

SERRF[1-0] SPI Flash ECC Error Flags
SERRF1 is set to 1 if unrecoverable error detected when read SPI Flash
SERRF0 is set to 1 if recoverable error detected when read SPI Flash

MERRF[1-0] DMA SRAM ECC Error Flags
MERRF1 is set to 1 if unrecoverable error detected when read DMA SRAM
MERRF0 is set to 1 if recoverable error detected when read DMA SRAM

DERRF[1-0] Data SRAM ECC Error Flags
DERRF1 is set to 1 if unrecoverable error detected when read data SRAM
DERRF0 is set to 1 if recoverable error detected when read data SRAM

ECCBYTE (0x5000_0104) ECC Byte (8-bit) Calculation Register RW (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	ECCCODE[31-24]	ECCCODE[23-16]	ECCCODE[15-8]	ECCCODE[7-0]
WR	ECCBYTE[31-24]	ECCBYTE[23-16]	ECCBYTE[15-8]	ECCBYTE[7-0]

Writing into ECCBYTE will yield the ECC data when read out. The ECC data is nibble based. In other words, the ECC is based on 4-bit data with ECC code of 4-bit being 1-bit parity plus length of 3-bit Hamming code. This operation takes about 6-cycles to complete with hardware ready control.

ECCWORDL (0x5000_0108) ECC Word (32-bit) Calculation Register L RW (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	-	-	-	ECCCODE[7-0]
WR	ECCWORD[31-24]	ECCWORD[23-16]	ECCWORD[15-8]	ECCWORD[7-0]

ECCWORDH (0x5000_010C) ECC Word (32-bit) Calculation Register H RW (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	-	-	-	-
WR	ECCWORD[63-56]	ECCWORD[55-48]	ECCWORD[47-40]	ECCWORD[39-32]

Writing into ECCWORD will yield the ECC code data when read out. The ECCCODE[7] is always parity bit plus variable length of Hamming code bits. The result format is summarized in the following table where P is parity, 0 is dummy bits and H is hamming code.

Data Width	ECCCODE[7-0]	Descriptions
11 - 8	P + 000 + H[3-0]	
26 - 12	P + 00 + H[4-0]	
57 - 27	P + 0 + H[5-0]	
64 - 58	P + H[6-0]	

ECCSRAMX (0x5000_0110) SRAM ECC Fault Injection Register X RW (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	ECCSRAMX[31-24]	ECCSRAMX[23-16]	ECCSRAMX[15-8]	ECCSRAMX[7-0]
WR	ECCSRAMX[31-24]	ECCSRAMX[23-16]	ECCSRAMX[15-8]	ECCSRAMX[7-0]

ECCSRAMX[31-0] SRAM ECC Fault Injection
Each bit inverses the polarity of SRAM output to ECC decoder logic. By setting this register, it can inject fault to ECC decoder logic for testing purpose. ECC encoder is not of concern because encoder can be verified by decoder.

ECCSRAMY (0x5000_0114) SRAM ECC Fault Injection Register Y RW (0x00)

	31-24	23-16	15-7	6-0
RD	-	-	-	ECCSRAMY[6-0]
WR	-	-	-	ECCSRAMY[6-0]

ECCSRAMY[6-0] SRAM ECC Fault Injection
Each bit inverses the polarity of SRAM output for ECC code to ECC decoder logic. By setting this register, it can inject fault to ECC decoder logic for testing purpose. ECC encoder is not of concern because encoder can be verified by decoder.

ECCSRAMS (0x5000_0118) SRAM ECC Fault Injection Select Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	ECCSRAMS[1-0]	
WR	-	-	-	-	-	-	ECCSRAMS[1-0]	

ECCSRAMS[1-0] SRAM ECC Fault Injection Select
00=disable
01=data SRAM
10=DMA SRAM
11=disable

6. Timer T0/T1/T2

There are three general-purpose counters. T0 and T1 are 16-bit with clock source of SYSCLK. T2 is 24-bit with selectable clock source of SYSCLK or SOSC32K. T2 can also be used to trigger wake-up.

TMR0CFG (0x5000_1000) Timer 0 Configure Registers R/W (0x00)

	7	6	5	4	3	2	1	0
RD	T0IF	-	-	TM0	TR0	-	-	T0IEN
WR	T0IF	-	-	TM0	TR0	-	-	T0IEN

T0IF	Timer 0 Overflow Interrupt Flag bit. T0IF is set by hardware when overflow condition occurs. This bit needs to be cleared by writing 1.
TM0	Timer 0 Mode Control bit. TM0=1 set timer 0 as auto reload, and TM0=0 set timer 0 as free-run.
TR0	Timer 0 Run Control bit. Set to enable Timer0, and clear to stop Timer0.
T0IEN	Timer 0 Interrupt Enable bit. T0IEN=0 disable the Timer 0 overflow interrupt T0IEN=1 enable the Timer 0 overflow interrupt

TMR0CNT (0x5000_1004) Timer 0 Counter R/W (0x0000)

	31-24	23-16	15-8	7-0
RD	-	-	TMR0CNT[15-8]	TMR0CNT[7-0]
WR	-	-	TMR0VAL[15-8]	TMR0VAL[7-0]

This register functions differently when beingread versus being written. When written, in reload mode, TMR0VAL[15-0] is for the reload value, and in free-run mode, TMR0VAL[15-0] is the counter value and is updated immediately. Also please note, in reload mode, TMR0VAL[15-0] should not be written during running.

When read, the return value TMR0CNT[15-0] is always the present counter value. There is no snapshot buffer in the read operation, so software is recommended to read with word or half-word operation.

TMR1CFG (0x5000_1008) Timer 1 Configure Registers R/W (0x00)

	7	6	5	4	3	2	1	0
RD	T1IF	-	-	TM1	TR1	-	-	T1IEN
WR	T1IF	-	-	TM1	TR1	-	-	T1IEN

T1IF	Timer 1 Overflow Interrupt Flag bit. T1IF is set by hardware when overflow condition occurs. This bit needs to be cleared by writing 1.
TM1	Timer 1 Mode Control bit. TM1=1 set timer 1 as auto reload, and TM1=0 set timer 1 as free-run.
TR1	Timer 1 Run Control bit. Set to enable Timer1, and clear to stop Timer1.
T1IEN	Timer 1 Interrupt Enable bit. T1IEN=0 disable the Timer 1 overflow interrupt T1IEN=1 enable the Timer 1 overflow interrupt

TMR1CNT (0x5000_100C) Timer 1 Counter R/W (0x0000)

	31-24	23-16	15-8	7-0
RD	-	-	TMR1CNT[15-8]	TMR1CNT[7-0]
WR	-	-	TMR1VAL[15-8]	TMR1VAL[7-0]

This register functions differently when being read versus being written. When written, in reload mode, TMR1VAL[15-0] is for the reload value, and in free-run mode, TMR0VAL[15-0] is the counter value and is updated immediately. Also please note, in reload mode, TMR1VAL[15-0] should not be written during running.

When read, the return value TMR1CNT[15-0] is always the present counter value. There is no snapshot buffer in the read operation, so software is recommended to read with word or half-word operation.

TMR2CFG (0x5000_1020) Timer 2 Configure Registers R/W (0x00)

	7	6	5	4	3	2	1	0
RD	T2IF	-	T2SEL	TM2	TR2	-	-	T2IEN
WR	T2IF	-	T2SEL	TM2	TR2	-	-	T2IEN

T2IF	Timer2 Overflow Interrupt Flag bit. T2IF is set by hardware when overflow condition occurs. This bit needs to be cleared by writing 1.
T2SEL	Timer 2 Clock Selection bits. 0: SYSCLK 1: SOSC32K
TM2	Timer 2 Mode Control bit. TM2=1 set timer 2 as auto reload, and TM2=0 set timer 2 as free-running mode.
TR2	Timer2 Run Control bit. Set to enable Timer2, and clear to stop Timer2.
T2IEN	Timer 2 Interrupt Enable bit. T2IEN=0 disable the Timer 2 overflow interrupt T2IEN=1 enable the Timer 2 overflow interrupt

TMR2CNT (0x5000_1024) Timer 2 Counter R/W (0x00_0000)

	31-24	23-16	15-8	7-0
RD	-	TMR2CNT[23-16]	TMR2CNT[15-8]	TMR2CNT[7-0]
WR	-	TMR2VAL[23-16]	TMR2VAL[15-8]	TMR2VAL[7-0]

TMR2CNT functions differently when being read versus being written. When written in reload mode, TMR2VAL[23-0] is for the reload value, and in free-run mode, the counter value is written (word mode) immediately. When read, the return value is always the present counter value. There is no snapshot buffer in the read operation, so software is recommended to read with word-mode.

7. Data SRAM

There are 8KB SRAM directly accessed by CPU. This is located starting at address 0x20000000. This memory is implemented as 2048 x 39. The access of the SRAM includes hardware ECC encoder and decoder for ECC error detection and correction. When reading, a correctable ECC error occurs, the data is corrected but not updated. It can be configured to generate an ECC interrupt. To write into the Data SRAM, it requires two SYSCLK cycles due to ECC check and rewrite.

SRAMECCE (0x4000_0000) SRAM Memory ECC Error Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	IEN1	IEN0	ERRF1	ERRF0
WR	-	-	-	-	IEN1	IEN0	ERRF1	ERRF0

IEN1 — Interrupt Enable for Un-correctable Error
If IEN1=1, an un-correctable error will generate an ECC interrupt.

IEN0 — Interrupt Enable for Correctable Error
If IEN0=1, a correctable error will generate an ECC interrupt.

ERRF1 — Un-correctable Error Flag
ERRF1 is set to 1 when un-correctable error occurs for IEN1=1. This error flag is also mapped to ECCFLAG register as read only. This bit needs to be cleared by software by writing 1.

ERRF0 — Correctable Error Flag
ERRF0 is set to 1 when correctable error occurs for IEN0=1. This error flag is also mapped to ECCFLAG register as read only. This bit needs to be cleared by software by writing 1.

SRAMECCA (0x4000_0004) SRAM Memory ECC Address Register R/W (0xXXXX)

	31-24	23-16	15-8	7-0
RD	-	-	ECCADR[15-0]	
WR	-	-	-	

ECCADR[15-0] — ECC Error Address
The Data SRAM address where ECC error occurs. This is read only. And it is cleared when ERRF1 and ERRF0 are cleared to 0.
This is in byte address format. Since the size of the SRAM is 2K x 39, only ECCADR[12-2] are meaningful, ECCADR[1-0] and ECCADR[15-13] should be ignored.

8. DMA SRAM and DMA Controller

The DMA controller is responsible for controlling the access of the 8KB data SRAM which resides at 0x3000000000. This SRAM can be accessed only with double word 32-bit width. The SRAM is implemented in 2048 x 39 with 7-bits for ECC.

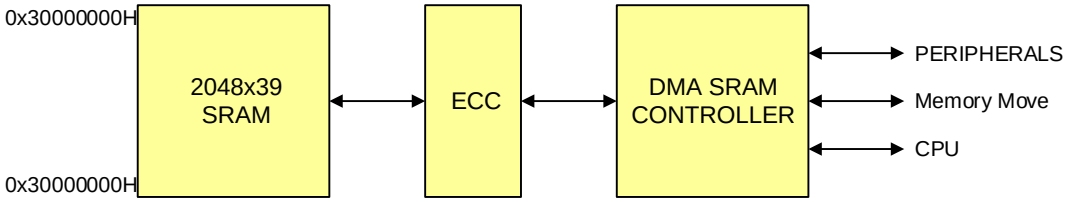


Figure 8-1 ECC for SRAM and DMA SRAM

The access is based on 16-slot round-robin arbitration. The even slots are always assigned to CPU access, and the odd slots are allocated for peripheral accesses. Each odd slot can be configured to be assigned to CPU or one of the DMA peripherals such as, CRC accelerator, SPIF, SPI1, EUART1, CANFD, etc. This guarantees CPU has at least 50% of memory access. And each slot can have 1/16th of the bandwidth, with 48MHz system clock it means 160Mbps. In addition, the ECC is performed when the data is written as well as reading. When reading, if a 1-bit correctable error is detected, then output data is automatically corrected without modifying stored data. If an uncorrectable error is detected, the stored data (possibly corrupted) is still outputted. ECC error (both 1-bit and 2-bit) triggers ECC interrupt. It is necessary the data should be initialized before usage to prevent unintended ECC error.

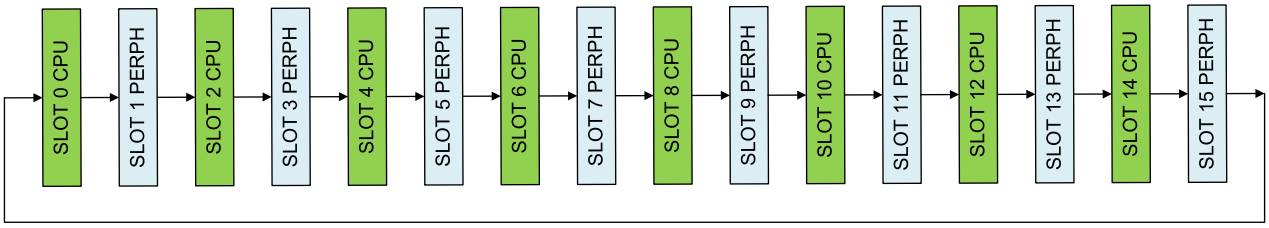


Figure 8-2 DMA Slot Assignment

DMACFGA (0x4000_0100) DMA Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SLOT3[3-0]				SLOT1[3-0]			
WR	SLOT3[3-0]				SLOT1[3-0]			

SLOT3[3-0] Slot 3 Assignment
Same definition as SLOT1[3-0]

SLOT1[3-0] Slot 1 Assignment
0000 = CPU
0001 = CRC/CS
0010 = SPF
0011 = SPV
0100 = LMC
0101 = CANFD
0110 = Memory Move
Other = Reserved

DMACFGB (0x4000_0104) DMA Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SLOT7[3-0]				SLOT5[3-0]			
WR	SLOT7[3-0]				SLOT5[3-0]			

SLOT7[3-0] Slot 7 Assignment
Same definition as SLOT1[3-0]

SLOT5[3-0] Slot 5 Assignment
Same definition as SLOT1[3-0]

DMACFGC (0x4000_0108) DMA Configuration Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SLOT11[3-0]				SLOT9[3-0]			
WR	SLOT11[3-0]				SLOT9[3-0]			

SLOT11[3-0] Slot 11 Assignment
Same definition as SLOT1[3-0]

SLOT9[3-0] Slot 9 Assignment
Same definition as SLOT1[3-0]

DMACFGD (0x4000_010C) DMA Configuration Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SLOT15[3-0]				SLOT13[3-0]			
WR	SLOT15[3-0]				SLOT13[3-0]			

SLOT15[3-0] Slot 15 Assignment
Same definition as SLOT1[3-0]

SLOT13[3-0] Slot 13 Assignment
Same definition as SLOT1[3-0]

DMAECC (0x4000_0110) DMA Memory ECC Error Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	IEN1	IEN0	ERRF1	ERRF0
WR	-	-	-	-	IEN1	IEN0	ERRF1	ERRF0

IEN1 Interrupt Enable for Un-correctable Error
If IEN1=1, an un-correctable error will generate an ECC interrupt.

IEN0 Interrupt Enable for Correctable Error
If IEN0=1, a correctable error will generate an ECC interrupt.

ERRF1 Un-correctable Error Flag
ERRF1 is set to 1 when un-correctable error occurs for IEN1=1. This error flag is also mapped to ECCFLAG register as read only. This bit needs to be cleared by software by writing 1.

ERRF0 Correctable Error Flag
ERRF0 is set to 1 when correctable error occurs for IEN0=1. This error flag is also mapped to ECCFLAG register as read only. This bit needs to be cleared by software by writing 1.

DMAECCA (0x4000_0120) DMA Memory ECC Address Register R/W (0xXXXX)

	31-24	23-16	15-8	7-0
RD	-	-	ECCADR[15-0]	
WR	-	-	-	

ECCADR[15-0] ECC Error Address
The DMA SRAM address where ECC error occurs. This is read only. And it is cleared when ERRF1 and ERRF0 are cleared to 0.
This is in byte address format. Since the size of the SRAM is 2K x 39, only ECCADR[12-2] are meaningful, ECCADR[1-0] and ECCADR[15-13] should be ignored.

8.1 DMA SRAM Move

The DMA controller has a built-in memory move function. When enabled, it moves a portion of the memory contents to another location. The move is in 32-bit alignment and up to 256 x 32.

DMAMOVC (0x4000_0200) DMA Memory MOVE Command Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	INTE	INTF	ASQREV	ENDIANMODE[1-0]		-	-	BUSY
WR	INTE	INTF	ASQREV	ENDIANMODE[1-0]		-	-	START

INTE — Interrupt Enable

INTF — Completion Interrupt Flag
This bit is set to 1 by hardware after completion of the move operation. It must be cleared by software by writing 1.

ASQREV — Move Address Sequencing Reverse
ASQREV=0, the move action is performed by incrementing both source and destination addresses.
ASQREV =1, the move action is performed by incrementing source address and decrementing destination address.

ENDIANMODE[1-0] — The ENDIANMODE[1-0].defined how the bit sequence was arranged during DMA move

ENDIANMODE[1-0]		Result
0	0	DATA[31-24] : DATA[23-16] : DATA[15-8] : DATA[7-0]
0	1	DATA[24-31] : DATA[16-23] : DATA[8-15] : DATA[0-7]
1	0	DATA[0-7] : DATA[8-15] : DATA[16-23] : DATA[24-31]
1	1	DATA[7-0] : DATA[15-8] : DATA[23-16] : DATA[31-24]

BUSY — Busy Status
Busy=1 indicates the move operation is ongoing.

START — Start Move Operation
This bit is auto cleared when operation is completed.
START=0 No operation
START=1 Start the move

DMAMOVL (0x4000_0201) DMA Memory MOVE Length Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LENGTH[7-0]							
WR	LENGTH[7-0]							

LENGTH[7-0] — Move Data Length
LENGTH[7-0]+1 defines the number of word (32-bit) to be moved.

DMAMOVA (0x4000_0204) DMA Memory MOVE Address Register R/W (0x0000_0000)

	31-16	15-0
RD	DADDR[15-0]	SADDR[15-0]
WR	DADDR[15-0]	SADDR[15-0]

8.2 CS/CRC Accelerator

To enhance performance, a hardware Checksum/CRC Accelerator is included and closely coupled with CPU and DMA. It provides most commonly used checksum and CRC operation for 8/16/32-bit data width. The input data stream is obtained from DMA SRAM through DMA access. The calculation is started by issuing START command. Alternatively, CPU can also perform CRC calculation by writing data into the CCDATA register if REGMODE is set.

CCCFGA (0x4000_0300) Checksum/CRC Accelerator Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DWIDTH[1-0]		REVERSE	NOCARRY	SEED	CRCMODE[2-0]		
WR	DWIDTH[1-0]		REVERSE	NOCARRY	SEED	CRCMODE[2-0]		

DWIDTH[1-0] — Data Width
00 – set input as 8-bit wide
01 – set input as 16-bit wide, Not supported.
10 – set the input as 24-bit wide, Not supported.
11 – set the input as 32-bit wide, Not supported.
The calculation is always 8-bit byte wise.

REVERSE — Reverse Input MSB/LSB Sequence
REVERSE=0 is for LSB first operations.
REVERSE=1 is for MSB first operation.

The reverse order is based on the data width. The CS/CRC encoding hardware is done in data width.

NOCARRY — Carry Setting for Checksum
NOCARRY=0 uses previous carry result for new result
NOCARRY=1 discard previous carry result.

SEED — Seed Entry
SEED=1 results writing into CRCDATA to become SEED value
SEED=0 for normal data inputs.

CRCMODE[2-0] — Defines CRC/Checksum Mode
000 – Accelerator is disabled and clock gated off
001 – 8-bit Checksum. Carry occurs at 8-bit boundary, and result is 8-bit.
010 – 32-bit Checksum. Carry occurs at 32-bit boundary, and result is 32-bit.
011 – CRC-16 0x8005
X16+X15+X2+1
100 – CRC-16 0x1021 CCITT
X16+X12+X5+1
101 – CRC-32 0x04C11DB7
X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X1+1
110 – Reserved.
111 – CRC and Checksum Clear
Writing “111” to CRCMODE[1-0] reset the CS/CRC states and restore default seed value (for checksum, seed value=0x00 or 0x0000, for CRC seed value = 0xFFFF). Writing “111” does not affect the previously set mode selection.
The first step for the programmer is to set the CRCMODE[2-0] for the Checksum or CRC operation and then write “111” to CRCMODE[2-0] to reset the Checksum/CRC states and restore the default seed value (for checksum, seed value=0x00 or 0x0000_0000, for CRC seed value = 0xFFFF or 0xFFFF_FFFF).

CCCFGB (0x4000_0301) Checksum/CRC Accelerator Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	REGMODE	-	OBYTESWP	OBITREV	-	-	-	BUSY
WR	REGMODE	-	OBYTESWP	OBITREV	-	-	-	START

REGMODE — Register Mode
REGMODE=0, CRC data is obtained automatically through DMA.
REGMODE=1, CRC data is input through writing of the CCDATA register.

OBYTESWP — Output Byte Swap
OBYTESWP=0, CCDATA is arranged as MSB byte, …, LSB byte.
OBYTESWP=1, CCDATA is arranged as LSB byte, …, MSB byte.

OBITREV — Output Bit Reverse
OBREV=0, CCDTAT is arrange as MSB, …., LSB
OBERV=1, CCDTAT is arrange as LSB, …., MSB
Please note, OBITREV is performed first then OBYTESWP.

BUSY — CRC Status
BUSY=1 indicates the results is not yet completed. Since only up to two cycles are used to calculate the Checksum or CRC, there is no need to check BUSY status before next data entry and reading the results.

START — CRC Start
Setting START=1 will start the CRC calculation with DMA access to the data. It is self-cleared to 0.

CCCFGC (0x4000_0302) Checksum/CRC Accelerator Configuration Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	INTEN	-	-	-	-	-	-	INTF
WR	INTEN	-	-	-	-	-	-	INTF

INTEN — Interrupt Enable
INTEN=1 enables interrupt when DMA CRC calculation is completed.

INTF — Interrupt Flag

INTF is set to 1 by hardware when DMA CRC calculation is completed. INTF must be cleared by software by writing 1.

CCADDR (0x0000_0304) CS/CRC DMA Address Register R/W (0x0000_0000)

	31	30-16	15-8	7-0
RD	DESC	BC[14-0]	CCADDR[15-8]	CCADDR[7-0]
WR	DESC	BC[14-0]	CCADDR[15-8]	CCADDR[7-0]

DESC — Address Descending/Ascending
DESC=0 the new data is fetched with incrementing address when performing calculation.
DESC=1 the new data is fetched with decrementing address when performing calculation.

BC[14-0] — Byte Count
The number of bytes to be calculated. This must match with the DWIDTH setting.

CCADDR[15-0] — DMA Start Address
DMA memory start address. This can be byte, word, or double word address. This should be word aligned.

CCDATA (0x0000_0308) CS/CRC Data Register R/W (0xFFFF_FFFF)

	31-24	23-16	15-8	7-0
RD	CCDATA[31-24]	CCDATA[23-16]	CCDATA[15-8]	CCDATA[7-0]
WR	CCDATA[31-24]	CCDATA[23-16]	CCDATA[15-8]	CCDATA[7-0]

CCDATA registers are the data port for Checksum/CRC Accelerator. For 8-bit data width only CCDATA[7-0] should be used. When SEED=1, the data been written goes to CS or CRC seed value. The result of accelerator can be directly read out from CCDATA registers.

Common CRC						CS920x Configuration						
Algorithm	Formula	Poly	Seed value	REFIN	REFOUT	CRCMODE	SEED bit	DWIDTH	REVERSE	OBITREV	Result	XOR output
CRC-16/IBM	$X^{16}+X^{15}+X^2+1$	0x8005	0x0000	TRUE	TRUE	3	0	0	1	Don't care	CCDATA[31:16]	
CRC-16/MAXIM	$X^{16}+X^{15}+X^2+1$	0x8005	0x0000	TRUE	TRUE	3	0	0	1	Don't care	CCDATA[31:16]	0xFFFF
CRC-16/USB	$X^{16}+X^{15}+X^2+1$	0x8005	0xFFFF	TRUE	TRUE	3	1	0	1	Don't care	CCDATA[31:16]	0xFFFF
CRC-16/MODBUS	$X^{16}+X^{15}+X^2+1$	0x8005	0xFFFF	TRUE	TRUE	3	1	0	1	Don't care	CCDATA[31:16]	
CRC-16/UMTS	$X^{16}+X^{15}+X^2+1$	0x8005	0x0000	FALSE	FALSE	3	0	0	0	Don't care	CCDATA[15:0]	
CRC-16/DDS-110	$X^{16}+X^{15}+X^2+1$	0x8005	0x800D	FALSE	FALSE	3	1	0	0	Don't care	CCDATA[15:0]	
CRC-16/CCITT	$X^{16}+X^{12}+X^5+1$	0x1021	0x0000	TRUE	TRUE	4	1	0	0	Don't care	CCDATA[31:16]	
CRC-16/CCITT-FALSE	$X^{16}+X^{12}+X^5+1$	0x1021	0xFFFF	FALSE	FALSE	4	0	0	1	Don't care	CCDATA[15:0]	
CRC-16/AUG-CCITT	$X^{16}+X^{12}+X^5+1$	0x1021	0x1D0F	FALSE	FALSE	4	1	0	1	Don't care	CCDATA[15:0]	
CRC-16/TMS37157	$X^{16}+X^{12}+X^5+1$	0x1021	0x89EC	TRUE	TRUE	4	1	0	0	Don't care	CCDATA[31:16]	
CRC-16/RIELLO	$X^{16}+X^{12}+X^5+1$	0x1021	0xB2AA	TRUE	TRUE	4	1	0	0	Don't care	CCDATA[31:16]	
CRC-16/IEC14443-3-A	$X^{16}+X^{12}+X^5+1$	0x1021	0xC6C6	TRUE	TRUE	4	1	0	0	Don't care	CCDATA[31:16]	
CRC-16/MCRF4XX	$X^{16}+X^{12}+X^5+1$	0x1021	0xFFFF	TRUE	TRUE	4	0	0	0	Don't care	CCDATA[31:16]	
CRC-16/X25	$X^{16}+X^{12}+X^5+1$	0x1021	0xFFFF	TRUE	TRUE	4	0	0	0	Don't care	CCDATA[31:16]	0xFFFF
CRC-16/XMODEM	$X^{16}+X^{12}+X^5+1$	0x1021	0x0000	FALSE	FALSE	4	1	0	1	Don't care	CCDATA[15:0]	
CRC-16/GENIBUS	$X^{16}+X^{12}+X^5+1$	0x1021	0x0000	FALSE	FALSE	4	1	0	1	Don't care	CCDATA[15:0]	0xFFFF
CRC-32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0xFFFFFFFF	TRUE	TRUE	5	0	Don't care	0	1	CCDATA[31:0]	0xFFFFFFFF
CRC-32/MPEG-2	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0xFFFFFFFF	FALSE	FALSE	5	0	0	1	0	CCDATA[31:0]	
CRC-32/BZIP2	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0xFFFFFFFF	FALSE	FALSE	5	0	0	1	0	CCDATA[31:0]	0xFFFFFFFF
CRC-32/JAMCRC	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0xFFFFFFFF	TRUE	TRUE	5	0	Don't care	0	1	CCDATA[31:0]	
CRC-32/POSIX	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0x00000000	FALSE	FALSE	5	1	0	1	0	CCDATA[31:0]	0xFFFFFFFF
CRC-32/SATA	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0x52325032	FALSE	FALSE	5	1	0	1	0	CCDATA[31:0]	
*REFIN : Whether each byte of input data is reversed by bit *REFOUT : Whether the whole data is reversed bit by bit after calculation and before the result is output						* If SEED bit = 1, fisrt write SEED as 1 then write seed value to CCDATA and then Write SEED 0						

CS920x CRC Default State

Algorithm	Formula	Poly	Seed value	REFIN	REFOUT
CRC-16	$X^{16}+X^{15}+X^2+1$	0x8005	0x0000	FALSE	FALSE
CRC-16 CCITT	$X^{16}+X^{12}+X^5+1$	0x1021	0xFFFF	TRUE	TRUE
CRC-32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0xFFFFFFFF	TRUE	TRUE

Figure 8-3 General setting for various CRC Calculation

CRC-16/IBM = CRC-16/LHA = CRC-16/ARC

CRC-16/UMTS = CRC-16/BUYPASS = CRC-16/VERIFONE

CRC-16/CCITT = CRC-16/KERMIT

CRC-16/CCITT False = CRC-16/IBM3740 = CRC-16/AUTOSTAR

CRC-16/CCITT False = CRC-16/IBM3740 = CRC-16/AUTOSTAR

CRC-16/GENIBUS = CRC-16/DARC = CRC-16/EPC = CRC-16/RPC-C1G2 = CRC-16/I-CODE

9. E-Flash Memory Controller

The embedded Flash memory contains two blocks –Main Memory and Information Block (IFB). The Main Memory is 64KX39 with uniform 256x39 page (1KB sector) size. There are four Information Blocks (IFB), each is also 256X39 and are located in two separate pages. IFB0 (also referred as IFBM) contains manufacture device information and calibration data, and should not be altered by the user. IFB1 (also referred as IFBU1), IFB2 (IFBU2), IFB3 (IFBU3) contain user data and can be accessed by the user.

The 39-bit access width of e-Flash contains 32-bit data and 7-bit ECC. ECC is hardware encoded and decoded. When the e-Flash is erased, all contents are read as “1”. When e-Flash is programmed, the content is read as “0”. Reading e-Flash content ECC is automatically checked. As the result, if a read operation is performed after erase, since all 39 bits are 1, ECC error will occur (DATA=FFFFFFFF ECC=0011000, DATA=00000000 ECC=0000000). After a write operation, the correct ECC content will be written by the hardware automatically. Then further reading operations should yield no ECC error.

The CPU access of the FLASH through AHB-Lite bus does not require any special attention. Due to limited FLASH access time (<40nsec), CPU running higher than 16MHz will need to set FLASH access wait state through WTST register. In the instruction fetch path, ECC is checked automatically. If ECC error is encountered, the ECC fault handling can be enabled to generate an interrupt.

Software and including SWI can also access the FLASH through Flash Controller. The commands performed by a Flash Controller are defined in FLSHCMD registers. The defined operations allow the user program to use on-chip flash as a program memory, and a non-volatile data memory in In-System-Programming as well as In-Application-Programming.

There are two locking mechanisms for data and code security. UKEY is stored in IFB1’s top 8-byte location, and MKEY is stored in IFB0’s top 8-byte location. MKEY governs IFB0 data, and UKEY governs IFB0 and main memory data. Regardless of access means, the corresponding KEY information must be provided and Key Verify command completed successfully to unlock the data protection. Under locked state, only limited commands can be issued and recognized. For command Locked State Erase commands are only accepted from SWI interface.

FLSHCMD0 (0x4000_1008) Flash Controller Command Register 0 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	UKOPEN	MKOPEN	TBIT	ECCST	FAIL[2-0]			BUSY
WR	CMD[7-0]							

UKOPEN — UKEY Lock Status
UKOPEN is set to 1 after a successful UKEY verify command is finished with matching key value. Any further unsuccessful UKEY verify command will reset UKOPEN to 0. UKOPEN is also cleared to 0 after any reset condition.

MKOPEN — MKEY Lock Status
MKOPEN is set to 1 after a successful MKEY verify command is finished with matching key value. Any further unsuccessful MKEY verify command will reset MKOPEN to 0. MKOPEN is also cleared to 0 after any reset condition.

TBIT — An e-Flash Internal signal.

ECCST — ECC status
ECCST is updated by hardware after a read command is finished. The corresponding ECC checking result is shown by ECCST. If ECC passed, ECCST=0, if ECC failed, ECCST=1. This ECC does not cause the command to fail or cause any interrupt.

FAIL[2-0] — Command execution result.
FAIL[2-0] is set and cleared by hardware after a command is finished.
000 Command Successful
001 Fail due to locked state
010 Fail due to time out
011 Fail due to protection zone
100 Invalid address range
101 Invalid commands
110 unsupported commands
111 write verify fail

BUSY — Flash Controller Status
BUSY is set to 1 by hardware indicating current command is still under execution.

FLSHCMD1 (0x4000_1009) Flash Controller Command Register 1 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-		-	-	DMODE[1-0]	
WR	-	-	1's comp of DMODE[1-0]		-		DMODE[1-0]	

DMODE[1-0] Command Data Mode
00 = ECC decode/encode data
01 = ECC decode/encode data
10 = Raw data
11 = Reserved

FLSHCMD2 (0x4000_100A) Flash Controller Command Register 2 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RDALL1	-	-	-	-	-	-	MMERASED
WR	2's Complement of CMD[7-0]							

RDALL1 This bit is set for a read command and all 39 bits of read data are one. This status helps to speed up for verify erase results.

MMERASED This bit is set by hardware after MM erase command is completed successfully. This bit is cleared by any reset condition.

FLSHCMD3 (0x4000_100B) Flash Controller Command Register 3 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	1's Complement of CMD[7-0]							

CMD[7-0] Command Definition is listed in the following table.

CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0	LOCK	COMMAND
0	0	0	0	0	0	0	0	Y	Verify MKEY[1-0]. MKey0 data is in FLSHDAT[31-0], MKey1 data is in FLSHADDR[31-0]
0	0	0	0	0	0	0	1	Y	Verify UKEY[1-0]. UKey0 data is in FLSHDAT[31-0], UKey1 data is in FLSHADDR[31-0]
0	0	0	0	0	0	1	0	Y	Main Memory Erase This command can only be executed through debug port access.
0	0	0	0	0	0	1	1	N/Y	Whole Chip Erase Only if MKOPEN=1. Use with caution. This erases manufacturing data and void any warranty. This command can only be executed through debug port access.
0	0	0	0	0	1	0	0	Y	Lock State IFB 1 Erase. Executable only Main Memory Erase Command has just been completed and passed.
0	0	0	0	0	1	0	1	N	Main Memory Sector Erase Sector 0 cannot be used, returns failure.

CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0	LOCK	COMMAND
0	0	0	0	0	1	1	0	N	IFB 0 Erase Only if MKOPEN=1 Use with caution. This erases manufacturing data and void any warranty.
0	0	0	0	0	1	1	1	N	IFB 1 Erase
0	0	0	0	1	0	0	0	N	Main Memory Read
0	0	0	0	1	0	0	1	N	Main Memory Read, Auto Increment.
0	0	0	0	1	1	0	0	N/Y	IFB 0 Word Read Locked state limit access to 0x00-0xFD.
0	0	0	0	1	1	0	1	N/Y	IFB 0 Word Read, Auto Increment. Locked state limit access to 0x00-0xFD.
0	0	0	0	1	1	1	0	N/Y	IFB 1/2/3 Word Read. IFB1 Locked state limit access to 0x00-0xFD
0	0	0	0	1	1	1	1	N/Y	IFB 1/2/3 Word Read, Auto Increment. IFB1 Locked state limit access to 0x00-0xFD
0	0	0	1	0	0	0	0	N	Main Memory Write Sector 0 cannot be used, returns failure.
0	0	0	1	0	0	0	1	N	Main Memory Write, Auto Increment. Sector 0 cannot be used, returns failure.
0	0	0	1	0	1	0	0	N	IFB 0 Word Write
0	0	0	1	0	1	0	1	N	IFB 0 Word Write, Auto Increment.
0	0	0	1	0	1	1	0	N	IFB 1/2/3 Word Write
0	0	0	1	0	1	1	1	N	IFB 1/2/3 Word Write, Auto Increment.

Table 9-1 Flash Controller command list

The command definition is listed in the above table. Any command not listed in the above is considered invalid and will cause FAIL to be set. Please note to provide command integrity, FLSHCMD[15-8] must have the 1's complement value of CMD[7-0].

FLSHDAT (0x4000_1004) Flash Controller Data Register R/W (0x0000_0000)

	31 - 0
RD	Flash Read Data Register [31-0]
WR	Flash Write Data Register [31-0]

FLSHECC (0x4000_100C) Flash Controller Data Register R/W (0x00)

	7	6-0
RD	-	ECC[6-0]
WR	-	ECC[6-0]

FLASHECC[6-0] contains raw data bits for the ECC code of the e-Flash. It is addressed by the FLSHADR and used when FLSHCMD's DMODE is set for raw accesses.

FLSHADR (0x4000_1000) Flash Controller Address Register R/W (0x0000_0000)

	31-0
RD	Flash Address Register ADDR[31-0]
WR	Flash Address Register ADDR[31-0]

Please note the least significant two bit of the address register is ignored as the flash interface is organized as 32-bit wide.

FLSHCFG0 (0x4000_1010) Flash Configuration Register 0 R/W (0x40)

	7	6	5	4	3	2	1	0
RD	ISPCLKF[7-0]							
WR	ISPCLKF[7-0]							

ISPCLKF[7-0] configures the clock time base for generation of Flash erase and write timing. ISPCLK = SYSCLK * (ISPCLKF[7-0]+1)/256. For correct timing, ISPCLK should be set to approximately at 4MHz.

FLSHCFG1 (0x4000_1011) Flash Configuration Register 1 R/W (0x04)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	CYC[2-0]		
WR	-	-	-	-	-	CYC[2-0]		

CYC[2-0] Flash Command Time Out Setting

CYC[2-0] defines command time out cycle count. Cycle period is defined by ISPCLK. The number of ISPCLK cycles for time out is tabulated as following.

CYC[2-0]			Write	Erase
0	0	0	100	10000
0	0	1	110	11000
0	1	0	120	12000
0	1	1	130	13000
1	0	0	140	14000
1	0	1	150	15000
1	1	0	160	16000
1	1	1	170	17000

Default is 100. For 4MHz ISPCLK, time out for write and erase is 35usec and 3.5msec respectively.

EFMECC (0x4000_1014) Embedded Flash Memory ECC Error Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ECCOFF	-	-	-	IEN1	IEN0	ERRF1	ERRF0
WR	ECCOFF	-	-	-	IEN1	IEN0	ERRF1	ERRF0

ECCOFF ECC Enable
ECCOFF=0 enables ECC
ECCOFF=1 disable ECC, this only affects direct read path from CPU.

IEN1 Interrupt Enable for Un-correctable Error
If IEN1=1, an un-correctable error will generate an ECC interrupt.

IEN0 Interrupt Enable for Correctable Error
If IEN0=1, a correctable error will generate an ECC interrupt.

ERRF1 Un-correctable Error Flag
ERRF1 is set to 1 when un-correctable error occurs. This error flag is also mapped to ECCFLAG register as read only. This bit needs to be cleared by software by writing 1.

ERRF0 Correctable Error Flag

ERRF0 is set to 1 when correctable error occurs. This error flag is also mapped to ECCFLAG register as read only. This bit needs to be cleared by software by writing 1.

EFMECCA (0x4000_1018) Embedded Flash Memory ECC Address Register R/W (0xXXXX_XXXX)

	31-24	23-18	17-8	7-0
RD	-	-	ECCADR[17-0]	
WR	-	-	-	

ECCADR[15-0] ECC Error Address
The e-Flash word address where ECC error occurs by CPU access.
For e-Flash controller read access, this address register does not reflect the error address. Therefore, when software performing e-Flash read, it should also check FERRF1 and FERRF0 status to determine if any ECC error has occurred. Please refer to ECC handling application note.
This is read only. And it is cleared when ERRF1 and ERRF0 are cleared to 0.
Since the e-Flash data width is 39, only ECCADR[17-2] are meaningful, and the LSB two bits are 0.

The ECC behavior also is affected by DMODE and differs between Flash Controller read and Instruction (CPU direct access) path. Following tables shows the behavior.

Read	ECCOFF=0			ECCOFF=1		
	Correction	EFMECC	EFMECCA	Correction	EFMECC	EFMECCA
Flash controller DMODE=0 (ECC)	Y	Y	N/A	Y	Y	N/A
Flash controller DMODE=3 (RAW)	N	Y	N/A	N	Y	N/A
Instruction	Y	Y	Y (A17:A2)	N	N/A	N/A

FLSHCFG2 (0x4000_101C) Flash Configuration Register 2 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	PRTRDY	PRTDLY[1-0]		-	
WR	-	-	-	-	PRTDLY[1-0]		-	

PRTRDY Protection Delay Status
PRTRDY=1 indicates protection delay is still on.
PRTRDY=0 indicates protection delay is off or no-effect.

PRTDLY[1-0] Main Memory Protection On/Off Delay Setting
This setting inserts a delay for turning on/off protection (changing setting of FLSHPRT[31-0]) to provide further protection of main memory content where resident program presides.
00 100msec
01 200msec
10 400msec
11 No delay
Default is 00 which is 100msec.
PRTDLY[1-0] only can be changed while PRTRDY is 0. When modifying PRTDLY[1-0] it will also cause a delay of setting.
This does not affect IFB protections where IFB are used for data storage.

The e-Flash has segment size of 256 x 39 in 1KB size. For CS9202, the e-Flash is 256KB, each of the main memory 256 segments and four IFBs has its own protection setting bits. The protection is controlled by two protection bits, PRT and PPT. PRT default after reset is 0, and PPT is 1 where 0 means protected, and 1 means unprotected. PPT (permanent) can be written 0 only and once written 0, it stays 0.

After power-on reset, all zones are protected for 100msec regardless of protection zone settings. After this period, any modification of protection bits to turn on/off the protection needs to elapse a protection period defined by PRTDLY[1-0] to become effective. Please note, in order to erase or modify the content of the Flash, the corresponding MKEY and UKEY has to be unlocked first. In addition, PRTDLY setting insert an additional delay when turning off the protection.

It is recommended that permanent protection should be set for critical program areas. For flash areas used for data that can be modified, the protection should be turned off. These should be done with software initialization after power on or reset.

FLSHPIFB (0x4000_101D) Flash IFB Protection Register R/W (0xF0)

	7	6	5	4	3	2	1	0
RD	PPTIFB3	PPTIFB2	PPTIFB1	PPTIFB0	PRTIFB3	PRTIFB2	PRTIFB1	PRTIFB0
WR	PPTIFB3	PPTIFB2	PPTIFB1	PPTIFB0	PRTIFB3	PRTIFB2	PRTIFB1	PRTIFB0

PPTIFB0	IFB3 Permanent Protection
PPTIFB1	IFB2 Permanent Protection
PPTIFB0	IFB1 Permanent Protection
PPTIFB1	IFB0 Permanent Protection
PRTIFB3	IFB3 Protection
PRTIFB2	IFB2 Protection
PRTIFB1	IFB1 Protection
PRTIFB0	IFB0 Protection

FLSHPRT0 (0x4000_1100) MM Protection Register R/W (0x0000_0000)

	31-0
RD	FLSHPRT[31-0]
WR	FLSHPRT[31-0]

FLSHPRT[31-0] Flash Memory Protection for sector 0 to 31
FLSHPRT[i] is the PRT bit for corresponding Flash sector [i].

FLSHPRT1 (0x4000_1104) MM Protection Register R/W (0x0000_0000)

	31-0
RD	FLSHPRT[63-32]
WR	FLSHPRT[63-32]
WR	FLSHPRT[63-32]

FLSHPRT[63-32] Flash Memory Protection for sector 32 to 64
FLSHPRT[i] is the PRT bit for corresponding Flash sector [i].

FLSHPRT2 (0x4000_1108) MM Protection Register R/W (0x0000_0000)

	31-0
RD	FLSHPRT[95-64]
WR	FLSHPRT[95-64]

FLSHPRT[95-64] Flash Memory Protection for sector 64 to 95
FLSHPRT[i] is the PRT bit for corresponding Flash sector [i].

FLSHPRT3 (0x4000_110C) MM Protection Register R/W (0x0000_0000)

	31-0
RD	FLSHPRT[127-96]
WR	FLSHPRT[127-96]

FLSHPRT[127-96] Flash Memory Protection for sector 96 to 127
FLSHPRT[i] is the PRT bit for corresponding Flash sector [i].

FLSHPRT4 (0x4000_1110) MM Protection Register R/W (0x0000_0000)

	31-0
RD	FLSHPRT[159-128]
WR	FLSHPRT[159-128]

FLSHPRT[159-128] Flash Memory Protection for sector 128 to 159
FLSHPRT[i] is the PRT bit for corresponding Flash sector [i].

FLSHPRT5 (0x4000_1114) MM Protection Register R/W (0x0000_0000)

	31-0
RD	FLSHPRT[191-160]
WR	FLSHPRT[191-160]

FLSHPRT[191-160] Flash Memory Protection for sector 160 to 191
FLSHPRT[i] is the PRT bit for corresponding Flash sector [i].

FLSHPRT6 (0x4000_1118) MM Protection Register R/W (0x0000_0000)

	31-0
RD	FLSHPRT[223-192]
WR	FLSHPRT[223-192]

FLSHPRT[223-192] Flash Memory Protection for sector 192 to 223
FLSHPRT[i] is the PRT bit for corresponding Flash sector [i].

FLSHPRT7 (0x4000_111C) MM Protection Register R/W (0x0000_0000)

	31-0
RD	FLSHPRT[255-224]
WR	FLSHPRT[255-224]

FLSHPRT[255-224] Flash Memory Protection for sector 224 to 255
FLSHPRT[i] is the PRT bit for corresponding Flash sector [i].

FLSHPPT0 (0x4000_1180) MM Permanent Protection Register R/W (0xFFFF_FFFF)

	31-0
RD	FLSHPPT[31-0]
WR	FLSHPPT[31-0]

FLSHPPT[31-0] Permanent Protection Setting of Flash sector 0 to 31
FLSHPPT[i] is the PPT bit for corresponding Flash sector [i].

FLSHPPT1 (0x4000_1184) MM Permanent Protection Register R/W (0xFFFF_FFFF)

	31-0
RD	FLSHPPT[63-32]
WR	FLSHPPT[63-32]

FLSHPPT[63-32] Permanent Protection Setting of Flash sector 32 to 63
FLSHPPT[i] is the PPT bit for corresponding Flash sector [i].

FLSHPPT2 (0x4000_1188) MM Permanent Protection Register R/W (0xFFFF_FFFF)

	31-0
RD	FLSHPPT[95-64]
WR	FLSHPPT[95-64]

FLSHPRT[95-64] Main Memory Zone Protection Setting

FLSHPPT3 (0x4000_118C) MM Permanent Protection Register R/W (0xFFFF_FFFF)

		31-0
RD		FLSHPPT[127-96]
WR		FLSHPPT[127-96]

FLSHPPT[127-96] Permanent Protection Setting of Flash sector 96 to 127
FLSHPPT[i] is the PPT bit for corresponding Flash sector [i].

FLSHPPT4 (0x4000_1190) MM Permanent Protection Register R/W (0xFFFF_FFFF)

	31-0
RD	FLSHPPT[159-128]
WR	FLSHPPT[159-128]

FLSHPPT[159-128] Permanent Protection Setting of Flash sector 128 to 160
FLSHPPT[i] is the PPT bit for corresponding Flash sector [i].

FLSHPPT5 (0x4000_1194) MM Permanent Protection Register R/W (0xFFFF_FFFF)

	31-0
RD	FLSHPPT[191-160]
WR	FLSHPPT[191-160]

FLSHPPT[191-160] Permanent Protection Setting of Flash sector 160 to 191
FLSHPPT[i] is the PPT bit for corresponding Flash sector [i].

FLSHPPT6 (0x4000_1188) MM Permanent Protection Register R/W (0xFFFF_FFFF)

	31-0
RD	FLSHPPT[223-192]
WR	FLSHPPT[223-192]

FLSHPPT[223-192] Permanent Protection Setting of Flash sector 192 to 223
FLSHPPT[i] is the PPT bit for corresponding Flash sector [i].

FLSHPPT7 (0x4000_119C) MM Permanent Protection Register R/W (0xFFFF_FFFF)

	31-0
RD	FLSHPPT[255-224]
WR	FLSHPPT[255-224]

FLSHPPT[255-224] Permanent Protection Setting of Flash sector 224 to 255
FLSHPPT[i] is the PPT bit for corresponding Flash sector [i].

10. SPI Flash Controller (SPF)

The SPI Flash Controller allows the CPU to access external SPI flash memory for data. In essence, SPF is a SPI master with dedicated function optimization for accessing external SPI Flash. The SPI Flash interface requires SPFCS, SPFCLK, and SPFDI/SPFDO pins for standard SPI, or SPFIO[3-0] for QPI (Quad Peripheral Interface) mode. Due to the speed of general purpose I/O these pins utilize, the SPI clock is typically limited to 10MHz. If QPI mode is set, SPF also needs to control the OE (Output Enable) of the I/O.

SPF transactions are started by issuing START command. There are four transaction types for various command formats of SPI Flash. The last transaction type is optimized for data transfer between SPI Flash and DMA SRAM. Once the transaction is started, software can monitor the BUSY status or enable interrupt to notify the completion of the transaction. Please note the transfer between SPI Flash and DMA SRAM is word (32-bit) base only.

SPF does not support DTR (Dual Transfer Rate) and AX read modes.

SPF supports QPI mode; however, software is responsible to issue configuration commands to initialize QPI mode of the SPI Flash before setting and entry of QPIMODE.

Software is also responsible for monitoring and checking the status of SPI Flash for successful erase/write operations as these will cause SPI Flash read operations to be ignored.

For data integrity, SPF allows ECC checking of the SPI Flash data. ECC only applies for the transaction for DMA transfer. When ECC is enabled, the 24-bit address is shifted left by one bit, and the data is transferred to the even byte address, and the ECC (in nibble base) data is written or read/checked on the odd byte. Please note ECC is not performed on the command, address, dummy bytes, and data bytes (if any). ECC encode and decode are only applied on the data stream to and from DMA SRAM. ECC can be turned on/off when there is no transaction; however, software needs to ensure the data/ECC coherence.

SPFCFGA (0x4000_2000) SPI Flash Controller Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SPFEN	SPICLKS[2-0]			SPIMODE	QPIMODE	OEDGE	IEDGE
WR	SPFEN	SPICLKS[2-0]			SPIMODE	QPIMODE	OEDGE	IEDGE

SPFEN — SPI Flash Controller Enable
SPFEN=1 Enables SPI Flash Controller.
SPFEN=0 Put SPI Flash Controller in disabled state. This also clears all internal state machine.

SPICLKS[3-0] — SPI Flash Clock Setting
000 = SYSCLK/2
001 = SYSCLK/4
010 = SYSCLK/6
011 = SYSCLK/8
100 = SYSCLK/10
101 = SYSCLK/12
110 = SYSCLK/16
111 = SYSCLK/32

SPIMODE — SPI Mode 0/3 Select
SPIMODE=0, Mode 0. In Mode 0, the idle state of SPFCLK is low. The sampling edges are defined by OEDGE and IEDGE setting.
SPIMODE=1, Mode 3. In Mode 1, the idle state of SPFCLK is high. The sampling edges are defined by OEDGE and IEDGE setting.

QPIMODE — QPI Mode Setting
0 = SDI and SDO standard SPI mode
1 = SDIO[3-0] QPI mode
Please note software must issue the command to configure the SPI Flash into corresponding mode before setting or clearing QPIMODE bit.

OEDGE — Output Sampling Edge
OEDGE=0, Output changes at SCK falling edge
OEDGE=1, Output changes at SCK rising edge

IEDGE — Input Sampling Edge
IEDGE=0, Input is sampled at SCK rising edge
IEDGE=1, Input is sampled at next SCK falling edge.

SPFCFGB (0x4000_2001) SPI Flash Controller Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	ADDRBC[1-0]		DMMYCL[3-0]			
WR	-	-	ADDRBC[1-0]		DMMYCL[3-0]			

ADDRBC[1-0] Address Field Byte Count
00 = 24-bit Address, ADDR[23-0] is used.
01 = 16-bit Address, only ADDR[15-0] is used.
10 = 8-bit Address, only ADDR[7-0] is used.
11 = 24-bit Address same as 00.

DMMYCL[3-0] Dummy Cycle Count
Define the transaction dummy cycle count. Default to 0.

SPFCMDA (0x4000_2004) SPI Flash Controller Command Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TRANX[3-0]				-		FAIL	BUSY
WR	TRANX[3-0]				-		-	START

TRANX[3-0] SPF Transaction
0000 CMD + DMMY
0001 CMD + ADDR + DMMY
0010 CMD + DMMY + DATAOUT
0011 CMD + DMMY + DATAIN
0100 CMD + ADDR + DATAOUT
0101 CMD + ADDR + DATAIN
0110 CMD + ADDR + DMMY + DATAOUT
0111 CMD + ADDR + DMMY + DATAIN
1000 CMD + ADDR + DMA DATAOUT
1001 CMD + ADDR + DMA DATAIN
1010 CMD + ADDR + DMMY + DMA DATAOUT
1011 CMD + ADDR + DMMY + DMA DATAIN
1100 CMD + ADDR + DMA DATAOUT, ECC
1101 CMD + ADDR + DMA DATAIN, ECC
1110 CMD + ADDR + DMMY + DMA DATAOUT, ECC
1111 CMD + ADDR + DMMY + DMA DATAIN, ECC
Please note,
1. For DATAOUT and DATAIN transactions, data register (DATA[31-0]) can support up to 4 bytes. For DATAIN transaction, if DATABC[7-0] is more than 4, only the last four bytes are kept in DATA[31-0]. For DATAOUT transaction, if DATABC[7-0] is more than 4, DATA[31-0] is transmitted in repetitive fashion.
2. For DMA related transaction, it should always be word aligned. The DATABC[1-0] is ignored.
3. For any transaction involving SPI flash write and erase, it is the software's responsibility to ensure the SPI flash write time requirement. It is recommended only one byte or word write should be issued in one transaction. This is especially true for 1010 and 1011.
4. For 1100 to 1111, ECC is assumed. Only the actual data is in the DMA SRAM. Nibble based ECC is automatically used.
5. There is no control of contents of data during dummy cycles, therefore any transaction format containing "mode bits" after address field is not supported.
6. based ECC is automatically used.

FAIL Transaction Result
This is a read-only bit reflecting the same value in FAIL bit of SPFCMDC

BUSY Transaction Status
BUSY=1 indicates transaction is still in progress.

START Transaction Start
START=1 starts the transaction.

SPFCMDB (0x4000_2005) SPI Flash Controller Command Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DATABC[7-0]							
WR	DATABC[7-0]							

DATABC[7-0] DATA Byte Count
For non-DMA transactions, the number of data bytes for the transaction is DATABC[7-0]+1 and it should be less than 8.
For DMA transactions, it must be on quad byte base, so DATABC[1-0] are ignored, and the number of data word count is DATABC[7-2]+1.

SPFCMDC (0x4000_2008) Interrupt and ECC Error Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	IEN	INTF	DMARXF	FAIL	IEN1	IEN0	ERRF1	ERRF0
WR	IEN	INTF	DMARXF	FAIL	IEN1	IEN0	ERRF1	ERRF0

IEN Interrupt Enable
IEN=1 enables interrupt when SPF transaction is completed.

INTF Interrupt Flag
INTF is set by hardware when interrupt is triggered. INTF must be cleared by software by writing 1.

DMARXF DMA Data In Interrupt Flag
For DMA DATA IN transactions, DMARXF is set to 1 after all DMA data has been transferred. For other type transactions, this flag may be set to 1 after the SPI transfer finished. It should be clear by software

FAIL Transaction Failure
FAIL is set by hardware when the transaction does not complete without error. This must be cleared by software. Same as the FAIL bit in SPFCMDA register.

IEN1 Interrupt Enable for Un-correctable Error
If IEN1=1, an un-correctable error will generate an ECC interrupt.

IEN0 Interrupt Enable for Correctable Error
If IEN0=1, a correctable error will generate an ECC interrupt.

ERRF1 Un-correctable Error Flag
ERRF1 is set to 1 when un-correctable error occurs for IEN1=1. This bit needs to be cleared by software writing 1.

ERRF0 Correctable Error Flag
ERRF0 is set to 1 when correctable error occurs for IEN0=1. This bit needs to be cleared by software writing 1.

SPFADR (0x4000_2010) SPF Flash ADDR Configuration Register R/W (0x0000)

	31-24	23-16	15-8	7-0
RD	CMD[7-0]	ADDR[23-16]	ADDR[15-8]	ADDR[7-0]
WR	CMD[7-0]	ADDR[23-16]	ADDR[15-8]	ADDR[7-0]

SPFDAT (0x4000_2014) SPF Flash DATA Configuration Register R/W (0x0000)

	31-24	23-16	15-8	7-0
RD	DATA[31-24]	DATA[23-16]	DATA[15-8]	DATA[7-0]
WR	DATA[31-24]	DATA[23-16]	DATA[15-8]	DATA[7-0]

SPFDMA (0x4000_201C) SPF DMA SRAM Configuration Register R/W (0x0000)

	31-24	23-16	15-8	7-0
RD	DMARAD [15-0]		DMAWAD15-0]	
WR	DMARAD [15-0]		DMAWAD[15-0]	

DMARAD[15-0] DMA SRAM Read Address
The data transfer is in word base. The LSB two bits are forced 0.

DMAWAD[15-0] DMA SRAM Write Address

The data transfer is in word base. The LSB two bits are forced 0.

Please note both read or write addresses are updated by hardware pointing to next start location to support continuous multiple transactions. This characteristic is different from other DMA peripherals.

10.1 SPF SDI/SDO Timing Waveform

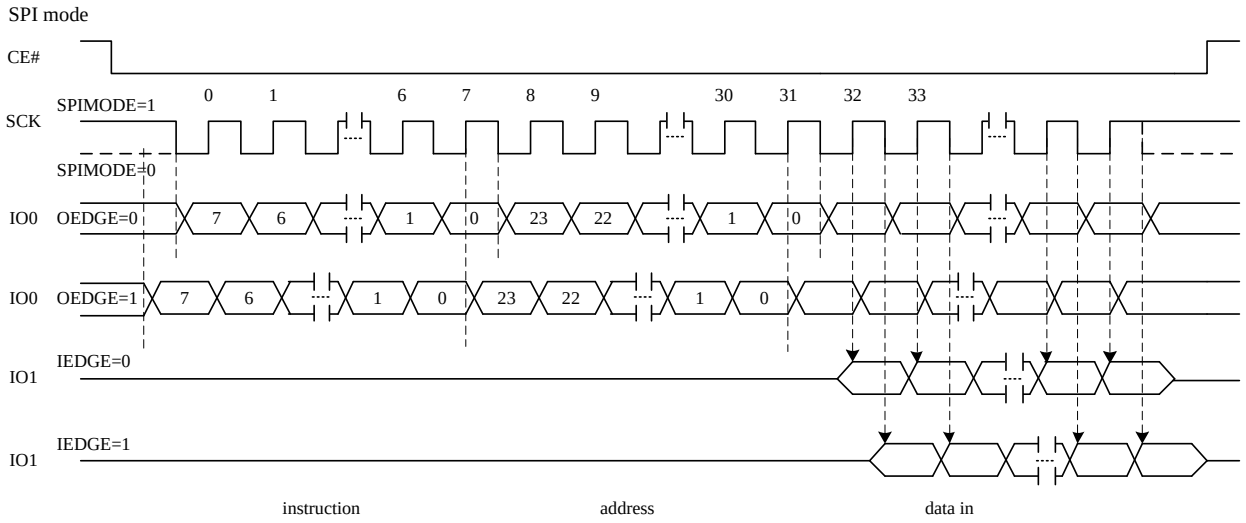


Figure 10-1 SPF SPI Mode Timing Diagram

10.2 SPF QPI Mode Timing

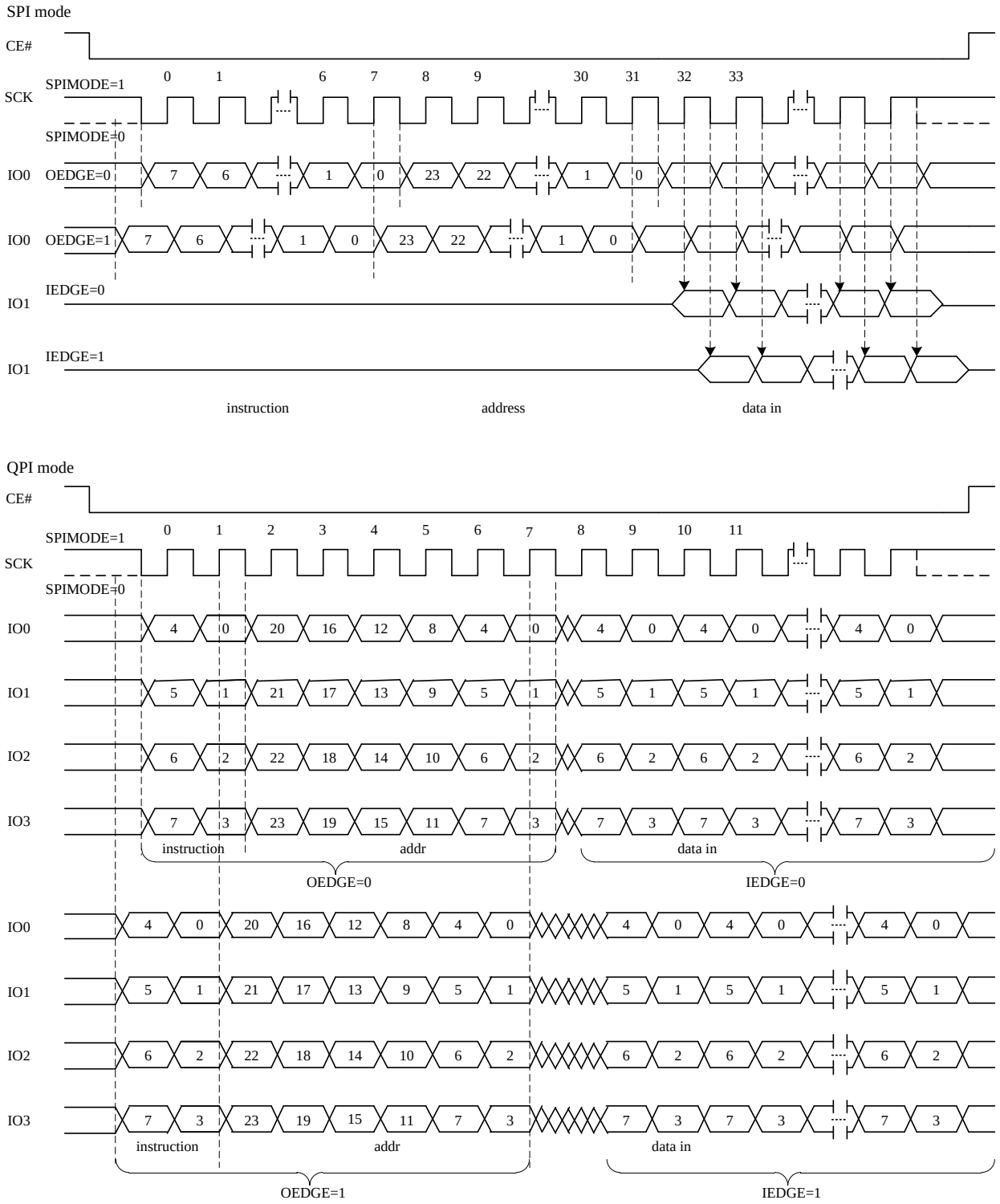


Figure 10-2 SPF QPI Mode Timing Diagram

11. Serial Video Bus Controller (SPV)

SPV is a SPI master similar to SPF but optimized for VSB (Video Serial Bus) applications. VSB is used to connect cascaded daisy-chain LED drivers with shift registers for display information. SPV usually have SCK, SDO/SDI and SCS (LE), and optional OE interface signal to control IO buffer. Several possible protocols exist for this purpose. SPV provides a flexible means to accommodate these variations.

The serial data send out is from SPVDAT then follows from DMA. Up to 4-bytes can be from SPVDAT and up to 256 bytes from DMA. SPV accesses DMA in 32-bit word width, however the interface is 8-bit byte width. Unaligned accesses are extended with zero.

SPVCFGA (0x4000_3000) SPV Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SPVEN	SDIOEN	SDOEDGE	SDIEDGE	SPICLKS[3-0]			
WR	SPVEN	SDIOEN	SDOEDGE	SDIEDGE	SPICLKS[3-0]			

SPVEN — SPV Controller Enable

SDIOEN — SDIO Enable
SDIOEN=0 use separate SDO and SDI as output and input pins
SDIOEN=1 use SDO as output and SDO as input as well. This in effect functions as SDIO bidirectional pin.

SDOEDGE — Output Sampling Reference Edge
SDOEDGE=0, Rising edge
SDOEDGE=1, Falling edge

SDIEDGE — Input Sampling Reference Edge Select
SDIEDGE = 0, SI samples at CLK rising edge
SDIEDGE = 1, SI samples at CLK falling edge

SPICLKS[3-0] — SPI Flash Clock Setting
0000 = SYSCLK/2
0001 = SYSCLK/4
0010 = SYSCLK/6
0011 = SYSCLK/8
0100 = SYSCLK/10
0101 = SYSCLK/12
0110 = SYSCLK/14
0111 = SYSCLK/16
1000 = SYSCLK/20
1001 = SYSCLK/24
1010 = SYSCLK/28
1011 = SYSCLK/32
1100 = SYSCLK/36
1101 = SYSCLK/40
1110 = SYSCLK/64
1111 = SYSCLK/128

SPVCFGB (0x4000_3001) SPV Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LEEN	CNTCLK	-	LEPOL	-	-	-	-
WR	LEEN	CNTCLK	-	LEPOL	-	-	-	-

LEEN — Latch Function Enable
LEEN=0 SCS function for normal SPI mode.
LEEN=1 LE(LAT) mode. Once LEEN is set, the LATX bit of SPVCMDA should be set as 1.

CNTCLK — Continuous Clock Enable
CNTCLK=1 enables continuous clock output.

LEPOL — LE Polarity
LEPOL=0, the LE output asserted state is logic high.
LEPOL=1, the LE output asserted state is logic low.

LEPOL effect is immediate. Please note changing LEPOL setting should only be done when transaction is not ongoing.

SPVCFGC (0x4000_3004) SPV Configuration Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LEDLY[7-0]							
WR	LEDLY[7-0]							

LEDLY[10-3] LE Delay

SPVCFGD (0x4000_3005) SPV Configuration Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-				LEDLY[11-8]			
WR	-				LEDLY[11-8]			

LEDLY[2-0] LE Delay
The total delay is LEDLY[10-0] clocks.

SPVCFGE (0x4000_3006) SPV Configuration Register E R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LEWID[7-0]							
WR	LEWID[7-0]							

LEWID[7-0] LE Width
The total width is LEWID[7-0] + 1 clocks.

SPVCMDA (0x4000_3008) SPV Command Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LATX	CMDBEN	CMDBL[1-0]		TRANX[2-0]			BUSY
WR	LATX	CMDBEN	CMDBL[1-0]		TRANX[2-0]			START

LATX Latch Pulse Enable
LATX=0 disables LE output for the transaction. This is default.
LATX=1 allows normal LE output function.

CMDBEN Command Bit Enable
CMDBEN=1 will insert command bits at the start of the transaction. The number of command bits is defined in CMDBL[1-0].
CMDBEN=0 is normal and nom command bit is sent.

CMDBL[1-0] Command Bit Length
00 = 1-bit, CMD[3] is sent
01 = 2-bit, CMD[3] + CMD[2] sent
10 = 3-bit, CMD[3] + CMD[2] + CMD[1] sent
11 = 4-bit, CMD[3] + CMD[2] + CMD[1] + CMD[0] sent

TRANX[2-0] SPV Transaction
000 Reserved
001 DATAOUT (DATABC)
010 DMAOUT (DMABC)
011 DATAOUT (DATABC) + DMAOUT (DMABC)
100 DATAOUT (DATABC) and DATAIN (DATABC)
101 DATAIN (DATABC), CMDBEN must be 0.
110 DMAIN (DMABC), CMDBEN must be 0.
111 Reserved

BUSY Busy Status
BUSY is 1 when the transaction is still ongoing

START Start SPV Transaction
START=0, no effect
START=1, start the transaction

SPVCMDB (0x4000_3009) SPV Command Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CMD[3-0]				DATABC[3-0]			
WR	CMD[3-0]				DATABC[3-0]			

CMD[3-0] Command Bits

DATABC DATA Register Byte Count

DATABC defines the number of bytes for data register output and input. Since the maximum data byte is only 8, DATABC should not be set more than 8 to prevent overrun.

The DATABC[3-0] couldn't be 0 if CMDBEN is 1

SPVCMDC (0x4000_300A) SPV Command Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DMABC[7-0]							
WR	DMABC[7-0]							

DMABC[7-0] DMA Access Byte Count

The number of bytes for DMA access is DMABC[7-0]+1. Maximum is 256 bytes.

SPVINT (0x4000_300C) Interrupt Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	IEN	-	-	-	-	-	-	INTF
WR	IEN	-	-	-	-	-	-	INTF

IEN Interrupt Enable

IEN=1 enables SPV interrupt

INTF Interrupt Flag

INTF is set to 1 by hardware when transaction completes. This bit needs to be cleared by software writing 1.

SPVDATOL (0x4000_3010) SPV Data Out Register L R/W (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	SPVDATO[31-24]	SPVDATO[23-16]	SPVDATO[15-0]	SPVDATO[7-0]
WR	SPVDATO[31-24]	SPVDATO[23-16]	SPVDATO[15-0]	SPVDATO[7-0]

SPVDATOH (0x4000_3014) SPV Data Out Register H R/W (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	SPVDATO[63-56]	SPVDATO[55-48]	SPVDATO[47-40]	SPVDATO[39-33]
WR	SPVDATO[63-56]	SPVDATO[55-48]	SPVDATO[47-40]	SPVDATO[39-33]

SPFDATO[63-0] SPF Data Out Register

SPVDATIL (0x4000_3018) SPV Data In Register L R/W (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	SPVDATI[31-24]	SPVDATI[23-16]	SPVDATI[15-0]	SPVDATI[7-0]
WR	SPVDATI[31-24]	SPVDATI[23-16]	SPVDATI[15-0]	SPVDATI[7-0]

SPVDATIH (0x4000_301C) SPV Data In Register H R/W (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	SPVDATI[63-56]	SPVDATI[55-48]	SPVDATI[47-40]	SPVDATI[39-33]
WR	SPVDATI[63-56]	SPVDATI[55-48]	SPVDATI[47-40]	SPVDATI[39-33]

SPFDATI[63-0] SPF Data In Register

SPVDMA (0x4000_3020) SPV DMA SRAM Configuration Register R/W (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	DMARAD [15-0]		DMAWAD[15-0]	
WR	DMARAD [15-0]		DMAWAD[15-0]	

DMARAD[15-0] DMA SRAM Read Address
The data transfer is in word base. The LSB two bits are forced 0.

DMAWAD[15-0] DMA SRAM Write Address
The data transfer is in word base. The LSB two bits are forced 0.

11.1 SPV Examples

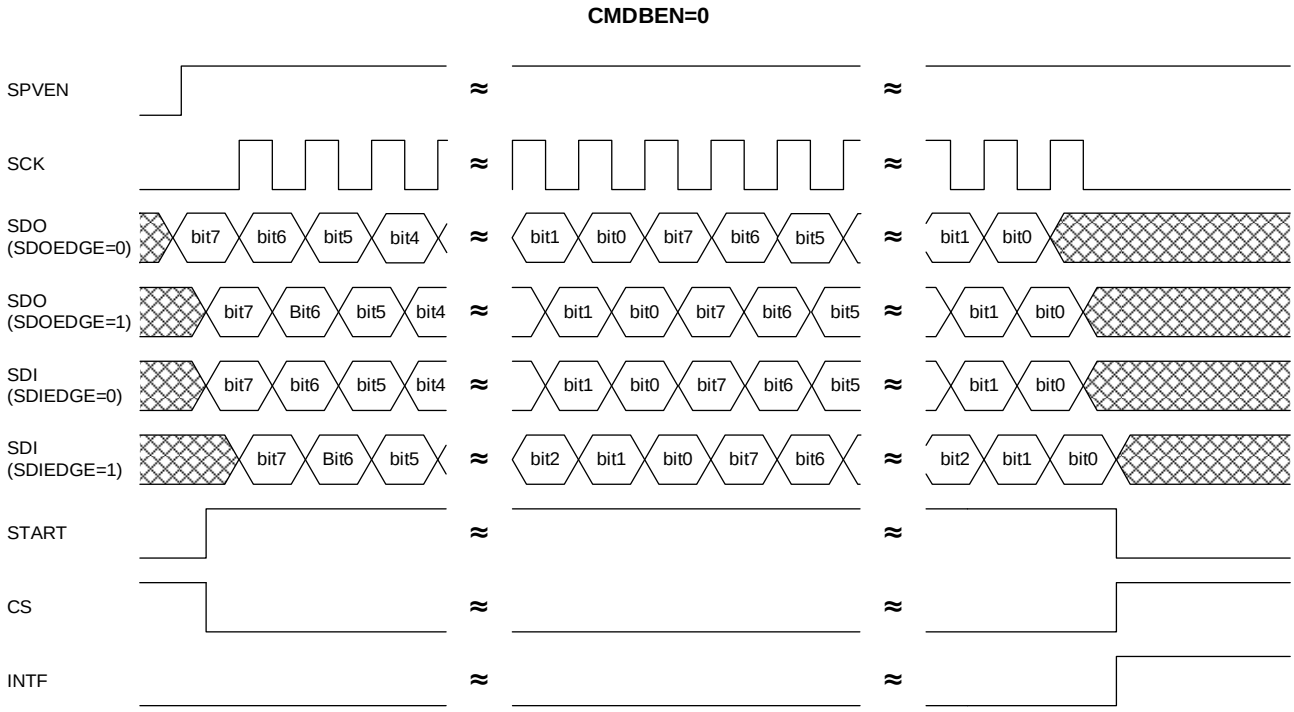


Figure 11-1 SPV waveform without command

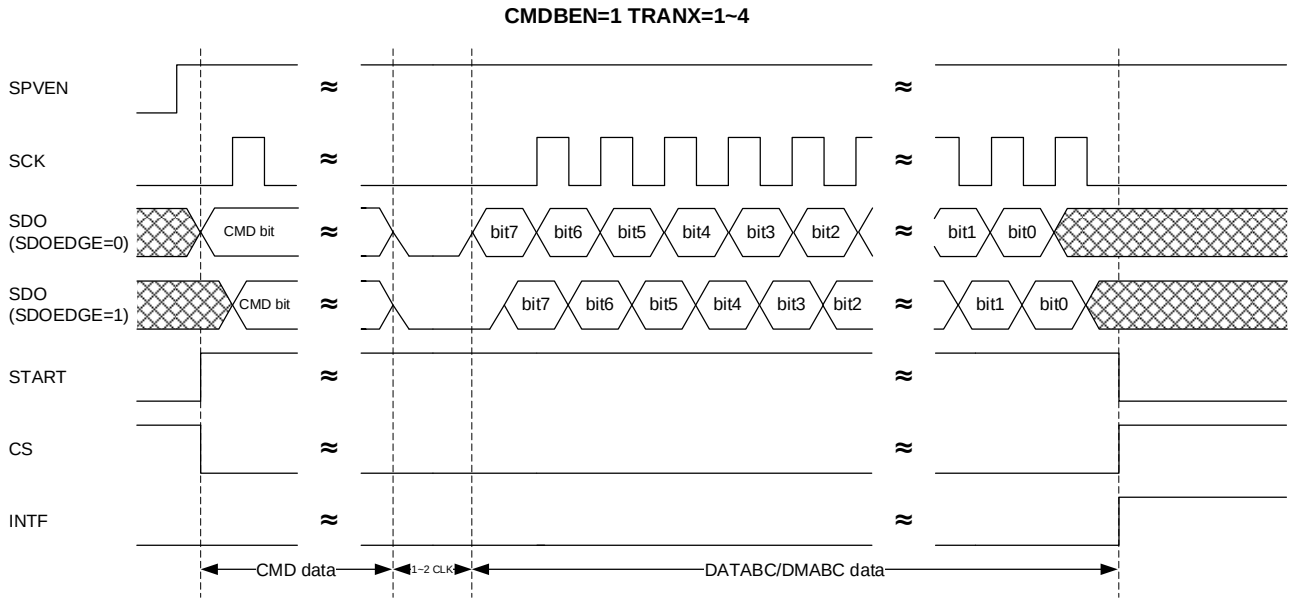


Figure 11-2 SPV waveform with command

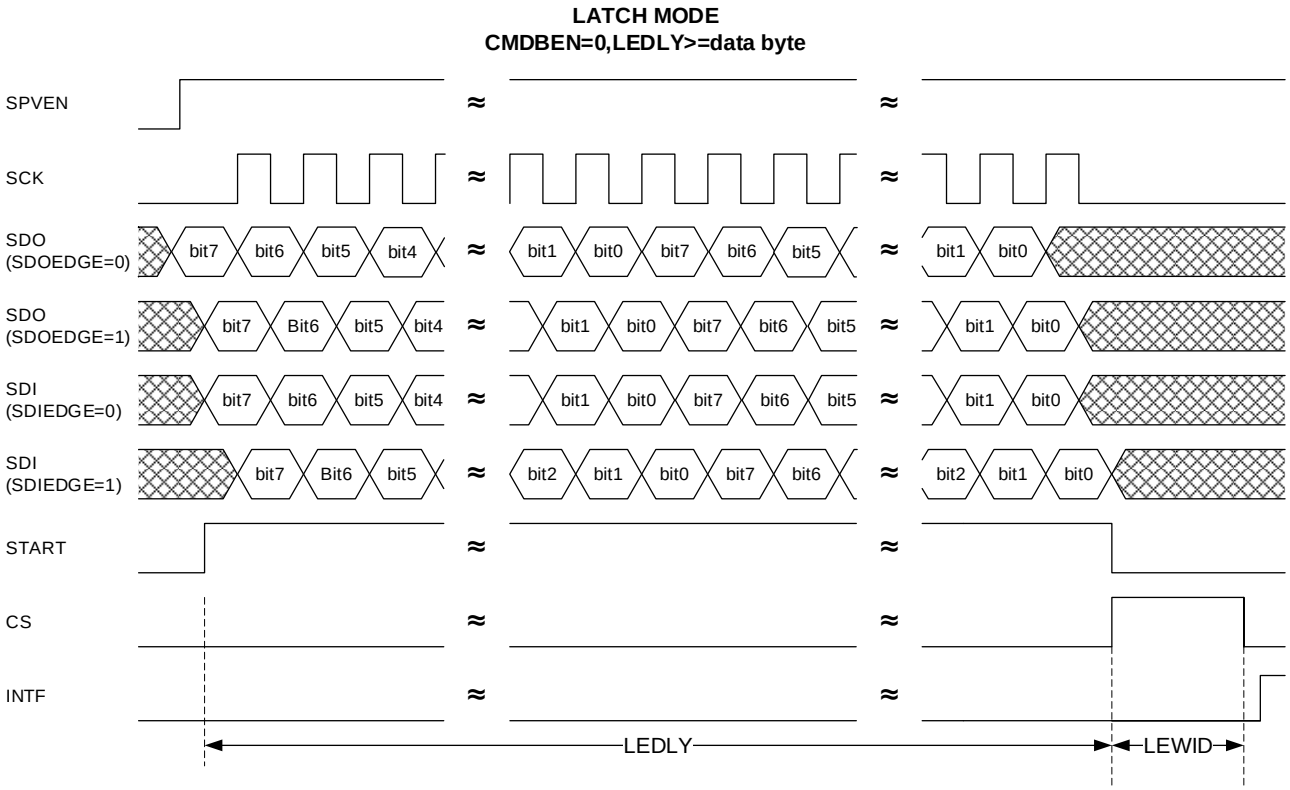


Figure 11-3 SPV waveform for latch mode without command and LEDLY >= data byte

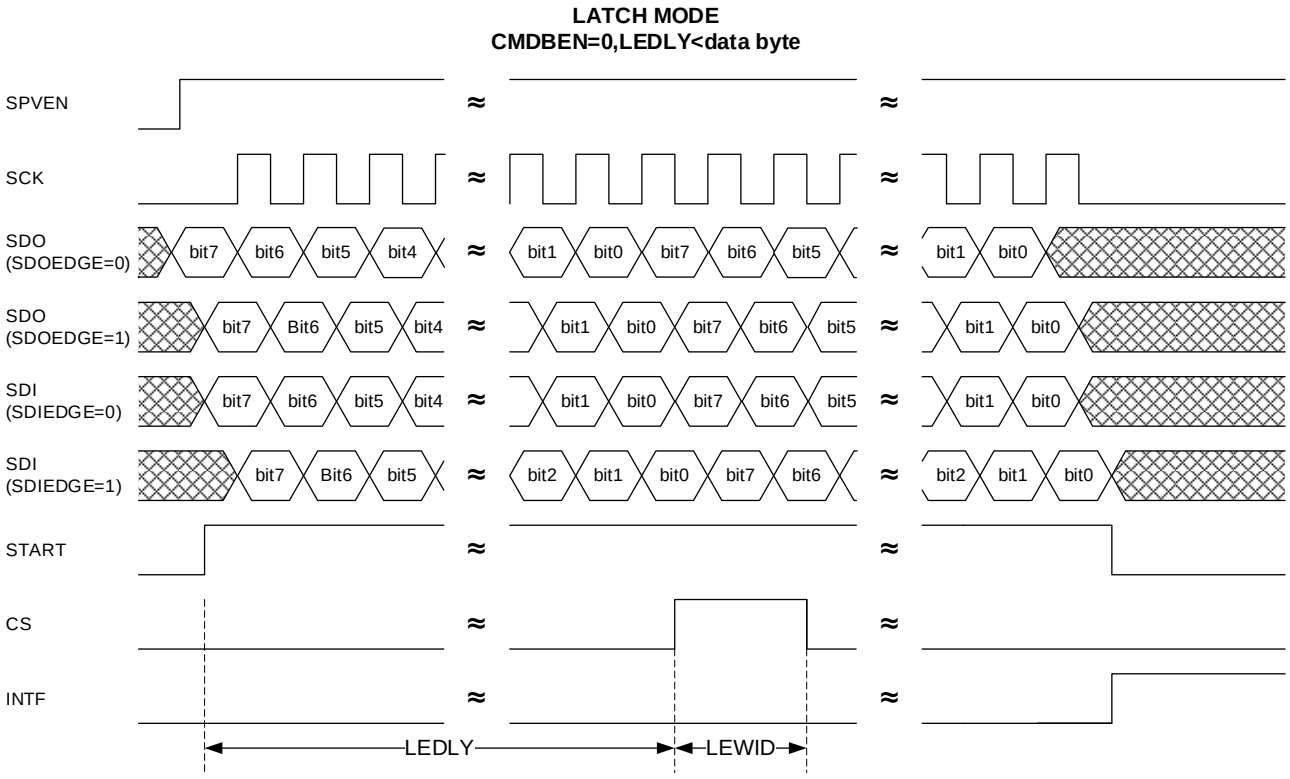


Figure 11-4 SPV waveform for latch mode without command and LEDLY < data byte

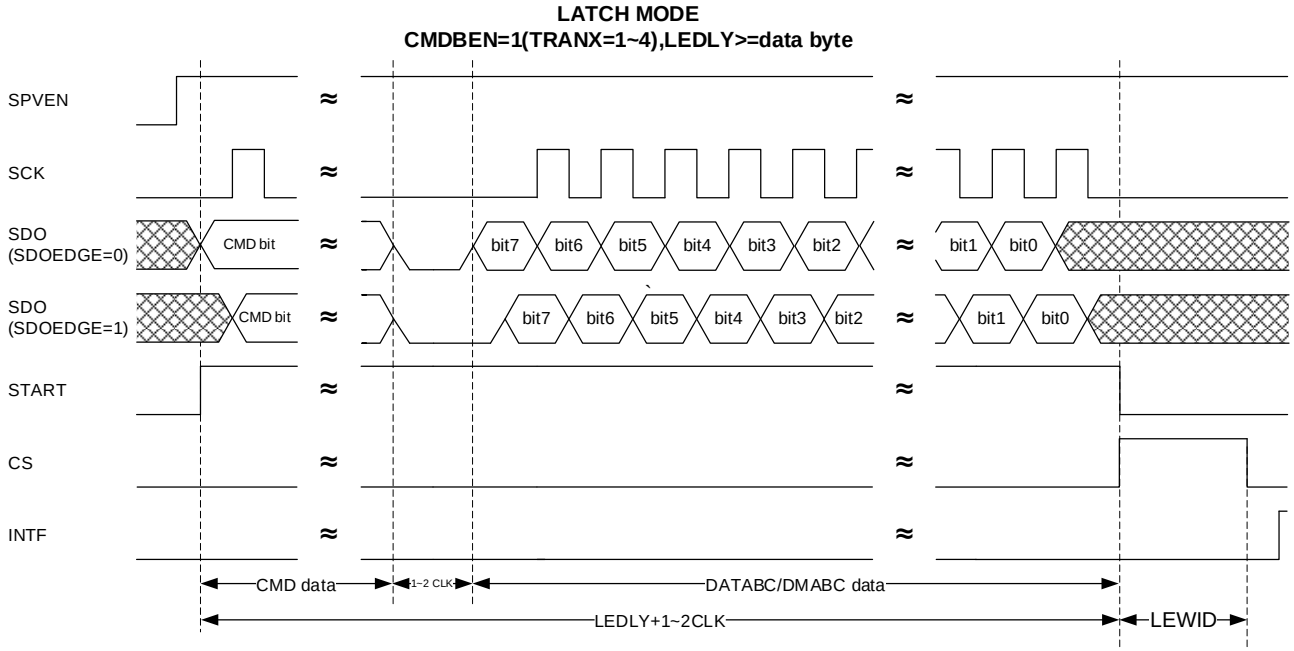


Figure 11-5 SPV waveform for latch mode with command and LEDLY >= data byte

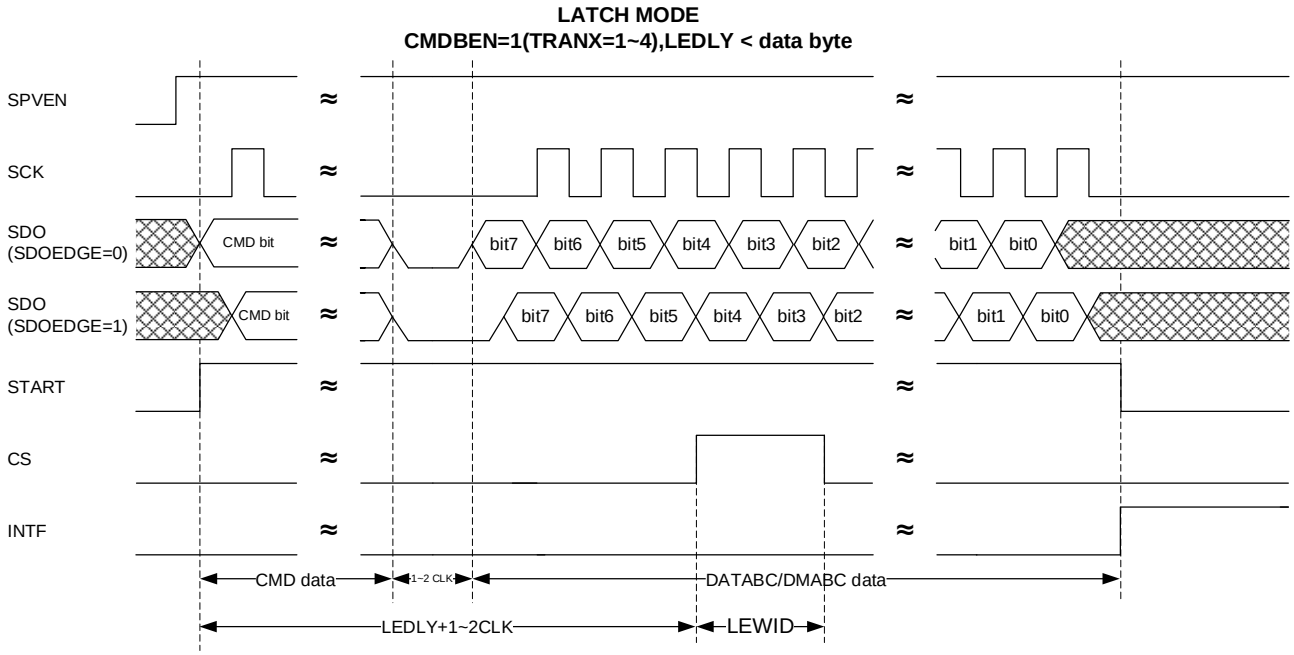


Figure 11-6 SPV waveform for latch mode with command and LEDLY < data byte

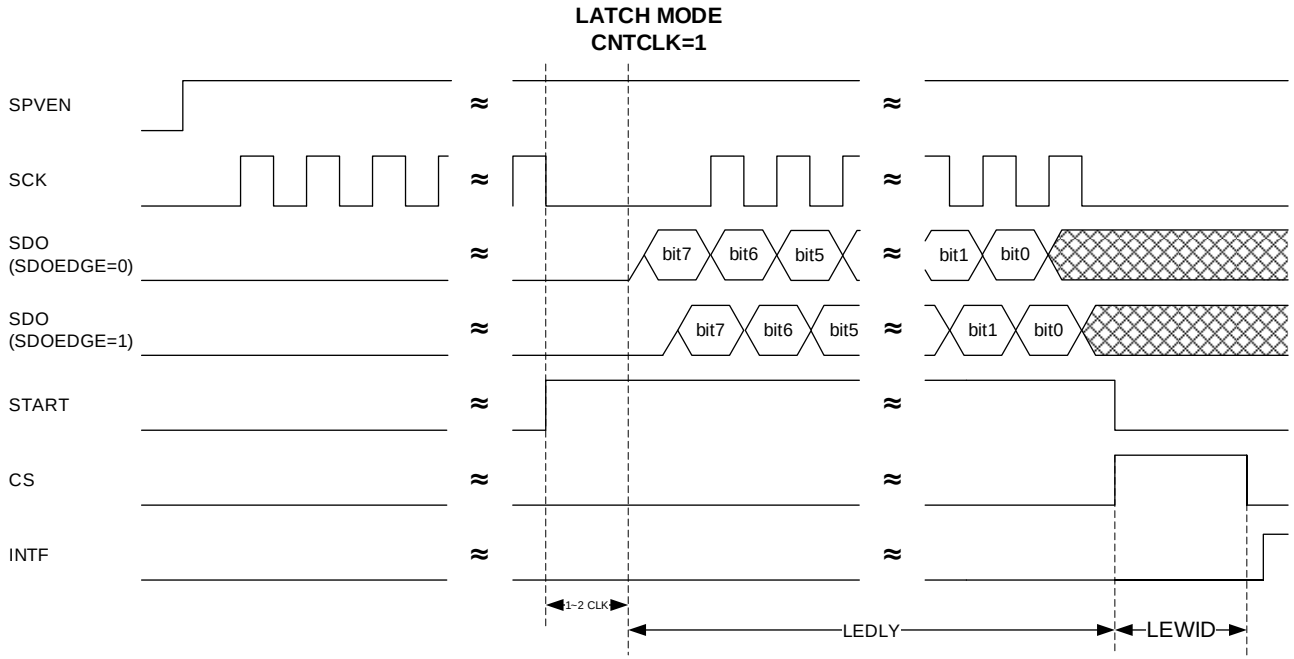


Figure 11-7 SPV waveform for latch mode and CNTCLK = 1

12. Lumibus Master Controller (LMC)

Lumibus is a proprietary bus developed by Lumissil for controlling LED drivers. It provides high speed bidirectional communication between the host and the slave LED drivers while offering reliable operation under harsh environments.

Lumibus can have several connection means depending on the specific application requirements. The first is a wired-or bus structure. This is illustrated in the following diagram.

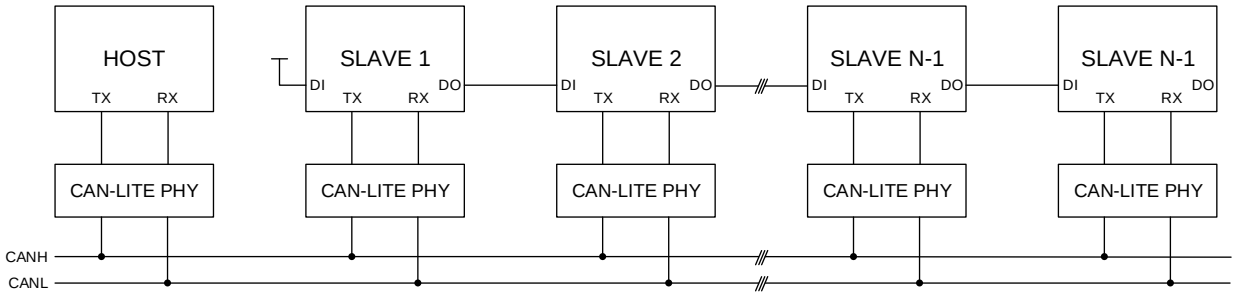


Figure 12-1 Lumibus Devices Connection 1

Using CAN-like bus structure, CAN-LITE bus can be either at 5V or 3.3V swing for reliable communication. DI and DO are daisy-chained to provide self-addressing capability. If self-addressing is not needed, DI/DO can be omitted.

An alternative Lumibus configuration is illustrated in the following diagram. In this configuration, each slave has two CAN-LITE PHY, and are connected in cascading daisy chain fashion. The interconnections between host and with each slave is short and local. The data stream is retimed by each slave to preserve timing accuracy, and also, the slave can control the direction of data flow, for example passing the data received from upstream to downstream or opposite direction. It can also block the downstream data and inject upstream data. The direction of flow is defined by the Lumibus protocol. Because the slaves are connected in daisy chain, self-addressing of slaves can be achieved through command protocol without additional pins. This configuration also provides up to 256 nodes, because the data is buffered and retimed by each node. The bit rate is limited by the total delay including the retime and buffers. The default configuration is bus equivalent i.e., each slave passing the data from upstream to downstream.

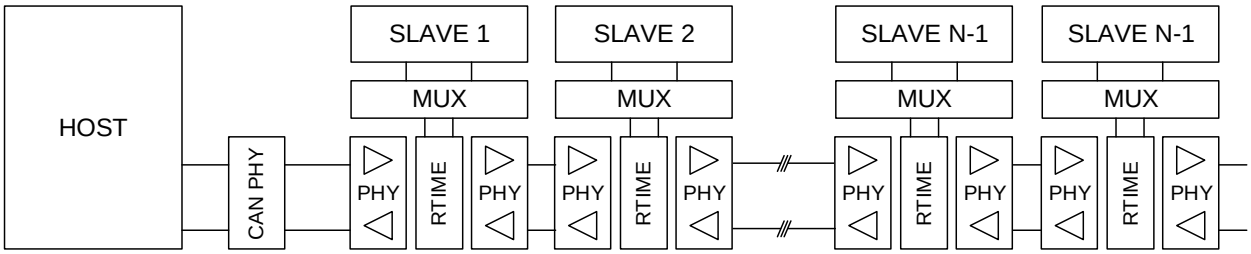


Figure 12-2 Lumibus Devices connection 2

12.1 Byte Unit

The basic data transmission format of Lumibus follows UART and LIN format. A transmission unit is byte based and preceded by a low start bit and ended with a high stop bit in total of 10-bit time. The data field is the same as UART where LSB precedes MSB.

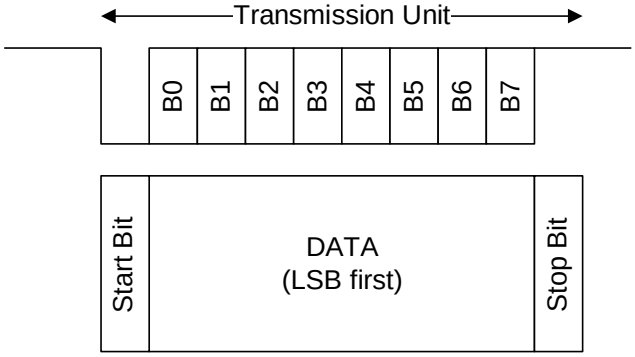


Figure 12-3 Byte Unit

12.2 Bus Reset

Bus reset is an extended period of bus low (dominant) time longer than 1msec. Master sends Bus Reset to reset all slaves.

12.3 Bus Break and Sync (BBS)

Master periodically sends BBS frame to synchronize the slaves for updating baud rate accuracy. The BBS consists of Bus Break (a bus low of 16BT of 100Kbps) followed by a DATA=55 in operation baud rate. This is shown in following timing diagram. Here BTL is defined by BRL register and should be set as close as possible to 100Kbps. BT is defined in BR register which is used for operation baud rate.

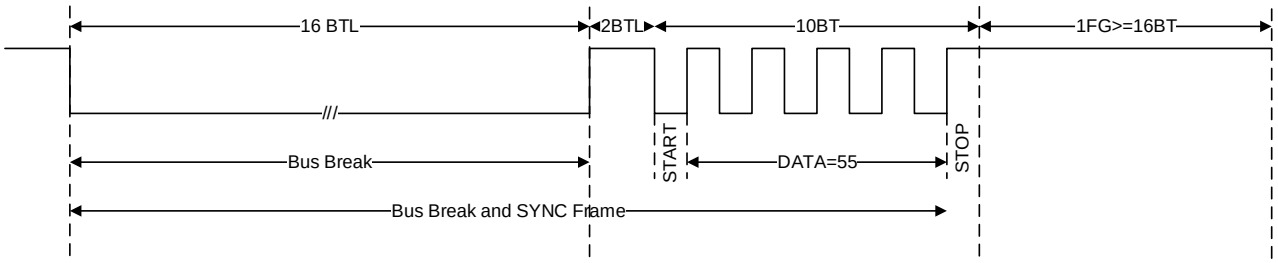


Figure 12-4 BBS Frame

Please also note, each frame should be separated by IFG (Inter-frame Gap) which is greater or equal to 16BT.

12.4 Frame Formats

For command and response protocol, Lumibus uses several formats, these include and are not limited to the following definitions. CRC checksum is a 16-bit value with LSB byte preceding MSB byte using CRC-16-IBM (X^16 + X^15 + X^2 + 1) with seed value of 0x0000. CRC calculation includes all fields of the packet. Please also note, the CRC data must be calculated using software and packed by software as a complete command frame. Same also applies for received response frame where CRC check is performed by software. Software can utilize CRC accelerator for this purpose.

Write Command Frame (N is 1 to 16)

CMD Frame Header	Device ID	Register Address	N Bytes of Data	CRC checksum

Special Command Frame

CMD Frame Header	Device ID	CRC checksum

Read Command Frame

CMD Frame Header	Device ID	Register Address	CRC checksum

Response Frame (N is 1 to 16)

RSP Frame Header	Device ID	N Bytes of Data	CRC checksum

Most common transactions are for host to send Write Command for slaves. Each slave receives the command frame and also retimes the frame and forward to downstream. The slave decodes the command and processes the command if it match its Device ID. The Device ID can also be set as broadcast ID, so all slaves will process the command. Typical timing of this transaction is shown in the following figure. Inter Frame Gap Host (IFGH) can be down to just 1BT, but it is recommended to be 16BT.

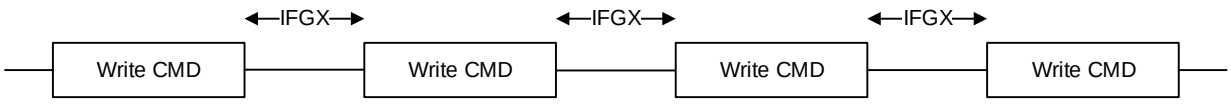


Figure 12-3 Lumibus Write Protocol

The host can also issue Read command to obtain status or data from the slaves. The response frames can consist of multiple slaves spaced by IFGS (Inter Frame Gap Slave) typically 12 BT. The read transaction completes with IFGH (>=32 BT) idle time. The host can issue next command frame after IFGH.

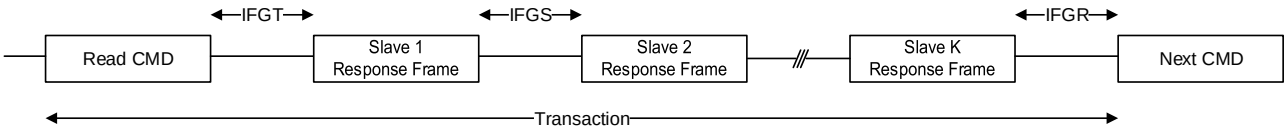


Figure 12-4 Lumibus Read Protocol

12.5 DMA Data Access

Lumibus controller uses DMA to access the command frame and stores the received response frames to DMA.

12.6 Registers

LMCCFGA (0x4000_4000) Lumibus Controller Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	BR[7-0]							
WR	BR[7-0]							

LMCCFGB (0x4000_4001) Lumibus Controller Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LMCEN	RCVSPL[2-0]			BR[11-8]			
WR	LMCEN	RCVSPL[2-0]			BR[11-8]			

LMCEN — Lumibus Controller Enable
LMCEN=0 disables the controller also clears the internal state machine

RCVSPL[1-0] — Adjust Receive Sampling Point
The receive sampling point is set at 0.5BT + 2 SYSCLK * RCVSPL[2-0]

BR[11-0] — Baud Rate Setting
Baud Rate is set as LMC clock/(BR[11-0]+1)
BR must be greater than 15. If less than 15, it is treated as 15. This means maximum baud rate is LMC clock/16.

LMCCFGC (0x4000_4002) Lumibus Controller Configuration Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	BRL[7-0]							
WR	BRL[7-0]							

BRL[7-0] — Low Baud Rate Setting
Low baud rate is set as LMC clock/(16*BRL[7-0]+1) should be as close as possible to 100KHz.
This is used to generate bus break timing as well as wake up frame.

LMCCFGD (0x4000_4003) Lumibus Controller Configuration Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	INTEN[1-0]		IFGX	IFGT	IFGR[1-0]		IFGS[1-0]	
WR	INTEN[1-0]		IFGX	IFGT	IFGR[1-0]		IFGS[1-0]	

INTEN[1-0] — Interrupt Enable
00 = Interrupt is disabled
01 = One response frame is received to trigger interrupt.
10 = four response frame is received to trigger interrupt.
11 = Interrupt is triggered only if IFGR is met
For 01/10 setting, the IFGR trigger is always valid.
For setting of 01/10/11, an interrupt is generated when either a transmit or receive command sequence (or frame) is completed.

IFGX — IFGX Setting
This is used to insert the delay for interrupt to ensure two consecutive transmit frames is spaced.
IFGX=0 1BT

IFGX=1 16BT

IFGT — IFGT Setting
0 = 256BT
1 = 1024BT
This is used for time out for the first response frame after Read CMD. If the first slave does not respond with this time limit, the Read CMD is aborted.

IFGR[1-0] — IFGR Setting
This setting is used to delimit the last response frame of read command frame. This is also added to the additional delay before command completion interrupt.
00 = 32BT
01 = 36BT
10 = 40BT
11 = 64BT

IFGS[1-0] — IFGS Setting
This is used for LMC to determine the delimitation of received response frame when read command is issued.
00 = 12BT
01 = 14BT
10 = 16BT
11 = 18BT

LMCCMDA (0x4000_4004) Lumibus Controller Command Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	STOPBC	-	-	-	-	-	-	BUSY
WR	STOPBC	-	-	-	-	CMD[2-0]		

STOPBC — Stop Bit Count
STOPBC=0, the byte boundary stop bit is 2 bit-time
STOPBC=1, the byte boundary stop bit is 1 bit-time

BUSY — Busy Status

CMD[2-0] — Command
000 = No Effect
001 = Send Bus Reset
010 = Send SYNC only
011 = Send Bus Break and SYNC
100 = Send Bus Break
101 = Initiate Command Sequence defined by DMA.
110 = Initiate Command Sequence defined by DMA and complete the receive transactions
111 = Abort current command sequence. This also clears all status bits.

LMCCMDB (0x4000_4005) Lumibus Controller Command Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CMDBC[7-0]							
WR	CMDBC[7-0]							

CMDBC[7-0] — Command Byte Count
The number of byte for the command is CMDBC[7-0]+4. Therefore, the minimum number of byte count is 4 for shortest command.

CMD HEADER 1B	Device ID 1B	CRC checksum 2B

LMCCMDC (0x4000_4006) Lumibus Controller Command Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	INTF	BERR	OVWC	OVFC	IFGTTO	IFGSTO	IFGRTO	IFGXTO
WR	INTF	-	-	-	-	-	-	-

INTF — Interrupt Flag

	INTF is set by hardware when interrupt is generated usually when command is executed. INTF must be cleared by software writing 1. This action also clears RCVWC[9-0] and RCVFC[7-0] as well as other status when OVWC or OCFC is 1.
BERR	Bit Error Flag If received RX is not equal to TX, BERR is set to 1. The bit error check is enabled by BEREN in LCMTST. Cleared when INTF is cleared.
OVWC	Receive Frame Word Count overflow, refer to LMCSTAA/LMCSTAB for detail. Cleared when INTF is cleared.
OVFC	Receive Frame Count overflow, refer to LMCSTAC for detail. Cleared when INTF is cleared.
IFGTTO	IFGT Time Out Cleared when INTF is cleared.
IFGSTO	IFGS Time Out IFGSTO triggering conditions are affected by INTEN[1-0]. For example, if INTEN[1-0]=10, then four IFGSTO have occurred to set IFGSTO, meaning four frames have been received. Cleared when INTF is cleared.
IFGRTO	IFGR Time Out IFGRTO is set to indicate a transaction has completed. Cleared when INTF is cleared.
IFGXTO	IFGX Time Out Cleared when INTF is cleared.

LMCCMDD (0x4000_4007) Lumibus Controller Command Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TXPOL	BEREN	LBKEN	SELFTST	CLRFIFO	RXACT	STSTF	FMERR
WR	TXPOL	BEREN	LBKEN	SELFTST	CLRFIFO	-	-	-

TXPOL	TX Polarity Setting TXPOL=0 for normal polarity. TXPOL=1 for reverse polarity.
BEREN	Bit Error Check Enable BEREN=1 turns on RX/TX comparison check.
LBKEN	Loopback Enable LBKEN=1 connects RX input to TX output forming self-test condition.
SELFTST	Self-Test Setting SELFTEST=1 starts the self-test through loopback TX to RX and compare. It is cleared to 0 by hardware when test is done. The test result is reflected on STSTF status bit. To perform self-test, LBKEN must be 1.
CLRFIFO	Clear FIFO Set CLRFIFO=1 will clear FIFO.
RXACT	Receive Active RXACT is read-only and when RXACT=1 reflects the ongoing receive action.
STSTF	Self-Test Result STSTF is set to 1 after self-test is completed and failed. STSTF is cleared when INTF is cleared.
FMERR	Frame Error FMERR is set to 1 if basic UART format is violated such as start-bit, stop-bit missing. FMERR is cleared when INTF is cleared.

LMCSTAA (0x4000_4008) Lumibus Controller Status Register A RO (0x00)

	7	6	5	4	3	2	1	0
RD	RCVWC[7-0]							
WR	-							

RCVWC[7-0]	Receive Word Count

LMCSTAB (0x4000_4009) Lumibus Controller Status Register B RO (0x00)

	7	6	5	4	3	2	1	0
RD	-				RCVWC[9-8]			
WR	-				-			

RCVWC[9-0] Received Word Count

Number of word received in the transaction is RCVWC[9-0]. RCVWC[9-0] is updated when IFGSTO or IFGRTO interrupts. These two conditions require software to process the received frames as soon as possible.

For a frame byte length not word aligned, the word count is rounded up because DMA is always word based.

Please note if actual RCVWC exceeds 1023 words, the counter value is maxed to 1023 and OVWC is asserted. Further receptions are ignored.

LMCSTAC (0x4000_400C) Lumibus Controller Status Register C RO (0x00)

	7	6	5	4	3	2	1	0
RD	RCVFC[7-0]							
WR	-							

RCVFC[7-0] Received Frame Count

Number of frames received since last IFGSTO or IFGRTO interrupts is RCVFC[7-0] frame. RCVFC[9-0] is updated when IFGSTO or IFGRTO interrupts.

The maximum number of response frames that can be received is 255. Frames exceeding 255 will be discarded and data ignored with OVFC asserted.

LMCDMA (0x4000_4010) LMC DMA SRAM Configuration Register R/W (0x0000_0000)

	31-24	23-16	15-8	7-0
RD	DMARAD[15-0]		DMAWAD[15-0]	
WR	DMARAD[15-0]		DMAWAD[15-0]	

DMARAD[15-0] DMA SRAM Read Address

The data transfer is in word base. The LSB two bits are forced 0.

DMAWAD[15-0] DMA SRAM Write Address

The data transfer is in word base. The LSB two bits are forced 0.

LMCPCS (0x4000_4014) Lumibus Controller Clock Pre-Scale Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CPS[7-0]							
WR	CPS[7-0]							

CPS[7-0] Clock Pre-Scaling Setting

LMC clock is derived as SYSCLK/(CPS[7-0]+1). For SYSCLK <= 32MHz, CPS[7-0] can be set to default 0. For higher SYSCLK frequency, CPS[7-0] can be set accordingly for proper baud rate.

13. CAN FD Controller

The Controller Area Network (CAN) is a serial asynchronous bus allowing multi-master communication. The CAN bus uses a single terminated twisted pair with maximum data rate of 1Mbps. Typical length of the bus can be up to 40 meters. The CAN protocol is message-based in nature. Nodes on the bus do not have specific addresses. Instead, the messages have unique identifiers which are used for determining message priority. Each node, depending on its function, transmits specific types of messages and also responds to specific related messages. CAN is especially popular in connecting electronic control modules, sensors, and actuators in automotive and industrial applications because of its simple bus structure, fault-tolerance, and extensive error checking.

The physical media of a typical CAN bus is a twisted pair terminated with a 120Ohm resistor on both ends. The twisted pair has two signals, CANH and CANL. CANH and CANL form a 5V pseudo differential signal thus having reliable transmission under high noise environments. There are two possible logic states on the bus. Dominant state (logic 0) is when CANH is actively pulled to 5V and CANL is actively pulled to 0V. Recessive state (logic 1) is when the bus is not actively driven and both CANH and CANL are pulled toward 2.5V by the termination resistors. The bus also exhibits the wire-AND characteristics, i.e., bus is in recessive state only if all nodes are recessive, and the bus is in dominant state if one or more nodes are in dominant state. Because of this pseudo differential nature, it is possible to maintain correct data transmission even if one of the wires is broken. To drive the CAN bus, typically an external transceiver is used to provide isolated drive.

Features

- Mixed CAN 2.0B and CAN Flexible Data (FD) mode
- Arbitration Bit Rate up to 1Mbps
- CAN FD Data Bit Rate up to 10Mbps
- Support Standard 11-bit and Extended 29-bit ID
- 64-byte Transmit Double Buffer for FD data
- DMA Interface to transfer Received Data
- Four Receive Acceptance ID Filters and Mask
- Bus Error Handling and Auto-Recovery
- Listen-Only mode
- Self-Test mode
- Self-Loop Test Mode

13.1 Register Map (Base Address 0x4000_5000)

ADDR	Operating Mode		Reset Mode	
	Read	Write	Read	Write
B+0x000	Mode Register	Mode Register	Mode Register	Mode Register
B+0x004	00H	Command Register	00H	——
B+0x008	Status Register	——	Status Register	——
B+0x00C	Interrupt Register	——	00H	——
	Interrupt Enable Register	Interrupt Enable Register	Interrupt Enable Register	Interrupt Enable Register
B+0x010	Bus Timing Register 0	——	Bus Timing Register 0	Bus Timing Register 0
	Bus Timing Register 1	——	Bus Timing Register 1	Bus Timing Register 1
B+0x014	FD Timing Register 0	——	FD Timing Register 0	FD Timing Register 0
	FD Timing Register 1	——	FD Timing Register 1	FD Timing Register 1
	FD SSP Select Register	——	FD SSP Select Register	FD SSP Select Register
B+0x018	Filter Enable Register 0	——	Filter Enable Register 0	Filter Enable Register 0
	Filter Status Register 0	——	——	——
	Arbitration Lost Capture Register	——	Arbitration Lost Capture Register	——
B+0x01C	Error Code Capture Register	——	Error Code Capture Register	——
	Error Warning Limit Register	——	Error Warning Limit Register	Error Warning Limit Register
	Receive Error Counter Register	——	Receive Error Counter Register	Receive Error Counter Register

ADDR	Operating Mode		Reset Mode	
	Read	Write	Read	Write
	Transmit Error Counter Register	——	Transmit Error Counter Register	Transmit Error Counter Register
B+0x020	Filter 0 Acceptance Code 0	——	Filter 0 Acceptance Code 0	Filter 0 Acceptance Code 0
	Filter 0 Acceptance Code 1	——	Filter 0 Acceptance Code 1	Filter 0 Acceptance Code 1
	Filter 0 Acceptance Code 2	——	Filter 0 Acceptance Code 2	Filter 0 Acceptance Code 2
	Filter 0 Acceptance Code 3	——	Filter 0 Acceptance Code 3	Filter 0 Acceptance Code 3
B+0x024	Filter 0 Mask Code 0	——	Filter 0 Mask Code 0	Filter 0 Mask Code 0
	Filter 0 Mask Code 1	——	Filter 0 Mask Code 1	Filter 0 Mask Code 1
	Filter 0 Mask Code 2	——	Filter 0 Mask Code 2	Filter 0 Mask Code 2
	Filter 0 Mask Code 3	——	Filter 0 Mask Code 3	Filter 0 Mask Code 3
B+0x028	Filter 1 Acceptance Code 0	——	Filter 1 Acceptance Code 0	Filter 1 Acceptance Code 0
	Filter 1 Acceptance Code 1	——	Filter 1 Acceptance Code 1	Filter 1 Acceptance Code 1
	Filter 1 Acceptance Code 2	——	Filter 1 Acceptance Code 2	Filter 1 Acceptance Code 2
	Filter 1 Acceptance Code 3	——	Filter 1 Acceptance Code 3	Filter 1 Acceptance Code 3
B+0x02C	Filter 1 Mask Code 0	——	Filter 1 Mask Code 0	Filter 1 Mask Code 0
	Filter 1 Mask Code 1	——	Filter 1 Mask Code 1	Filter 1 Mask Code 1
	Filter 1 Mask Code 2	——	Filter 1 Mask Code 2	Filter 1 Mask Code 2
	Filter 1 Mask Code 3	——	Filter 1 Mask Code 3	Filter 1 Mask Code 3
B+0x030	Filter 2 Acceptance Code 0	——	Filter 2 Acceptance Code 0	Filter 2 Acceptance Code 0
	Filter 2 Acceptance Code 1	——	Filter 2 Acceptance Code 1	Filter 2 Acceptance Code 1
	Filter 2 Acceptance Code 2	——	Filter 2 Acceptance Code 2	Filter 2 Acceptance Code 2
	Filter 2 Acceptance Code 3	——	Filter 2 Acceptance Code 3	Filter 2 Acceptance Code 3
B+0x034	Filter 2 Mask Code 0	——	Filter 2 Mask Code 0	Filter 2 Mask Code 0
	Filter 2 Mask Code 1	——	Filter 2 Mask Code 1	Filter 2 Mask Code 1
	Filter 2 Mask Code 2	——	Filter 2 Mask Code 2	Filter 2 Mask Code 2
	Filter 2 Mask Code 3	——	Filter 2 Mask Code 3	Filter 2 Mask Code 3
B+0x038	Filter 3 Acceptance Code 0	——	Filter 3 Acceptance Code 0	Filter 3 Acceptance Code 0
	Filter 3 Acceptance Code 1	——	Filter 3 Acceptance Code 1	Filter 3 Acceptance Code 1
	Filter 3 Acceptance Code 2	——	Filter 3 Acceptance Code 2	Filter 3 Acceptance Code 2
	Filter 3 Acceptance Code 3	——	Filter 3 Acceptance Code 3	Filter 3 Acceptance Code 3
B+0x03C	Filter 3 Mask Code 0	——	Filter 3 Mask Code 0	Filter 3 Mask Code 0
	Filter 3 Mask Code 1	——	Filter 3 Mask Code 1	Filter 3 Mask Code 1
	Filter 3 Mask Code 2	——	Filter 3 Mask Code 2	Filter 3 Mask Code 2
	Filter 3 Mask Code 3	——	Filter 3 Mask Code 3	Filter 3 Mask Code 3
B+0x040	Transmit Frame Information Register	Transmit Frame Information Register	——	——
B+0x044	Transmit ID Register	Transmit ID Register	——	——
B+0x048 | B+0x084	Transmit data register 0 | Transmit data register 63	Transmit data register 0 | Transmit data register 63	——	——
B+0x090	Receive Frame Information Register	——	——	——

ADDR	Operating Mode		Reset Mode	
	Read	Write	Read	Write
B+0x094	Receive ID Register	——	——	——
B+0x098 | B+0x0D4	Receive data register 0 | Receive data register 63	——	——	——
B+0x0F0	DMA Start Address 0	——	DMA Start Address 0	DMA Start Address 0
	DMA Start Address 1	——	DMA Start Address 1	DMA Start Address 1
	DMA Memory Size	——	DMA Memory Size	DMA Memory Size
B+0x0F4	DMA RX Current Address 0	——	DMA RX Current Address 0	——
	DMA RX Current Address 1	——	DMA RX Current Address 1	——
	DMA RX Current Length	——	DMA RX Current Length	——
	DMA RX Frame Number	——	DMA RX Frame Number	——
B+0x0FC	——	——	——	Loop Test Security Key
	——	——	——	Loop Test Enable

Table 13-1 CAN Bus Register Map

B = Base Address

13.2 Configuration, Control and Status Registers

CANMODE0 (0x4000_5000) CAN Mode Configuration Register R/W (0x01)

	7	6	5	4	3	2	1	0
RD	FDEN	FDTOL	PTOEXC	PHYOFF	ERAR	STE	LOM	RSTM
WR	FDEN	FDTOL	PTOEXC	PHYOFF	ERAR	STE	LOM	RSTM

FDEN — FD Mode Enable
FDEN=1 enable FD mode, able to receive and to transmit FD frames, as well as Classical frames
FDEN=0 disable FD mode
This register can only be modified in reset mode (RSTM=1).

FDTOL — FD Tolerant Mode
FDTOL=1 and FDEN=0 enable FD tolerant mode. Support of the classical CAN frame format only, but not disturbing FD frames.
FDTOL=0 and FDEN=0 only support the classical CAN frame and will disturbing FD frames.
This register can only be modified in reset mode (RSTM=1).

PTOEXC — Protocol Exception Option Enable
PTOEXC =1 enable the protocol exception option in FD mode. CAN controller shall enter the bus integration state and wait bus idle condition when it detects the res bit to be recessive instead of the expected dominant value.
PTOEXC=0 disable the protocol exception option. The FD enabled node, detecting the res bit following the FDF bit to be recessive shall treat this as a form error.
This register can only be modified in reset mode (RSTM=1).

PHYOFF — CAN Transceiver Bus Off
PHYOFF directs to the can_bus_off output. The can_bus_off indicates to turn-off the external CAN transceiver.
This register can only be modified in reset mode (RSTM=1).

ERAR — Error Auto Recovery
ERAR=1 will allow auto recovery from bus off state
When this bit is set, auto recovery from Bus Off state to Error Active state is enabled. The auto recovery condition is met when 128 occurrences of bus-free time (11 consecutive ‘recessive’ bits) is received. When exiting the Bus Off state, TEC and REC are cleared to 0 by hardware if ERAR=1.

ERAR=0 for disable the auto recovery function.
If ERAR=0, software must clear REC and TEC to exit of Bus Off state to Error Active state. Please note in the entry of Bus Off state, if ERAR=0, RSTM will be set by hardware, so modification of REC and TEC is allowed. After clearing REC and TEC, the software must also clear RSTM bit to return to normal operation.
This register can only be modified in reset mode (RSTM=1).

STE — Self Test Enable
STE=1 enables self-test mode.
When STE=1, the only difference from normal operation mode is that all ACK errors are ignored.
STE=0 for normal mode operation
This register can only be modified in reset mode (RSTM=1).

LOM — Listen-Only Mode
LOM=1 will set the CAN controller as listen-only mode
When LOM=1, CAN controller will work in Listen-Only mode. In this mode, the CAN controller will only perform receiver functions and does not engage any transmission (including ACK signaling). All other functions can be used like in Operating mode. In this mode, the error counters are stopped at the current value and message transmission is not possible. But when detecting bus error, listen-only node will still transmit error frame.
LOM=0 for normal mode operation
This register can only be modified in reset mode (RSTM=1).

RSTM — Reset Mode
RSTM=1 will force the CAN Controller into rest mode regardless of other setting, and will initialize all state machine. It does not affect the register contents. RSTM defaults to 1 after system reset. RSTM is set to 1 by hardware when entering into Bus-Off state if ERAR=0.
RSTM must be cleared to 0 to allow the CAN Controller into normal operation.
Please note RSTM must be set to 1 first in order to modify the other registers in CANMODE. The write cycle of RSTM must be preceding the modification cycles of FDEN, PTOEXC, PHYOFF, ERAR, STE, and LOM.

CANMODE1 (0x4000_5001) CAN Mode Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	SLEEP	DMAEN
WR	-	-	-	-	-	-	SLEEP	DMAEN

SLEEP — SLEEP Mode
Set SLEEP to 1 to indicate that CAN Controller is going to enter the SLEEP mode. This register is used to enable the CAN bus wakeup detector. Hardware will clear this register automatically when detect the rx_i falling edge from CAN transceiver. Meanwhile, if the WKIE is set, the WKI interrupt will be issued. In order to detect the valid wakeup signal from CAN bus, this register shall be programed during CAN bus idle (rx_i is voltage high).
This register can only be modified in reset mode (RSTM=1).

DMAEN — RX DMA Mode Enable
DMAEN=1 enable RX DMA mode
DMAEN=0 disable RX DMA mode
In RX DMA mode (DMAEN=1), the dma_req is asserted when there are one or more valid frames stored in the RX-FIFO. The DMA host shall read out all received data through APB bus from address B+90h to B+D4h. One PHCLK delay is inserted by pulling PREADYOUT to low when reading the RX Buffer. After receive the dma_ack and dma_finish, hardware will release the receive buffer automatically.
If RX DMA mode is disabled (DMAEN=0), the received frame only be read through APB read command. One PHCLK delay is inserted by pulling PREADYOUT to low when reading the RX Buffer SFRs which address is from B+90h to B+D4h. The dma_req, dma_single and dma_last are always driving zero. Software must release the RX Buffer by setting RRB command after read out all necessary data.
This register can only be modified in reset mode (RSTM=1).

CANMODE3 (0x4000_5003) CAN Mode Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CANCLK_SEL		CANCLK_DIV					
WR	CANCLK_SEL		CANCLK_DIV					

CANCLK_SEL — CAN clock source selection. The tolerance of clock oscillator frequency shall be less than ±0.2%. The maximum operating frequency of the CANCLK is 80MHz.
CANCLK_SEL = 00, CAN clock source is System clock.
CANCLK_SEL = 01, CAN clock source is PLL output.
CANCLK_SEL = 10, CAN clock source is external clock (XOSC).
CANCLK_SEL = 11, CAN clock source is internal clock (IOSC).

CANCLK_DIV — CAN clock source divider. CANCLK = CAN clock source / (CANCLK_DIV + 1)

CANCMD (0x4000_5004) CAN Command Register WO (0x00)

	7	6	5	4	3	2	1	0
RD	0	0	0	0	0	0	0	0
WR	-	CTB	OFR	SRR	CDO	RRB	AT	TR

CTB — Clear Transmit Buffer
Set CTB=1 will cancel the unexecuted Transmit Request (TR) command and release the transmit buffer for any message revision. This command is only valid when the transmit buffer is occupied by an unsent message with TBS=0. This bit is self-cleared by hardware when TBS is set. Since CTB is only cancel the unexecuted transmission in the transmit buffer, the extra AT command is necessary to abort the pending transmission in CAN MAC controller if you are going to abort all transmissions.

OFR — Overload Frame Request
OFR=1 will request transmitting an overload frame. The overload frame is transmitted the first bit time of next expected INTERMISSION. This bit is self-cleared by hardware after the overload frame is transmitted. Since the overload frame will be transmitted regardless of the bus condition, no interrupt is generated at the completion.

SRR — Self Receive Request
SRR=1 will result a message be transmitted and received simultaneously. Upon self reception request command, a message is transmitted and simultaneously received if the acceptance filter is set to the corresponding identifier. This command is used mainly for self-test of the controller. If STE=1 which enables the self-test mode, the transmit and receive can be successful without other nodes present. This bit is cleared by hardware when transmit and receive operations are completed and both transmit and receive interrupts will be generated. SRR is only valid when both CTB=0 and TR=0.

CDO — Clear Data Overrun
CDO=1 will clear data overrun status bit. This command bit is used to clear the data overrun condition indicated by the data overrun status bit. This command bit is self cleared by hardware.

RRB — Release Receive Buffer
After reading the contents of the receive buffer, the CPU can release this memory space in the RX-FIFO by setting the Release Receive Buffer bit to logic 1. This may result in another message becoming immediately available within the receive buffer. If there is no other message available, the Receive Buffer Status (RBS) bit is reset. RRB is relevant only when DMAEN=0.

AT — Abort Transmitting
AT=1 will abort the any pending transmission request. AT does not stop an on-going message transmission. But the next transmission attempt will be aborted. The completion of the abortion will cause TCS=1 in CANSTAT register. AT should be used only for aborting a message transmission with errors (mostly in ACK error). As AT cannot guarantee to stop an on-going message transmission, the software should check Bus Error Flag (BEI) bit and Arbitration Lost Flag (ALI) when TCS=1. TCS=1 can be resulted by either abortion or success of transmission. If BEI=0 and ALI=0, then the message is sent successfully even abort command is issued.

Otherwise, the transmission was actually aborted because of bus error or arbitration lost.

TR — Transmit Request

TR=1 will start the transmit process of the message in the transmit buffer.

While setting TR =1, the message in the transmit buffer including CANTFIR, CANTXID0~3 and CANTXDTA0~63 will be delivered to CAN MAC controller for transmit process if the CAN bus was idle. When the message is delivered to CAN MAC controller, the transmit buffer will be free for next message from CPU. Meanwhile the TBS and the TBI will be set to notice Users. TR is only valid when CTB=0.

CANSTAT (0x4000_5008) CAN Status Register RO (0x0C)

	7	6	5	4	3	2	1	0
RD	BOS	ES	TS	RS	TCS	TBS	DOS	RBS
WR	-	-	-	-	-	-	-	-

BOS — Bus Off State Status

BOS=1 indicates the CAN controller is in the Bus Off State.

BOS=0 indicates the CAN controller is involved in bus activities.

Bus Off state is defined in the error handling state diagram. Possible exit of Bus Off state is by Bus Auto recovery, or by writing a value less than 255 into TEC under reset mode.

ES — Error Status

ES=1 indicates at least one of the error counters has reached or exceeded the warning limit defined by the Error Warning Limit register.

ES=0 indicates warning limits are not exceeded.

Errors detected during reception or transmission will affect the error counters according to the CAN 2.0B protocol specification. ES is set when at least one of the error counters has reached or exceeded the warning limit setting (EWLR). An error warning interrupt is generated, if enabled. The default value of EWLR after hardware reset is 0x60.

TS — Transmit Status

TS=1 indicates CAN controller is transmitting a message.

RS — Receive Status

RS=1 indicates CAN controller is receiving a message.

If both TS and RS are 0, the CAN controller is idle.

TCS — Transmit Complete Status

TCS=1 indicates last requested message transmission has been successfully completed.

TCS is cleared to 0 when TR or SRR command is issued. And TCS is set to 1 by hardware when the message is sent successfully. TCS remains to be 0 during the message transmission, and TCS is set to 1 by hardware if the message is sent successfully. Please note TCS=1 can also be resulted by AT command.

TBS — Transmit Buffer Status

TBS=1 indicates the transmit buffer is free and a new message can be written.

TBS=0 indicates the transmit buffer is locked and the content in the transmit buffer is either waiting for transmission or in the process of being transmitted. And no new message could be written.

DOS — Data Overrun Status

DOS=1 indicates a received message was lost due to RX-FIFO overrun. The overrun interrupt is generated with DOI flag set.

DOS must be cleared by setting CDO (Clear Data Overrun) command bit or by setting to reset mode.

RBS — Receive Buffer Status.

RBS=1 indicates one or more valid messages are available in RX-FIFO.

RBS is cleared when the RX-FIFO is empty.

CANINTF0 (0x4000_500C) CAN Interrupt Flag Register0 RO (0x00)

	7	6	5	4	3	2	1	0
RD	BEI	ALI	EPI	BOI	DOI	EI	TI	RI
WR	-	-	-	-	-	-	-	-

This is a read-only register. The corresponding interrupt enable bit must be enabled for individual flag to be effective. All flags (except BOI) are event change triggering so only the change of the condition will cause the interrupt and corresponding flag to be set. All bits are cleared after read except BOI.

BEI — Bus Error Interrupt Flag
BEI is set when the CAN controller detects error(s) on the CAN-bus. The error types include format error, bit stuff error, bit error (when data or CRC transmitted does not match with received), CRC error, and ACK error. The BEIE bit must be set to enable the BEI function.

ALI — Arbitration Lost Interrupt
ALI is set when a message was not sent successfully due to loss of arbitration. The ALIE bit must be set to enable the ALI function.

EPI — Error Passive Interrupt
EPI is set when the CAN controller enters the error passive state, or the CAN controller is already in the error passive state and enters the error active state. EPIE bit must be set to enable EPI function.

BOI — Bus Off State Interrupt
BOI is set when the CAN controller in the bus off state or equivalently BOS=1. Since BOI and BOS is 1 whenever in Bus Off state, the software must take care to disable BOIE when entering BOI ISR and then correct the BOS status otherwise the interrupt will occur indefinitely. BOIE bit must be set to enable BOI function.

DOI — Data Overrun Interrupt
DOI is set when a received message was lost due to RX-FIFO overrun. The function is equivalently when DOS changes from 0 to 1. DOIE bit must be set to enable DOI function. DOI is cleared when read.

EI — Error Warning Interrupt
EI is set on when REC or TEC is greater or equal to the CANEWLR setting or equivalently when ES changes from 0 to 1. EIE bit must be set to enable EI function.

TI — Transmit Interrupt
TI is set whenever the transmit buffer becomes free or equivalently the transmission of the message has finished and TBS changes from 0 to 1. TIE bit must be set to enable TI function.

RI — Receive Interrupt
RI is set when a new message has been put into the RX-FIFO by the CAN controller. RI is cleared when read. RIE bit must be set to enable RI function.

CANINTF1 (0x4000_500D) CAN Interrupt Flag Register1 RO (0x00)

	7	6	5	4	3	2	1	0
RD	SFALMI	0	0	0	OFI	TBI	DMAI	WKI
WR	-	-	-	-	-	-	-	-

SFALMI — Safety Alarm Interrupt Flag
SFALMI is only implemented in the safety CAN FD controller. In general CAN FD controller, this bit is always tied to 0. SFALMI is set when the safety CAN controller detects any defect in the internal circuit of the CAN FD controller. If SFALMI is set, users shall turn-off the CAN controller then verify the functions by Test Modes. SFALMI is cleared when read.
The behavior of the sf alarm irq output is the same as SFALMI.

OFI — Overload Frame Interrupt Flag
OFI is set when CAN controller detects or issues an overload frame on the CAN-bus. OFI is cleared when read. OFIE bit must be set to enable OFI function.

TBI — Transmit Buffer Free Interrupt Flag
TBI is set when Transmit Buffer is free for new message. The transmit buffer including CANTFIR, CANTXID0~3 and CANTXDTA0~63 are double buffers. TBS

and TBI indicate the transmit buffer is free to accept a new message from CPU. TBI is cleared when read. TBIE bit must be set to enable TBI function

DMAI — RX DMA Complete Interrupt Flag
WKI is set when receive dma_finsih signal under RX DMA mode. DMAI is cleared when read. Both DMAEN and DMAIE must be set to enable DMAI function.

WKI — Wake Up Interrupt Flag
WKI is set when the CAN controller detects a dominant bit on the CAN-bus in the SLEEP mode (SLEEP=1). WKI is cleared when read. Both WKIE and SLEEP bit must be set to enable the WKI function. This interrupt can notice and wake up the controller from CAN-bus.

CANIE0 (0x4000_500E) CAN Interrupt Enable Register0 RW (0x00)

	7	6	5	4	3	2	1	0
RD	BEIE	ALIE	EPIE	BOIE	DOIE	EIE	TIE	RIE
WR	BEIE	ALIE	EPIE	BOIE	DOIE	EIE	TIE	RIE

The interrupt enable register allows indicating different types of interrupt source. The register appears to the CPU as a read/write memory.

BEIE — Bus Error Interrupt Enable
BEI=1 enables bus error interrupt

ALIE — Arbitration Lost Interrupt Enable
ALIE=1 enables arbitration loss interrupt

EPIE — Error Passive Interrupt Enable

BOIE — Bus Off Interrupt Enable

DOIE — Data Overrun Interrupt Enable

EIE — Error Warning Interrupt Enable

TIE — Transmit Interrupt Enable

RIE — Receive Interrupt Enable

CANIE1 (0x4000_500F) CAN Interrupt Enable Register1 RW (0x00)

	7	6	5	4	3	2	1	0
RD	SFALMIE	0	0	0	OFIE	TBIE	DMAIE	WKIE
WR	SFALMIE	-	-	-	OFIE	TBIE	DMAIE	WKIE

SFALMIE — Safety Alarm Interrupt Enable. SFALMIE is only implemented in the safety CAN FD controller. The can_irq will be asserted to inform Host when both SFALMIE and SFALMI are set.

OFIE — Overload Frame Interrupt Enable

TBIE — Transmit Buffer Free Interrupt Enable

DMAIE — RX DMA Complete Interrupt Enable

WKIE — Wake Up Interrupt Enable

CANBTR0 (0x4000_5010) CAN Bus Timing Register 0 R/W (0x01)

	7	6	5	4	3	2	1	0
RD	SJW[1-0]		CANCS[5-0]					
WR	SJW[1-0]		CANCS[5-0]					

SJW[1-0] — Synchronization Jump Length
This defines the maximum adjustment in TQCLK period units for re-synchronization. The receiver uses the falling edges to determine the physical delay of the network. The synchronization is achieved by inserting a delay compensation segment in the bit time after SYNC SEG and thus also delays the sampling points of the local receiver. SJW[1-0]+1 is used to set the maximum allowed delay segment in TQCLK period.

CANCS[5-0] — CANCS[5-0] defines the CAN Controller Clock
The CAN controller clock TQCLK = CANCLK/(CANCS[5-0]+1). This defines the CAN Time Quanta (TQCLK) from the CAN system clock.

CANBTR1 (0x4000_5011) CAN Bus Timing Register 1 R/W (0x67)

	7	6	5	4	3	2	1	0
RD	SAM	TSEG2[2-0]			TSEG1[3-0]			
WR	SAM	TSEG2[2-0]			TSEG1[3-0]			

SAM — SAM=1 will use triple-sampling method. The triple-sampling method uses three instances between TSEG1 and TSEG2 as shown in the following timing diagram. The data is determined by the majority of the sampled results.
SAM=0 will use single-sampling method. The single sampling occurs at the time instance between TSEG1 and TSEG2

TSEG2[2-0] — TSEG2[2-0] defines the bit time component 2. TSEG2 = TSEG2[2-0] + 1.

TSEG1[3-0] — TSEG1[3-0] defines the bit time component 1. TSEG1 = TSEG1[3-0] + 1.

CANFDTR0 (0x4000_5014) CAN FD Timing Register 0 R/W (0x01)

	7	6	5	4	3	2	1	0
RD	FDSJW[1-0]		CANFDCS[5-0]					
WR	FDSJW[1-0]		CANFDCS[5-0]					

FDSJW[1-0] — Data Phase Synchronization Jump Length
This defines the maximum adjustment in FDTQCLK period units for re-synchronization. The receiver uses the falling edges to determine the physical delay of the network. The synchronization is achieved by inserting a delay compensation segment in the bit time after SYNC SEG and thus also delays the sampling points of the local receiver. FDSJW[1-0]+1 is used to set the maximum allowed delay segment in FDTQCLK period.

CANFDCS[5-0] — CANFDCS[5-0] defines the CAN Controller Clock in FD data phase
The CAN controller clock FDTQCLK = CANCLK/(CANFDCS[5-0]+1). This defines the CAN FD Time Quanta (FDTQCLK) from the CAN system clock.

CANFDTR1 (0x4000_5015) CAN FD Timing Register 1 R/W (0x67)

	7	6	5	4	3	2	1	0
RD		FDTSEG2[2-0]			FDTSEG1[3-0]			
WR		FDTSEG2[2-0]			FDTSEG1[3-0]			

FDTSEG2[2-0] — FDTSEG2[2-0] defines the bit time component 2. TSEG2 = FDTSEG2[2-0] + 1.

FDTSEG1[3-0] — FDTSEG1[3-0] defines the bit time component 1. TSEG1 = FDTSEG1[3-0] + 1.

A CAN bit time is partitioned by TQCLK periods as the following diagram. The first CAN TQCLK period is used for edge synchronization, and TSEG1 is used for defining the sampling edge. For SAM=0, the sampling edge is the first edge of TSEG2. For SAM=1, the sampling edges are the last two edges of TSEG1, and the first edge of TSEG2. For typical CAN bus design, TSEG1 and TSEG2 is organized for sampling at the last two quanta (period of TQCLK).

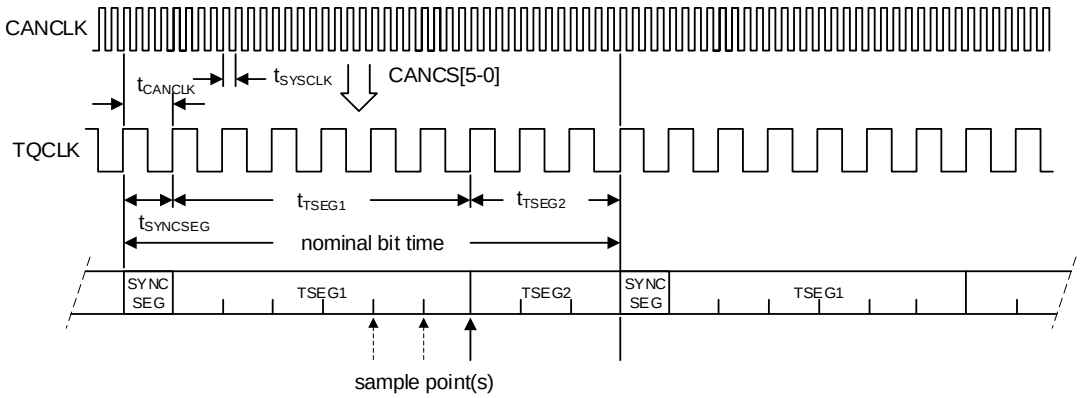


Figure 13-1 CAN bus timing

CAN baud rate thus can be calculated as TQCLK/((TSEG2[2-0]+1)+(TSEG1[3-0]+1)+1). Or in terms of CANCLK,

CANCLK/(CANCS[5-0]+1)/{(TSEG2[2-0]+1)+(TSEG1[3-0]+1)+1}. Also note, with re-synchronization, a delay compensation segment is inserted between SYNC SEG and TSEG1. The maximum length of delay compensation

segment is limited by SJW[1-0].

CANFDSSP (0x4000_5017) CAN FD Secondary Sample Point Select Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SW_SSP	SSP_SEL[2-0]			0	0	0	0
WR	SW_SSP	SSP_SEL[2-0]			-	-	-	-

The transmitter delay mechanism only be used by transmitters in the data phase of FD Frames, when BRS is recessive. In default SW_SSP is disabled, the hardware mechanism calculates and compensate the transceiver's loop delay time automatically. In the general application, SW_SSP shall be disabled unless software gets the accurate loop delay from tx_o to rx_i.

SW_SSP — Software transmitter delay compensation
SW_SSP=0, this mechanism calculates the transceiver's loop delay time and compensate the delay time for comparing the tx_o and rx_i automatically. Reading the SSP_SEL[2-0] is the transmitter delay compensation value that calculated by hardware mechanism.
SW_SSP=1, the transmitter delay compensation SSP_SEL[2-0] is according to software programmed.

SSP_SEL[2-0] — Transmitter delay compensation value
If SW_SSP=0, SSP_SEL[2-0] is a read-only register, and reading the SSP_SEL[2-0] is the transmitter delay compensation value that calculated by hardware mechanism.
If SW_SSP=1, software must configure an accurate loop delay by setting this register. The delay value is depended on the loop delay from tx_o to rx_i which through an external transceiver when transmitting.

SSP_SEL	Transmitter delay compensation time
000	Zero transmitter delay compensation. For loop delay time < TSEG1
001	1 bit time transmitter delay compensation. For TSEG1 <= loop delay < 1 bit time + TSEG1
010	2 bit time transmitter delay compensation. For 1 bit time + TSEG1 <= loop delay < 2 bit time + TSEG1
011	3 bit time transmitter delay compensation. For 2 bit time + TSEG1 <= loop delay < 3 bit time + TSEG1
100	4 bit time transmitter delay compensation. For 3 bit time + TSEG1 <= loop delay < 4 bit time + TSEG1
101	5 bit time transmitter delay compensation. For 4 bit time + TSEG1 <= loop delay < 5 bit time + TSEG1
110, 111	Reserved

CANFTER (0x4000_5018) CAN ID Filter Enable Register 1 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	FTER3	FTER2	FTER1	FTER0
WR	-	-	-	-	FTER3	FTER2	FTER1	FTER0

There are four Acceptance Filters for message ID matching. This register controls the enabling of each filter, multiple filters can be enabled. This register can be modified only in reset mode and appears as read-only in normal operating mode.

FTER3 — Acceptance Filter 3 Enable
FTER3=1 enables Acceptance Filter 3

FTER2 — Acceptance Filter 2 Enable
FTER2=1 enables Acceptance Filter 2

FTER1 — Acceptance Filter 1 Enable
FTER1=1 enables Acceptance Filter 1

FTER0 — Acceptance Filter 0 Enable
FTER0=1 enables Acceptance Filter 0

CANDSR (0x4000_5019) CAN Acceptance Filter Match Status Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	FMS[3]	FMS[2]	FMS[1]	FMS[0]
WR	-	-	-	-	-	-	-	-

This register is read-only and the corresponding match bit is set to 1 when a match condition occurs for the acceptance filer.

FMS[3] Acceptance Filter 3 Match Status
FSM[3]=1 Acceptance Filter 3 match
FSM[3]=0 Acceptance Filter 3 not match

FMS[2] Acceptance Filter 2 Match Status
FSM[2]=1 Acceptance Filter 2 match
FSM[2]=0 Acceptance Filter 2 not match

FMS[1] Acceptance Filter 1 Match Status
FSM[1]=1 Acceptance Filter 1 match
FSM[1]=0 Acceptance Filter 1 not match

FMS[0] Acceptance Filter 0 Match Status
FSM[0]=1 Acceptance Filter 0 match
FSM[0]=0 Acceptance Filter 0 not match

CANALC (0x4000_501A) CAN Arbitration Lost Capture Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	ALC4	ALC3	ALC2	ALC1	ALC0
WR	-	-	-	-	-	-	-	-

This register is read only and reports the bit locations of a lost arbitration when transmitting a message. The following table defines the corresponding relationship.

Bits					Decimal Value	Description
ALC4	ALC3	ALC2	ALC1	ALC0		
0	0	0	0	0	0	Arbitration lost in bit ID28
0	0	0	0	1	1	Arbitration lost in bit ID27
0	0	0	1	0	2	Arbitration lost in bit ID26
0	0	0	1	1	3	Arbitration lost in bit ID25
0	0	1	0	0	4	Arbitration lost in bit ID24
0	0	1	0	1	5	Arbitration lost in bit ID23
0	0	1	1	0	6	Arbitration lost in bit ID22
0	0	1	1	1	7	Arbitration lost in bit ID21
0	1	0	0	0	8	Arbitration lost in bit ID20
0	1	0	0	1	9	Arbitration lost in bit ID19
0	1	0	1	0	10	Arbitration lost in bit ID18
0	1	0	1	1	11	Arbitration lost in bit SRTR (Note 1)
0	1	1	0	0	12	Arbitration lost in bit IDE
0	1	1	0	1	13	Arbitration lost in bit ID17 (Note 2)
0	1	1	1	0	14	Arbitration lost in bit ID16 (Note 2)
0	1	1	1	1	15	Arbitration lost in bit ID15 (Note 2)
1	0	0	0	0	16	Arbitration lost in bit ID14 (Note 2)
1	0	0	0	1	17	Arbitration lost in bit ID13 (Note 2)
1	0	0	1	0	18	Arbitration lost in bit ID12 (Note 2)

Bits					Decimal Value	Description
ALC4	ALC3	ALC2	ALC1	ALC0		
1	0	0	1	1	19	Arbitration lost in bit ID11 (Note 2)
1	0	1	0	0	20	Arbitration lost in bit ID10 (Note 2)
1	0	1	0	1	21	Arbitration lost in bit ID9 (Note 2)
1	0	1	1	0	22	Arbitration lost in bit ID8 (Note 2)
1	0	1	1	1	23	Arbitration lost in bit ID7 (Note 2)
1	1	0	0	0	24	Arbitration lost in bit ID6 (Note 2)
1	1	0	0	1	25	Arbitration lost in bit ID5 (Note 2)
1	1	0	1	0	26	Arbitration lost in bit ID4 (Note 2)
1	1	0	1	1	27	Arbitration lost in bit ID3 (Note 2)
1	1	1	0	0	28	Arbitration lost in bit ID2 (Note 2)
1	1	1	0	1	29	Arbitration lost in bit ID1 (Note 2)
1	1	1	1	0	30	Arbitration lost in bit ID0 (Note 2)
1	1	1	1	1	31	Arbitration lost in bit RTR (Note 2)

Table 13-2 Error code for CAN bus Arbitration Lost

Note 1: Bit RTR for standard frame messages.

Note 2: Extended frame messages only.

On arbitration lost, the corresponding arbitration lost interrupt is generated. At the same time, the current bit position of the bit stream processor is captured into the Arbitration Lost Capture register. The content in this register is locked until the software read out its contents. The capture mechanism is then activated again. The corresponding interrupt flag located in the interrupt register is cleared during the read access to the interrupt flag register. A new arbitration lost interrupt is not possible until the arbitration lost capture register is read out. The following diagram shows the bit locations of the arbitration lost. And an ALC=0x08 is shown as an example, where arbitration lost is detected when outputting a recessive bit but receiving a dominant bit at ID20.

CANECC (0x4000_501C) CAN Error Code Capture Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	ERRC[1]	ERRC[0]	DIR	SEG[4]	SEG[3]	SEG[2]	SEG[1]	SEG[0]
WR	-	-	-	-	-	-	-	-

This register is read only. This register always keeps the last error status on CAN bus.

ERRC[1-0] Error Code

This reflects the error type as defined in the following table

ERCC[1]	ERRC[0]	Descriptions
0	0	Bit error
0	1	Form error
1	0	Stuff error
1	1	Other type of error

DIR Direction of Transfer Error

DIR=1 means errors occurred during reception

DIR=0 means errors occurred during transmission

SEG[4-0] Error Locations

SEG[4-0]					Function
0	0	0	1	1	Start of frame
0	0	0	1	0	ID28 to ID21

SEG[4-0]					Function
0	0	1	1	0	ID20 toID18
0	0	1	0	0	Bit SRTR
0	0	1	0	1	Bit IDE
0	0	1	1	1	ID17 to ID13
0	1	1	1	1	ID12 to ID5
0	1	1	1	0	ID4 to ID0
0	1	1	0	0	Bit RTR
0	1	1	0	1	Reserved bit 1
0	1	0	0	1	Reserved bit 0
0	1	0	1	1	Data length code
0	1	0	1	0	Data field
0	1	0	0	0	CRC sequence
1	1	0	0	0	CRC delimiter
1	1	0	0	1	Acknowledge slot
1	1	0	1	1	Acknowledge delimiter
1	1	0	1	0	End of frame
1	0	0	1	0	Intermission
1	0	0	0	1	Active error flag
1	0	1	1	0	Passive error flag
1	0	0	1	1	Tolerate dominant bits (Note)
1	0	1	1	1	Error delimiter
1	1	1	0	0	Overload flag

Note: Any node tolerates up to 7 consecutive ‘dominant’ bits after send Active Error Flag, Passive Error Flag or Overload Flag.

If a bus error occurs, the corresponding bus error interrupt is generated. At the same time, the current position of the bit stream processor is captured into the Error Code Capture register. The content in this register is locked until the software read out its content. A new bus error capture is prohibited until the capture register is read. The capture mechanism is then activated again. The corresponding interrupt flag located in the interrupt flag register is cleared by reading of the interrupt flag register.

CANEWLR (0x4000_501D) CAN Error Warning Limit Register RW (0x60)

	7	6	5	4	3	2	1	0
RD	CANEWLR[7-0]							
WR	CANEWLR[7-0]							

This register is modifiable on in reset mode and appears as read-only in normal operating mode. This register set the warning limit for the error counter. When CANREC or CANTEC value exceed CANEWLR value, an error warning interrupt is generated, and the EI flag is set in the interrupt flag register. CANEWLF defaults to 0x60 after hardware reset.

CANREC (0x4000_501E) CAN Receive Error Count Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANREC[7-0]							
WR	CANREC[7-0]							

This register is modifiable on in reset mode and appears as read-only in normal operating mode. CANREC contains the current error count. When the controller enters bus-off state, CANREC is re-initialized as 0, and during bus-off, the writing to CANREC has no effect.

CANTEC (0x4000_501F) CAN Transmit Error Count Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTEC[7-0]							
WR	CANTEC[7-0]							

This register is modifiable on in reset mode and appears as read-only in normal operating mode. CANTEC contains the current error count, and is initialized as 0 after hardware rest. If a bus-off event occurs, the transmit error counter is initialized to 128 if auto-recovery is not enabled. The software can just use the reset mode to allow recovery to the normal error-active state.

13.3 Transmit Buffer

The global layout of the transmit buffer in XFR is shown in figure below. The transmit buffer contains 13-bytes of XFR space can be in the form of Standard Frame Format (SFF) or Extended Frame Format (EFF) configuration.

Standard Format Frame	
B+040h	TX Frame Information
B+044h	TX Identifier[10:0]
B+048h	TX Data 3 - TX Data 0
B+04Ch	TX Data 7 - TX Data 4
B+050h	TX Data 11 - TX Data 8
B+054h	TX Data 15 - TX Data 12
B+058h	TX Data 19 - TX Data 16
B+05Ch	TX Data 23 - TX Data 20
B+060h	TX Data 27 - TX Data 24
B+064h	TX Data 31 - TX Data 28
B+068h	TX Data 35 - TX Data 32
B+06Ch	TX Data 39 - TX Data 36
B+070h	TX Data 43 - TX Data 40
B+074h	TX Data 47 - TX Data 44
B+078h	TX Data 51 - TX Data 48
B+07Ch	TX Data 55 - TX Data 52
B+080h	TX Data 59 - TX Data 56
B+084h	TX Data 63 - TX Data 60

Extended Format Frame	
B+040h	TX Frame Information
B+044h	TX Identifier[28:0]
B+048h	TX Data 3 - TX Data 0
B+04Ch	TX Data 7 - TX Data 4
B+050h	TX Data 11 - TX Data 8
B+054h	TX Data 15 - TX Data 12
B+058h	TX Data 19 - TX Data 16
B+05Ch	TX Data 23 - TX Data 20
B+060h	TX Data 27 - TX Data 24
B+064h	TX Data 31 - TX Data 28
B+068h	TX Data 35 - TX Data 32
B+06Ch	TX Data 39 - TX Data 36
B+070h	TX Data 43 - TX Data 40
B+074h	TX Data 47 - TX Data 44
B+078h	TX Data 51 - TX Data 48
B+07Ch	TX Data 55 - TX Data 52
B+080h	TX Data 59 - TX Data 56
B+084h	TX Data 63 - TX Data 60

Figure 13-2 Transmit Buffer memory map

The transmit buffer layout is subdivided into descriptor and data fields where the first byte of the descriptor field is the frame information byte (frame information). It describes the frame format (SFF or EFF), remote or data frame and the data length. Four identifier bytes for both SFF and EFF messages follow. But only the lower 11 bits in the four identifier bytes are valid for SFF and the upper 21 bits are invalid. For EFF, the lower 29 bits in the four identifier bytes are valid and the upper 3 bits are invalid. The data field contains up to eight data bytes.

CANTFIR (0x4000_5040) CAN Transmit Frame Information Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	FF	RTR	FDF	BRS	DLC[3]	DLC[2]	DLC[1]	DLC[0]
WR	FF	RTR	FDF	BRS	DLC[3]	DLC[2]	DLC[1]	DLC[0]

FF — Frame Format
FF=1 indicates Extended format frame
FF=0 indicates Standard format frame

RTR — Remote Frame Transmit Request
RTR=1 indicates remote frame
RTR=0 indicates regular data frame

FDF — FD Format indicator
FDF=1 indicates FD frame format
FDF=0 indicates Classical frame format

BRS — Bit Rate Switch
BRS=1 shall switch to the preconfigured data bit rate of the data phase
BRS=0 keep the nominal bit rate of the arbitration phase

DLC[3-0] — Data Byte Count = DLC[3-0]
The number of bytes in the data field of a message is coded by the data length code. At the start of a remote frame transmission the data length code is not considered due to the RTR bit being logic 1 (remote). This forces the number of transmitted/received data bytes to be 0. Nevertheless, the data length code must be specified correctly to avoid bus errors, if two CAN controllers start a remote frame transmission with the same identifier simultaneously.

Frames	Data length code				Number of data bytes
	DLC3	DLC2	DLC1	DLC0	
Classical Frames and FD Frames	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	2
	0	0	1	1	3
	0	1	0	0	4
	0	1	0	1	5
	0	1	1	0	6
	0	1	1	1	7
	1	0	0	0	8
Classical Frames	1	0 or 1	0 or 1	0 or 1	8
FD Frames	1	0	0	1	12
	1	0	1	0	16
	1	0	1	1	20
	1	1	0	0	24
	1	1	0	1	32
	1	1	1	0	48
	1	1	1	1	64

Figure 13-3 Coding of the numbers of data bytes by the DLC

In standard Frame Format (SFF) the identifier consists of 11 bits (ID.10 to ID.0) and in Extended Frame Format (EFF) messages the identifier consists of 29 bits (ID.28 to ID.0). ID.28 is the most significant bit, which is transmitted first on the bus during the arbitration process. The identifier acts as the message's name and is used for bus arbitration and used in a receiver for acceptance filtering. The smaller the binary value of the identifier is, the higher the priority is. This is due to the larger number of leading dominant bits during arbitration. Also please

note the R0, R1 and SRR bits in the SFF and EFF are transmitted as R0=R1=0 and SRR=1.

CANTXID0 (0x4000_5044) CAN Transmit Identifier Register 0 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXID0[7-0]							
WR	CANTXID0[7-0]							

CANTXID0[7-0] For SFF ID, CANTXID0[7-0] = ID[7-0].
For EFF ID, CANTXID0[7-0] = ID[7-0].

CANTXID1 (0x4000_5045) CAN Transmit Identifier Register1 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXID1[7-0]							
WR	CANTXID1[7-0]							

CANTXID1[7-0] For SFF ID, CANTXID1[2-0] = ID[10-8]. CANTXID1[7-3] is not used.
For EFF ID, CANTXID1[7-0] = ID[15-8].

CANTXID2 (0x4000_5046) CAN Transmit Identifier Register 2 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXID2[7-0]							
WR	CANTXID2[7-0]							

CANTXID2[7-0] For SFF ID, CANTXID2[7-0] is not used.
For EFF ID, CANTXID2[7-0] = ID[23-16].

CANTXID3 (0x4000_5047) CAN Transmit Identifier Register 3 RW (0x00)

	7	6	5	4	3	2	1	0
RD	0	0	0	CANTXID3[4-0]				
WR	-	-	-	CANTXID3[4-0]				

CANTXID3[4-0] For SFF ID, CANTXID3[4-0] is not used.
For EFF ID, CANTXID3[4-0] = ID[28-24].

CANTXDATA0 (0x4000_5048) CAN Transmit DATA Register 0 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA0[7-0]							
WR	CANTXDATA0[7-0]							

CANTXDATA1 (0x4000_5049) CAN Transmit DATA Register 1 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA1[7-0]							
WR	CANTXDATA1[7-0]							

CANTXDATA2 (0x4000_504A) CAN Transmit DATA Register 2 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA2[7-0]							
WR	CANTXDATA2[7-0]							

CANTXDATA63 (0x4000_5087) CAN Transmit DATA Register 63 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA63[7-0]							
WR	CANTXDATA63[7-0]							

Data field registers CANTXDATA0 ~ CANTXDATA63 are mapping to address 0x4000_5048 to 0x4000_5087. The number of transferred data bytes is defined by the data length code. The first bit transmitted is the most significant bit of data byte 0 at address 0x4000_5048.

13.4 Acceptance Filter

There are four Acceptance Filters for matching with the ID of a message on the bus. Each filter can be enabled or disabled independently. Each acceptance filter consists of 4 ID registers and corresponding ID Mask register to mask the match comparing the specific bit in the ID. The layout of the ID registers corresponding to forming ID28 to ID0 for SFF and EFF is the same as Transmit ID registers. Please note the ID registers only define the ID and do not define the frame type, therefore it is possible a value in the ID registers corresponding to two possible matches for SFF and EFF separately. The software should check if unintended accepted ID is possible for another frame type. For the Filter's Mask register, when mask bit is 0 the corresponding bit is compared for matching, and when the mask bit is 1 the corresponding bit is ignored. All Acceptance Filter ID registers and Mask register can only be modified in the reset mode, and appear as read-only in normal operating mode.

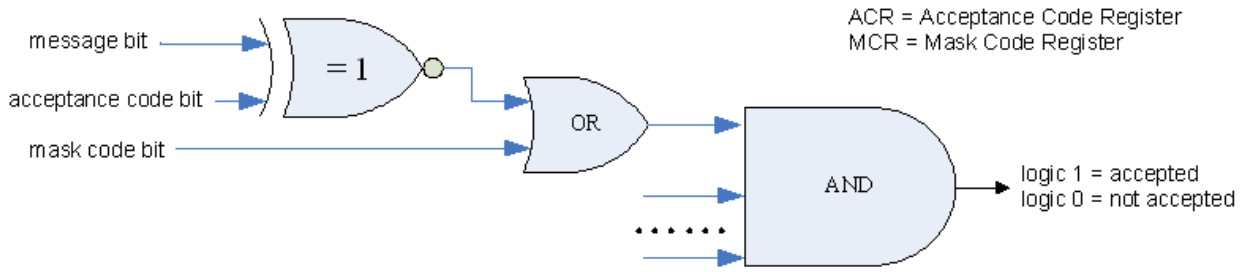


Figure 13-4 Acceptance Filter function diagram

CANAFID00 (0x4000_5020) Acceptance Filter 0 ID Byte 0 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID00[7-0]							
WR	CANFID00[7-0]							

This register is modifiable on in reset mode and appears as read-only in normal operating mode. This register set the warning limit for the error counter. When CANREC or CANTEC value exceed CANEWLR value, an error warning interrupt is generated, and the EI flag is set in the interrupt flag register. CANEWLF defaults to 0x60 after hardware reset.

CANAFID01 (0x4000_5021) Acceptance Filter 0 ID Byte 1 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID01[7-0]							
WR	CANFID01[7-0]							

CANFID01[7-0]

CANAFID02 (0x4000_5022) Acceptance Filter 0 ID Byte 2 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID02[7-0]							
WR	CANFID02[7-0]							

CANFID02[7-0]

CANAFID03 (0x4000_2023) Acceptance Filter 0 ID Byte 3 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID03[7-0]							
WR	CANFID03[7-0]							

CANFID03[7-0]

CANAFMK00 (0x4000_5024) Acceptance Filter 0 Mask Byte 0 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK00[7-0]							
WR	CANFMK00[7-0]							

CANFMK00[7-0]

CANAFMK01 (0x4000_5025) Acceptance Filter 0 Mask Byte 1 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK01[7-0]							
WR	CANFMK01[7-0]							

CANFMK01[7-0]

CANAFMK02 (0x4000_5026) Acceptance Filter 0 Mask Byte 2 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK02[7-0]							
WR	CANFMK02[7-0]							

CANFMK02[7-0]

CANAFMK03 (0x4000_5027) Acceptance Filter 0 Mask Byte 3 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK03[7-0]							
WR	CANFMK03[7-0]							

CANFMK03[7-0]

CANAFID10 (0x4000_5028) Acceptance Filter 1 ID Byte 0 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID10[7-0]							
WR	CANFID10[7-0]							

CANFID10[7-0]

CANAFID11 (0x4000_5029) Acceptance Filter 1 ID Byte 1 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID11[7-0]							
WR	CANFID11[7-0]							

CANFID11[7-0]

CANAFID12 (0x4000_502A) Acceptance Filter 1 ID Byte 2 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID12[7-0]							
WR	CANFID12[7-0]							

CANFID12[7-0]

CANAFID13 (0x4000_502B) Acceptance Filter 1 ID Byte 3 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID13[7-0]							
WR	CANFID13[7-0]							

CANFID13[7-0]

CANAFMK10 (0x4000_502C) Acceptance Filter 1 Mask Byte 0 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK10[7-0]							
WR	CANFMK10[7-0]							

CANFMK10[7-0]

CANAFMK11 (0x4000_502D) Acceptance Filter 1 Mask Byte 1 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK11[7-0]							
WR	CANFMK11[7-0]							

CANFMK11[7-0]

CANAFMK12 (0x4000_502E) Acceptance Filter 1 Mask Byte 2 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK12[7-0]							
WR	CANFMK12[7-0]							

CANFMK12[7-0]

CANAFMK13 (0x4000_502F) Acceptance Filter 1 Mask Byte 3 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK13[7-0]							
WR	CANFMK13[7-0]							

CANFMK13[7-0]

CANAFID20 (0x4000_5030) Acceptance Filter 2 ID Byte 0 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID20[7-0]							
WR	CANFID20[7-0]							

CANFID20[7-0]

CANAFID21 (0x4000_5031) Acceptance Filter 2 ID Byte 1 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID21[7-0]							
WR	CANFID21[7-0]							

CANFID21[7-0]

CANAFID22 (0x4000_5032) Acceptance Filter 2 ID Byte 2 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID22[7-0]							
WR	CANFID22[7-0]							

CANFID22[7-0]

CANAFID23 (0x4000_5033) Acceptance Filter 2 ID Byte 3 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID23[7-0]							
WR	CANFID23[7-0]							

CANFID23[7-0]

CANAFMK20 (0x4000_5034) Acceptance Filter 2 Mask Byte 0 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK20[7-0]							
WR	CANFMK20[7-0]							

CANFMK20[7-0]

CANAFMK21 (0x4000_5035) Acceptance Filter 2 Mask Byte 1 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK21[7-0]							
WR	CANFMK21[7-0]							

CANFMK21[7-0]

CANAFMK22 (0x4000_5036) Acceptance Filter 2 Mask Byte 2 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK22[7-0]							
WR	CANFMK22[7-0]							

CANFMK22[7-0]

CANAFMK23 (0x4000_5037) Acceptance Filter 2 Mask Byte 3 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK23[7-0]							
WR	CANFMK23[7-0]							

CANFMK23[7-0]

CANAFID30 (0x4000_5038) Acceptance Filter 3 ID Byte 0 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID30[7-0]							
WR	CANFID30[7-0]							

CANFID30[7-0]

CANAFID31 (0x4000_5039) Acceptance Filter 3 ID Byte 1 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID31[7-0]							
WR	CANFID31[7-0]							

CANFID31[7-0]

CANAFID32 (0x4000_503A) Acceptance Filter 3 ID Byte 2 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID32[7-0]							
WR	CANFID32[7-0]							

CANFID32[7-0]

CANAFID33 (0x4000_503B) Acceptance Filter 3 ID Byte 3 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFID33[7-0]							
WR	CANFID33[7-0]							

CANFID33[7-0]

CANAFMK30 (0x4000_503C) Acceptance Filter 3 Mask Byte 0 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK30[7-0]							
WR	CANFMK30[7-0]							

CANFMK30[7-0]

CANAFMK31 (0x4000_503D) Acceptance Filter 3 Mask Byte 1 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK31[7-0]							
WR	CANFMK31[7-0]							

CANFMK31[7-0]

CANAFMK32 (0x4000_503E) Acceptance Filter 3 Mask Byte 2 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK32[7-0]							
WR	CANFMK32[7-0]							

CANFMK32[7-0]

CANAFMK33 (0x4000_503F) Acceptance Filter 3 Mask Byte 3 (0x00)

	7	6	5	4	3	2	1	0
RD	CANFMK33[7-0]							
WR	CANFMK33[7-0]							

CANFMK33[7-0]

13.5 Receive Buffer

There is not dedicated SRAM as CAN FD Receive FIFO (RX-FIFO). Software needs to allocate a specific DMA SRAM as CAN FD RX-FIFO by setting Receive DMA configuration registers then enable the RX DMA mode (DMAEN=1).

Under the RX DMA mode (DMAEN=1), MCU can read the received date from DMA SRAM according the CRADR and CRLEN.

If RX DMA mode is disable (DMAEN=0), the received data only be read from the Receive Buffer SFRs which address is from B+0x090 to B+0x0D4. Software must read out the received data and release the Receive Buffer by setting RRB command before next receive frame incoming.

LOCATION	DESCRIPTION	31 24	23 16	15 8	7 0
*B + 0x90	ACP FILTER INFO	0x00	0100 Filter Stat[3-0]	0x00 ESI	FF RTR FDF BRS DLC[3-0]
*B + 0x94	ID	000	ID [28 – 0]		
*B + 0x98	RX DATA Max. 64-byte The valid data length is defined by DLC[3-0]	RX DATA 3 [7-0]	RX DATA 2 [7-0]	RX DATA 1 [7-0]	RX DATA 0 [7-0]
*B + 0x9C		RX DATA 7 [7-0]	RX DATA 6 [7-0]	RX DATA 5 [7-0]	RX DATA 4 [7-0]
*B + 0xA0		RX DATA 11 [7-0]	RX DATA 10 [7-0]	RX DATA 9 [7-0]	RX DATA 8 [7-0]
*B + 0xA4		RX DATA 15 [7-0]	RX DATA 14 [7-0]	RX DATA 13 [7-0]	RX DATA 12 [7-0]
*B + 0xA8		RX DATA 19 [7-0]	RX DATA 18 [7-0]	RX DATA 17 [7-0]	RX DATA 16 [7-0]
*B + 0xAC		RX DATA 23 [7-0]	RX DATA 22 [7-0]	RX DATA 21 [7-0]	RX DATA 20 [7-0]
*B + 0xB0		RX DATA 27 [7-0]	RX DATA 26 [7-0]	RX DATA 25 [7-0]	RX DATA 24 [7-0]
*B + 0xB4		RX DATA 31 [7-0]	RX DATA 30 [7-0]	RX DATA 29 [7-0]	RX DATA 28 [7-0]
*B + 0xB8		RX DATA 35 [7-0]	RX DATA 34 [7-0]	RX DATA 33 [7-0]	RX DATA 32 [7-0]
*B + 0xBC		RX DATA 39 [7-0]	RX DATA 38 [7-0]	RX DATA 37 [7-0]	RX DATA 36 [7-0]
*B + 0xC0		RX DATA 43 [7-0]	RX DATA 42 [7-0]	RX DATA 41 [7-0]	RX DATA 40 [7-0]
*B + 0xC4		RX DATA 47 [7-0]	RX DATA 46 [7-0]	RX DATA 45 [7-0]	RX DATA 44 [7-0]
*B + 0xC8		RX DATA 51 [7-0]	RX DATA 50 [7-0]	RX DATA 49 [7-0]	RX DATA 48 [7-0]
*B + 0xCC		RX DATA 55 [7-0]	RX DATA 54 [7-0]	RX DATA 53 [7-0]	RX DATA 52 [7-0]
*B + 0xD0		RX DATA 59 [7-0]	RX DATA 58 [7-0]	RX DATA 57 [7-0]	RX DATA 56 [7-0]
*B + 0xD4		RX DATA 63 [7-0]	RX DATA 62 [7-0]	RX DATA 61 [7-0]	RX DATA 60 [7-0]

*B = Base Address

Figure 13-5 Receive Buffer memory map

CANRFIR0 (0x4000_5090) CAN Receive Frame Information Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	FF	RTR	FDF	BRS	DLC[3]	DLC[2]	DLC[1]	DLC[0]
WR	-	-	-	-	-	-	-	-

FF Frame Format

	FF=1 indicates Extended format frame FF=0 indicates Standard format frame
RTR	Remote Frame Transmit Request RTR=1 indicates remote frame RTR=0 indicates regular data frame
FDF	FD Format indicator FDF=1 indicates FD frame format FDF=0 indicates Classical frame format
BRS	Bit Rate Switch BRS=1 shall switch to the preconfigured data bit rate of the data phase BRS=0 keep the nominal bit rate of the arbitration phase
DLC[3-0]	Data Byte Count = DLC[3-0] The number of bytes in the data field of a message is coded by the data length code. At the start of a remote frame transmission the data length code is not considered due to the RTR bit being logic 1 (remote). This forces the number of transmitted/received data bytes to be 0. Nevertheless, the data length code must be specified correctly to avoid bus errors, if two CAN controllers start a remote frame transmission with the same identifier simultaneously.

Frames	Data length code				Number of data bytes
	DLC3	DLC2	DLC1	DLC0	
Classical Frames and FD Frames	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	2
	0	0	1	1	3
	0	1	0	0	4
	0	1	0	1	5
	0	1	1	0	6
	0	1	1	1	7
	1	0	0	0	8
Classical Frames	1	0 or 1	0 or 1	0 or 1	8
FD Frames	1	0	0	1	12
	1	0	1	0	16
	1	0	1	1	20
	1	1	0	0	24
	1	1	0	1	32
	1	1	1	0	48
	1	1	1	1	64

Figure 13-6 Coding of the numbers of data bytes by the DLC

In standard Frame Format (SFF) the identifier consists of 11 bits (ID.10 to ID.0) and in Extended Frame Format (EFF) messages the identifier consists of 29 bits (ID.28 to ID.0). ID.28 is the most significant bit, which is transmitted first on the bus during the arbitration process. The identifier acts as the message's name and is used for bus arbitration and used in a receiver for acceptance filtering. The smaller the binary value of the identifier is, the higher the priority is. This is due to the larger number of leading dominant bits during arbitration. Also please note the R0, R1 and SRR bits in the SFF and EFF are transmitted as R0=R1=0 and SRR=1.

CANRFIR1 (0x4000_5091) CAN Receive Frame Information Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	0	0	0	0	0	0	0	ESI
WR	-	-	-	-	-	-	-	-

ESI — Error Status Indicator
ESI=1 indicates the transmitter is an error passive node.
ESI=0 indicates the transmitter is an error active node.

CANRFIR2 (0x4000_5092) CAN Receive Frame Information Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	0	1	0	0	RXFMS[3]	RXFMS[2]	RXFMS[1]	RXFMS[0]
WR	-	-	-	-	-	-	-	-

RXFMS[3] — Acceptance Filter 3 Match Status
RXFSM[3]=1 Acceptance Filter 3 match
RXFSM[3]=0 Acceptance Filter 3 not match

RXFMS[2] — Acceptance Filter 2 Match Status
RXFSM[2]=1 Acceptance Filter 2 match
RXFSM[2]=0 Acceptance Filter 2 not match

RXFMS[1] — Acceptance Filter 1 Match Status
RXFSM[1]=1 Acceptance Filter 1 match
RXFSM[1]=0 Acceptance Filter 1 not match

RXFMS[0] — Acceptance Filter 0 Match Status
RXFSM[0]=1 Acceptance Filter 0 match
RXFSM[0]=0 Acceptance Filter 0 not match

CANRXID0 (0x4000_5094) CAN Receive Identifier Register 0 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CANRXID0[7-0]							
WR	-							

CANTXID0[7-0] — For SFF ID, CANRXID0[7-0] = ID[7-0].
For EFF ID, CANRXID0[7-0] = ID[7-0].

CANRXID1 (0x4000_5095) CAN Receive Identifier Register1 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CANRXID1[7-0]							
WR	-							

CANRXID1[7-0] — For SFF ID, CANRXID1[2-0] = ID[10-8]. CANRXID1[7-3] is not used.
For EFF ID, CANRXID1[7-0] = ID[15-8].

CANRXID2 (0x4000_5096) CAN Receive Identifier Register 2 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CANRXID2[7-0]							
WR	-							

CANRXID2[7-0] — For SFF ID, CANRXID2[7-0] is not used.
For EFF ID, CANRXID2[7-0] = ID[23-16].

CANRXID3 (0x4000_5097) CAN Receive Identifier Register 3 RO (0x00)

	7	6	5	4	3	2	1	0
RD	0	0	0	CANRXID3[4-0]				
WR	-	-	-	-				

CANRXID3[4-0] — For SFF ID, CANRXID3[4-0] is not used.
For EFF ID, CANRXID3[4-0] = ID[28-24].

CANRXDATA0 (0x4000_5098) CAN Receive DATA Register 0 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CANRXDATA0[7-0]							
WR	-							

CANRXDATA1 (0x4000_5099) CAN Receive DATA Register 1 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CANRXDATA[07-0]							
WR	-							

CANRXDATA2 (0x4000_509A) CAN Receive DATA Register 2 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CANRXDATA2[7-0]							
WR	-							

CANRXDATA63 (0x4000_50D7) CAN Receive DATA Register 0-63 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CANRXDATA63[7-0]							
WR	-							

Data field registers CANRXDATA0 ~ CANRXDATA63 are mapping to address 0x4000_5098 to 0x4000_50D7.

The number of received data bytes is defined by the data length code.

13.6 Receive DMA

The Receive DMA Controller is as a CAN Receive FIFO to store the received frames into DAM SRAM. The CAN DMA Start Address STADR[15-0] and the CAN DMA Memory Size DMASIZE[9-2] should be updated only when DMAEN=0.

If RX DMA Mode is disabled (DMAEN=0), the DMA controller will be reset and the Receive FIFO will be cleared. There is only one receive buffer for single frame, and only can be read from SFR base+0x090 to SFR base+0x0D4. After read-out the data from receive buffer, MCU must release these registers by setting Release Receive Buffer (RRB) command.

When RX DMA Mode is enabled (DMAEN=1), the receive frame will be stored into DMA SRAM according the STADR and DMASIZE configuration. The DMA interrupt DMAI will be asserted after upload a complete receive frame into DMA SRAM. MCU must check-out the receive frame from the DMA SRAM according to CAN Current Receive Address CRADR[15-0]. The byte length of current frame is specified in CRLEN[6-0]. After read-out a frame by DMA, MCU must release the memory space by setting Release Receive Buffer (RRB) command. And MCU can check-out the next valid frame from updated CRADR[15-0] if FRNUM[6-0] is not zero.

CANSTADR0 (0x4000_50F0) CAN DMA Start Address Register 0 RW (0x00)

	7	6	5	4	3	2	1	0
RD	STADR[7-2]						0	0
WR	STADR[7-2]						-	-

CANSTADR1 (0x4000_50F1) CAN DMA Start Address Register 1 RW (0x00)

	7	6	5	4	3	2	1	0
RD	STADR[15-8]							
WR	STADR[15-8]							

STADR[15-2] CAN DMA Memory Start Address
Because DMA Memory is 32-bit width, the value of STADR[1-0] is useless
This register can only be modified in reset mode (RSTM=1).

CANDMASIZE (0x4000_50F2) CAN DMA Memory Size Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	DMASIZE[9-2]							
WR	DMASIZE[9-2]							

DMASIZE[9-2] DMA Memory Maximal Size.

This register indicates the total memory size reserved for CAN receive buffer. Because the minimum unit is 32-bit, the DMASIZE[1-0] will be ignored, The allocated memory size is (DMASIZE[9-2] + 1) X 32-bit started from STADR[15-0].
This register can only be modified in reset mode (RSTM=1).

CANCRADR0 (0x4000_50F4) CAN Current Receive Address Register 0 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CRADR[7-2]						0	CRRDY
WR	-						-	-

CANCRADR1 (0x4000_50F5) CAN Current Receive Address Register 1 RO (0x00)

	7	6	5	4	3	2	1	0
RD	CRADR[15-8]							
WR	-							

CRADR[15-2] CAN DMA Current Receive Address
This register indicates the address of the top received frame stored in the DMA memory. The DMA memory space will be released by setting Release Receive Buffer (RRB) command. Meanwhile CRADR will be relocated to the next received frame. This register is read-only and will be initialized as STADR[15-0] when RX DMA Mode Enable is disabled (DMAEN=0).

CRRDY CAN Current Receive Address and CANCRLEN Ready
CRRDY=1 means that CRADR[15-2] and CRLEN[6-0] is valid.
CRRDY=0 means that CRADR[15-2] and CRLEN[6-0] is waiting for updated or the receive FIFO is empty now.
CRRDY is cleared by Release Receive Buffer (RRB) command, and will be set when a valid CRADR and CRLEN are updated by CAN DMA controller.

CANCRLEN (0x4000_50F6) CAN Current Receive Data Length Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	0	CRLEN[6-0]						
WR	-	-						

CRLEN[6-0] CAN Current Receive Data Length
This register indicates the valid data byte count of current receive frame stored in the DMA SRAM. The minimum value is 0x08 while receive a remote frame or a zero-length frame. The maximal value is 0x48 while data length is 64-byte.

CANFRNUM (0x4000_50F7) CAN Receive Identifier Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	DMABUSY	FRNUM[6-0]						
WR	-	-						

DMABUSY DMABUSY=1 means that a new received frame is writing to DMA memory.
DMABUSY=0 means that CAN DMA bus is idle.

FRNUM[6-0] Valid Receive Frame Number
This register indicates the valid unread receive frame number stored in the DMA SRAM. The FRNUM is increased after upload a complete receive frame successfully and is decreased by the Release Receive Buffer (RRB) command. If FRNUM equals zero, that means all receive frames was check-out and the receive FIFO is empty. FRNUM will be reset to zero if DMA mode is disabled (DMAEN=0).

13.7 Basic Operations

The basic function of CAN controller will be discussed in this section. The following chapter will describe the advanced functions of CAN controller. To communicate with other nodes on the CAN-bus, CAN controller must be configured properly as described. Following diagram is the flowchart of basic communication.

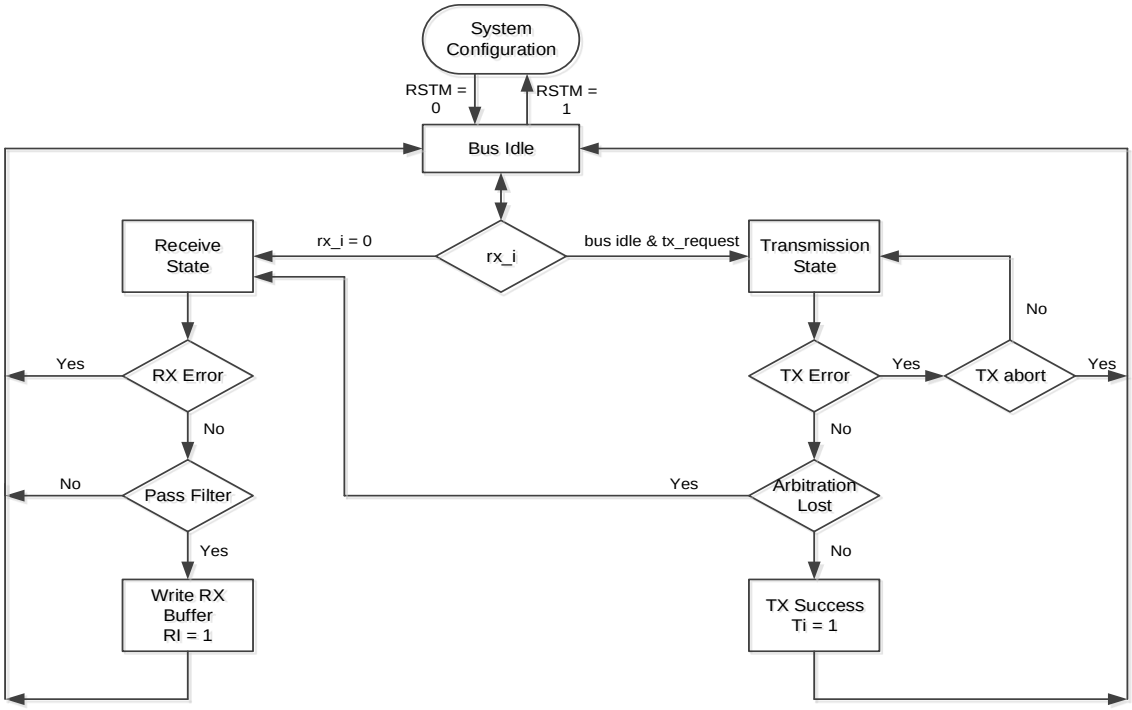


Figure 13-7 Operation Principle of CAN controller

13.8 System Configuration

Following system configurations must be done under reset mode (RSTM=1).

13.8.1 Enter Reset Mode

Some registers are writable only in reset mode such as Bus Timing registers and Acceptance Filter registers etc., so we must write these registers in the reset mode. Then the CAN controller must quit reset mode when these registers are configured properly to make the controller work.

13.8.2 Baud Rate Setting

The baud rate in the CAN-bus is determined by the two registers CANBTR0 and CANBTR1. The following example explains how to calculate the baud rate of 1Mbps assuming system clock is 80MHz, and CANBTR0=04h CANBTR1=58h. Then the TQ1_CLK is CANCLK/(CANCS[5-0]+1) = 80MHz/5= 16MHz. Thus each time quanta (TQ1) is 62.5nsec. Then the one CAN bit time (BT) of arbitration phase is (1 + (TSEG1+1) + (TSEG2+1)) x 62.5 nsec = (1+6+9) x 62.5 nsec = 1usec. As the result the CAN bus baud rate of arbitration phase is 1Mbps.

The baud rate in CAN FD data phase is determined by CANFDTR0 and CANFDTR1. The following example explains how to calculate the FD data baud rate of 5MHz assuming system clock is 80MHz, and CANFDTR0=00h CANFDTR1=58h. Then the TQ2_CLK is CANCLK/(CANFDCS[5-0]+1) = CANCLK/1 = 80MHz. Thus each time quanta in FD data phase is 16.67nsec. Then the one CAN FD bit time of data phase is (1 + (FDTSEG1+1) + (FDTSEG2+1)) x 12.5 nsec = (1+6+9) x 12.5 nsec = 200nsec. As the result the CAN bus baud rate of the FD data phase is 5Mbps.

13.8.3 Setting of Acceptance Filter

With the help of the acceptance filter, the CAN controller can receive proper frames from CAN-bus only when the identifier bits of the received frames are equal to the predefined ones within the acceptance filter registers. The acceptance filter is defined by the Acceptance Code Registers (ACR) and the Mask Code Registers (MCR). The bit patterns of frames to be received are defined within the Acceptance Code registers. The corresponding mask code registers allow define certain bit positions to be ‘don’t care’. After hardware reset, all the Acceptance Code registers and Mask Code registers are cleared.

There are 4 groups of acceptance filter. We must enable at least one group of filter previously. Each group of filter has 4 Acceptance Code bits and 4 Mask Code bits. Acceptance code 3 register is the highest byte in the Acceptance Code; the Acceptance code 0 register is the lowest byte in the Acceptance Code. Mask code 3 register is the highest byte in the Mask Code and Mask Code 0 register is the lowest byte in the Mask Code.

When received message is standard format frame, the lower 11 bits of the Acceptance Code and the Mask Code are valid. When received message is extended format frame, the lower 29 bits of the Acceptance Code and the Mask Code are valid.

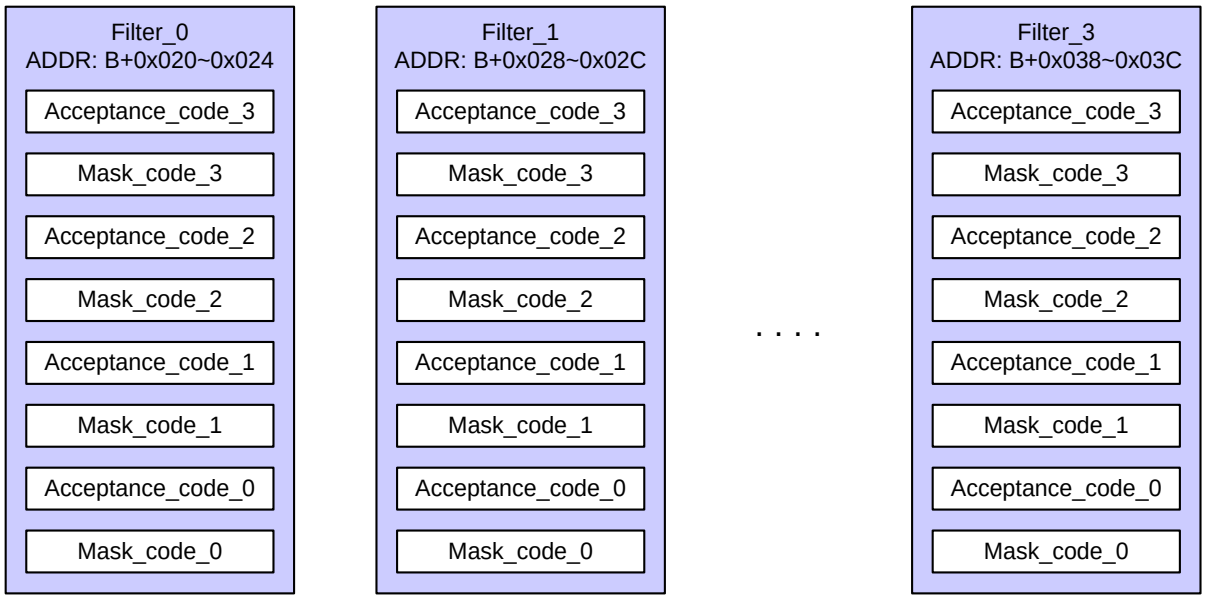


Figure 13-8 CAM Acceptance Filter

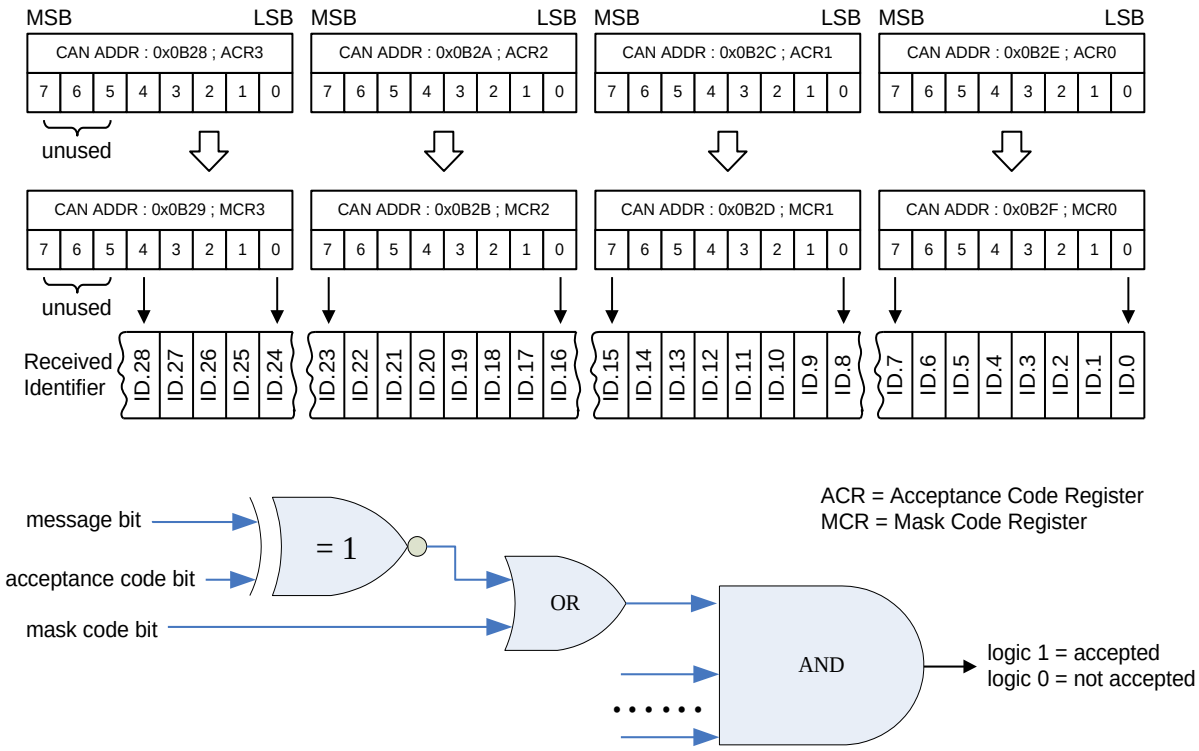


Figure 13-9 Filter In CAN Is Receiving Extended Frame Messages

Extended frame: if an extended format frame is received, the complete identifier including the RTR bit is used for acceptance filtering. For a successful reception of a message, all single bit comparisons have to signal acceptance. It should be noted that the 3 most significant bits of ACR3 and MCR3 are not used. In order to be compatible with future products these bits should be ‘don’t care’ by setting MCR3.7, MCR3.6 and MCR3.5 to logic 1.

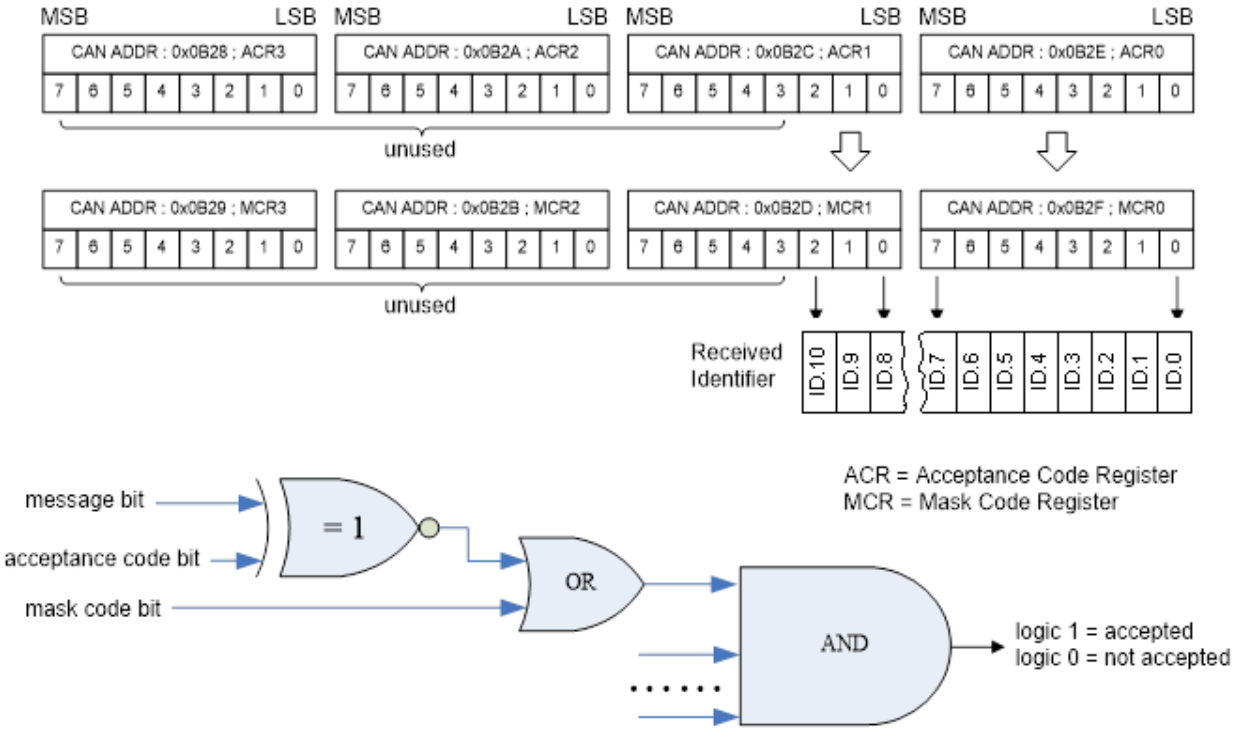


Figure 13-10 Filter in CAN Is Receiving Standard Frame Messages

Standard frame: if a standard format frame message is received, the completely identifier are used for acceptance filtering. For a successful reception of a frame, all single bit comparisons have to signal acceptance. Note that ACR3, MCR3, ACR2, MCR2 and the 5 most significant bits of ACR1 and MCR1 are not used. In order to be compatible with future products, these unused bits in Mask Code registers should be programmed to be ‘don’t care’ by setting these bits to logic 1.

13.8.4 Enter Operation Mode

After all system configurations are done, we must enter Operation Mode by setting RSTM to zero. In the meanwhile, the CAN controller enters the idle state waiting for transmitting and receiving frames.

13.9 Transmitting a Frame

You must write transmit buffer to send a frame to CAN-bus, then set the transmit request bit to logic 1. The controller will wait for the bus idle and sends the frame to the bus. For example, CAN will send a standard format frame to CAN-bus. Frame identifier is 0x123, the data have a length of 8 and the data are 11h, 22h, 33h, 44h, 55h, 66h, 77h and 88h.The program written in Keil-C is as follows:

Write_reg(0x040,{0x00, 0x00, 0x00, 0x08}); //write the frame information register, FF=0, RTR=0, DLC=8

Write_reg(0x044,{0x00, 0x00, 0x01, 0x23}); //write the TX ID Register

Write_reg(0x048,{0x44, 0x33, 0x22, 0x11}); //write the TX data 1-4 Register

Write_reg(0x04C,{0x88, 0x77, 0x66, 0x55}); // write the TX data 5-8 Register

Write_reg(0x004,{0x00, 0x00, 0x00, 0x01}); //write the transmit request bit

If the transmission completed successfully, the Transmit Interrupt (TI) will be set. If there is any error happened, the Bus Error Interrupt (BEI) will be set and the present frame will be retransmitted after the error frame. User can check the Error Code Capture Register (CANECC) for the detailed error condition and can abort the unsuccessful transmission by setting Abort Transmitting (AT) command.

13.10 Receiving a Frame

Only the frame which ID passed the Acceptance Filter can be received by controller. When CAN controller receives a frame successfully and the Receive Interrupt Enable (RIE) is enabled, the Receive Interrupt (RI) bit will be set logic 1.

If RX DMA mode (DMAEN) is disable, MCU is informed by the receive interrupt and reads out the data in the receive buffer, then releases the receive buffer. Please pay special attention here that the data must be read before releasing receive buffer, because the value in the receive buffer has changed after releasing receive buffer.

If RX DMA mode (DMAEN) is enable, the DMA request (dma_req) will be active and start the DMA protocol. When complete the DMA protocol by receiving dma_finish, the receive buffer is released and the RX DMA Complete Interrupt (DMAI) will be set to logic 1.

13.11 Test Modes

13.11.1 Self Test Mode

Enable the STE bit in CANMODE[2] will enable the self-test mode .The self-test mode is of the CAN control is the same normal operation mode except all ACK errors are ignored. This allows the software to initiate a “self receive request” command. Upon this command, a message is transmitted to the bus and will be treated as successfully transmitted without check the ACK flag. The transmitted message is also received by the sending CAN controller if the ID field matches with the acceptance filter.

13.11.2 Self Receive Request

Enable the SRR in CANCMD[4] will result a message be transmitted and received simultaneously. Upon Self Receive Request command, a message is transmitted and simultaneously received if the acceptance filter is set to the corresponding identifier. This command is used mainly for self test of the controller. If STE=1 which enables the Self Test Mode, the transmit and receive can be successful without other nodes present.

When transmit and receive operations are completed and both transmit and receive interrupts will be generated. The software shall read and verify the received data from RX-FIFO. Then set the Release Receive Buffer (RRB) command to release the memory space.

13.11.3 Self-Loop Test Mode

The software can then check the correctness of the received message and perform loop-back test. The loop-back can be done by shorting CANTX and CANRX, or can be done through the external transceiver on the actual CAN bus. When loop back from the bus, the CAN bus needs to be properly terminated to ensure correct electrical level for normal bus operation.

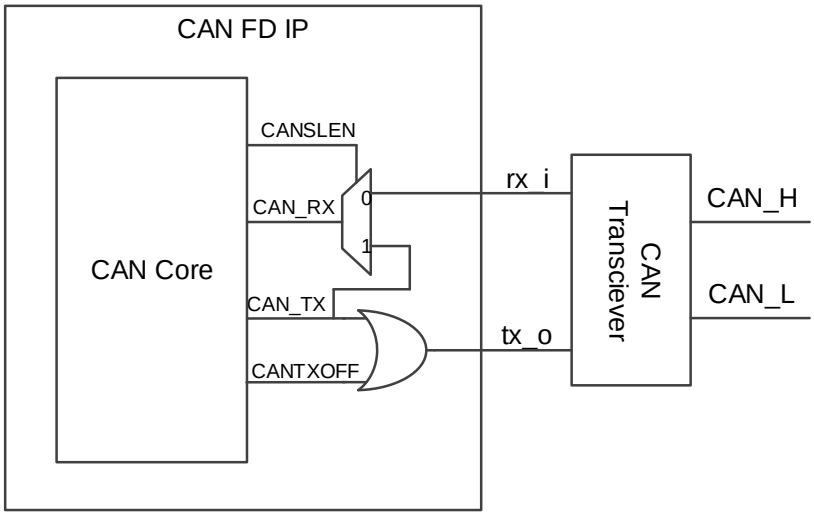


Figure 13-11 CAN bus loop test

CANSLKEY (0x4000_50FE) CAN Self-Loop Security Key Register WO (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-			
WR	-	-	-	-	CANSLKEY[3-0]			

CANSLKEY is used to protect the CANSLEN. Writing 0x6 into CANSLKEY[3-0] can unlock the access of CANSLEN. This register can only be modified in reset mode (RSTM=1).

CANSLEN (0x4000_50FF) CAN Self-Loop Enable Register (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	CANTXOFF	CANSLEN
WR	IPRESET	-	-	-	-	-	CANTXOFF	CANSLEN

IPRESET — CAN FD IP Reset. This register can be programed only when Self-Loop Security Key is unlocked in reset mode (RSTM=1).
Setting IPRESET=1 will generate an internal reset signal to initiate all registers and all mechanisms in the CAN FD controller. IPRESET is cleared automatically.

CANTXOFF — CAN TX Off. This register can be programed only when Self-Loop Security Key is unlocked in reset mode (RSTM=1).
CANTXOFF =1 Force tx_o always output logical high to CAN transceiver.
CANTXOFF =0 tx_o is normal function to CAN transceiver

CANSLEN — CAN Self-Loop Test Mode. This register can be programed only when Self-Loop Security Key is unlocked in reset mode (RSTM=1).
CANSLEN=1 Enable CAN Self-Loop Test Mode
CANSLEN=0 Disable CAN Self-Loop Test Mode

13.12 Self Test Example

13.12.1 Step 1

In Reset Mode (RSTM=1), unlock the CANSLKEY then enable CAN Self-Loop Test Mode and set CANTXOFF to bypass the external CAN Transceiver.

13.12.2 Step 2

Enable Self Test Mode (STE=1) to ignore all ACK errors.

13.12.3 Step 3

In Reset Mode (RSTM=1), configure all bit rate timing registers, operation setting registers and RX acceptance filters same as normal operation. In order to read and compare the TX/RX data by MCU, the RX DMA Mode (DMAEN) shall be disable.

13.12.4 Step 4

Enter normal operation mode (RSTM=0), enable both Transmit Interrupt (TIE) and Receive Interrupt (RIE). Program the test ID and Data into the Transmit Buffers then enable Self Receive Request (SRR) command to start the transmit and receive.

13.12.5 Step 5

When can_irq is issued, software shall read out the received data from the Receive Buffer and compare with the transmitted data.

13.13 Bit Time Configuration Suggestion

In the general application, recommend assigning both SJW[1-0] and FDSJW[1-0] a nonzero value, such as 0x1 or 0x2. Following tables are the examples of bit time configuration under 120MHz, 80Mhz, and 16MHz CANCLK frequency. In real CAN system, these setting shall be readjusted with transceiver's timing.

13.13.1 CANCLK = 120MHz

Bit Rate	Nominal bit time configuration			Data bit time configuration		
	CANCS[5-0]	TSEG2[2-0]	TSEG1[3-0]	CANFDCS[5-0]	FDTSEG2[2-0]	FDTSEG1[3-0]
10 Mbps	N/A	N/A	N/A	0	3	6
8 Mbps	N/A	N/A	N/A	0	4	8
6 Mbps	N/A	N/A	N/A	0	5	12
5 Mbps	N/A	N/A	N/A	1	3	6
4 Mbps	N/A	N/A	N/A	1	4	8
3 Mbps	N/A	N/A	N/A	1	5	12
2 Mbps	N/A	N/A	N/A	2	5	12

Bit Rate	Nominal bit time configuration			Data bit time configuration		
	CANCS[5-0]	TSEG2[2-0]	TSEG1[3-0]	CANFDCS[5-0]	FDTSEG2[2-0]	FDTSEG1[3-0]
1 Mbps	5	5	12	5	5	12
500 Kbps	11	5	12	11	5	12

N/A: The maximal nominal bit rate is 1Mbps

13.13.2 CANCLK = 80MHz

Bit Rate	Nominal bit time configuration			Data bit time configuration		
	CANCS[5-0]	TSEG2[2-0]	TSEG1[3-0]	CANFDCS[5-0]	FDTSEG2[2-0]	FDTSEG1[3-0]
8 Mbps	N/A	N/A	N/A	0	2	5
5 Mbps	N/A	N/A	N/A	0	4	9
4 Mbps	N/A	N/A	N/A	0	5	12
2 Mbps	N/A	N/A	N/A	1	5	12
1 Mbps	3	5	12	3	5	12
500 Kbps	7	5	12	7	5	12

N/A: The maximal nominal bit rate is 1Mbps.

13.13.3 CANCLK = 16MHz

Bit Rate	Nominal bit time configuration			Data bit time configuration		
	CANCS[5-0]	TSEG2[2-0]	TSEG1[3-0]	CANFDCS[5-0]	FDTSEG2[2-0]	FDTSEG1[3-0]
1 Mbps	0	4	9	0	4	9
500 Kbps	1	4	9	1	4	9

N/A: The maximal nominal bit rate is 1Mbps.

14. I²C Master Controller (I2CM)

The I²C master controller provides the interface to I²C slave devices. It can be programmed to operate with arbitration and clock synchronization to allow it to operate in multi-master configurations. The master uses SCL and SDA pins. The controller contains a built-in 8-bit timer to allow various I²C bus speeds. The maximum I²C bus speed is limited to SYSCLK/12.

I2CMTP (0x4000_6000) I²C Master Time Period R/W (0x01)

	7	6	5	4	3	2	1	0
RD	I2CMTP[7-0]							
WR	I2CMTP[7-0]							

I2CMTP — I²C Clock Speed Setting.
This register set the period time of I²C bus clock – SCL. The SCL period time is set according to
SCLPERIOD = 8 * (1 + I2CMTP) * CPUCLK_PERIORD

I2CMSA (0x4000_6004) I²C Master Slave Address R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SA[6-0]							RS
WR	SA[6-0]							RS

SA[6-0] — Slave Address.
SA[6-0] defines the slave address the I²C master uses to communicate.

RS — Receive/Send Bit.
RS determines if the following operation is to RECEIVE (RS=1) or SEND (RS=0).

I2CMBUF (0x4000_6008) I²C Master Data Buffer Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RD[7-0]							
WR	TD[7-0]							

I2CMBUF functions as a transmit-data register when written and as a receive-data register when read. When written, TD is sent to the bus by the next SEND or BURST SEND operations. TD[7] is sent first. When read, RD contains the 8-bit data received from the bus upon the last RECEIVE or BURST RECEIVE operation.

I2CMCR (0x4000_600C) I²C Master Control and Status Register R/W (0x20)

	7	6	5	4	3	2	1	0
RD	-	BUSBUSY	IDLE	ARBLOST	DACKERR	AACKERR	ERROR	BUSY
WR	CLEAR	INFILEN	-	-	ACK	STOP	START	RUN

The I2CMCR register is used for setting control when it is written, and as a status signal when read.

CLEAR — Clear I2CM state machine.
Writing 1 to CLEAR, clears the state machine and also clears I2CMINT flags.

BUSBUSY — This bit indicates that the external I²C bus is busy and access to the bus is not possible. This bit is set/reset by START and STOP conditions.

INFILEN — Input Noise Filter Enable. When IFILEN is set, pulses shorter than 50 nsec on inputs of SDA and SCL are filtered out.

IDLE — This bit indicates that I²C master is in the IDLE mode.

ARBLOST — This bit is automatically set when the last operation I²C master controller loses the bus arbitration.

DACKERR — This bit is automatically set when the last operation transmitted data is not acknowledged.

ACK — Refer to Table 14-1, Table 14-2, Table 14-3

AACKERR — This bit is automatically set when the last operation slave address transmitted is not acknowledged.

STOP — Refer to Table 14-1, Table 14-2, Table 14-3

ERROR — This bit indicates that an error occurs in the last operation. The errors include slave address was not acknowledged, or transmitted data is not acknowledged, or the master controller loses arbitration.

START	Refer to Table 14-1, Table 14-2, Table 14-3
BUSY	This bit indicates that I²C master is receiving or transmitting data, and other status bits are not valid.
RUN	Refer to Table

START, STOP, RUN and RS, ACK bits are used to drive I²C Master to initiate and terminate a transaction. The Start bit generates START, or REPEAT START protocol. The Stop bit determines if the cycle stops at the end of the data cycle or continues to a burst. To generate a single read cycle, the designated address is written in SA, RS is set to 1, ACK=0, STOP=1, START=1, RUN=1 are set in I2CMCR to perform the operation and then STOP. When the operation is completed (or aborted due to errors), I²C master generates an interrupt. The ACK bit must be set to 1. This causes the controller to send an ACK automatically after each byte transaction. The ACK bit must be reset to 0 when the master operates in receive mode and not to receive further data from the slave devices. The following table lists the permitted control bits combinations in master IDLE mode.

RS	ACK	Stop	Start	Run	Operations
0	-	0	1	1	START condition followed by SEND. Master remains in TRANSMITTER mode
0	-	1	1	1	START condition followed by SEND and STOP
1	0	0	1	1	START condition followed by RECEIVE operation with negative ACK. Master remains in RECEIVER mode
1	0	1	1	1	START condition followed by RECEIVE and STOP
1	1	0	1	1	START condition followed by RECEIVE. Master remains in RECEIVER mode
1	1	1	1	1	Illegal command

Table 14-1 Master I²C Command in Idle Mode

The following table lists the permitted control bits combinations in master TRANSMITTER mode.

RS	ACK	Stop	Start	Run	Operations
-	-	0	0	1	SEND operation. Master remains in TRANSMITTER mode
-	-	1	0	0	STOP condition
-	-	1	0	1	SEND followed by STOP condition
0	-	0	1	1	REPEAT START condition followed by SEND. Master remains in TRANSMITTER mode
1	-	1	1	1	REPEAT START condition followed by SEND and STOP condition
1	0	0	1	1	REPEAT START condition followed by RECEIVE operation with negative ACK. Master remains in TRANSMITTER mode
1	0	1	1	1	REPEAT START condition followed by SEND and STOP condition.
1	1	0	1	1	REPEAT START condition followed by RECEIVE. Master remains in RECEIVER mode.
1	1	1	1	1	Illegal command

Table 14-2 Master I²C Command in Transmitter Mode

The following table lists the permitted control bits combinations in master RECEIVER mode.

RS	ACK	Stop	Start	Run	Operations
-	0	0	0	1	RECEIVE operation with negative ACK. Master remains in RECEIVE mode
-	-	1	0	0	STOP condition
-	0	1	0	1	RECEIVE followed by STOP condition
-	1	0	0	1	RECEIVE operation. Master remains in RECEIVER mode
-	1	1	0	1	Illegal command

RS	ACK	Stop	Start	Run	Operations
1	0	0	1	1	REPEAT START condition followed by RECEIVE operation with negative ACK. Master remains in RECEIVER mode
1	0	1	1	1	REPEAT START condition followed by RECEIVE and STOP conditions
1	0	1	1	1	REPEAT START condition followed by RECEIVE. Master remains in RECEIVER mode
0	-	0	1	1	REPEAT START condition followed by SEND. Master remains in TRANSMITTER mode.
0	-	1	1	1	REPEAT START condition followed by SEND and STOP conditions.

Table 14-3 Master I²C Command in Receiver Mode

All other control-bit combinations not included in three tables above are NOP. In Master RECEIVER mode, STOP should be generated only after data negative ACK executed by Master or address negative ACK executed by slave. Negative ACK means SDA is pulled low when the acknowledge clock pulse is generated.

I2CMTO0 (0x4000_6010) I²C Time Out Control Register 0 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	I2CMTO[7-0]							
WR	I2CMTO[7-0]							

I2CMTO1 (0x4000_6011) I²C Time Out Control Register 1 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	I2CMTOEN	I2CMTO[14-8]						
WR	I2CMTOEN	I2CMTO[14-8]						

I2CMTOEN — I2CM Time Out Enable
I2CMTO[14-0] — I2CM Time Out Setting
The TO time is set to (I2CMTO[14-0]+1)*2*SCLPERIOD. When time out occurs, an I2CM interrupt will be generated.

I2CMINT (0x4000_6014) I²C Master Interrupt Control R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-						I2CMTOF	I2CMINTF
WR	-						-	I2CMINTF

I2CMTOF — I2CM Time Out Flag
I2CMTOF is set to 1 if I2CM time out occurs. I2CMTOF is cleared when I2CMINTF is cleared.
I2CMINTF — I2CM Interrupt Status
I2CMINTF is set to 1 when I2CM interrupt occurs. Write 1 to clear.

15. I²C Slave Controller (I2CS1)

The I²C Slave Controller 1 is a regular I²C Slave controller with enhanced functions such as clock-stretching and programmable hold time. These enhancements provide significant improvement on compatibilities. I2CS1 also can be configured to respond to two I²C addresses – I2CADR1 and I2CADR2. These two addresses can be enabled separately. In addition, I2CS1 can be configured as wakeup source under system sleep mode when address match condition occurs.

In receive mode, the controller detects a valid matching address and issues an ADDRMI interrupt. At the same time, the data bit on SDA line is shifted into receive buffer. The RCBI interrupt is generated whenever a complete byte is received and is ready to be read from I2CSDAT. If for any reason, the software does not respond to RCBI interrupt in time (i.e. RCBI is not cleared), and a new byte is received, the controller either forces an NACK response on I²C (if CLKSTREN bit is not set) or by pulling and holding SDA low (if CLKSTREN bit is set) to stretch the SCL low duration to force the master into a wait state. In clock stretching mode, SCL is released when the software responds to RCBI interrupt and cleared RCBI flag.

In transmit mode, the controller detects a valid matching address and issue an ADDRMI interrupt. At the same time, the data preloaded in the transmit data register through I2CSDAT is transferred to the transmit shift register and is serially shifted out onto SDA line. When this occurs, the controller generates a TXBI interrupt to inform the software that a new byte can be written into I2CSDAT. When the shift register is empty and ready for the next transmit, the slave controller checks if the new byte is written to the I2CSDAT. If TXBI is not cleared, it indicates lack of new data and the slave controller holds SCL line low to stretch the current clock cycle if CLKSTREN is set. If the clock stretching is not enabled, the slave controller takes the old byte into the shift register and replies with NACK, thus causing data corruption. On the other hand, if the master returns the NACK after the byte transfer, this indicates the end of data to the I²C slave. In this case, the I²C slave releases the data line to allow the master to generate a STOP or REPEAT START.

I2CS1CONA (0x4000_6100) I2CS1 Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	EADRWK	EADDRMI	ESTOPI	ERPSTARTI	ETXBI	ERCBI	CLKSTREN	EACKWK
WR	EADRWK	EADDRMI	ESTOPI	ERPSTARTI	ETXBI	ERCBI	CLKSTREN	EACKWK

EADRWK — Enable Address matched wakeup from sleep mode,

EADDRMI — ADDRMI Interrupt Enable bit.
Set this bit to set ADDRMI interrupt as the I²C slave interrupt. This interrupt is generated when I²C slave received a matching address.

ESTOPI — STOPI Interrupt Enable bit.
Set this bit to set STOPI interrupt as the I²C slave interrupt.

ERPSTARTI — RPSTARTI Interrupt Enable Bit.
Set this bit to set RPSTARTI interrupt as the I²C slave interrupt.

ETXBI — TXBI Interrupt Enable bit.
Set this bit to allow TXBI interrupt as the I²C slave interrupt.

ERCBI — RCBI Interrupt Enable bit.
Set this bit to allow RCBI interrupt as the I²C slave interrupt.

CLKSTREN — Clock Stretching Enable bit.
Set to enable the clock stretching function of the slave controller. Clock stretching is an optional feature defined in I²C specification.
If the clock stretching option is enabled (for slave I²C), the data written into transmit buffer is shifted out only after the occurrence of clock stretching, and the data cannot be loaded to transmit shift register. The programmer must write the same data again to the transmit buffer.

EACKWK — 1: Enable clock stretching during system wakeup from sleep and wait until system wakeup completed and asks controller send ACK to master.
0: controller send NACK when address matched

I2CS1CONB (0x4000_6104) I2CS1 Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	STRTOF	SADR2M	XMT	START	SDAFLT[1-0]		GDFLT[1-0]	
WR	I2CSRST	-	-	-	SDAFLT[1-0]		GDFLT[1-0]	

STRTOF — Clock Stretch Time Out Flag

STRTOF is set to 1 by hardware when clock stretch exceeds STRTO setting. STRTOF can be cleared only by I2CRST.

I2CSRST — I^2C Slave Reset bit.
Set this bit causes the Slave Controller to reset all internal state machine. Auto clear by controller.

SARD2M — Slave Address Match Flag bit.
This bit is meaningful only when ADDRMI is set.
SARD2M=0 indicates the received I^2C address matches with I2CSADR1.
SARD2M=1 indicates the received I^2C address matches with I2CSADR2.
This bit is cleared when ADDRMI is cleared.

XMT — This bit is set by the controller when the I^2C slave is in transmit operation. This bit is cleared when the I^2C slave controller is in receive operation.

START — Start Condition.
This bit is set when the slave controller detects a START condition on the SCL and SDA lines. This bit is not very useful as the start of transaction can be indicated by address match interrupt. This read-only bit is cleared when STOP condition is detected

SDAFLT[1-0] — Delay filter for SDA input
0: 10ns RC filter delay
1: 20ns RC filter delay
2: 20ns RC filter delay
3: 30ns RC filter delay

GDFLT[1-0] — Global delay filter for SCL and SDA input
0: 10ns RC filter delay
1: 20ns RC filter delay
2: 20ns RC filter delay
3: 30ns RC filter delay

I2CS1ST (0x4000_6108) I2CS1 Status Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ADRWKF	ADDRMI	STOPI	RPSTARTI	TXBI	RCBI	FIRSTBT	NACK
WR	ADRWKF	ADDRMI	STOPI	RPSTARTI	-	-	-	NACK

ADRWKF — Address Matched Wakeup Flag
This bit is set when an address match has caused the wakeup event from SLEEP mode.
Write 1 to clear ADRWKF.

ADDRMI — Slave Address Match Interrupt Flag bit.
This bit is set when the received address matches the address defined in I2CSADR1. If EADDMI is set, this generates an interrupt. This bit must be cleared by software.
Write 1 to clear ADDRMI flag.

STOPI — Stop Condition Interrupt Flag bit.
This bit is set when the slave controller detects a STOP condition on the SCL and SDA lines. This bit must be cleared by software.
Write 1 to clear STOPI flag.

RPSTARTI — Repeat Start Condition Interrupt Flag bit.
This bit is set when the slave controller detects a REPEAT START condition on the SCL and SDA lines. This bit must be cleared by software.
Write 1 to clear RPSTARTI flag.

TXBI — Transmit Buffer Interrupt Flag. This bit is set when the slave controller is ready to accept a new byte for transmission. This bit is cleared when new data is written into I2CSDAT register.

RCBI — Receiver Buffer Interrupt Flag bit. This bit is set when the slave controller puts new data in the I2CSDAT and ready for software reading. This bit is cleared after the software reads I2CSDAT.

FIRSTBT — This bit is set to indicate the data in the data register as the first byte received after address match. This bit is cleared after the second byte received. The bit is read only and generated by the slave controller.

NACK — NACK Condition bit.

This bit is set when the host responds with NACK in the byte transaction. This bit is only meaningful for slave-transmit operation. Please note if the master returns with NACK on the byte transaction, the slave does not upload new data into the shift register. And the slave transmits the old data again as the next transfer, and this re-transmission continues if NACK is repeated until the transmission is successful and returned with ACK. This bit is cleared when a new ACK is detected or it can be cleared by software.
Write 1 to clear NACK flag

I2CS1ADR1 (0x4000_610C) I2CS1 1st Slave Address Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	I2CSEN1	RXADDR[6-0]						
WR	I2CSEN1	ADDRA1[6-0]						

I2CSEN1 Set this bit to enable the I²C slave controller and ADDRA1[6-0] for address matching. Please note either I2CSEN1 or I2CSEN2 is 1 will enable I2CS.

RXADDR Received slaved address

ADDRA1[6-0] 7-bit slave address.

I2CS1DAT (0x4000_6110) I2CS1 Data Register R/W (0xFF)

	7	6	5	4	3	2	1	0
RD	I²C Slave Receive Data Register							
WR	I²C Slave Transmit Data Register							

I2CS1ADR2 (0x4000_6114) I2CS1 2nd Slave Address Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	I2CSEN2	ADDRA2[6-0]						
WR	I2CSEN2	ADDRA2[6-0]						

I2CSEN2 Set this bit to enable the I²C slave controller and ADDRA2[6-0] for address matching
Please note either I2CSEN1 or I2CSEN2 is 1 will enable I2CS.

ADDRA2[6-0] 7-bit slave address.

I2CS1STO (0x4000_6118) Stretch Time Out Register R/W (0xFF)

	7	6	5	4	3	2	1	0
RD	STRTO[7-0]							
WR	STRTO[7-0]							

STRTO[7-0] Clock Stretch Time Out Setting
To prevent extended clock stretching duration interferes with normal I²C bus operation, a time out mechanism is implemented. When time out occurs, an interrupt is generated and STRTOF bit is set. Since this is an adverse and dire situation, I2CRST must be done to resume I2CS operation. The time out setting is SOSC32K/4 * STRTO[7-0].
Please note STRTO should be set longer than REGRDY delay to prevent error when address matching wakeup is used.

15.1 Clock Stretching

Clock stretching is used to allow a slow I²C slave to hold the transaction from the master by driving SCL low after ACK bit and before the start of 1st bit of next byte. In another word, the SCL is driven low by the slave after it sees SCL is driven low by the host at ACK field. Maximum delay the slave can stretch does not have a limit but should be limited to 35msec if SMBus conformance is required. Clock stretching has an impact on the slave response and behaves differently for host read or write. The waveform for clock stretching disabled and enabled are shown in the following diagram.

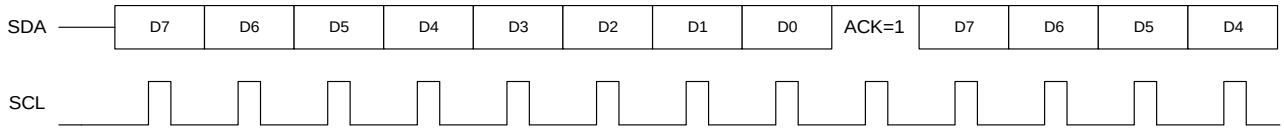


Figure 15-1 I²C without lock stretch

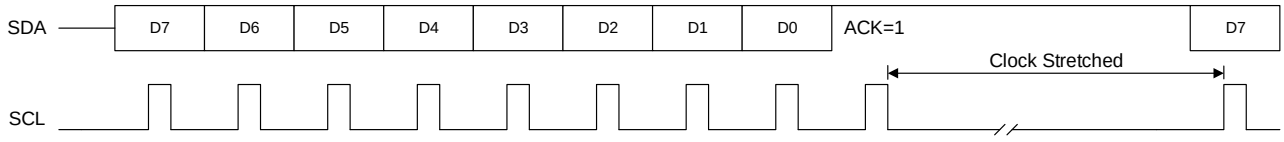


Figure 15-2 I²C with clock stretch

For host write, the address byte is always responded to with ACK=1. The data byte will respond with ACK=1 if data buffer is empty. If the data buffer is not empty, I2CS1 will respond with ACK=0 (NACK), if clock stretching is not enabled. If clock stretching is enabled, after responding with ACK=1, I2CS1 will pull SCL low to facilitate clock stretching. And the software needs to read out data buffer, and I2CS1 then release SCL for host to continue.

For host read, the address byte is responded with ACK=1 if the data buffer is ready. However, if data buffer is not ready and clock stretching is not enabled, then ACK=0 is responded. If clock stretching is enabled, ACK=1 is responded but clock stretching is commenced by I2CS1 until data buffer is written by software. Similar action is taken for the data transaction that ACK=1 is first responded with clock stretching.

16. EUART1

EUART1 is an enhanced UART controller (EUART) with separate transmit and receive FIFO. Both transmit and receive FIFO are 15-bytes deep and can be parameterized for interrupt triggering. The addition of FIFO significantly reduces the CPU load to handle high-speed serial interface. Transmit FIFO and receive FIFO have respective interrupt trigger levels that can be set based on optimal CPU performance adjustment. The EUART also has dedicated 16-bit Baud Rate generator and thus provides accurate baud rate under wide range of system clock frequency. The following registers are used for configurations of and interface with EUART1.

SCON1A (0x4000_7100) EUART1 Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	EUARTEN	SB	WLS[1-0]		BREAK	OP	PERR	SP
WR	EUARTEN	SB	WLS[1-0]		BREAK	OP	PE	SP

EUARTEN — Transmit and Receive Enable bit
Set to enable EUART1 transmit and receive functions: To transmit messages in the TX FIFO and to store received messages in the RX FIFO.

SB — Stop Bit Control
Set to enable 2 Stop bits, and clear to enable 1 Stop bit.

WLS[1-0] — The number of bits of a data byte. This does not include the parity bit when parity is enabled.
00 - 9 bits
01 - 6 bits
10 - 7 bits
11 - 8 bits

BREAK — Start Sending BREAK and followed by SYNC byte (if BRKSYNC=1).
Set to initiate a break condition on the UART interface by holding UART output at low for duration of BRKLEN, and then followed by a SYNC byte (if BRKSYNC=1).
When read, 1 indicates it is still ongoing. It is self-cleared by hardware when completed.
At completion, it also generates an EUART1 interrupt.
Software can start putting data into TX FIFO. The data will start transmission after SYNC byte is transmitted.

OP — Odd/Even Parity Control Bit

PE/PERR — Parity Enable / Parity Error status
Set to enable parity and clear to disable parity checking functions. If read, PERR=1 indicates a parity error in the current data of RX FIFO.

SP — Parity Set Control Bit
When SP is set, the parity bit is always transmitted as 1.

SCON1B (0x4000_7101) EUART1 Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RXST	BITERR	BECLRX	BECLRR	LBKEN	BERIE	-	TXPOL
WR	-	BITERR	BECLRX	BECLRR	LBKEN	BERIE	CLRFIFO	TXPOL

RXST — Receive Status
RXST controlled by hardware. RXST is set by hardware when a START bit is detected. It is cleared when STOP condition is detected.

BITERR — Bit Error Flag
BITERR is set by hardware when received bit does not match with transmit bit, if BERIE=1, then this error generates an interrupt. BITERR must be cleared by software. Write 1 to clear.

BECLRX — Bit Error Force Clear Transmit Enable
If BECLRX=1, when BITERR is set by hardware, hardware also immediately disables current transmission and clears TX state machines and FIFO.

BECLRR — Bit Error Force Clear RECEIVE Enable
If BECLRX=1, when BITERR is set by hardware, hardware also immediately disables current reception and clears RX state machines and FIFO.

LBKEN — Enable EUART Loopback Test,

When LBKEN=1, EUART1 enters loopback mode, with its TX output connected to RX input. When in loopback mode, to prevent the TX output to pin, corresponding MFCFG bit must be cleared.

BERIE — Bit Error Interrupt Enable (1:Enable / 0:Disable)

CLRFIFO — Set to clear transmit/received FIFO level and state machine. CLRFIFO bit is auto clear by hardware

TXPOL — EUART output polarity

SCON1C (0x4000_7102) EUART1 Configuration Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	RCVSPL[1-0]	
WR	-	-	CLRRX	CLRTX	-	-	RCVSPL[1-0]	

CLRRX — Clear RX FIFO level and RX state

CLRTX — Clear TX FIFO level and TX state

RCVSPL[1-0] — Adjust Receive Sampling Point
00 = 50%
01 = 62.5%
10 = 69%
11 = 75%

SCON1D (0x4000_7103) EUART1 Configuration Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	BRKIEN	BRKBR[2-0]			BRKF	BRKSYNC	BRKLEN[1-0]	
WR	BRKIEN	BRKBR[2-0]			BRKF	BRKSYNC	BRKLEN[1-0]	

BRKIEN — BREAK Completion Interrupt Enable
BRKIEN=1 enables EUART1 interrupt when BRK/SYNC transmission is completed

BRKBR[2-0] — Break Field Baud Rate
Break Baud rate is based on BR.
000 = BR
001 = BR/2
010 = BR/4
011 = BR/8
100 = BR/16
101 = BR/64
110 = BR/128
111 = BR/256

BRKF — BREAK Completion Flag
BRKF is set by hardware when BRK/SYNC transmission completes. It must be cleared by software.

BRKSYNC — Send SYNC after Break
If BRKSYNC=0, then only the Break field is sent.
If BRKSYNC=1, then after Break field, a SYNC byte is also sent.

BRKLEN[1-0] — BREAK Length Setting
00 = 13 BT
01 = 14 BT
10 = 15 BT
11 = 16 BT

SBAUD1 (0x4000_7104) EUART1 Baud Rate Register RW (0x00XX_0000)

	31-24	23-16	15-0
RD	BRCS[7-0]	-	BR[15-0]
WR	BRCS[7-0]	-	BR[15-0]

BRCS[7-0] — Baud Rate Pre-Scaler

BR[15-0] — The Baud Rate Setting of EUART.
BUAD RATE = SYSCLK/(BRCS[7-0]+1)/(BR[15-0]+1).

SFIFO1 (0x4000_7108) EUART1 FIFO Status/Control Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RFL[3-0]				TFL[3-0]			
WR	RFLT[3-0]				TFLT[3-0]			

RFL[3-0] Current Receive FIFO level. This is read only and indicate the current receive FIFO byte count.

RFLT[3-0] Receive FIFO trigger threshold. This is write-only. RDA interrupt will be generated when RFL[3-0] is greater than RFLT[3-0].

RFLT[3-0]	Description
0000	RX FIFO trigger level = 0
0001	RX FIFO trigger level = 1
0010	RX FIFO trigger level = 2
0011	RX FIFO trigger level = 3
0100	RX FIFO trigger level = 4
0101	RX FIFO trigger level = 5
0110	RX FIFO trigger level = 6
0111	RX FIFO trigger level = 7
1000	RX FIFO trigger level = 8
1001	RX FIFO trigger level = 9
1010	RX FIFO trigger level = 10
1011	RX FIFO trigger level = 11
1100	RX FIFO trigger level = 12
1101	RX FIFO trigger level = 13
1110	RX FIFO trigger level = 14
1111	NA

TFL[3-0] Current Transmit FIFO level. This is read only and indicate the current transmit FIFO byte count.

TFLT[3-0] Transmit FIFO trigger threshold. This is write-only. TRA interrupt will be generated when TFL[3-0] is less than TFLT[3-0].

TFLT[3-0]	Description
0000	NA
0001	TX FIFO trigger level = 1
0010	TX FIFO trigger level = 2
0011	TX FIFO trigger level = 3
0100	TX FIFO trigger level = 4
0101	TX FIFO trigger level = 5
0110	TX FIFO trigger level = 6
0111	TX FIFO trigger level = 7
1000	TX FIFO trigger level = 8
1001	TX FIFO trigger level = 9
1010	TX FIFO trigger level = 10
1011	TX FIFO trigger level = 11
1100	TX FIFO trigger level = 12
1101	TX FIFO trigger level = 13

TFLT[3-0]	Description
1110	TX FIFO trigger level = 14
1111	TX FIFO trigger level = 15

Receive and transmit FIFO level can be reset by clear RX/TX FIFO operation.

SINT1L (0x4000_710C) EUART1 Interrupt Enable Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	INTEN	TRAEN	RDAEN	RFOEN	RFUEN	TFOEN	FERREN	TIEN
WR	INTEN	TRAEN	RDAEN	RFOEN	RFUEN	TFOEN	FERREN	TIEN

INTEN — Interrupt Enable
Set to enable EUART1 interrupt. Clear to disable interrupt. Default is 0.

TRAEN — Transmit FIFO Ready Interrupt Enable.

RDAEN — Receive FIFO Ready Interrupt Enable.

RFOEN — Receive FIFO Overflow interrupt Enable

RFUEN — Receive FIFO Underflow Interrupt Enable

TFOEN — Transmit FIFO Overflow Interrupt Enable

FERREN — Framing Error Interrupt Enable

TIEN — Transmit Message Completion Interrupt Enable

SINT1H (0x4000_710D) EUART1 Interrupt Status Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	TRA	RDA	RFO	RFU	TFO	FERR	TI
WR	-	TRA	RDA	RFO	RFU	TFO	FERR	TI

TRA — Transmit FIFO Ready Flag
This bit is set when transmit FIFO has been emptied below FIFO threshold.
The flag is cleared by software writing “1”.

RDA — Receive FIFO Ready Flag
This bit is set by hardware when receive FIFO exceeds the FIFO threshold. RDA will also be set when RFL < RFLT for bus idle duration longer than (RFLT +1) * 16 * Bit time. This is to inform software that there are remaining unread received bytes in the FIFO.
The flag is cleared by software writing “1”.

RFO — Receive FIFO Overflow Flag
This bit is set when overflow condition of receive FIFO occurs.
The flag is cleared by software writing “1” or by FIFO reset action.

RFU — Receive FIFO Underflow Flag
This bit is set when underflow condition of receive FIFO occurs.
The flag is cleared by software writing “1” or by FIFO reset action.

TFO — Transmit FIFO Overflow Flag
This bit is set when overflow condition of transmit FIFO occurs.
The flag is cleared by software writing “1” or by FIFO reset action.

FERR — Framing Error Flag
This bit is set when framing error occurs as the byte is received.
The flag is cleared by software writing “1”.

TI — Transmit Message Completion Flag
This bit is set when all messages in the TX FIFO are transmitted and thus the TX FIFO becomes empty.
The flag is cleared by software writing “1”

SBUF1L (0x4000_7110) EUART1 Data Buffer Low Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	EUART1 Receive Data Register RXDATA[7-0]							
WR	EUART1 Transmit Data Register TXDATA[7-0]							

SBUF1H (0x4000_7111) EUART1 Data Buffer High Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	RXDATA[8]
WR	-	-	-	-	-	-	-	TXDATA[8]

This register is the virtual data buffer register for both receive and transmit FIFO. When being read, it reads out the top byte of the RX FIFO; when written, it writes into the top byte of the TX FIFO.

Writing SBUF1 causes TFL + 1 and any reading SBUF1 cause RFL - 1, the 9 bit data transfer have to use word or half-word to write/read.

17. EUART2 with LIN Controller (EUART2)

LIN-capable 16550-like EUART2 is an enhanced UART controller (EUART) with separate transmit and receive FIFO. Both transmit and receive FIFO are 15-bytes deep and can be parameterized for interrupt triggering. The addition of FIFO significantly reduces the CPU load to handle high-speed serial interface. Transmit FIFO and receive FIFO have respective interrupt trigger levels that can be set based on optimal CPU performance adjustment. The EUART also has dedicated 16-bit Baud Rate generator and thus provides accurate baud rate under wide range of system clock frequency. The EUART2 also provides LIN extensions that incorporate message handling and baud-rate synchronization. The Block Diagram of EUART2 is shown in the following.

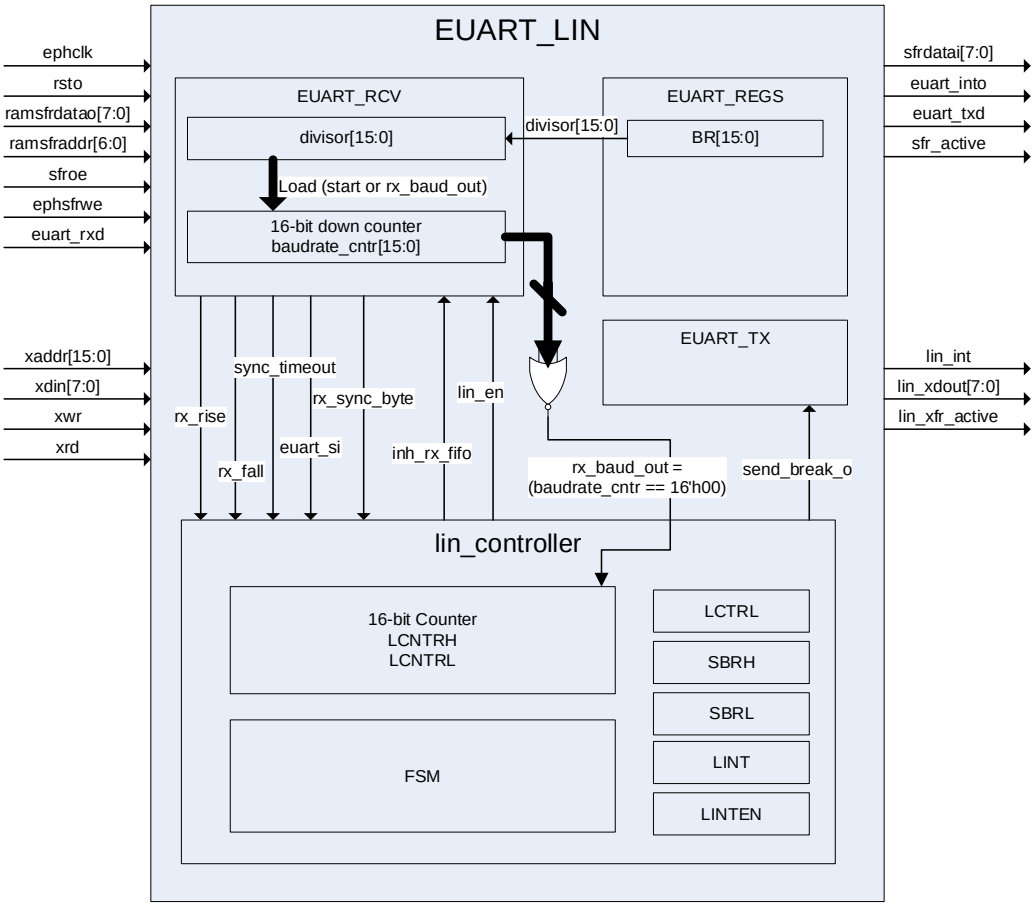


Figure 17-1 EUART LIN Block Diagram

SCON2A (0x4000_7200) EUART2 Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	EUARTEN	SB	WLS[1-0]		BREAK	OP	PERR	SP
WR	EUARTEN	SB	WLS[1-0]		BREAK	OP	PE	SP

EUARTEN — Transmit and Receive Enable bit
Set to enable EUART2 transmit and receive functions: To transmit messages in the TX FIFO and to store received messages in the RX FIFO.

SB — Stop Bit Control
Set to enable 2 Stop bits, and clear to enable 1 Stop bit.

WLS[1-0] — The number of bits of a data byte. This does not include the parity bit when parity is enabled.
00 - 9 bits
01 - 6 bits
10 - 7 bits
11 - 8 bits

BREAK — BREAK Condition Control Bit.
Set to initiate a break condition on the UART interface by holding UART output at low until BREAK bit is cleared.

OP — Odd/Even Parity Control Bit

PE/PERR — Parity Enable / Parity Error status
Set to enable parity and clear to disable parity checking functions. If read, PERR=1 indicates a parity error in the current data of RX FIFO.

SP — Parity Set Control Bit
When SP is set, the parity bit is always transmitted as 1.

SCON2C (0x4000_7202) EUART2 Interrupt Status/Enable Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	RCVSPL[1-0]	
WR	-	-	-	-	-	-	RCVSPL[1-0]	

RCVSPL[1-0] — Adjust Receive Sampling Point
00 = 50%
01 = 62.5%
10 = 69%
11 = 75%

SBAUD2 (0x4000_7204) EUART2 Baud Rate Register RW (0x00XX_0000)

	31-24	23-16	15-0
RD	BRCS[7-0]	-	CBR[15-0]
WR	BRCS[7-0]	-	BR[15-0]

BRCS[7-0] — Baud Rate Pre-Scaler

CBR[15-0] — The current Baud Rate used for EUART

BR[15-0] — The Baud Rate Setting of EUART.
BUAD RATE = SYSCLK/(BRCS[7-0]+1)/(BR[15-0]+1).
When a slave receives a BREAK followed by a valid SYNC field, an RSI interrupt is generated and the acquired baud rate from SYNC field is stored in SBR[15-0]. The acquired baud rate is BAUD RATE = SYSCLK/(BRCS[7-0]+1)/(SBR[15-0]+1). The software can just update this acquired value SBR[15-0] into BR[15-0] to achieve synchronization with the master. If Auto-Sync Update (ASU) register bit is enabled under LIN slave mode, LIN controller will automatically update BR[15-0] with SBR[15-0] and issue another ASUI interrupt when received a valid SYNC field.

SFIFO2 (0x4000_7208) EUART2 FIFO Status/Control Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RFL[3-0]				TFL[3-0]			
WR	RFLT[3-0]				TFLT[3-0]			

RFL[3-0] — Current Receive FIFO level. This is read only and indicate the current receive FIFO byte count.

RFLT[3-0] — Receive FIFO trigger threshold. This is write-only. RDA interrupt will be generated when RFL[3-0] is greater than RFLT[3-0].

RFLT[3-0]	Description
0000	RX FIFO trigger level = 0
0001	RX FIFO trigger level = 1
0010	RX FIFO trigger level = 2
0011	RX FIFO trigger level = 3
0100	RX FIFO trigger level = 4
0101	RX FIFO trigger level = 5
0110	RX FIFO trigger level = 6
0111	RX FIFO trigger level = 7
1000	RX FIFO trigger level = 8
1001	RX FIFO trigger level = 9
1010	RX FIFO trigger level = 10

RFLT[3-0]	Description
1011	RX FIFO trigger level = 11
1100	RX FIFO trigger level = 12
1101	RX FIFO trigger level = 13
1110	RX FIFO trigger level = 14
1111	NA

TFL[3-0] Current Transmit FIFO level. This is read only and indicate the current transmit FIFO byte count.

TFLT[3-0] Transmit FIFO trigger threshold. This is write-only. TRA interrupt will be generated when TFL[3-0] is less than TFLT[3-0].

TFLT[3-0]	Description
0000	NA
0001	TX FIFO trigger level = 1
0010	TX FIFO trigger level = 2
0011	TX FIFO trigger level = 3
0100	TX FIFO trigger level = 4
0101	TX FIFO trigger level = 5
0110	TX FIFO trigger level = 6
0111	TX FIFO trigger level = 7
1000	TX FIFO trigger level = 8
1001	TX FIFO trigger level = 9
1010	TX FIFO trigger level = 10
1011	TX FIFO trigger level = 11
1100	TX FIFO trigger level = 12
1101	TX FIFO trigger level = 13
1110	TX FIFO trigger level = 14
1111	TX FIFO trigger level = 15

Receive and transmit FIFO level can be reset by clear TX/RX FIFO operation.

SBR2 (0x4000_720A) EUART2 Baud Rate Register RO (0x0000)

	15-0
RD	SBR[15-0]
WR	-

SBR[15-0] The acquired Baud Rate under LIN protocol. This is read-only.
SBR[15-0] is the acquired baud rate from last received valid sync byte. SBR is meaningful only in LIN-Slave mode. The actual baud rate acquired is SYSCLK/(BRCS[7-0]+1)/(SBR[15-0]+1).

SINT2L (0x4000_720C) EUART2 Interrupt Enable Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	INTEN	TRAEN	RDAEN	RFOEN	RFUEN	TFOEN	FERREN	TIEN
WR	INTEN	TRAEN	RDAEN	RFOEN	RFUEN	TFOEN	FERREN	TIEN

INTEN Interrupt Enable
Set to enable EUART2 interrupt. Clear to disable interrupt. Default is 0.

TRAEN Transmit FIFO Ready Interrupt Enable.

RDAEN Receive FIFO Ready Interrupt Enable.

RFOEN Receive FIFO Overflow interrupt Enable

RFUEN	Receive FIFO Underflow Interrupt Enable
TFOEN	Transmit FIFO Overflow Interrupt Enable
FERREN	Framing Error Interrupt Enable
TIEN	Transmit Message Completion Interrupt Enable

SINT2H (0x4000_720D) EUART2 Interrupt Status Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	TRA	RDA	RFO	RFU	TFO	FERR	TI
WR	-	TRA	RDA	RFO	RFU	TFO	FERR	TI

TRA	Transmit FIFO Ready Flag This bit is set when transmit FIFO has been emptied below FIFO threshold. The flag is cleared by software writing “1”.
RDA	Receive FIFO Ready Flag This bit is set by hardware when receive FIFO exceeds the FIFO threshold. RDA will also be set when RFL < RFLT for bus idle duration longer than (RFLT +1) * 16 * bit time. This is to inform software that there are still remaining unread received bytes in the FIFO. The flag is cleared by software writing “1”.
RFO	Receive FIFO Overflow Flag This bit is set when overflow condition of receive FIFO occurs. The flag is cleared by software writing “1” or by FIFO reset action.
RFU	Receive FIFO Underflow Flag This bit is set when underflow condition of receive FIFO occurs. The flag is cleared by software writing “1” or by FIFO reset action.
TFO	Transmit FIFO Overflow Flag This bit is set when overflow condition of transmit FIFO occurs. The flag is cleared by software writing “1” or by FIFO reset action.
FERR	Framing Error Flag This bit is set when framing error occurs as the byte is received. The flag is cleared by software writing “1”.
TI	Transmit Message Completion Flag This bit is set when all messages in the TX FIFO are transmitted and thus the TX FIFO becomes empty. The flag is cleared by software writing “1”.

SBUF2L (0x4000_7210) EUART2 Data Buffer Low Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	EUART2 Receive Data Register RXDATA[7-0]							
WR	EUART2 Transmit Data Register TXDATA[7-0]							

SBUF2H (0x4000_7211) EUART2 Data Buffer High Register (0x00) R/W

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	RXDATA[8]
WR	-	-	-	-	-	-	-	TXDATA[8]

This register is the virtual data buffer register for both receive and transmit FIFO. When being read, it reads out the top byte of the RX FIFO; when written, it writes into the top byte of the TX FIFO.

Writing SBUF2 causes TFL + 1 and any reading SBUF2 cause RFL - 1, the 9 bit data transfer have to use word or half-word to write/read.

EUART2 can be configured to add LIN capability. The major enhancement of LIN includes master/slave configurations, auto baud-rate synchronization, and frame-based protocol with header. Under LIN extension mode, all EUART2 registers and functions are still effective and operational. LIN is a single-wire bus and it requires external components to combine RX and TX signals externally. LIN is frame based and consists of message protocols with master/slave configurations. The following diagram shows the basic composition of a header message sent by the master. It starts with BREAK, the SYNC byte, ID bytes, DATA bytes, and CRC bytes.

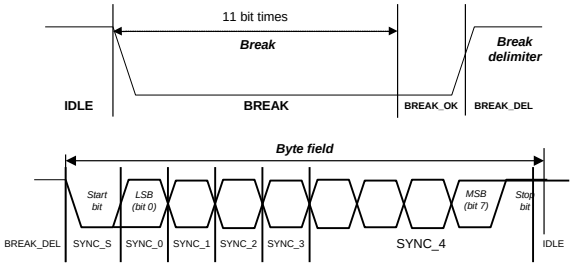


Figure 17-2 LIN Bus data format for Break and Sync field

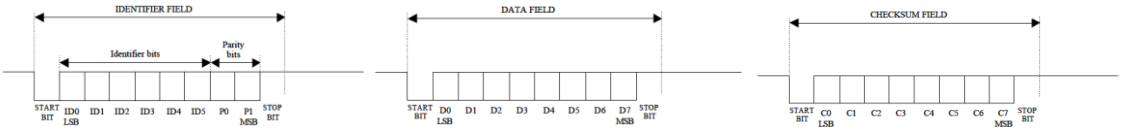


Figure 17-3 LIN Bus Data Format for ID, Data and Checksum field

A LIN frame structure is shown and the frame time matches the number of bits sent and has a fixed timing.

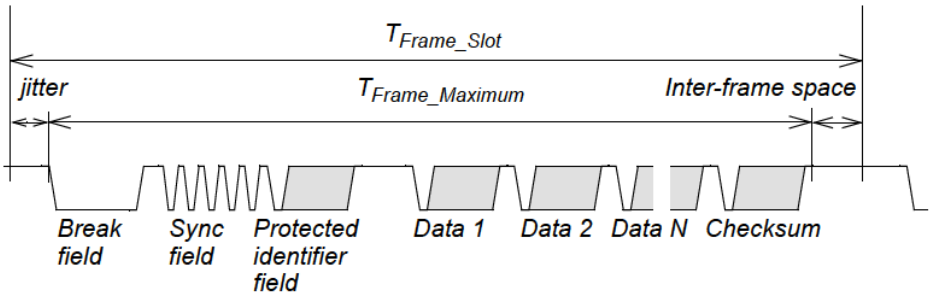


Figure 17-4 LIN Bus Frame Format

LIN bus protocol is based on frame. Each frame is partitioned into several parts as shown above. For master to initiate a frame, the software follows the following procedure.

Initiate a SBK command. (SW needs to check if the bus is in idle state, and there is no pending transmit data).

Write “55” into SBUF2.

Write “PID” into SBUF2.

Wait for SBK to complete interrupts and then write the following transmit data if applicable. (This is optional).

The following diagram shows Finite State Machine (FSM) of the LIN extension and is followed by registers within EUART2.

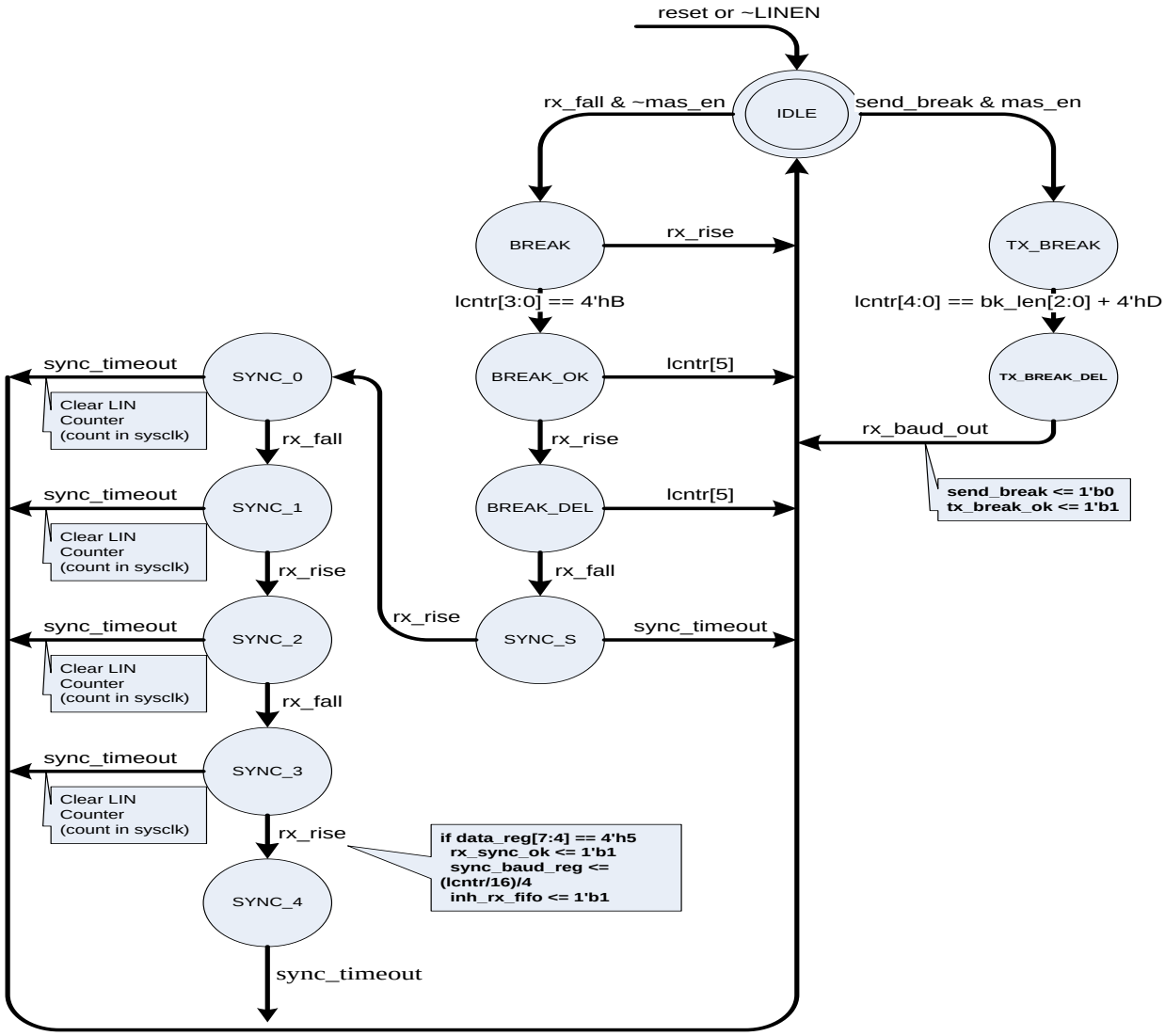


Figure 17-5 Diagram for Finite State Machine (FSM) of the LIN extension

LCTRL2 (0x4000_7214) EUART2 LIN Status/Control Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LINEN	MASEN	ASU	MASU	SBK	BL[2-0]		
WR	LINEN	MASEN	ASU	MASU	SBK	BL[2-0]		

LINEN — LIN Enable (1: Enable / 0: Disable)
LIN header detection / transmission is functional when LINEN = 1.
※ Before enabling LIN functions, the EUART2 registers must be set correctly：0xB0 is recommended for SCON2.

MASEN — Master Enable bit (1: Master / 0: Slave) LIN operating mode selection. This bit is changeable only when LINEN = 0 (must clear LINEN before changing MASEN).

ASU — Auto-Sync Update Enable (1: Enable / 0: Disable), Write Only
If ASU is 1, the LIN controller will automatically overwrite BR[15-0] with SBR[15-0] and issue an ASUI interrupt when received a valid SYNC field.
If ASU is 0, the LIN controller will only notice the synchronized baud rate in SBR[15-0] by issuing an RSI interrupt.
Please note, ASU should not be set under UART mode. ASU capability is based on the message containing BREAK and SYNC field in the beginning.
When ASU=1, the auto sync update is performed on every receiving frame and is updated frame by frame.

MASU — Message Auto Sync Update Enable.

MASU is meaningful only if ASU=0. MASU=1 will enable the auto sync update on the next received frame only. It is self-cleared when the sync update is completed. The software must set MASU again if another auto sync operation is desired.

SBK — Send Break (1: Send / 0: No send request)
LINEN and MASEN should be set before setting SBK. When LINEN and MASEN are both 1, set SBK to send a bit sequence of 13+BL[2-0] consecutive dominant bits and 1 recessive bit (Break Delimiter). Once SBK is set, this bit represents the "Send Break" status and CANNOT be cleared by writing to "0"; instead, clearing LINEN cancels the "Send Break" action. In normal cases, SBK is cleared automatically when the transmission of Break Delimiter is completed.

BL[2-0] — Break Length Setting
Break Length = 13 + BL[2-0]. Default BL[2-0] is 3'b000.

LINCFG2 (0x4000_7215) EUART2 LIN Configure Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	SYNCLRX
WR	-	-	-	-	-	-	-	SYNCLRX

SYNCLRX — Sync Auto Clear RX FIFO
If SYNCLRX is set, the RX FIFO will be cleared automatically while LIN Slave controller receives a valid Sync byte following a Break (RSI=1). This function is valid only when SYNCMD is set under LIN Slave operating mode.

LINTMR2 (0x4000_7218) LIN Timer Register High R/W (0x3FFFF)

	31-18	17-0
RD	-	LCNTR[17-0]
WR	-	LINTMR[17-0]

LCNTR[17-0] is read only and is an internal 16-bit counter clocked by the baud rate clock. LINTMR[17-0] is write only and is the timer limit for LCNTR[17-0]. If MASEN=1 as LIN master mode, this timer is used to generate Frame time base. The internal counter LCNTR[17-0] is cleared whenever a "SEND BREAK" command is executed, and when the counter reaches LINTMR [17-0] (LCNTR[17-0] >= LINTMR[17-0]), a LCNTRO interrupt is generated. Thus the software can write a Frame Time value into LINTMR and use interrupts to initiate frames. If MASEN=0 as LIN slave mode, this timer is used for determining the accumulated bus idle time. The internal counter is cleared whenever a RX transition occurs. When the internal counter reaches LINTMR[17-0], an LCNTRO interrupt is generated. The software can use this interrupt to enter sleep mode by writing the required bus idling time into LINTMR[17-0].

LININTF2 (0x4000_721C) EUART2 LIN Interrupt Flag Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RXSTAT	BITERR	LSTAT	LIDLE	ASUI	SBKI	RSI	LCNTRO
WR	-	BITERR			ASUI	SBKI	RSI	LCNTRO

RXSTAT — Receive status
RXSTAT = 1 indicates that the LIN controller is in receive state.

BITERR — Bit Error
BITERR is set by hardware when received bit does not match with transmit bit , If BERIE=1, then this error generates an interrupt. BITERR must be cleared by software with write "1".

LSTAT — LIN Bus Status bit (1: Recessive / 0: Dominant), Read only.
LSTAT = 1 indicates that the LIN bus (RX pin) is in recessive state.
LSTAT = 0 indicates that the LIN bus (RX pin) is in dominant state.

LIDLE — LIDLE is 1 when LIN bus is idle and not transmitting/receiving LIN header or data bytes. This bit read only. It is 1 when LINEN = 0.

ASUI — Receive valid Sync Interrupt bit (1: Set / 0: Clear)
This flag is set when a valid Sync byte is received following a Break. It must be cleared by writing "1" in the bit.

SBKI — If MASEN=1, SBKI is Send Break Completion Interrupt bit (1: Set / 0: Clear)
This flag is set when Send Break completes. It must be cleared by writing "1" in the bit.

If MASEN=0, SBKI is Receive Break Completion Interrupt bit
This flag is set when a Break condition is detected and completed by a rising edge of bus signal. It must be cleared by writing “1” in the bit.

RSI — RSI Receive 4bit Sync Interrupt bit (1: Set / 0: Clear)
This flag is set when a half valid Sync byte is received following a Break. It must be cleared by writing “1” in the bit.

LCNTRO — LIN Counter Overflow Interrupt bit (1: Set / 0: Clear).
This flag is set when the LIN counter reaches LINTMR [17-0] (LCNTR[17-0] >= LINTMR[17-0]). It must be cleared by writing “1” in the bit.

LININTE2 (0x4000_7220) EUART2 LIN Interrupt Enable Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LINTEN	BERIE	SYNCMD	-	ASUIE	SBKIE	RSIE	LCNTRIE
WR	LINTEN	BERIE	SYNCMD	-	ASUIE	SBKIE	RSIE	LCNTRIE

LINTEN — LIN Interrupt Enable (1: Enable / 0: Disable)
Set to enable all LIN interrupts. LINT flags should be checked before setting or modifying.

BERIE — Bit Error Interrupt Enable (1: Enable / 0: Disable)

SYNCMD — SYNCMD=0 will only allow automatic synchronization of baud rate within +/- 6% deviations from current baud rate setting. Larger than +/- 6% deviation may cause error of reception.
SYNCMD=1 will automatically synchronize and update SBR with newly acquired baud rate (if ASU/MASU = 1). SYNCMD should be set to 1 when either ASU or MASU is 1. Although under this setting, the tolerant range of deviation can be up to +/- 50%, it is recommended to set the BR[15-0] as close as target baud rate.
The new baud rate can be successfully synchronized and frame received correctly must meet the following conditions at the same time.
1. Within +/- 50% of the current baud rate setting.
2. The incoming Break Length satisfies following two conditions at the same time
   A. Break length is less than 32 current baud rate bit times
   B. Break length is less than 253952 system clock
3. For the application with multi-baud rates, software should set the BR[15-0] using the lowest baud rate. Since after each LIN transaction, BR[15-0] is automatically updated with newly synchronized value, software needs to reset BR[15-0] to the lowest baud rate again if new baud rate is used.

ASUIE — Enable receive valid SYNC Interrupt (1: Enable / 0: Disable)

SBKIE — If MASEN=1, SBKIE is Send Break Completion Interrupt Enable (1: Enable / 0: Disable)
If MASEN=0, SBKIE is Receive Break Completion Interrupt Enable (1: Enable / 0: Disable)

RSIE — Receive Sync Completion Interrupt Enable (1: Enable / 0: Disable)

LCNTRIE — LIN Counter Overflow Interrupt Enable (1: Enable / 0: Disable)

LININTE2B (0x4000_7221) EUART2 LIN Interrupt Enable Register R/W (0x00)

	15	14	13	12	11	10	9	8
RD	TLOOPEN	BETXCLR	BERXCLR	-	-	-	RXPOL	TXPOL
WR	TLOOPEN	BETXCLR	BERXCLR	CLRTX	CLRRX	CLRFIFO	RXPOL	TXPOL

TLOOPEN — Set to enable loopback mode by connecting TX output to internal RX input. To prevent the TX output to pin, corresponding MFCFG bit must be cleared

BETXCLR — Clear TX FIFO and TX state while bit error occurs

BERXCLR — Clear RX FIFO and RX state while bit error occurs

CLRTX — Clear TX FIFO level and TX state

CLRRX — Clear RX FIFO level and RX state

CLRFIFO — Clear FIFO level and clears flags and LIN state machine
Set to clear transmit/received FIFO buffer and CLRFIFO is auto clear by hardware. In addition, this will also reset the LIN state machine, i.e., also clears RFO, RFU and TFO interrupt flags without writing the interrupt register. The LIN counter LCNTR is also cleared.

RXPOL EUART/LIN input polarity
TXPOL EUART/LIN output polarity

LINTCON2 (0x4000_7224) EUART2 LIN Time Out Configuration R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	LRXDFEN	-	-	RXDD_F	TXDD_F	RXDDEN	TXDDEN
WR	-	LRXDFEN	-	-	RXDD_F	TXDD_F	RXDDEN	TXDDEN

LRXDFEN Set to enable LIN break check condition added received dominant fault. LIN break check fail if RXD Dominant Fault occurs.
RXDD_F RXD Dominant Fault Interrupt Flag
RXDD_F is set to 1 by hardware and must be cleared by software with write “1”
TXDD_F TXD Dominant Fault Interrupt Flag
TXDD_F is set to 1 by hardware and must be cleared by software with write “1”
RXDDEN RXD Dominant Fault Interrupt Enable
TXDDEN TXD Dominant Fault Interrupt Enable

TXDTO2 (0x4000_7228) EUART2 LIN TXD Dominant Time Out Registers R/W (0x0000)

	31-16	15-0
RD	-	TXDTO[15-0]
WR	-	TXDTO[15-0]

TXDTO TXD Dominant Time Out (TXDTO +1) * SYSCLK*(BRCS[7-0]+1). The value of TXDTO[15-0] shall be greater than zero.

RXDTO2 (0x4000_722C) EUART2 LIN RXD Dominant Time Out Registers R/W (0x0000)

	31-16	15-0
RD	-	RXDTO[15-0]
WR	-	RXDTO[15-0]

RXDTO RXD Dominant Time Out (2*RXDTO[15-0] + 1) * SYSCLK*(BRCS[7-0]+1). The value of RXDTO[15-0] shall be greater than zero.

BSDCLR2 (0x4000_7230) EUART2 Bus Stuck Dominant Clear Width Registers R/W (0x0000)

	31-16	7-0
RD	-	BSDCLR[7-0]
WR	-	BSDCLR[7-0]

BSDCLR Bus Stuck Dominant Clear Time
0: 1 SOSC128K to 2 SOSC128K
1: 1 SOSC128K to 2 SOSC128K
2: 2 SOSC128K to 3 SOSC128K
N: N SOSC128K to N+1 SOSC128K, N=3 to 255.
BSDCLR detects the recessive state duration after a dominant to recessive transition. BSDCLR is used in conjunction with BSDACT setting.

BSDACT2 (0x4000_7234) EUART2 Bus Stuck Dominant Active Width Registers R/W (0x0000)

	7-0
RD	BSDACT[7-0]
WR	BSDACT[7-0]

BSDACT Bus Stuck Dominant Active Time
0: 1 SOSC128K to 2 SOSC128K
1: 1 SOSC128K to 2 SOSC128K
2: 2 SOSC128K to 3 SOSC128K
N: N SOSC128K to N+1 SOSC128K, N=3 to 255.
BSDACT detects the dominant state duration after a recessive state meeting BSDCLR transition to dominant state.
If BSDCLR and BSDACT timings are both met, a BSDWEN interrupt is generated.

Purpose of BSDCLR and BSDACT is to prevent false wakeup if sleep mode entry occurs during system fault with bus stuck in dominant state where transceiver is integrated with LIN controller to control the supply voltage. In a typical application, it should not be used and BSDWEN interrupt should be turned off.

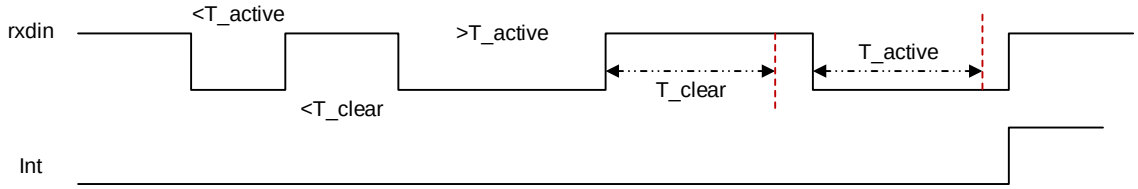


Figure 17-6 Timing for BSDCLR and BSDACT

WKFLT2 (0x4000_7235) EUART2 Wakeup Time Registers R/W (0x0000)

	7-0
RD	WKFLT[7-0]
WR	WKFLT[7-0]

WKFLT — LIN Wakeup Time
0: 1 SOSC128K to 2 SOSC128K
1: 1 SOSC128K to 2 SOSC128K
2: 2 SOSC128K to 3 SOSC128K
N: N SOSC128K to N+1 SOSC128K, N=3 to 255.

BSDWKC2 (0x4000_7238) EUART2 Bus Stuck Wakeup Configuration R/W (0x00)

	7	6	5	4	3	2	1	0
RD	BSDW_F	BFW_F	BSDWEN	BFWEN	-			
WR	BSDW_F	BFW_F	BSDWEN	BFWEN	-			

BSDW_F — LIN Bus Stuck Wake Up Interrupt Flag
BSDW_F is set to 1 by hardware and must be cleared by software with write “1”

BFW_F — LIN Wake Up Interrupt Flag
BFW_F is set to 1 by hardware and must be cleared by software with write “1”

BSDWEN — LIN Bus Stuck Wake Up/ Interrupt Enable
BSDWEN controls the BSDCLR/BSDACT interrupt. In typical applications, it should be turned off.

BFWEN — LIN Wake Up/Interrupt Enable. If Bus dominant time > (WKFLT[7-0] + 1) * SOSC128K, the MCU will wake up and generate an interrupt.

Please note for auto synchronization of baud rate in slave configuration, ASU and SYNCMD should be set to 1. In addition, SBR[15-0] should be initialized to 0xFFFF (as the slowest baud rate). If SBR[15-0] is not initialized properly, the incoming break field may cause LNTRO timeout and interrupt the baud rate detection logic state, or lead to erroneous extra data bytes received.

18. Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) is an enhanced synchronous serial hardware which is compatible with Motorola's SPI specifications. The SPI Controller includes 16-byte FIFO for both transmit and receive. SPI Interface uses Master-Out-Slave-In (MOSI), Master-In-Slave-Out (MISO), Serial Clock (SCK) and Slave Select (SSN) for interface. SSN is low active, and SSN can also be configured as latch output in master mode to serve in video bus applications. The maximum SPI clock rate is SYSCLK/4.

SPICRA (0x4000_8000) SPI Configuration Register R/W (0x20)

	7	6	5	4	3	2	1	0
RD	SPEN	-	MSTR	CPOL	CPHA	SCKE	SSNLAT	-
WR	SPEN	-	MSTR	CPOL	CPHA	SCKE	SSNLAT	-

SPEN — SPI interface Enable bit.

MSTR — SPI Master/Slave Switch.
MSTR=1: Master.
MSTR=0: Slave.

CPOL — SPI interface Polarity bit: Set to configure the SCK to stay HIGH while the SPI interface is idling and clear to keep it LOW.

CPHA — Clock Phase Control bit: If CPOL=0, set to shift output data at rising edge of SCK, and clear to shift output data at falling edge of SCK. If CPOL=1, set to shift output data at falling edge of SCK and clear to shift output data at rising edge of SCK.

SCKE — Clock Selection bit in Master Mode
Set to delay 0.5 period of SCK to sample the input data.
Clear to use normal edge of SCK to sample the input data.
The sampling phase is determined by the combinations of CPOL and CPHA setting shown in the following table.

CPOL	CPHA	DATAIN Edge			DATAOUT Edge
		Slave	Master, SCKE=0	Master, SCKE=1	
0	0	Rising edge	Rising edge	Falling edge	Falling edge
0	1	Falling edge	Falling edge	Rising edge	Rising edge
1	0	Falling edge	Falling edge	Rising edge	Rising edge
1	1	Rising edge	Rising edge	Falling edge	Falling edge

SSNLAT — Use SSN as LATCH Signal.
SSNLAT=0, configure SSN as normal SPI interface.
SSNLAT=1, configure SSN as LATCH signal. The LATCH start position and width are defined in SLTCHSP and SLTCHWD registers.
Note the Latch function is only available while CPOL = 0 and CPHA = 0.

SPICRB (0x4000_8001) SPI Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	FCLR	SIFLTEN	CSFLTEN	CKFLTEN	SPR[2]	SPR[1]	SPR[0]	DIR
WR	FCLR	SIFLTEN	CSFLTEN	CKFLTEN	SPR[2]	SPR[1]	SPR[0]	DIR

FCLR — FIFO Clear/Reset
Set to clear and reset transmit and receive FIFO, auto clear by hardware

SIFLTEN — Enable SDI digital filter

CSFLTEN — Enable CS digital filter

CKFLTEN — Enable SCK digital filter

SPR[2-0] — Slave SPI Clock Rate up to SYSCLK/4,
Master SPI Clock Rate Setting.
000 – SCK = SYSCLK/4;
001 – SCK = SYSCLK/6;
010 – SCK = SYSCLK/8;
011 – SCK = SYSCLK/16;
100 – SCK = SYSCLK/32;

101 – SCK = SYSCLK/64;
110 – SCK = SYSCLK/128;
111 – SCK = SYSCLK/256.

DIR — Transfer Format
DIR=1 uses MSB-first format.
DIR=0 uses LSB-first format.

SPIMRA (0x4000_8004) SPI Access Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	LASTACC	-	-	-	SPISTART
WR	-	-	-	LASTACC	-	-	-	SPISTART

LASTACC — Last CPU Access
For Master only, CPU write 1 to this to indicate the last access to the XMT FIFO has completed. Controller will terminate the SPI transaction after emptying XMT FIFO. This bit is cleared by hardware when the transaction is completed. When read, this bit also reflect the XMT is in progress.

SPISTART — SPI Start
For Master only, CPU write 1 to this bit to start transaction.

SPIMRB (0x4000_8005) SPI XMT FIFO Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-			XFIFO[4-0]				
WR	-			XFIFOTH[4-0]				

XFIFO[4-0] — XMIT FIFO Current Value
XFIFO[3-0] indicates the current XMT FIFO occupied location in bytes.

XFIFOTH[4-0] — XMT FIFO Interrupt Threshold
The bytes count in XMT FIFO for generating SPI interrupt.
The XFIFO should be reset by set FCLR of SPICRB if a transfer is complete while SPI in slave mode

SPIMRC (0x4000_8006) SPI RCV FIFO Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-			RFIFO[4-0]				
WR	-			RFIFOTH[4-0]				

RFIFO[4-0] — RMIT FIFO Current Value
RFIFO[4-0] indicates the current RCV FIFO occupied location in bytes.

RFIFOTH[4-0] — RCV FIFO Interrupt Threshold

SPIMRD (0x4000_8007) SPI FIFO Status Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	RFULL	REMPT	TFULL	TEMPT
WR	-	-	-	-	-	-	-	-

RFULL — Receive FIFO Full Status bit. Set when receiver FIFO is full. Read only.
REMPT — Receive FIFO Empty Status bit. Set when receiver FIFO is empty. Read only.
TFULL — Transmitter FIFO Full Status bit. Set when transfer FIFO is full. Read only.
TEMPT — Transmitter FIF0 Empty Status bit. Set when transfer FIFO is empty. Read only.

SPISTA (0x4000_8008) SPI Interrupt Enable Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ROVREN	RUDREN	TOVREN	TUDREN	FBERREN	EOFEN	RFIFOEN	XFIFOEN
WR	ROVREN	RUDREN	TOVREN	TUDREN	FBERREN	EOFEN	RFIFOEN	XFIFOEN

ROVREN — Enable Receive FIFO over run interrupt
RUDREN — Enable Receive FIFO under run interrupt
TOVREN — Enable Transmit FIFO over run interrupt
TUDREN — Enable Transmit FIFO under run interrupt

FBERREN	Enable Frame bit error interrupt
EOFEN	Enable End of Frame interrupt
RFIFOEN	Enable Receive FIFO threshold interrupt
XFIFOEN	Enable Transmit FIFO threshold interrupt

SPISTB (0x4000_8009) SPI Interrupt Flag Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ROVR	RUDR	TOVR	TUDR	FBERR	EOFIF	RFIFO	XFIFO
WR	ROVR	RUDR	TOVR	TUDR	FBERR	EOFIF	RFIFO	XFIFO

ROVR	Receive FIFO-overrun Error Flag bit. When Receiver FIFO Full Status occurs and SPI receives new data, ROVR is set and generates an interrupt. Clear by assigning this bit to 1 or disabling SPI.
RUDR	Receive FIFO-underrun Error Flag bit. When Receive FIFO Empty status occurs and reading of RCV FIFO, RUDR is set and generates an interrupt. Clear by assigning this bit to 1 or disabling SPI.
TOVR	Transmit FIFO-overrun Error Flag bit. When Transfers FIFO Full Status occurs and new data is written, TOVR is set and generates an interrupt. Clear by assigning this bit to 1 or disabling SPI.
TUDR	Transmit Under-run Error Flag bit. When Transfers FIFO Empty Status and new data transmission occur, TUDR is set and generates an interrupt. Clear by written 1 to this bit or disable SPI.
FBERR	Frame Bit Error FBERR is set by hardware when transaction does not end with byte boundaries. Clear by assigning this bit to 1.
EOFIF	SPI End of Frame Interrupt Flag bit. Set by hardware when complete data transfer and generates an interrupt if EOFOE =1. Clear by assigning this bit to 1.
RFIFO	RFIFO Threshold Interrupt Set by hardware when RCV FIFO reaches threshold. Clear by assigning this bit to 1.
XFIFO	XFIFO Threshold Interrupt Set by hardware when XMT FIFO reaches threshold. Clear by assigning this bit to 1.

SPIDATA (0x4000_800C) SPI Data Register R/W (0xXX)

	7	6	5	4	3	2	1	0
RD	SPIDATA[7-0]							
WR	SPIDATA[7-0]							

SPIDATA is byte wide access. Other access types such as word, or double word are invalid and returns undefined data.

SLTCHSP0 (0x4000_8010) SSN Latch Start Position Register 0 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SLTCHSP[7-0]							
WR	SLTCHSP[7-0]							

SLTCHSP1 (0x4000_8011) SSN Latch Start Position Register 1 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SLTCHSP[15-8]							
WR	SLTCHSP[15-8]							

SLTCHSP[15-0]	SSN Latch Signal Start Position. This defines the start location in SPI bit time.

SLTCHWD (0x4000_8012) SSN Latch Width Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SLTCHWD[7-0]							
WR	SLTCHWD[7-0]							

SLTCHWD[7-0] SSN Latch Signal Width. This defines the Latch width in SPI bit time.

18.1 SPI Master Timing Illustration

18.1.1 CPOL=0 CPHA=0

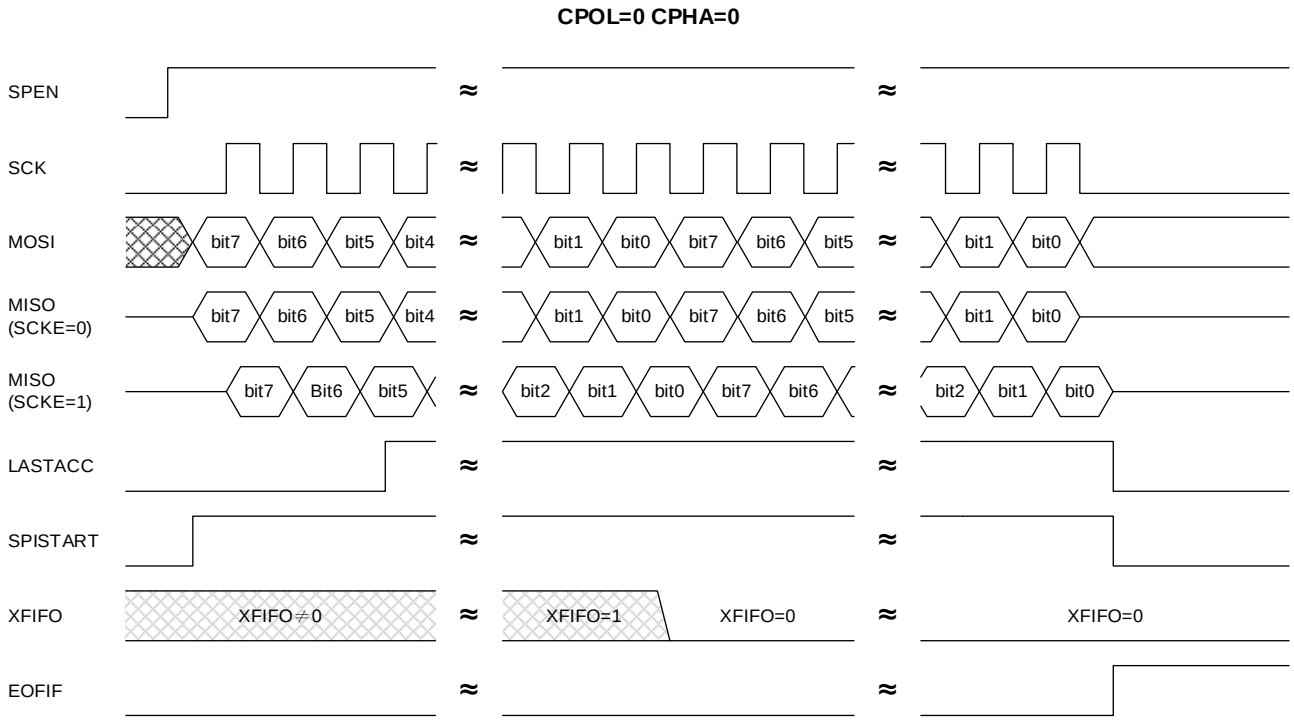


Figure 18-1 SPI Master Timing with CPOL=0, CPHA=0

18.1.2 CPOL=0 CPHA=1

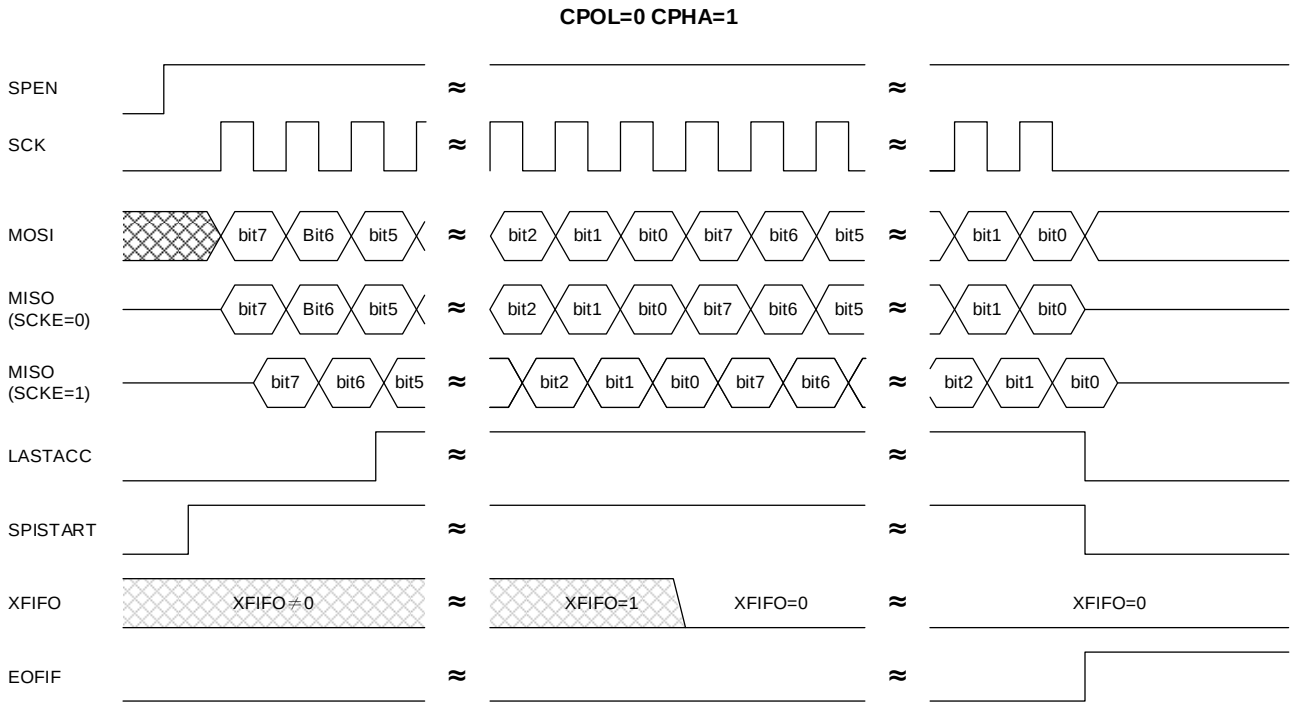


Figure 18-2 SPI Master Timing with CPOL=0, CPHA=1

18.1.3CPOL=1 CPHA=0

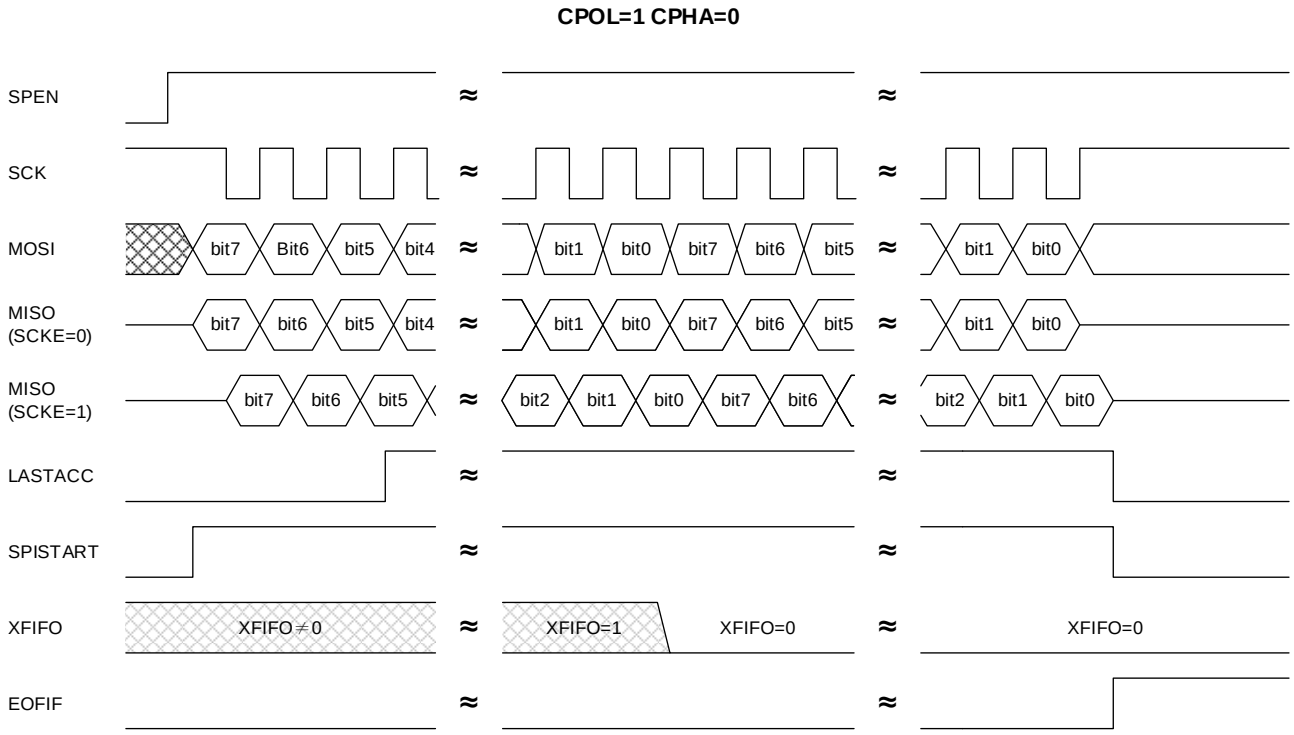


Figure 18-3 SPI Master Timing with CPOL=1, CPHA=0

18.1.4CPOL=1 CPHA=1

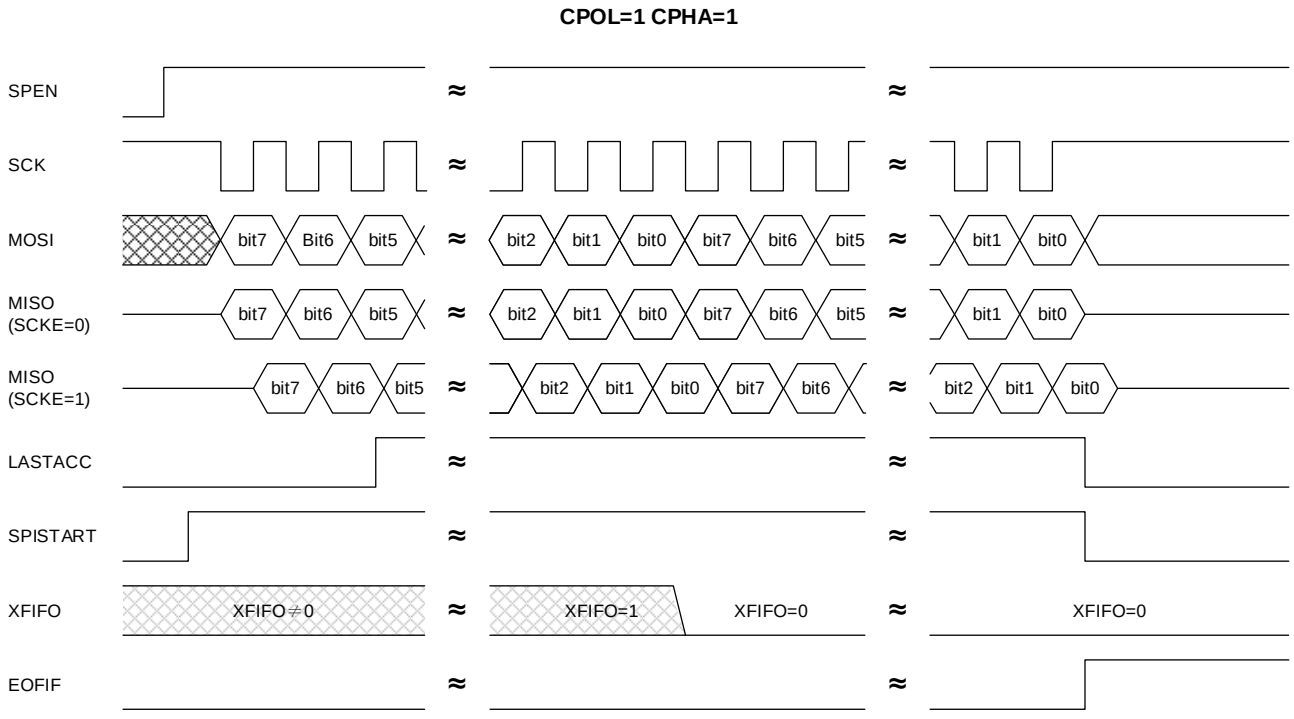


Figure 18-4 SPI Master Timing with CPOL=1, CPHA=1

18.2 SPI Slave Timing Illustration

18.2.1CPOL=0 CPHA=0

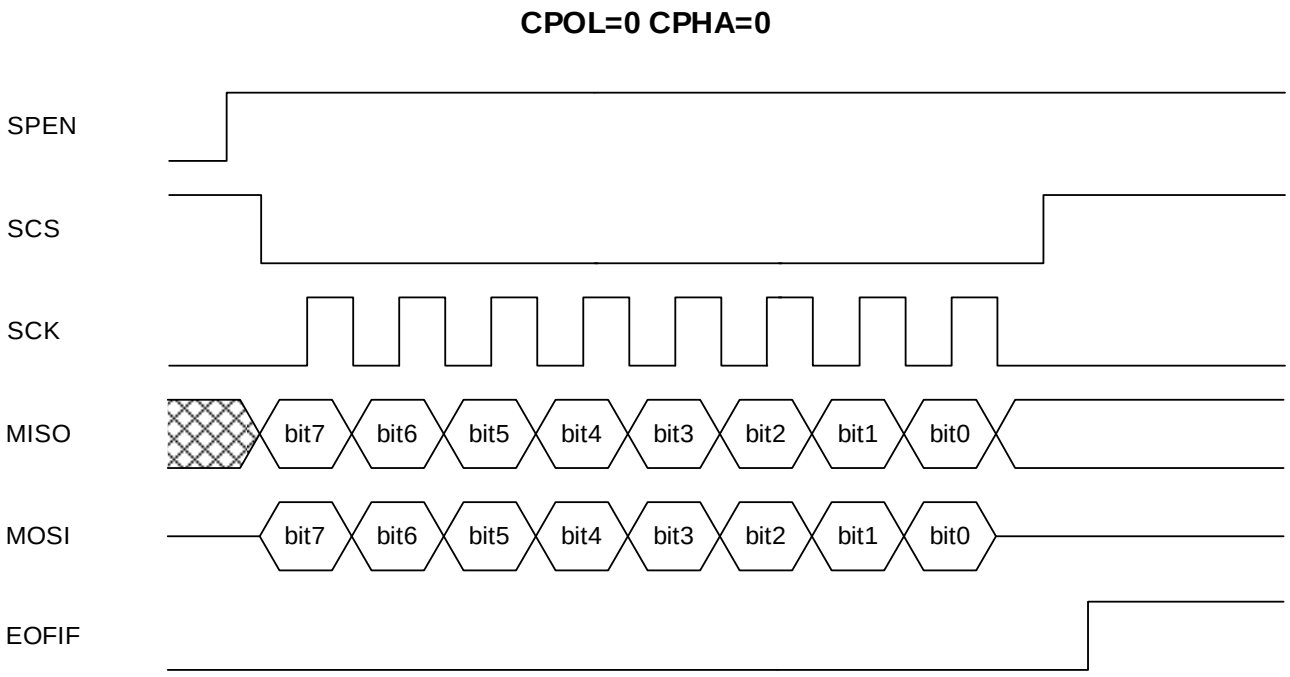


Figure 18-5 SPI Slave Timing with CPOL=0, CPHA=0

18.2.2CPOL=0 CPHA=1

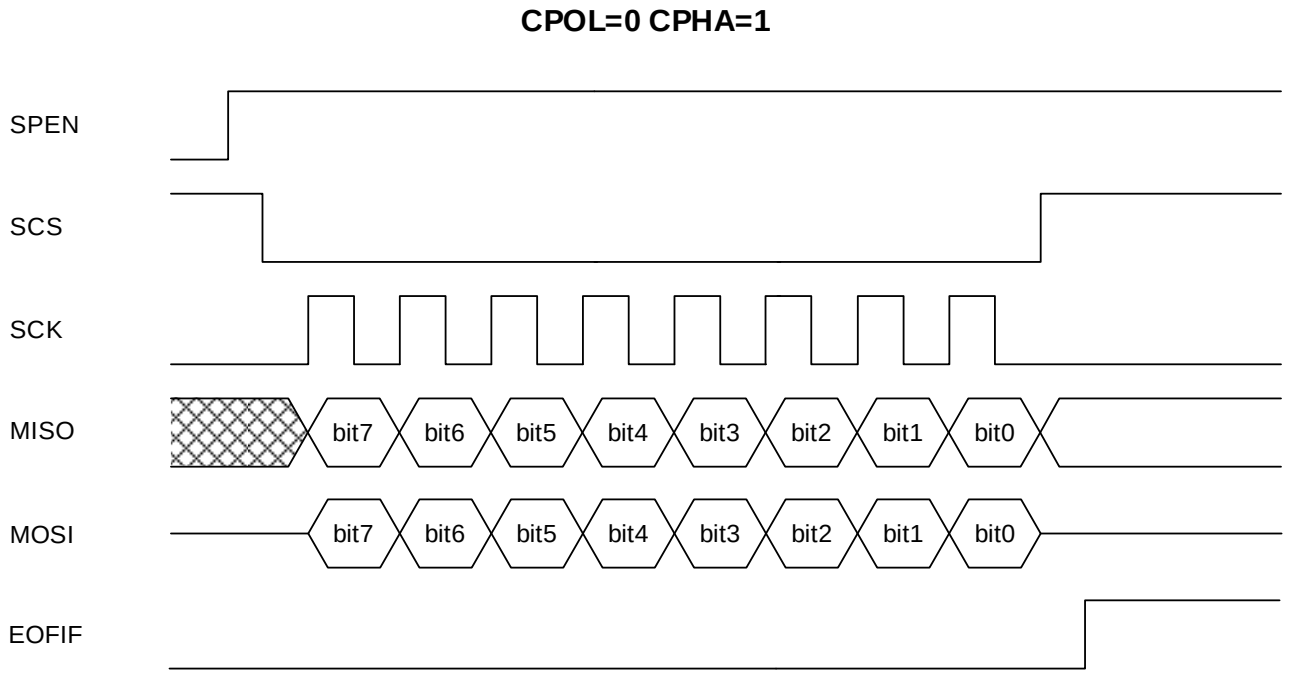


Figure 18-6 SPI Slave Timing with CPOL=0, CPHA=1

18.2.3CPOL=1 CPHA=0

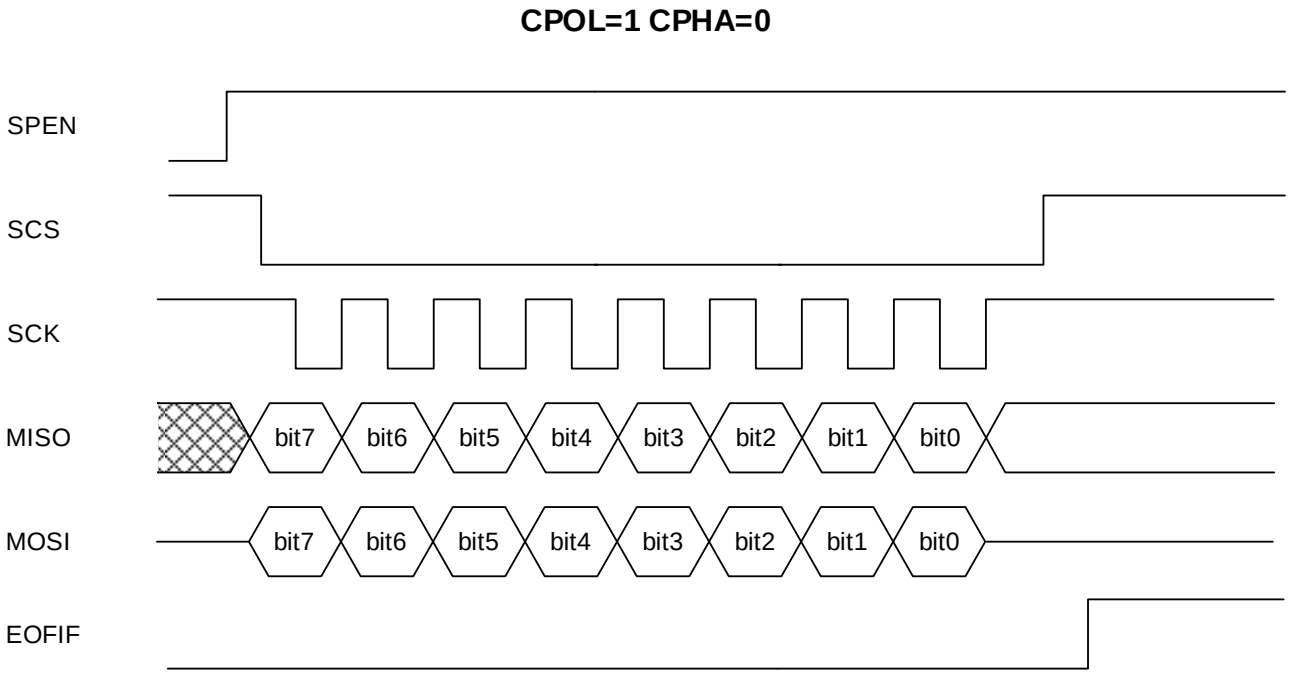


Figure 18-7 SPI Slave Timing with CPOL=1, CPHA=0

18.2.4CPOL=1 CPHA=1

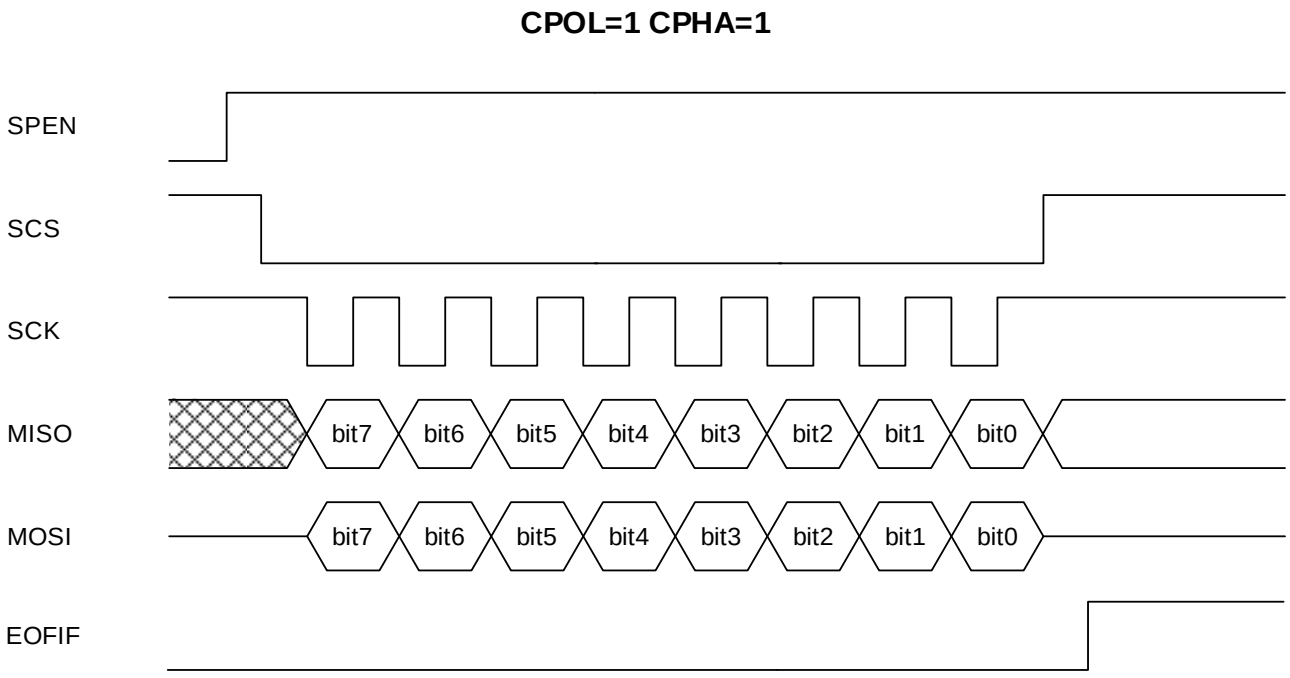


Figure 18-8 SPI Slave Timing with CPOL=1, CPHA=1

18.2.5SSN as LATCH (Master only)

Only available for mode 0 (CPOL and CPHA both are 0)

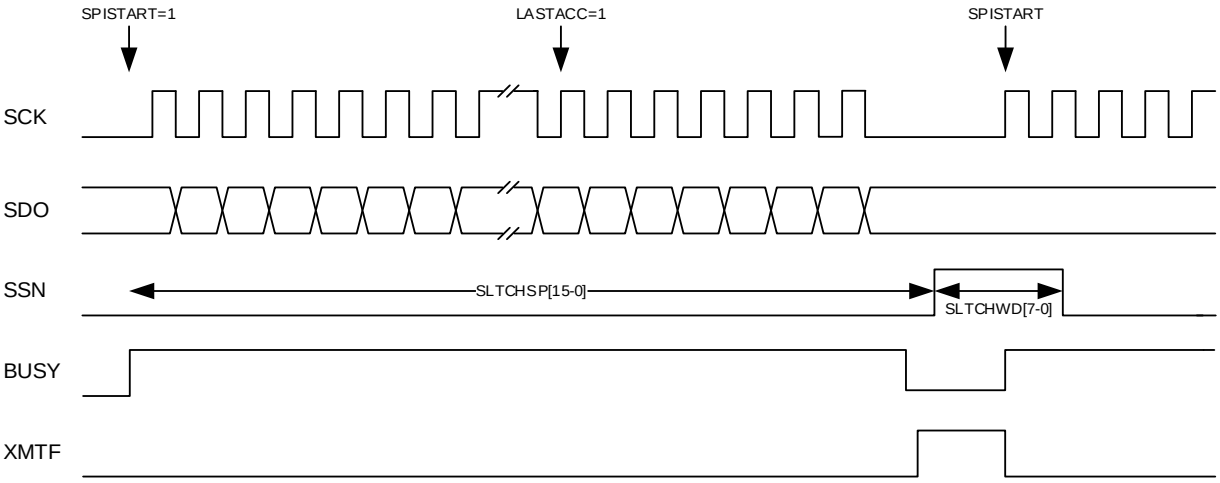


Figure 18-9 SSN as Latch signal

19. Timer with Compare/Capture TCC1/TCC2

The Timer/Capture unit is based on a 16-bit counter with pre-scalable SYSCLK as counting clock. The count starts from 0 and reloads when it reaches TC (terminal count). TC is met when the count equals the period value. During counting, the count value is compared with COMP and when it matches, a CC condition is met. Note that both PERIOD and COMP register are double buffered, therefore any new value is updated after the current period ends. TC and CC can be used for triggering an interrupt and also routed to GPIO. The output pulse width of TC and CC is programmable. For CC, it can also be configured as a PWM output. There are two data registers for capture events. The capture event can be from external signals from GPIO with edge selection option, or from QE block, or triggered by software. The software can also select to reset the counter or not, this option give simpler calculation of consecutive capture events without an offset. The following Block Diagram shows the TCC implementations.

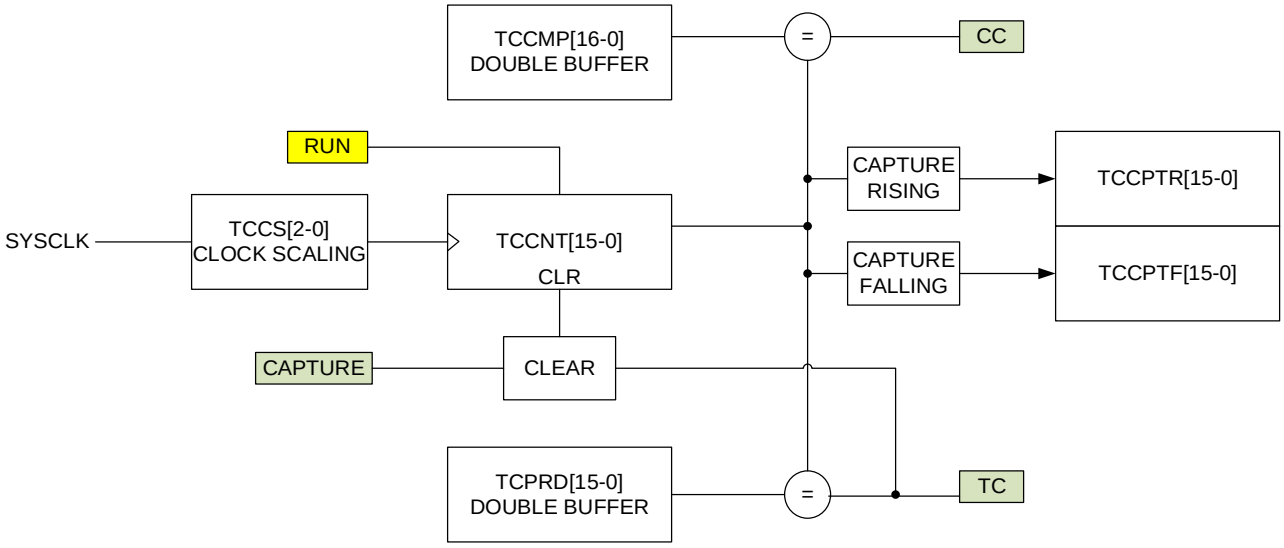


Figure 19-1 TCC Block Diagram

TC1CFG1A (0x4000_9000) TC1 Configuration Register 1A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TCEN	TCCS[2-0]			CCSEL[1-0]		TCSEL	RUNST
WR	TCEN	TCCS[2-0]			CCSEL[1-0]		TCSEL	RUN

TCEN — TC Enable

TC = 0 disables TC. In disabled state, TCCNT, and TCCPTR/TCCPTF are cleared to 0. TC and CC are also set to low.

TC = 1 enable TC. RUN bit also needs to set to 1 to start the counter, otherwise if RUN=0 then counter is in pause mode.

Please note setting of TCCS, CCSEL, TCSEL should be configured or modified when TCEN=0 to avoid undefined behavior.

TCCS[2-0] — TC Clock Scaling

000 SYSCLK
001 SYSCLK/2
010 SYSCLK/4
011 SYSCLK/8
100 SYSCLK/16
101 SYSCLK/32
110 SYSCLK/64
111 SYSCLK/128

CCSEL[1-0] — CC Output Pulse Select

00 PW = 16 TCCLK
01 PW = 64 TCCLK
10 PWM Waveform (CC = low when TCCNT < CMP, CC = high when TCCNT >= CMP).
11 PWM Toggle waveform (CC toggles when TCCNT = CMP).

TCSEL — TC Output Pulse Select

0 PW = 16 TCCLK

1 PW = 64 TCCLK

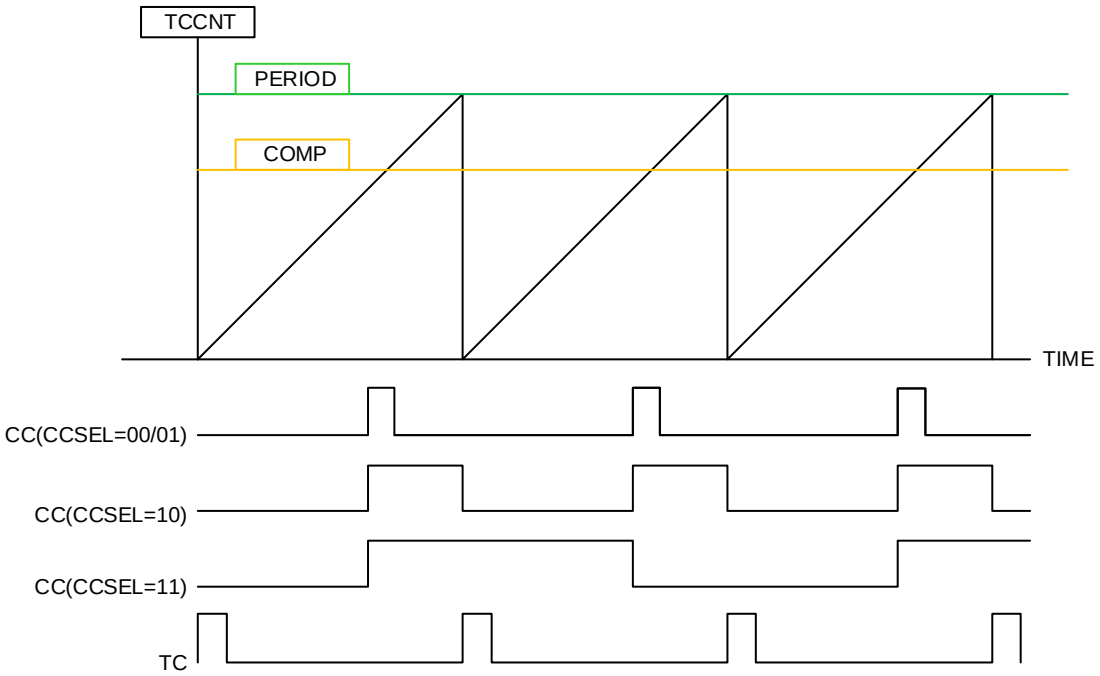


Figure 19-2 TCC timing diagram

RUNST Run Status
Set by hardware to indicate running TC counter. RUNST=1 indicates running.

RUN Run or Pause TC Counter
Writing “0” to RUN will pause the TC counting.
Writing “1” to RUN will resume the TC counting.

TC1CFG2A (0x4000_9004) TC1 Configuration Register 2A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	IDXST	PHAST	PHBST	XRCF	XFCF	TCF	CCF
WR	RSTTC	-	-	-	XRCF	XFCF	TCF	CCF

RSTTC Reset TC
Writing RSTTC “1” will reset the TC counter and capture registers. Once counter is cleared, TC counter is put in STOP mode. To resume counting, RUN bit must be set by software.

IDXST Index Input real-time status

PHAST PHA input real-time status

PHBST PHB input real-time status

XRCF External Rising Edge Capture Flag
XRCF is set to “1” by hardware when an external rising edge capture event has occurred. XRCF must be cleared by software by writing “1”. The captured value is stored in CPTR register.

XFCF External Falling Edge Capture Flag
XFCF is set to “1” by hardware when an external falling edge capture event has occurred. XFCF must be cleared by software by writing “1”. The captured value is stored in CPTF register.

TCF Terminal Count Interrupt Flag
TCF is set to “1” by hardware when terminal count occurs. TCF must be cleared by software by writing “1”.

CCF Compare Match Interrupt Flag
CCF is set to “1” by hardware when compare match occurs. CCF must be cleared by software by writing “1”.

TC1CFG2B (0x4000_9005) TC1 Configuration Register 2B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SYNCTC2	SYNCPWM	-	-	-	-	-	-
WR	SYNCTC2	SYNCPWM	-	-	-	-	SWCPTR	SWCPTF

SYNCTC2 Synchronize with TC2
Synchronize the counter start with TC2 counter.
This bit is set by software and cleared by hardware. Once this bit is set, the counter restart counting when TC2 count reaches restart point.
To maintain synchronization, TC1 and TC2's clock scaling and period setting must be the same.

SYNCPWM Synchronize with PWM
Synchronize the counter start with PWM counter.
This bit is set by software and cleared by hardware. Once this bit is set, the counter restart counting when PWM count reaches restart point.
To maintain synchronization, TC1 and PWM clock scaling must be the same. Due to PWM counting is center aligned, the period setting of TC1 must be 2X of PWM period setting.

SWCPTR Software Capture R
Writing "1" to SWCPTR will generate a capture event and capture the count value into CPTR register. This bit is cleared by hardware.

SWCPTF Software Capture F
Writing "1" to SWCPTF will generate a capture event and capture the count value into CPTF register. This bit is cleared by hardware

TC1CFG3A (0x4000_9008) TC1 Configuration Register 3A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	IENTC	IENCC	QECEN	CPTCLR	XRCEN	XFCEN	TCPOL	CCPOL
WR	IENTC	IENCC	QECEN	CPTCLR	XRCEN	XFCEN	TCPOL	CCPOL

IENTC TC Interrupt Enable

IENCC CC Interrupt Enable

QECEN QE Capture Enable
QECEC=1 use QE output event as capture event.

CPTCLR Enable Clear Counter after Capture
If CPTCLR=1, the TCCNT is cleared to 0 after each capture event. This allows continuous capture value with identical initial value.
If CPTCLR=0, the capture event does not affect the TCCNT counting.

XRCEN External Rising Edge Capture Enable
XRCEN=1 use external input rising edge as capture event. The captured value is stored in CPTR register.

XFCEN External Falling Edge Capture Enable
XFCEN=1 use external input falling edge as capture event. The captured value is stored in CPTF register.

TCPOL TC output polarity

CCPOL CC output polarity

Please note all capture sources are not mutually exclusive, i.e. allow several capture sources can coexist.

TC1CFG3B (0x4000_9009) TC1 Configuration Register 3B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	IENXRC	IENXFC
WR	-	-	-	-	-	-	IENXRC	IENXFC

IENXRC XC Rising Edge Interrupt Enable
IENXRC=1 enables interrupt when XRC event occurs.

IENXFC XC Falling Edge Interrupt Enable
IENXFC=1 enables interrupt when XFC event occurs.

TC1PRDL (0x4000_900C) TC1 Period and Count Register LOW Double Buffer R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC1CNT[7-0]							
WR	TC1PRD[7-0]							

TC1CNT[7-0] TC1 Counter value
TC1PRD[7-0] TC1 Period setting

TC1PRDH (0x4000_900D) TC1 Period and Count Register High Double Buffer R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC1CNT[15-8]							
WR	TC1PRD[15-8]							

TC1CNT[15-8] TC1 Counter value
TC1PRD[15-8] TC1 Period setting

TC1CMPL (0x4000_9010) TC1 Compare Register Low Double Buffer R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC1CMP[7-0]							
WR	TC1CMP[7-0]							

TC1CMP[7-0] TC1 Compare setting

TC1CMPH (0x4000_9011) TC1 Compare Register High Double Buffer R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC1CMP[15-8]							
WR	TC1CMP[15-8]							

TC1CMP[15-8] TC1 Compare setting

TC1CPTRL (0x4000_9014) TC1 Rising Edge Capture Register R Low R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC1CPTR[7-0]							
WR	-							

TC1CPTR[7-0] TC1 Rising Edge Capture value

TC1CPTRH (0x4000_9015) TC1 Rising Edge Capture Register R High R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC1CPTR[15-8]							
WR	-							

TC1CPTR[15-8] TC1 Rising Edge Capture value

TC1CPTFL (0x4000_9018) TC1 Falling Edge Capture Register F Low R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC1CPTF[7-0]							
WR	-							

TC1CPTF[7-0] TC1 Falling Edge Capture value

TC1CPTFH (0x4000_9019) TC1 Falling Edge Capture Register F High R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC1CPTF[15-8]							
WR	-							

TC1CPTF[15-8] TC1 Falling Edge Capture value

TC2 is the same as TC1. Its control registers start with 0x4000_9020.

TC2CFG1A (0x4000_9020) TC2 Configuration Register 1A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TCEN	TCCS[2-0]			CCSEL[1-0]		TCSEL	RUNST
WR	TCEN	TCCS[2-0]			CCSEL[1-0]		TCSEL	RUN

TCEN — TC Enable

TC = 0 disables TC. In disabled state, TCCNT, and TCCPTR/TCCPTF are cleared to 0. TC and CC are also set to low.

TC = 1 enable TC. RUN bit also needs to set to 1 to start the counter, otherwise if RUN=0 then counter is in pause mode.

TCCS[2-0] — TC Clock Scaling

000 SYSCLK
001 SYSCLK/2
010 SYSCLK/4
011 SYSCLK/8
100 SYSCLK/16
101 SYSCLK/32
110 SYSCLK/64
111 SYSCLK/128

CCSEL[1-0] — CC Output Pulse Select

00 PW = 16 TCCLK
01 PW = 64 TCCLK
10 PWM Waveform (CC = low when TCCNT < CMP, CC = high when TCCNT >= CMP).
11 PWM Toggle waveform (CC toggles when TCCNT = CMP).

TCSEL — TC Output Pulse Select

0 PW = 16 TCCLK
1 PW = 64 TCCLK

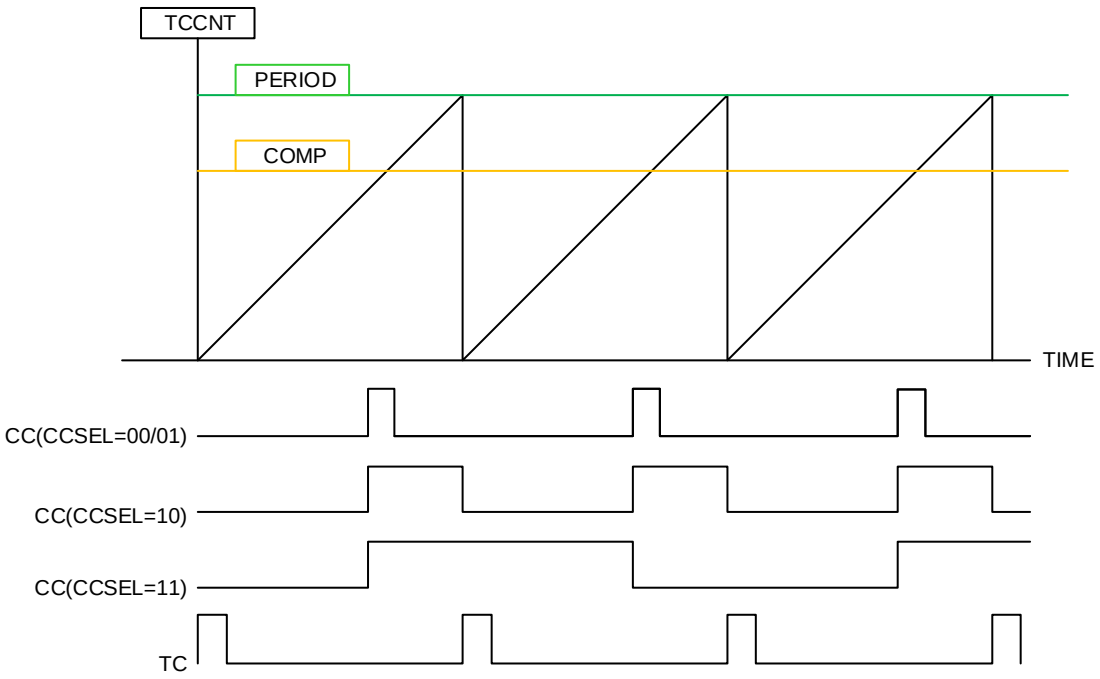


Figure 19-3 TCC timing diagram

RUNST — Run Status

Set by hardware to indicate running TC counter. RUNST=1 indicates running.

RUN — Run or Pause TC Counter

Writing “0” to RUN will pause the TC counting.
Writing “1” to RUN will resume the TC counting.

TC2CFG2A (0x4000_9024) TC2 Configuration Register 2A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	XRCF	XFCF	TCF	CCF
WR	RSTTC	-	-	-	XRCF	XFCF	TCF	CCF

RSTTC — Reset TC
Writing RSTTC “1” will reset the TC counter and capture registers. Once counter is cleared, TC counter is put in STOP mode. To resume counting, RUN bit must be set by software.

XRCF — External Rising Edge Capture Flag
XRCF is set to “1” by hardware when an external rising edge capture event has occurred. XRCF must be cleared by software by writing “1”. The captured value is stored in CPTR register.

XFCF — External Falling Edge Capture Flag
XFCF is set to “1” by hardware when an external falling edge capture event has occurred. XFCF must be cleared by software by writing “1”. The captured value is stored in CPTF register.

TCF — Terminal Count Interrupt Flag
TCF is set to “1” by hardware when terminal count occurs. TCF must be cleared by software by writing “1”.

CCF — Compare Match Interrupt Flag
CCF is set to “1” by hardware when compare match occurs. CCF must be cleared by software by writing “1”.

TC2CFG2B (0x4000_9025) TC2 Configuration Register 2B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SYNCTC1	SYNCPWM	-	-	-	-	-	-
WR	SYNCTC1	SYNCPWM	-	-	-	-	SWCPTR	SWCPTF

SYNCTC1 — Synchronize with TC1
Synchronize the counter start with TC1 counter.
This bit is set by software and cleared by hardware. Once this bit is set, the counter restart counting when TC1 count reaches restart point.
To maintain synchronization, TC1 and TC2’s clock scaling and period setting must be the same.

SYNCPWM — Synchronize with PWM
Synchronize the counter start with PWM counter.
This bit is set by software and cleared by hardware. Once this bit is set, the counter restart counting when PWM count reaches restart point.
To maintain synchronization, TC2and PWM clock scaling must be the same. Due to PWM counting is center-aligned, the period setting of TC1 must be 2X of PWM period setting.

SWCPTR — Software Capture R
Writing “1” to SWCPTR will generate a capture event and capture the count value into CPTR register. This bit is cleared by hardware.

SWCPTF — Software Capture F
Writing “1” to SWCPTF will generate a capture event and capture the count value into CPTF register. This bit is cleared by hardware

TC2CFG3A (0x4000_9028) TC2 Configuration Register 3A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	IENTC	IENCC	QECEN	CPTCLR	XRCEN	XFCEN	TCPOL	CCPOL
WR	IENTC	IENCC	QECEN	CPTCLR	XRCEN	XFCEN	TCPOL	CCPOL

IENTC — TC Interrupt Enable
IENCC — CC Interrupt Enable
QECEN — QE Capture Enable
QECEC=1 use QE output event as capture event.
CPTCLR — Enable Clear Counter after Capture

If CPTCLR=1, the TCCNT is cleared to 0 after each capture event. This allows continuous capture value with identical initial value.
If CPTCLR=0, the capture event does not affect the TCCNT counting.

XRCEN External Rising Edge Capture Enable
XRCEN=1 use external input rising edge as capture event.

XFCEN External Falling Edge Capture Enable
XFCEN=1 use external input falling edge as capture event.

TCPOL TC output polarity

CCPOL CC output polarity

Please note all capture sources are not mutually exclusive, i.e. allow several capture sources can coexist.

TC2CFG3B (0x4000_9029) TC2 Configuration Register 3B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	IENXRC	IENXFC
WR	-	-	-	-	-	-	IENXRC	IENXFC

IENXRC XC Rising Edge Interrupt Enable
IENXRC=1 enables interrupt when XRC event occurs.

IENXFC XC Falling Edge Interrupt Enable
IENXFC=1 enables interrupt when XFC event occurs.

TC2PRDL (0x4000_902C) TC2 Period and Count Register Low Double Buffer R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC2CNT[7-0]							
WR	TC2PRD[7-0]							

TC2PRDH (0x4000_902D) TC2 Period and Count Register High Double Buffer R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC2CNT[15-8]							
WR	TC2PRD[15-8]							

TC2CMPL (0x4000_9030) TC2 Compare Register Low Double Buffer R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC2CMP[7-0]							
WR	TC2CMP[7-0]							

TC2CMPH (0x4000_9031) TC2 Compare Register High Double Buffer R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC2CMP[15-8]							
WR	TC2CMP[15-8]							

TC2CPTRL (0x4000_9034) TC2 Rising Edge Capture Register R Low R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC2CPTR[7-0]							
WR	-							

TC2CPTRH (0x4000_9035) TC2 Rising Edge Capture Register R High R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC2CPTR15-8]							
WR	-							

TC2CPTFL (0x4000_9038) TC2 Falling Edge Capture Register F Low R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC2CPTF[7-0]							
WR	-							

TC2CPTFH (0x4000_9039) TC2 Falling Edge TC2 Capture Register F High R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TC2CPTF[15-8]							

20. Quadrature Encoder

The quadrature encoder is clocked by a scaled SYSCLK and has three external inputs through GPIO multi-functions. The three inputs include two signals of 90 degree phase difference, PHA and PHB, and an index indicating the terminal of the encoder. QE can function as an independent function block and can be configured to couple with TCC and use TCC to calculate the speed information of the encoder. Using TCC to capture TCC count value using the Index input of QE or terminal count of QE, the speed of QE input can be calculated. The QE unit implementation is shown in the following Block Diagram.

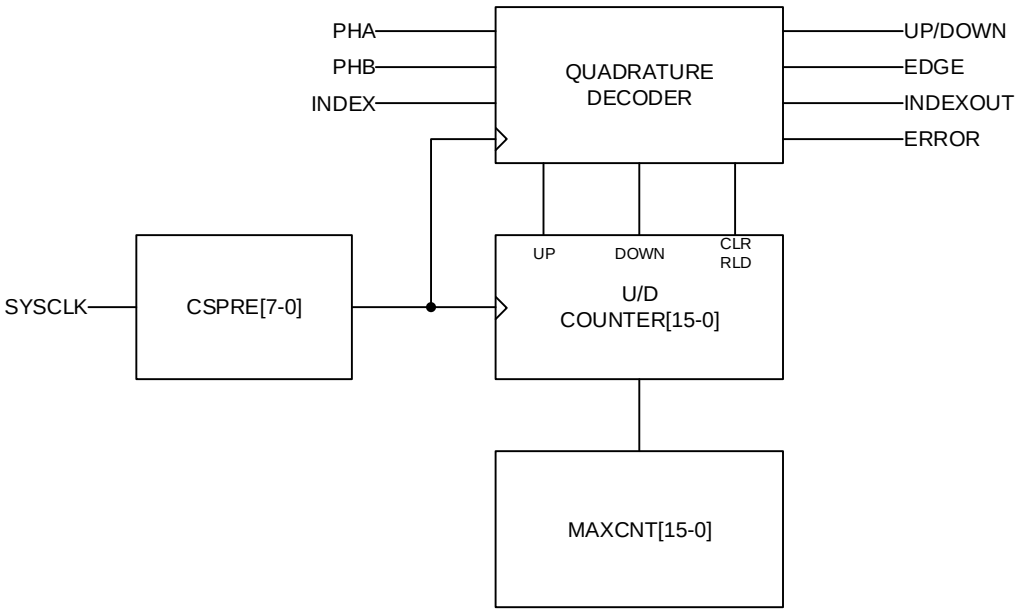


Figure 20-1 Quadrature Encoder Block Diagram

Please note QE Counter is in signed integer format, the MSB (Bit 15) indicates the sign, and reload action cause the counter loads a default value of 0x8000. The corresponding maximum count register thus only have 15 valid bits, MSB bit 15 is not used. The reload action is triggered either by external INDEX event or terminal count condition when counter absolute value reaches (equal) to MAXCNT value.

QECFG0 (0x4000_9040) QE Configuration Register 0 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	QEMODE[1-0]		QECS[1-0]		SWAP	DBCS[2-0]		
WR	QEMODE[1-0]		QECS[1-0]		SWAP	DBCS[2-0]		

MODE[1-0] QE Mode
00 – Disable QE
01 – 1X mode
10 – 2X mode
11 – 4X mode

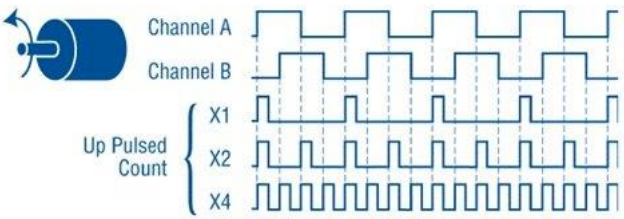


Figure 20-2 Quadrature Encoder input Timing for 1X, 2X and 4X

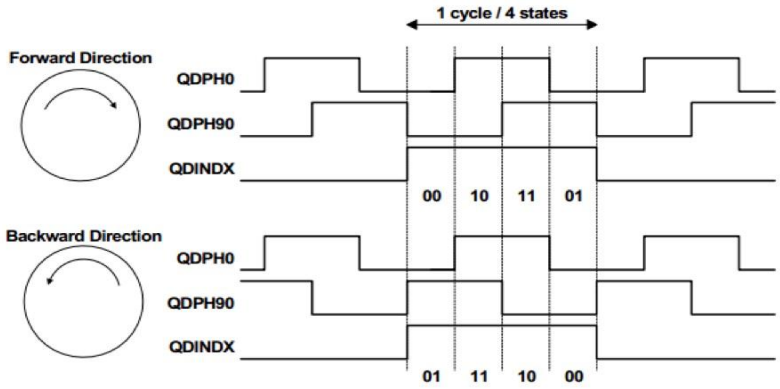


Figure 20-3 Quadrature Encoder input Timing for Forward and Backward

QECS[1-0]	QE Clock Scaling	
	00	SYSCLK/4
	01	SYSCLK/16
	10	SYSCLK/64
	11	SYSCLK/256
SWAP	Swap PHA and PHB	
DBCS[2-0]	De-Bounce Clock Scaling	
	000	Disable de-bounce
	001	SYSCLK/2
	010	SYSCLK/4
	011	SYSCLK/8
	100	SYSCLK/16
	1/32	SYSCLK/32
	1/64	SYSCLK/64
	1/128	SYSCLK/128
	1/256	SYSCLK/256

The debounce take 2 sample clocks to evaluate input, it usually takes 2 ~ 3 clocks delay

QECFG1 (0x4000_9041) QE Configuration Register 1 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	RLDM[1-0]		IDXEN	IDXM[1-0]	
WR	-	-	-	RLDM[1-0]		IDXEN	IDXM[1-0]	

RLDM[1-0] QE Counter Reload Mode

RLDM[1-0] = 00 No Reload, QECNT will count up/down cycling through 0x0000 or 0xFFFF

RLDM[1-0] = 01 Reload using Index event.

Reload QECNT=0, when Index==1 && UP

Reload QECNT=QEMAX, when Index==1 && DOWN

RLDM[1-0] = 10 Reload using TC event.

Reload QECNT=0, when QECNT==QEMAX && UP

Reload QECNT=QEMAX, when QECNT==0 && DOWN

RLDM[1-0] = 11 Reload using both Index and TC events

Combine Index and TC events and reload whichever occurs earlier

IDXEN Index Input Enable

IDXEN=0 gates out the external INDEX input is gated to 0.

IDXEN=1 allows external INDEX.

IDXM[1-0] Index Match Selection, this is applicable only for X2 and X4 modes.

00 = up phase 00 → 10 down phase 10 → 00

01 = up phase 10 → 11 down phase 11 → 10

10 = up phase 01 → 00 down phase 00 → 01

11 = up phase 11 → 01 down phase 01 → 11

QECFG2 (0x4000_9044) QE Configuration Register 2 R/W (0x80)

	7	6	5	4	3	2	1	0
RD	DIR	-	-	ERRF	TCF	IDXF	DIRF	CNTF
WR	-	-	-	ERRF	TCF	IDXF	DIRF	CNTF

DIR — Direction Status
1 is up
0 is down

ERRF — Phase Error Flab
ERRF is set to 1 by hardware if PHA and PHB change value at the same time. ERRF must be cleared by software.

TCF — TC Event Interrupt Flag
TCF is set by hardware when a TC event interrupt has occurred. TCF needs to be cleared by software by writing “1”.

IDXF — Index Event Interrupt Flag
IDXF is set by hardware when an Index event interrupt has occurred. IDXF needs to be cleared by software by writing “1”.

DIRF — Direction Change Event Interrupt Flag
DIRF is set by hardware when a Direction change event interrupt has occurred. DIRF needs to be cleared by software by writing “1”.

CNTF — Count Change Event Interrupt Flag
CNTF is set by hardware when a QE count change event interrupt has occurred. CNTF needs to be cleared by software by writing “1”.

QECFG3 (0x4000_9048) QE Configuration Register 3 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	IENERR	IENTC	IENIDX	IENDIR	IENTC
WR	-	-	-	IENERR	IENTC	IENIDX	IENDIR	IENTC

IENERR — Interrupt Enable for Error

IENTC — Interrupt Enable for TC
TC condition for QE is defined as the following three conditions
1. QECNT=QEMAX when UP
2. QECNT=0 when down

IENIDX — Interrupt Enable for Index event

IENDIR — Interrupt Enable for Direction Change

IENCNT — Interrupt Enable for any QECNT change

IDXEN — Index Input Enable
IDXEN=0 gates out the external INDEX input is gated to 0.
IDXEN=1 allows external INDEX.

QECNTL (0x4000_9050) QE Counter Low R/W (0x00)

	7	6	5	4	3	2	1	0
RD	QECNT[7-0]							
WR	QECNTINI[7-0]							

QECNTH (0x4000_9051h) QE Counter High R/W (0x00)

	7	6	5	4	3	2	1	0
RD	QECNT[15-8]							
WR	QECNTINI[15-8]							

Reading QECNT will return the current QE counter value. Writing QECNT will set the current count value. Writing QECNT is allowed only when QE is in disabled state.

QEMAXL (0x4000_9054) QE Maximum Counter Low R/W (0x00)

	7	6	5	4	3	2	1	0
RD	QEMAX[7-0]							
WR	QEMAX[7-0]							

QEMAXH (0x4000_9055) Maximum QE Counter High R/W (0x00)

	7	6	5	4	3	2	1	0
RD	QEMAX[15-8]							
WR	QEMAX[15-8]							

QEMAX hold the maximum count of the QE counter. When QEMAX is reached a TC event is triggered and QE counter is reloaded.

21. PWM Controller

PWM controller provides programmable 8 channels 14/12/10/8-bit PWM center-aligned duty cycle outputs. The duty cycle setting is always double buffered and PWM outputs are multiplexed with GPIO ports.

PWMCFG1A (0x4000_A080) PWM Clock Scaling Setting Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PWMEN	-	-	-	PRSEN	SYNCEN	MODE[1-0]	
WR	PWMEN	-	-	-	PRSEN	SYNCEN	MODE[1-0]	

PWMEN — PWM Controller Enable
PWMEN=0 clears the counter, reset the PWM state and all channel outputs are forced to 0.
PWMEN=1 allows normal running operation of PWM controller.

PRSEN — Pseudo-Random Sequence Enable
PRSEN=1 will enable a pseudo random sequence to the PWM output width. This can be an effective way to reduce EMI for output. When PRSEN=1, the instantaneous duty cycle will be affected cycle by cycle, but the average duty cycle remains the same.

SYNCEN — SYNC double update enable
SYNCEN=1, allow using SYNC to synchronize all channel duty value update.
SYNCEN=0, duty double buffer update immediately at next PWM start.

MODE[1-0] — PWM Mode
00 = 8-bit
01 = 10-bit
10 = 12-bit
11 = 14-bit

PWMCFG1B (0x4000_A081) PWM Clock Scaling Setting Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CS[7-0]							
WR	CS[7-0]							

CS[7-0] — PWM Counting Clock Scaling
PWM clock is SYSCLK/[CS[7-0]+1]

There are 8 independent PWMDTY registers to define the duty cycle. If PWMDTY = 0, the output is 0. If PWMDTY > the maximum for the specific mode setting the output duty cycle is full. For example, in 12-bit mode, PWMDTY is set higher than 0x10000, the output is in full duty. PWMDTY is always double buffered and is loaded to duty cycle comparator when the current counting cycle is completed.

PWMCFG2 (0x4000_A084) PWM Interrupt Enable and Flag Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SYNC	-	-	-	-	TINTF	ZINTF	CINTF
WR	SYNC	-	-	-	-	TINTF	ZINTF	CINTF

SYNC — Channel DTY Synchronize Control
If SYNCEN=1, SYNC is used to synchronize all channels DTY value update timing. When SYNC is written 1, the DTY values of all channels are updated together at next PWM cycle start. SYNC is cleared by hardware after the update occurs.

TINTF — Trigger Interrupt Flag
TINTF is set to 1 by hardware to indicate a trigger interrupt has occurred. TINTF must be cleared by software.

ZINTF — Zero Interrupt Flag
ZINTF is set to 1 by hardware to indicate a Zero interrupt has occurred. ZINTF must be cleared by software.

CINTF — Center Interrupt Flag
CINTF is set to 1 by hardware to indicate a center interrupt has occurred. CINTF must be cleared by software.

PWM counter =cccc 8bit cccc=255
10bit cccc=1023
12bit cccc=4095
14bit cccc=16383

PWMnDTY =aaaa

PWMTRG =bbbb

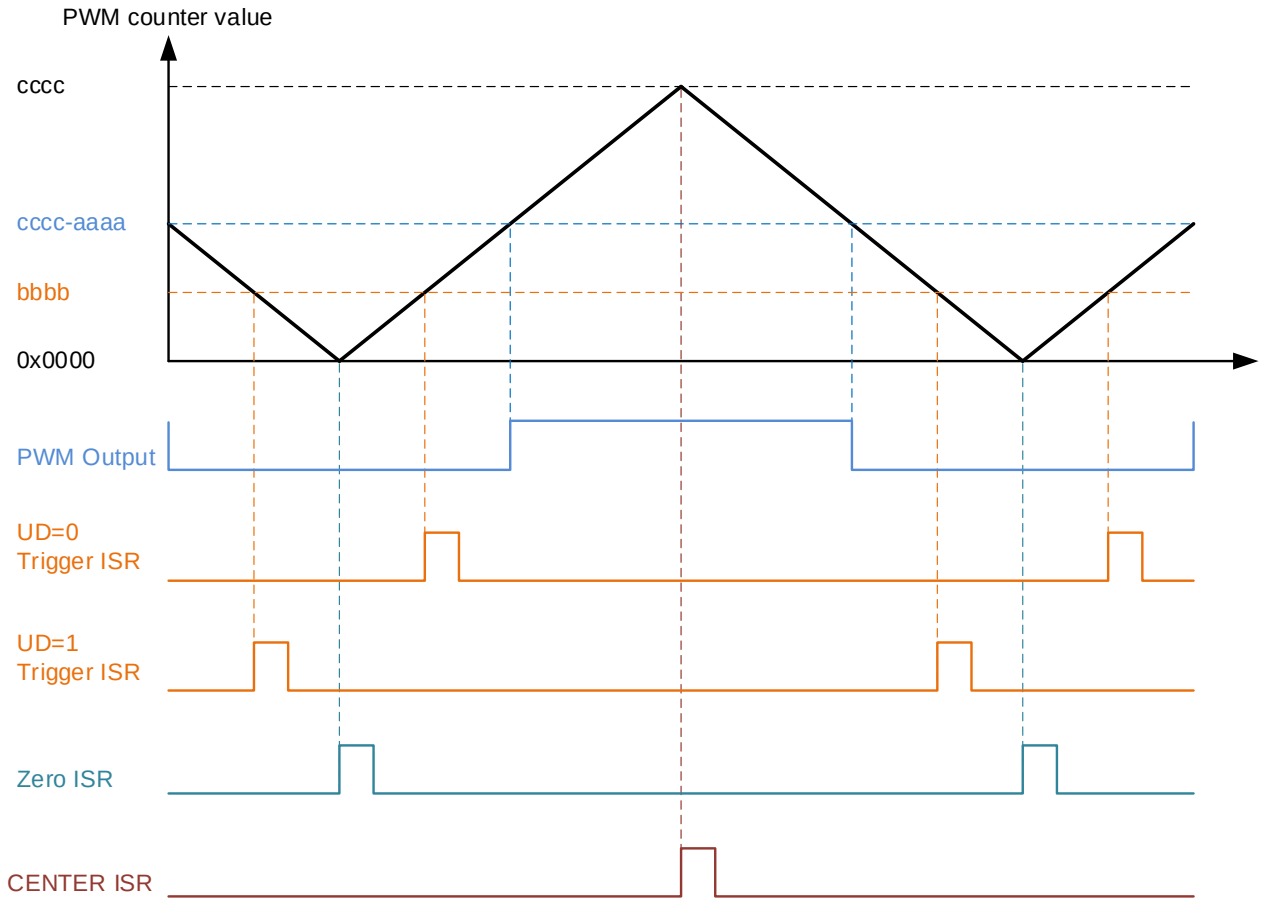


Figure 21-1 Relationship for Zero/Center/Trigger and PWM counter and output

PWMCFG3 (0x4000_A088) PWM Configuration 3 Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	TTRGEN	ZTRGEN	CTRGEN	-	TINTEN	ZINTEN	CINTEN
WR	-	TTRGEN	ZTRGEN	CTRGEN	-	TINTEN	ZINTEN	CINTEN

TTRGEN Trigger ADC Trigger Enable
ZTRGEN Zero ADC Trigger Enable
CTRGEN Center ADC Trigger Enable, only support 12 bit ADC
TINTEN Trigger Interrupt Enable
TINTEN=1 allows PWM Controller to generate interrupt when count reach PWMTRG.
ZINTEN Zero Interrupt Enable
ZINTEN=1 allows PWM Controller to generate interrupt at start of the PWM cycle.
CINTEN Center Interrupt Enable
CINTEN=1 allows PWM Controller to generate interrupt at center of the PWM cycle.

PWM0DTY (0x4000_A090) PWM0 Duty Register R/W (0x0000)

	31-14	13-0
RD	-	PWM0DTY[13-0]
WR	-	PWM0DTY[13-0]

PWM0DTY[13-0] PWM Channel 0 Duty setting

PWM1DTY (0x4000_A094) PWM1 Duty Register R/W (0x0000)

	31-14	13-0
RD	-	PWM1DTY[13-0]
WR	-	PWM1DTY[13-0]

PWM1DTY[13-0] PWM Channel 1 Duty setting

PWM2DTY (0x4000_A098) PWM2 Duty Register R/W (0x0000)

	31-14	13-0
RD	-	PWM2DTY[13-0]
WR	-	PWM2DTY[13-0]

PWM2DTY[13-0] PWM Channel 2 Duty setting

PWM3DTY (0x4000_A09C) PWM3 Duty Register R/W (0x0000)

	31-14	13-0
RD	-	PWM3DTY[13-0]
WR	-	PWM3DTY[13-0]

PWM3DTY[13-0] PWM Channel 3 Duty setting

PWM4DTY (0x4000_A0A0) PWM4 Duty Register R/W (0x0000)

	31-14	13-0
RD	-	PWM4DTY[13-0]
WR	-	PWM4DTY[13-0]

PWM4DTY[13-0] PWM Channel 4 Duty setting

PWM5DTY (0x4000_A0A4) PWM5 Duty Register R/W (0x0000)

	31-14	13-0
RD	-	PWM5DTY[13-0]
WR	-	PWM5DTY[13-0]

PWM5DTY[13-0] PWM Channel 5 Duty setting

PWM6DTY (0x4000_A0A8) PWM6 Duty Register R/W (0x0000)

	31-14	13-0
RD	-	PWM6DTY[13-0]
WR	-	PWM6DTY[13-0]

PWM6DTY[13-0] PWM Channel 6 Duty setting

PWM7DTY (0x4000_A0AC) PWM7 Duty Register R/W (0x0000)

	31-14	13-0
RD	-	PWM7DTY[13-0]
WR	-	PWM7DTY[13-0]

PWM7DTY[13-0] PWM Channel 7 Duty setting

It is also possible to set interrupt trigger and ADC trigger through register setting.

PWMTRG (0x4000_A0D0) PWM Trigger Setting Register R/W (0x0000)

	15	14	13-0
RD	-	UD	PWMTRG[13-0]
WR	-	UD	PWMTRG[13-0]

UD Up/Down Setting
UD=0 indicates up counting aligned
UD=1 indicates down counting aligned

PWMTRG[13-0] Trigger Point setting

22. Melody Controller

The melody controller can be used to generate a melody. The tone frequency is derived from SYSCLK divided by either 32 or 64, and the tone frequency is generated with a resolution of 14-bits to support precision tone generation with wide octave span. The duration/pause timers can be programmed in 1ms/2ms/4ms/8ms steps.

Melody Controller has two outputs, BZ and POW. BZ is used to drive the speaker or buzzer. POW is used to enable the driver which allows a gradual diminishing tonal effect. BZ and POW can be multiplexed to GPIO pin using GPIO multifunction select.

The parameters for each note of a melody are: note frequency, note duration, note pause time, and driver on window (POW time). These parameters are double buffered so a new note can be written when previously set note is being played. As these are double buffered, the writing sequence should follow FRQL, DUR, PAU, POW (these can be written in byte, word, or double word) then last FRQH. When FRQH is written, it completes the setting of a note. The timing waveform of a note is shown in the following diagram.

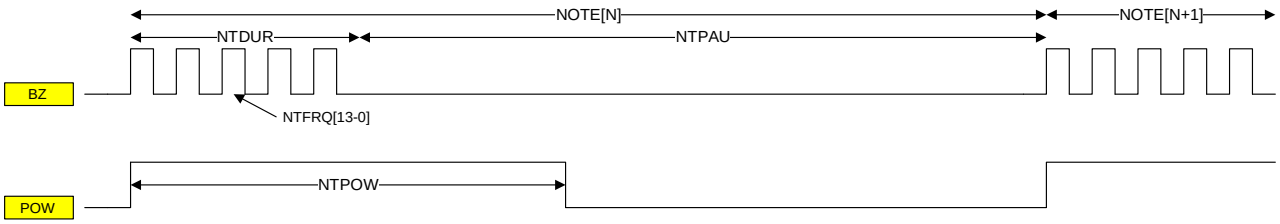


Figure 22-1 Melody Controller Output Waveform

NTFRQL (0x4000_B000) Note Frequency Register L RW DB (0x00)

	7	6	5	4	3	2	1	0
RD	NTFRQ[7-0]							
WR	NTFRQ[7-0]							

NTFRQ[7-0] Note Frequency

NTFRQH (0x4000_B001) Note Frequency Register H DB R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PLAYQUE	MBZINTF	NTFRQ[13-8]					
WR	PLAYQUE	MBZINTF	NTFRQ[13-8]					

PLAYQUE Play Queue
Writing this bit “1” will put the parameters of the note (frequency, duration, pause, and POW) into the play queue.
This bit is self-cleared by hardware when the queue is ready to update.

MBZINTF Melody Buzzer Controller Interrupt Flag
MBZINTF is set to 1 if play queue is ready to be updated. In other words, the flag is set by self-clearing action of PLAYQUE bit. MBZINTF must be cleared by software by writing “1”.

NTFRQ[13-0] Note Frequency
Tone frequency is Tone Base Clock/(NTFRQ[13-0]+1).
Please note due to timing restriction, NTFRQ[13-0] must be >= 2.

NTDUR (0x4000_B004) Note Duration Register WR DB (0x00)

	7	6	5	4	3	2	1	0
RD	NTDUR[7-0]							
WR	NTDUR[7-0]							

NTDUR[7-0] Note Duration
Note duration is TU * NTDUR[7-0]

NTPAU (0x4000_B005) Note Pause Register RW DB (0x00)

	7	6	5	4	3	2	1	0
RD	NTPAU[7-0]							
WR	NTPAU[7-0]							

NTPAU[7-0] Note Pause
Note pause is TU * NTPAU[7-0]

NTPOW (0x4000_B006) Note Power-On Window Register RW DB (0x00)

	7	6	5	4	3	2	1	0
RD	NTPOW [7-0]							
WR	NTPOW [7-0]							

NTPOW[7-0] Note POW Time
POW time is TU * NTPOW[7-0]

BZCFG (0x4000_B008) Buzzer Configure Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	MBZEN	MBZIE	BUSY	POWPOL	BZPOL	TBASE	TU[1-0]	
WR	MBZEN	MBZIE	-	POWPOL	BZPOL	TBASE	TU[1-0]	

MBZEN Melody Buzzer Controller Enable
MBZEN=1 enables the buzzer controller
MBZEN=0 disables the buzzer controller

MBZIE Melody Buzzer Controller Interrupt Enable
IE = 0 disables interrupt.
IE = 1 enables interrupt.
There is only one interrupt for Melody controller. It is generated when the play queue is ready for input after a note was written previously. The interrupt flag is MBZIF in NTFRQH register.

BUSY Melody/Buzzer Status
BUSY is set to 1 by hardware if Melody controller is playing.

POWPOL POW Polarity Setting
POWPOL=1, POW output logic is inverted, i.e., POW is low at the start of the note.
POWPOL=0, normal POW polarity as shown in the timing diagram.

BZPOL BZOUT Polarity Setting
BZPOL=1, BZOUT is inverted
BZPOL=0, normal polarity

TBASE Tone Base Frequency Select
TBASE=0 uses (SYSCLK or APB Clock)/32 as tone base clock
TBASE=1 uses (SYSCLK or APB Clock)/64 as tone base clock

TU[1-0] Time Unit
TU[1-0] defines the time unit for duration and pause, and POW timer
00 = 1msec
01 = 2msec
10 = 4msec
11 = 8msec

23. Essential Analog Blocks

23.1 On-Chip 1.5V Regulator

An on-chip regulator is used to provide supply for core logic and internal E-Flash memory (VDDC). The regulator is partitioned into a back-up regulator and a main regulator. The main regulator is enabled only in normal, and SLEEP modes, and disabled in DEEP SLEEP mode when the backup regulator is turned on. The main regulator consumes about 100uA, while the back-up regulator consumes about 1uA. The back-up regulator can supply up to 500uA; therefore, it is important all peripheral circuits should be kept off during DEEP SLEEP mode. The regulator requires an external capacitor, which should be connected to VDDC pin. A minimum of 1uF plus a 0.1uF in parallel is required for the stability of the regulator. The main regulator outputs about can be adjusted by REGTRM (with default at its maximum) and the back-up regulator outputs about 1.40V. A manufacturer calibrated value of REGTRM for 1.5V is stored in IFB.

REGTRM (0x4000_F100) Regulator Trim Register RW (0xFF)

	7	6	5	4	3	2	1	0
RD	REGTRM[7-0]							
WR	REGTRM[7-0]							

REGTRM[7-0] Trim Register for main 1.5V regulator.
REGTRM[7-0]=FF corresponds to maximum output level. REGTRM[7-0]=00 corresponds to minimum output level. The in-between value in general is linear to the output level. Typically the maximum is around 1.65V while the minimum is around 1.35V.

REGRDY (0x4000_F104) Regulator Ready Delay Register RW (0x06)

	7	6	5	4	3	2	1	0
RD	-					REGRDY[2-0]		
WR	-					REGRDY[2-0]		

REGRDY[2-0] Regulator Ready Delay Setting
REGRDY[2-0] defines the number of SOSC32K cycle for allowing regulator stabilize at exit of sleep mode.
000 N=0, Please don't set N = 0
001 N=1
010 N=4
011 N=16
100 N=64
101 N=128
110 N=256
111 N=510

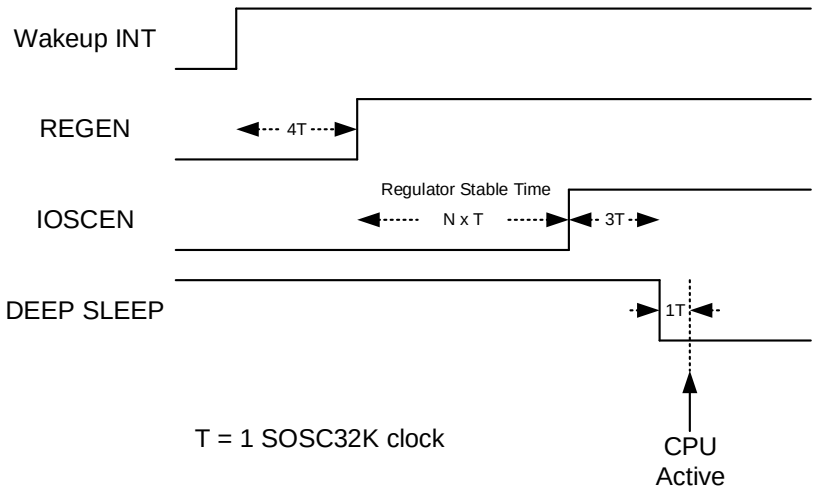


Figure 23-1 Regulator on/off control under sleep / wakeup

23.2 Precision Internal Oscillator (IOSC)

The internal oscillator is a very important peripheral as it provides the default clock source after reset and other critical timing. The internal oscillator remains stabil during the enable and disable transients. No clock glitches or extra clock edge is generated during the on/off transition, and the oscillator can reach stable oscillations, typically within 10 cycles. The IOSC consumes around 300uA when enabled. The IOSC is always enabled except when entering into SLEEP / DEEP SLEEP mode. And in DEEP SLEEP mode when it is disabled, IOSC consumes less than 0.1uA standby current.

Similar to the on-chip regulator, IOSC also exhibits chip-to-chip variations. A calibrated value that set IOSC at 16MHz/32MHz +/- 1% is stored in IFB. The user program can set this value to IOSC trimming register, IOSCITRM and IOSCVTRM. The IOSC frequency has very little variations over the operation range (-40°C– 85°C and VDD = 2.5V – 5.5V). The variation is typically less than +/-2% over the operation conditions. It is possible that user program to set a different frequency as long as user program provide a calibration method to set IOSC frequency at the desired value at typical operation condition, and it will be stable and accurate over the entire operation range. Please note that the trimming register will be set to its default value after resets, the user program must reinitialize to its calibrated value.

The IOSC is also equipped with Spread Spectrum capability for EMI sensitive applications. The SS deviation can be controlled to fit various requirements. However, once SS is enabled, the accuracy of IOSC cannot be maintained.

IOSCITRM (0x4000_F200) IOSC Coarse Trim Register R/W (0x01)

	7	6	5	4	3	2	1	0
RD	SSC[3-0]				SSA[1-0]		ITRM[1-0]	
WR	SSC[3-0]				SSA[1-0]		ITRM[1-0]	

SSC[3-0] SSC[3-0] defines the spread spectrum sweep rate. If SSC[3-0] = 0000, then the spread spectrum is disabled.

SSA[1-0] SSA[1-0] defines the amplitude of spread spectrum frequency change. The frequency is changed by adding SSA[1-0] range to actual IOSCVTRM[7-0].
SSA[1-0] = 11, +/- 32
SSA[1-0] = 10, +/- 16
SSA[1-0] = 01, +/- 8
SSA[1-0] = 00, +/- 4

ITRM[1-0] ITRM[1-0] is the coarse trimming of the IOSC.

IOSCVTRM (0x4000_F201) IOSC Fine Trim Register R/W (0x80)

	7	6	5	4	3	2	1	0
RD	IOSCVTRM[7-0]							
WR	IOSCVTRM[7-0]							

This register provides fine trimming of the IOSC frequency. The higher the value of IOSCVTRM, the lower the frequency is.

IOSCPDIV (0x4000_F204) IOSC Post Divider Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-				-	IOSCPDIV[2-0]		
WR	-				-	IOSCPDIV[2-0]		

IOSCDIV[2-0] IOSC Post Divider

IOSCPDIV[2-0]	IOSC Post Divider Output
0	IOSC
1	IOSC/2
2	IOSC/4
3	IOSC/6
4	IOSC/8
5	IOSC/10

IOSCPDIV[2-0]	IOSC Post Divider Output
6	IOSC/12
7	IOSC/16

The post divider output is used for SYSCLK selection.

The manufacturer trim value is stored in IFB and is trimmed to 16MHz/32MHz. The user program provides the freedom to set the IOSC at a preferred frequency as long as the program is able to calibrate the frequency. Once set, the IOSC frequency has accuracy deviation within +/- 2% over the operation conditions. The following lists the range of the typical IOSC frequency for each trimming setting.

ITRM[1-0]=2'b11, IOSC=27.4—36.8MHz

ITRM[1-0]=2'b10, IOSC=25.5—34.3MHz

ITRM[1-0]=2'b01, IOSC=14.1—19.2MHz

ITRM[1-0]=2'b00, IOSC=12.2—16.5MHz

A hardware Spread Spectrum can be enabled for the IOSC. This is controlled by SSC[3-0]. When SSC[3-0] = 0, the spread spectrum is disabled, and IOSC functions normally as a fixed frequency oscillator. If SSC[3-0] is not 0, then Spread Spectrum is enabled and IOSC frequency is swept according to the setting of SSC[3-0] and SSA[1-0]. The spread is achieved by varying the actual VTRM output to the oscillator circuit thus effectively changes the oscillation frequency. The effect of SSC[3-0] and SSA[1-0] is shown in the following graph.

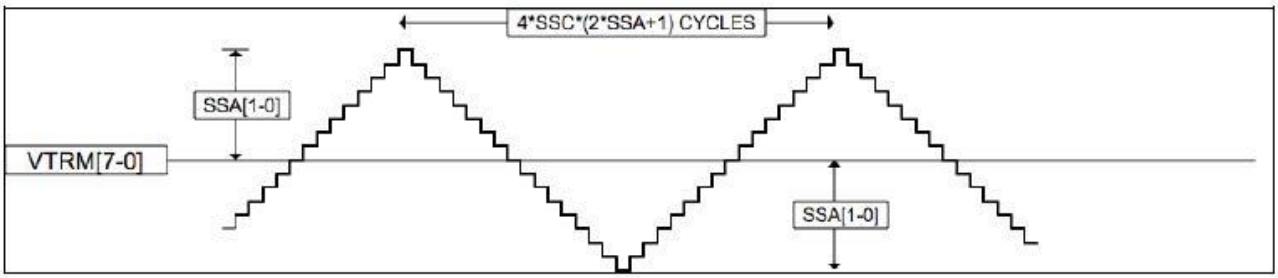


Figure 23-2 IOSC Frequency sweep for Spread Spectrum enable

When Spread Spectrum is enabled, the actual controlling output to IOSC is VTRM[7-0] +/- SSA. This is shown in the graph above as the bold curve. The above example shows SSA[1-0] = 01, and the deviation is +/- 8. SSC[3-0] defines the update time in IOSC cycles. Then we can calculate the period of a complete sweep is 4*SSC*(2SSA+1) IOSC cycles, and we can obtain the sweep frequency from this period. When SS is enabled, the frequency of IOSC varies according to time and setting, therefore, the accuracy of IOSC frequency cannot be guaranteed. Please also note that VTRMOUT is VTRM[7-0] +/- SSA but is bounded by 0 and 255. Therefore for a linear non-clipped sweep, VTRM[7-0] needs to be within the range of SSA – 256-SSA, for example, SSA[10] = 01, then SSA is 8. VTRM[7-0] should be in the range from 8 to 248 to prevent the sweep from being clipped. As Spread Spectrum suggests, the total EMI energy is not reduced, but the energy is spread over wider frequency. It is recommended that SS usage should be carefully evaluated and the setting of spread amplitude and the sweep frequency should be chosen carefully for reducing EMI effect.

23.3 SOSC 128KHz

An ultralow power slow oscillator of 128KHz is also present for use as wake-up or sleep mode system clock. SOSC is never powered down and consumes about 1uA from VDDC. SOSC frequency is temperature dependent typically +/- 20% over the operating range. It can be trimmed using SOSCTRM register.

SOSCTRM (0x4000_F300) SOSC Trim Register R/W (0x08)

	7	6	5	4	3	2	1	0
RD	-			SOSCTRM[4]	SOSCTRM[3-0]			
WR	-			SOSCTRM[4]	SOSCTRM[3-0]			

SOSCTRM[4] 256KHz Select
If SOSCTRM[4]=1, SOSC oscillator centers at 256KHz.
If SOSCTRM[4]=0, SOSC oscillator centers at 128KHz.

SOSCTRM[3-0] SOSC Trim Setting
SOOSCTRM[3-0] is used to fine tune the oscillation frequency.

Please note regardless of SOSCTRM[4] setting, SOSC is always at 128KHz and SOSC32KHz is 32KHz to ensure consistent timing based on SOSC. The implementation is shown as the following Block Diagram.

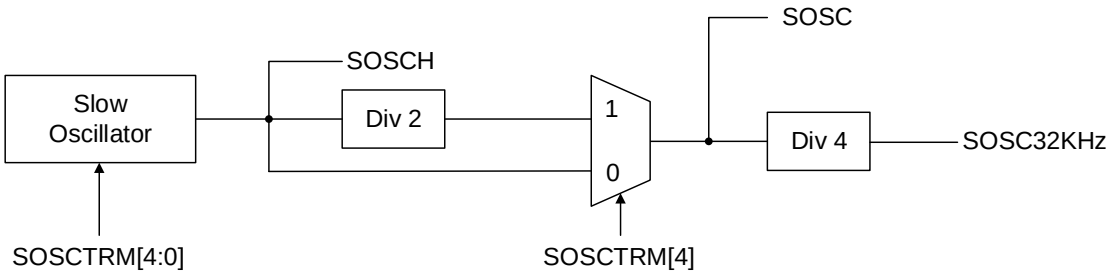


Figure 23-3 SOSC VS. SOCK32K

23.4 XOSC

The on-chip crystal oscillator is suitable for using external crystal with frequency range between 1MHz to 25MHz. XIN can also be used for external clock input. When enabled, the XOSC consumes between 0.5mA to 1.5mA depending on power setting. It takes about 1msec to be stable after startup.

XIN and XOUT are parallel connections to P17 and P20. When used as crystal oscillator input, P17 and P20 must be put in Hi-Z mode. Vice versa, when P17 and P20 are used, XOSC must be in disabled state with all enable bits set to 0.

XOSCCFG (0x4000_F400h) Crystal Oscillator Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	EXEN	-	XOSCEN1	XOSCEN0
WR	-	-	-	-	EXEN	-	XOSCEN1	XOSCEN0

EXEN	External Input Enable EXEN=0 for oscillator function EXEN=1 for external clock input function. External clock is applied from XIN pin. In this mode, XOSCEN1/0 should be set to 0.
XOSCEN1	XOSC Enable 1 XOSCEN1=1 enables 1mA oscillator current
XOSCEN0	XOSC Enable 0 XOSCEN0=1 enables 0.5mA oscillator current XOSCEN1=XOSCEN0=0 disables the crystal oscillator

23.5 PLL

An on-chip PLL is included for generating system clock (SYSCLK). The maximum operating frequency of the SYSCLK is 48MHz. The Block Diagram of the PLL is shown in picture below. The VCO has limited range of 40MHz to 100MHz for better jitter performance. For lower target frequency of PLLCLK, post divider can be used. PLL takes about 1msec to be stable and consumes about 4mA. PLL enable stable time is about 10 µsec and should allow for 100 µsec for fully stabilized.

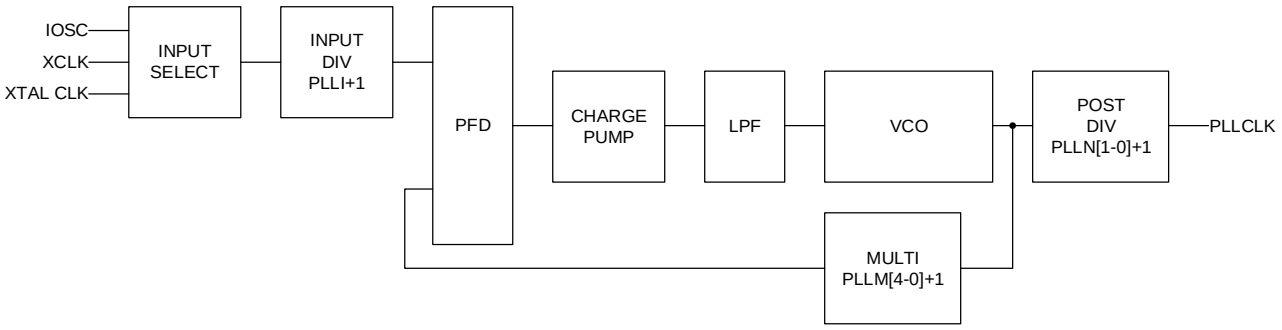


Figure 23-4 PLL Block Diagram

The effective PLL output frequency $Fout = Fin * (PLLM[4-0]+1)/(PLLI+1)/(PLLN[1-0]+1)$.

PLLCFGA (0x4000_F500h) PLL Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PLLEN	INSEL[1-0]		VCOBIAS[1-0]		CPBIAS[1-0]		FAIL
WR	PLLEN	INSEL[1-0]		VCOBIAS[1-0]		CPBIAS[1-0]		-

PLLEN — PLL Enable

INSEL — Input Selection
00 = IOSC
01 = XCLK (external CLK input)
10 = XOSC (Crystal CLK)
11 = XOSC (Crystal CLK)

VCOBIAS[1-0] — VCO Bias, VCO frequency setting

CPBIAS[1-0] — Charge Pump Bias, 0b10 is the recommend setting

FAIL — PLL FAIL
VCO control voltage is not within design range. This indicates PLL failed to lock. This is a real-time status. PLL can be used only if FAIL=0.

PLLCFGB(0x4000_F501h) PLL Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PLLI	PLLN[1-0]		PLLM[4-0]				
WR	PLLI	PLLN[1-0]		PLLM[4-0]				

PLLI — PLL Input Divider
The input clock is divided by PLLI+1

PLLN[1-0] — PLL Output Divider
The output clock is divided by PLLN[1-0]+1

PLLM[4-0] — PLL Multiplication
The VCO output frequency is input frequency times PLLM[4-0]+1

PLLCFGC(0x4000_F502h) PLL Configuration Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-		-		IBIAS[2-0]		
WR	-	-		-		IBIAS[2-0]		

IBAIS[2-0] — PLL Bias Setting, 0b010 is the recommend setting

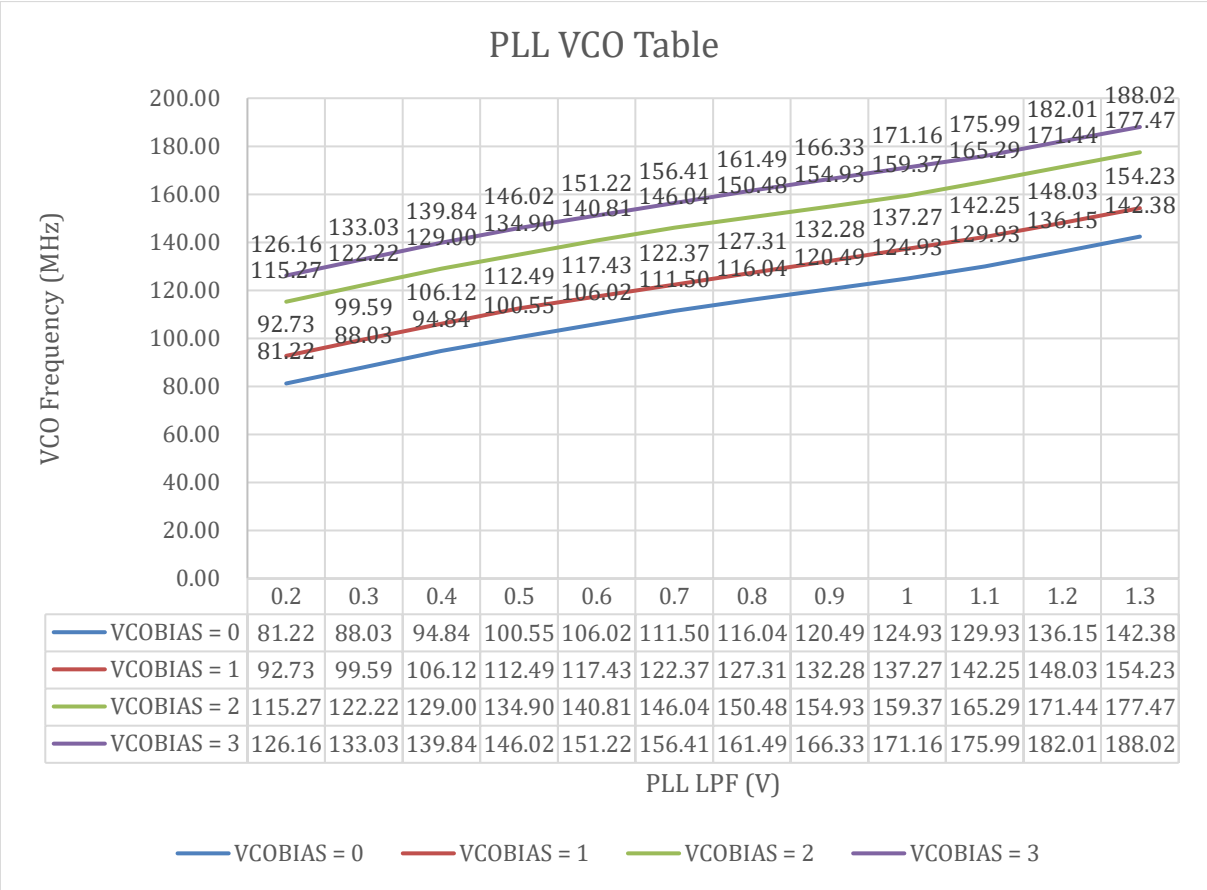


Figure 23-5 VCO Frequency Table

The PLL LPF voltage (measured from pin P21 by set ANEN2 of GPIO07C as 1) should be range from (1/2 VDDC +- 0.25V (0.5V ~ 1.0V) and as close to 1/2 VDDC (0.75V) as possible.

VCOBIAS = 0: 100.55 ~ 124.93MHz

VCOBIAS = 1: 112.49 ~ 137.27MHz

VCOBIAS = 2: 134.9 ~ 159.37MHz

VCOBIAS = 3: 151.22 ~ 171.16MHz

Example to set PLL parameters, assume the target spec is X'tal input 4MHz and PLL output 40MHz.

a. Choose PLLN, PLL output is 40MHz, VCO frequency output is 40MHz * PLLN

b. PLLN should be 2 or 3 which results to VCO frequency output is 120MHz (VCOBIAS = 0) or 160MHz (VCOBIAS = 3)

c. PLLN = 2 need a multiplier 40 which exceed the maximum value of PLLM[4-0]

d PLLN = 3 need a multiplier 40, then the PLLM[4-0] is 29 (30-1) and PLLI should be 1 (2-1)

e. We get all the parameters for PLL setting such as PLLN, PLLM, PLL I and VCOBIAS

f. Check bit 0 of PLLCFGA

g. Measure PLL LPF to ensure the voltage is within 1/2 VDDC +- 0.25V

Note: The solution may be more than one, it's recommended to choose the one which PLL LPF voltage is closest to 1/2 VDDC (by set GPIO21 as 0x00020000 and measure P21) The P21 must be set as PLLLPF (GPIO21 set as 0x00020000) and connect a 1uF capacitor to GND if VCO clock is higher than 96MHz).

23.6 Real Time Clock (32768Hz) Oscillator

RTC oscillator is also an ultra-low power oscillator that consumes less than 1uA from VDDC. The power level can be programmed to suit for different 32768 crystal characteristics. RTC oscillator includes a ripple divider that divides the 32768Hz down to 4Hz/2Hz/1Hz. These can be used as a periodic interrupt or wake-up sources. It also includes a 30-bit counter with unit of second that is used to maintain calendar. A corresponding 30-bit alarm

register can be used to generate wake-up and interrupt when the second counter reaches the value in the alarm register. The RTC is equipped with digital calibration means. The range of calibration is approximately +/- 200ppm and the resolution is about 1ppm.

RTCCFGA (0x4000_F800) RTC Oscillator Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RTCEN	RTCPW	-	-	ALMIEN	4HZIEN	2HZIEN	1HZIEN
WR	RTCEN	RTCPW	-	RTCSET	ALMIEN	4HZIEN	2HZIEN	1HZIEN

RTCEN RTC Oscillator Enable
RTCPW RTC Oscillator Power Setting
RTCPW=0 uses lower power (0.6uA)
RTCPW=1 uses higher power (1.2uA)
RTCSET RTC Second Counter Set
Setting RTCSET=1 will reset RTCSND[29-0] to all 0. This is self-cleared by hardware.
ALMIEN Alarm Interrupt Enable
4HZIEN 4Hz Interrupt Enable
2HZIEN 2Hz Interrupt Enable
1HZIEN 1Hz Interrupt Enable

RTCCFGB (0x4000_F804) RTC Oscillator Calibration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RTCCAL[7-0]							
WR	RTCCAL[7-0]							

RTCCAL[7-0] RTC Calibration Setting
Each LSB of RTCCAL defines the adjustment of approximately 4ppm.

RTCCFGC (0x4000_F805) RTC Oscillator Calibration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	RTCCAL[8]
WR	-	-	-	-	-	-	-	RTCCAL[8]

RTCCAL[8] Calibration Polarity
RTCCAL[8] =0, adjust RTCCAL[7-0] steps to slow down RTC. Each steps adjust approximately 4ppm. Total range is +/- 4*255 = +/- 1025ppm, approximately +/- 0.1%.
RTCCAL[8] =1, adjust RTCCAL[7-0] steps to speed up RTC.

RTCIF (0x4000_F808) RTC Oscillator Interrupt Flag Register R/W 0x00

	7	6	5	4	3	2	1	0
RD	-	-	-	-	ALMF	4HZF	2HZF	1HZF
WR	-	-	-	-	ALMF	4HZF	2HZF	1HZF

ALMF Alarm Interrupt Flag
ALMF is set to 1 by hardware. It must be cleared by software by writing 1.
4HZF 4Hz Interrupt Flag
4HZF is set to 1 by hardware. It must be cleared by software by writing 1.
2HZF 2Hz Interrupt Flag
2HZF is set to 1 by hardware. It must be cleared by software by writing 1.
1HZF 1Hz Interrupt Flag
1HZF is set to 1 by hardware. It must be cleared by software by writing 1.

RTCSND (0x4000_F810) RTC Second Count Register RO (0x0000_0000)

	31-0
RD	RTCSND[31-0]
WR	-

RTCSND[31-0] RTC Second count value

RTCALM (0x4000_F814) RTC Alarm Register RW (0x00) (0x0000_0000)

	31-0
RD	RTCALM[31-0]
WR	-

RTCALM[31-0] Alarm Setting
The alarm interrupt is triggered at RTCALM[31-0]+1.

23.7 Supply Low Voltage Detection (LVD)

The supply Low Voltage Detection (LVD) circuit detects VDD < VTH condition and can be used to generate an interrupt or reset condition. An enabled LVD circuit consumes about 100uA to 200uA. The LVDTHD[6-0] sets the compare threshold according to the following equation when LVDTHV is the detection voltage.

LVDTHV = VDDC * (1 + 2 * (1 – LVDTHD[6-0]/128))
= 1.5 + 3 * (1 – LVDTHD[6-0]/128), if VDDC is calibrated to 1.5V.

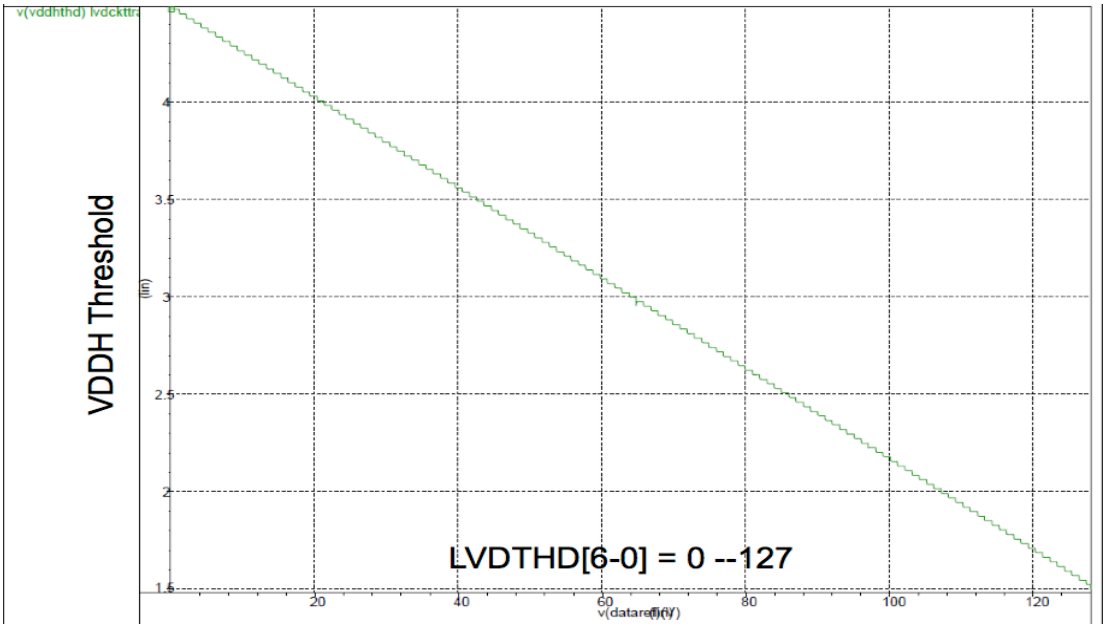


Figure 23-6 LVDTHD

LVDCFG (0x4000_F700) Low Voltage Detection Configuration Register RW (0xA0)

	7	6	5	4	3	2	1	0
RD	LVDEN	LVREN	LVTEN	LVDFLTEN	-	-	-	LVTIF
WR	LVDEN	LVREN	LVTEN	LVDFLTEN	-	-	-	LVTIF

LVDEN LVD Enable bit. Set to turn on supply voltage detection circuits.
LVREN LVR Enable bit. LVREN = 1 allows low voltage detect condition to cause a system reset.
LVTEN LVT Enable bit. LVTEN = 1 allows low voltage detect condition to generate an interrupt.
LVDFLTEN LVD Filter Enable
LVDFLTEN = 1 enables a noise filter on the supply detection circuits. The filter is set at around 30usec.
LVTIF Low Voltage Detect Interrupt Flag
LVTIF is set by hardware when LVD detection occurs and must be cleared by software writing 1.

LVDTHD (0x4000_F704) LVD Threshold Register R/W (0x7F)

	7	6	5	4	3	2	1	0
RD	-	LVDTHD6	LVDTHD5	LVDTHD4	LVDTHD3	LVDTHD2	LVDTHD1	LVDTHD0
WR	-	LVDTHD6	LVDTHD5	LVDTHD4	LVDTHD3	LVDTHD2	LVDTHD1	LVDTHD0

LVDTHD = 0x00 will set the detection threshold at its maximum, and LVDTHD = 0x7F will set the detection threshold at its minimum (approximately 1.8V).

LVDHYS (0x4000_F708) LVD Threshold Hysteresis Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LVDHYEN	LVDHYS6	LVDHYS5	LVDHYS4	LVDHYS3	LVDHYS2	LVDHYS1	LVDHYS0
WR	LVDHYEN	LVDHYS6	LVDHYS5	LVDHYS4	LVDHYS3	LVDHYS2	LVDHYS1	LVDHYS0

To ensure a solid Low Voltage detect, a digital controlled hysteresis is used. If LVDHYEN=1, when LVD is asserted a new threshold defined by LVDHYS[6-0] replaces LVDTHD[6-0]. In typical applications, LVDHYS[6-0] should be set to be smaller than LVDTHD[6-0] such that the recovery voltage is higher than detection voltage..

Please note LVDCFG, LVDTHD, and LVDHYS are set to their default values only with power-on-reset or its equivalent. Other reset condition does not restore these to their default values.

24. 12-Bit SAR ADC

The on-chip ADC is a 12-bit two stage SAR ADC. The ADC has two full-scale reference selection, internal VDDC (1.5V typical) as reference with maximum ADC clock rate of 250K, or external reference, ADCREF, though GPIO analog input with maximum ADC clock up to 2MHz. The external reference level should be higher than 2.0V and lower than VDD-1V.

When enabled, the ADC consumes about 1mA of current. Typical ADC accuracy is about 9.5 bit to 10 Bit. The accuracy will degrade to 8.5-bit to 9-bit if the input is too close to 0V (0V to 0.25V).

There are four input channels of ADC. CHA and CHB are further connected to GPIO's analog I/O switches to expand multiplexed inputs. The third channel, TPS, is connected to internal temperature sensors and can be further selected using a negative temperature coefficient source of a diode-connected NPN, or a positive temperature coefficient source derived from band-gap reference. The fourth channel, VPS, is connected to 1/5th of VDD for measuring the supply voltage. There is a T/H front-end of ADC input and can be set to enable or bypass.

ADC conversion can be software triggered or hardware triggered. Hardware trigger sources include TC1 and TC2 CC events, and PWM Center and Zero events. The conversion takes 14 to 16 ADCCLK cycles. The conversion result can go through hardware averaging of 1 to 8 conversions.

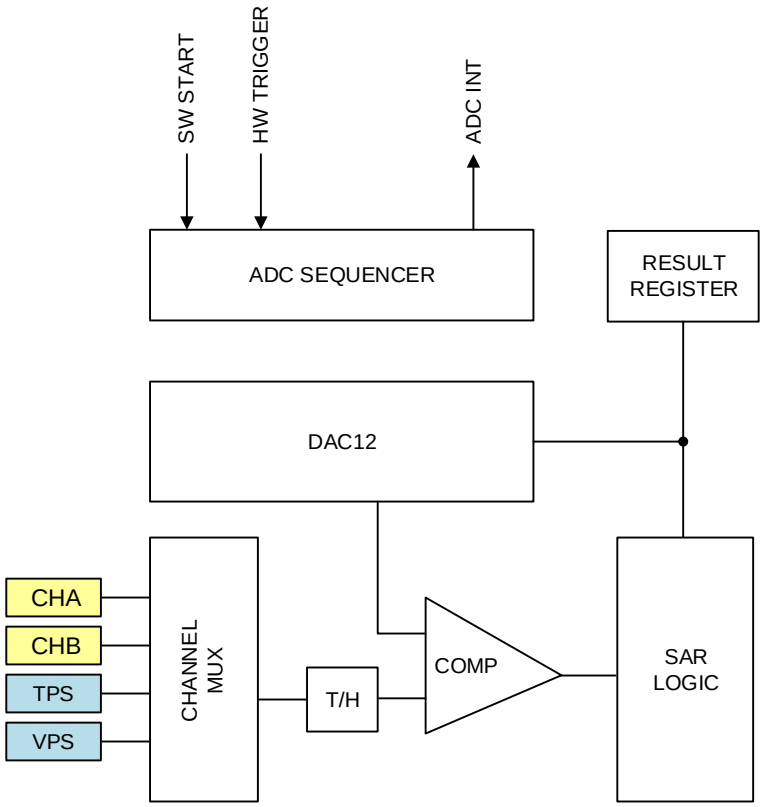


Figure 24-1 12bit SAR ADC

ADCCFG0 (0x4000_E000) ADC Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ADCEN	ADCCS[2-0]			TRGTC2	TRGTC1	TRGPT	TRGPZ
WR	ADCEN	ADCCS[2-0]			TRGTC2	TRGTC1	TRGPT	TRGPZ

ADCEN — ADC Enable bit
ADCEN=1 enables ADC.
ADCEN=0 puts ADC into power down mode. Please set CHSEL[2-0] of ADCCTLA to 7 while ADCEN set as 0.
When ADCEN is set from 0 to 1, the program needs to wait at least 20us to allow analog bias to stabilize to ensure ADC's proper functionality.

ADCCS[2-0] — ADC Clock Divider

ADCCS[2-0]	ADC CLOCK
0	SYSCLK
1	SYSCLK/2
2	SYSCLK/8
3	SYSCLK/12
4	SYSCLK/16
5	SYSCLK/20
6	SYSCLK/24
7	SYSCLK/32

ADCCLK should be set below 500K if ADC FS is less than 2V.
ADCCLK can be set above 500KHz up to 2MHz if ADC FS is higher than 2V and less than VDD-1V range.

TRGTC2 TC2 CC Event Trigger Enable
TRGTC1 TC1 CC Event Trigger Enable
TRGPT PWM Trigger Event Trigger Enable
TRGPZ PWM Zero Event Trigger Enable

ADCCFG1 (0x4000_E001) ADC Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	REFSEL	-	-	-	-	-	-	-
WR	REFSEL	-	-	-	-	-	-	-

REFSEL ADC Reference Select
REFSEL=0 ADC uses internal VDDC 1.5V as reference
REFSEL=1 ADC uses external ADCREF as reference. External ADCREF should in the range of 2V to VDD-1V.

ADCCTLA (0x4000_E004) ADC Control Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	AVG[1-0]		CHSEL[2-0]			SHEN	ADCINTE	BUSY
WR	AVG[1-0]		CHSEL[2-0]			SHEN	ADCINTE	CSTART

AVG[1-0] AVG[2-0] controls the hardware averaging logic of ADC readout. It is recommended the setting is changed only when ADC is stopped. If multiple channels are enabled, then each channel is averaged in sequence. The default is 00.

AVG1	AVG0	ADC Result
0	0	1 Times Average
0	1	2 Times Average
1	0	4 Times Average
1	1	8 Times Average

CHSEL[2-0] ADC Channel Select

CHSEL[2]	CHSEL[1]	CHSEL[0]	ADC Channel
0	0	0	CHA
0	0	1	CHB
0	1	0	Negative Temperature
0	1	1	PGA Output
1	0	0	Positive Temperature
1	0	1	1/5 VDD
1	1	0	Reserved

1	1	1	GND

SAR ADC's channel select cannot be the same as SD ADC to avoid interferences between two ADC.

SHEN — Sample and Hold Enable
SHEN=0 Pass Through
SHEN=1 2 ADCCLK

ADCINTE — ADC Interrupt Enable

BUSY — ADC Status
BUSY is set to 1 by hardware when ADC is in conversion.

CSTART — Software Start Conversion bit
Set this CSTART=1 to trigger an ADC conversion on selected channels. This bit is self-cleared when the conversion is done.

ADCCTLB (0x4000_E005) ADC Control Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ADCT2F	ADCT1F	ADCPTF	ADCPZF	ADCPCF	-	-	ADCIF
WR	-	-	-	-	-	-	-	ADCIF

ADCT2F — TC2 Trigger Completion Flag
ADCT2F is set by hardware if the completed conversion is triggered by TC2.

ADCT1F — TC1 Trigger Completion Flag
ADCT1F is set by hardware if the completed conversion is triggered by TC1.

ADCPTF — PWM Trigger Completion Flag
ADCPTF is set by hardware if the completed conversion is triggered by PWM trigger setting.

ADCPZF — PWM Zero Trigger Completion Flag
ADCPZF is set by hardware if the completed conversion is triggered by PWM zero.

ADCPCF — PWM Center Trigger Completion Flag
ADCPCF is set by hardware if the completed conversion is triggered by PWM Center.

ADCIF — ADC Conversion Completion Interrupt Flag bit
ADCIF is set by hardware when conversion complete. If ADC interrupt is enabled, this also generates an interrupt. This bit must be cleared by software writing 1. Clearing ADCIF also clears all flags.

ADCDAT (0x4000_E008) ADC Result Register RO (0xXX)

	31-12	11-0
RD	-	ADCDATA[11-0]
WR	-	

Please note, if ADC is in conversion and another start or trigger is initiated, the result is undefined. Typically, the new start and trigger is ignored.

25. 14-Bit Incremental SD ADC

The on-chip ADC is a 14-bit incremental delta sigma ADC using 2nd order modulator and 128 oversampling rates. The ADC uses internal 1.5V as reference with 0V to 1.0V as full scale. When enabled, the ADC consumes about 2mA of current running at 4MHz. The Block Diagram is shown below. Please note the SD ADC and SAR ADC share an input. This means for example, ADCA and ADCB from GPIO analog switches are connected together respectively. And ADCA and ADCB go to both SAR ADC and SD ADC. Therefore, software can decide which ADC to use. To prevent interference of SAR ADC and SD ADC, It is not recommended to use both ADC at the same time for same channel.

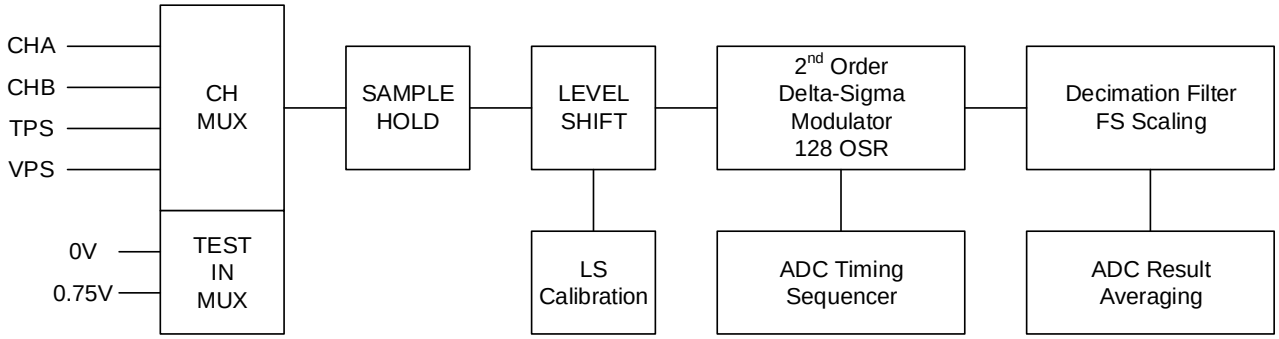


Figure 25-1 14 bit SD ADC Block Diagram

The ADC has four intrinsic channels. CHA and CHB are further connected to GPIO's analog I/O switches to expand multiplexed inputs. TPS is connected to internal temperature sensor within the band-gap reference with positive temperature coefficient. VPS is 1/6th of VDDH. When enabled, the ADC consumes about 1mA of current. The ADC also includes hardware to perform result average. Average can be set to 1 to 8 times. ADC conversion can be software triggered or by hardware triggered. Hardware trigger sources include TC1 and TC2 CC events, and PWM Center and Zero events.

ADC14CFG (0x4000_E100) ADC Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ADCEN	ADCCS[2-0]			TRGTC2	TRGTC1	TRGPT	TRGPZ
WR	ADCEN	ADCCS[2-0]			TRGTC2	TRGTC1	TRGPT	TRGPZ

ADCEN — ADC Enable bit

ADCEN=1 enables ADC.

ADCEN=0 puts ADC into power down mode. Please set CHSEL[2-0] of ADC14CTLA to 7 while ADCEN set as 0.

When ADCEN is set from 0 to 1, the program needs to wait at least 20us to allow analog bias to stabilize to ensure ADC's proper functionality.

ADCCS[2-0] — ADC Clock Divider

ADCCS[2-0]	ADC CLOCK
0	SYSCLK
1	SYSCLK/2
2	SYSCLK/8
3	SYSCLK/12
4	SYSCLK/16
5	SYSCLK/20
6	SYSCLK/24
7	SYSCLK/32

TRGTC2 — TC2 CC Event Trigger Enable

TRGTC1 — TC1 CC Event Trigger Enable

TRGPT — PWM Trigger Event Trigger Enable

TRGPZ — PWM Zero Event Trigger Enable

ADC14CTLA (0x4000_E104) ADC Control Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	AVG[1-0]		-	CHSEL[2-0]			ADCINTE	BUSY
WR	AVG[1-0]		-	CHSEL[2-0]			ADCINTE	CSTART

AVG[1-0] AVG[2-0] controls the hardware averaging logic of ADC readout. It is recommended the setting is changed only when ADC is stopped. If multiple channels are enabled, then each channel is averaged in sequence. The default is 00.

AVG1	AVG0	ADC Result
0	0	1 Times Average
0	1	2 Times Average
1	0	4 Times Average
1	1	8 Times Average

CHSEL[2-0] ADC Channel Select

CHSEL[2]	CHSEL[1]	CHSEL[0]	ADC Channel
0	0	0	CHA
0	0	1	CHB
0	1	0	Reserved
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Reserved
1	1	0	0.5*VDDC
1	1	1	GND

SD ADC's channel select cannot be the same as SAR ADC to avoid interferences between two ADC.

ADCINTE

BUSY ADC Status
BUSY is set to 1 by hardware when ADC is in conversion.

CSTART Software Start Conversion bit
Set this CSTART=1 to trigger an ADC conversion on selected channels. This bit is self-cleared when the conversion is done.

ADC14CTLB (0x4000_E105) ADC Control Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ADCT2F	ADCT1F	ADCPTF	ADCPZF	-	-	-	ADCIF
WR	-	-	-	-	-	-	-	ADCIF

ADCT2F TC2 Trigger Completion Flag
ADCT2F is set by hardware if the completed conversion is triggered by TC2.

ADCT1F TC1 Trigger Completion Flag
ADCT1F is set by hardware if the completed conversion is triggered by TC1.

ADCPTF PWM Trigger Completion Flag
ADCPTF is set by hardware if the completed conversion is triggered by PWM trigger setting.

ADCPZF PWM Zero Trigger Completion Flag
ADCPZF is set by hardware if the completed conversion is triggered by PWM zero.

ADCIF ADC Conversion Completion Interrupt Flag bit
ADCIF is set by hardware when conversion complete. If ADC interrupt is enabled, this also generates an interrupt. This bit must be cleared by software writing 1. Clearing ADCIF also clears all flags.

ADC14CTLC (0x4000_E106) ADC Control Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-		-	-	SHEN	SH[2-0]		
WR	-		-	-	SHEN	SH[2-0]		

SHEN — S/H Enable

SH[2-0] — S/H Duration Setting
000 = Pass Through
001 = 1 ADCCLK
010 = 2 ADCCLK
011 = 4 ADCCLK
100 = 8 ADCCLK
101 = 16 ADCCLK
110 = 32 ADCCLK
111 = 64 ADCCLK

ADC14CTLD (0x4000_E107) ADC Control Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	LSEN	LS[6-0]						
WR	LSEN	LS[6-0]						

LSEN — Level Shift Enable
LSEN=1 to enable the level shift circuit. This adds a 0.25V shift to the input to allow DS modulator in normal operation.

LS[6-0] — Level Shift Fine Trim
LS[6-0] is used to fine trim the level shift value to calibrate ADC output be 0x0000 when input is forced to 0V.

ADC14DAT (0x4000_E108) ADC Result Register RO (0xXX)

	31-14	13-0
RD	-	ADCDATA[13-0]
WR	-	

Please note, if ADC is in conversion and another start or trigger is initiated, the result is undefined. Typically, the new start and trigger is ignored.

26. Programmable Gain Amplifier

The programmable gain amplifier provides two gain settings of 5 or 10. It can serve as a differential to single-ended signal conversion especially for high-side voltage measurements. A R2R DAC is used to adjust the PGA output offset. The output of PGA is fed into ADC's input. The OP Amp used in PGA has PMOS differential input. Therefore, the common-mode input of PGAINN and PGAINP is limited to (VDD – 1.2V) to 0.5V for proper gain operations.

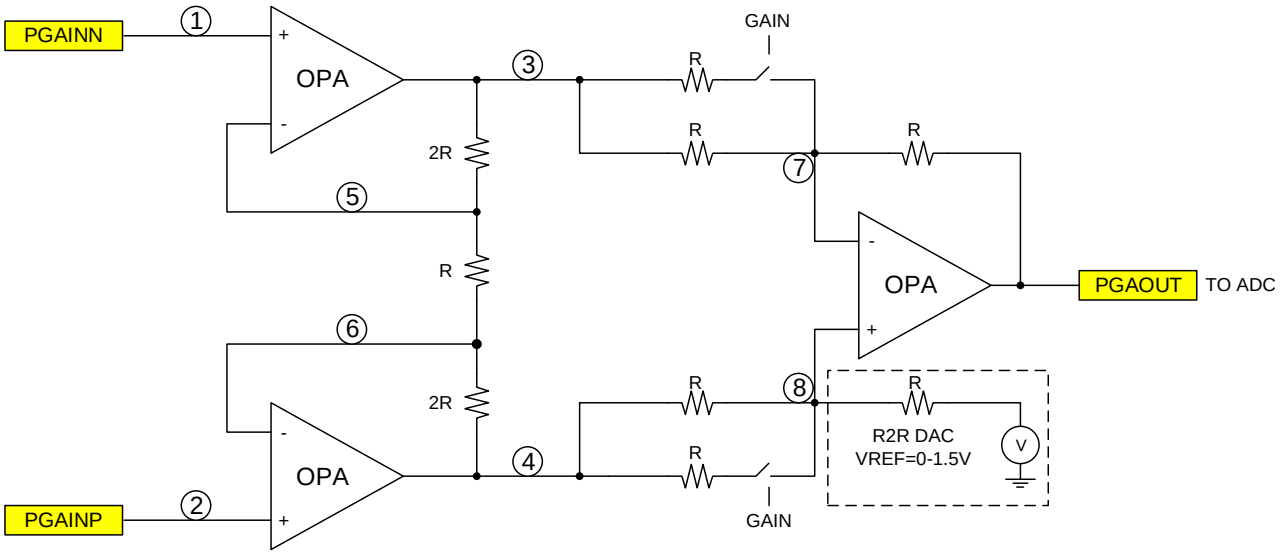


Figure 26-1 Programable Gain Amplifier

PGACFGA (0x4000_E220) PGA Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PGAEN	GAIN	-	-	INNSEL[1-0]		INPSEL[1-0]	
WR	PGAEN	GAIN	-	-	INNSEL[1-0]		INPSEL[1-0]	

PGAEN — PGA Enable bit

GAIN — Gain setting
GAIN=1 sets PGA gain to 10
GAIN=0 sets PGA gain to 5

INNSEL[1-0] — PGA input select
00 = PGAINN
01 = force to 0V
10 = force to VDDC (1.5V)
11 = force to VDD

INPSEL[1-0] — PGA input select
00 = PGAINP
01 = force to 0V
10 = force to VDDC (1.5V)
11 = force to VDD

PGACFGB (0x4000_E224) PGA Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	VRDAC[7-0]							
WR	VRDAC[7-0]							

VRDAC[7-0] — VREF for offset adjustment
0x00 to set the DAC to maximum (VDDC), 0xFF to set DAC to 0

27. Analog Comparator and DAC8

There are four analog comparators as its on-chip external peripherals. When enabled, each comparator consumes about 250uA. The input signal range is from 0V to VDD. There are two 8-bit R2R DACs associated with the comparators to generate the compare threshold. The R2R DAC uses the internal 1.5V supply as the full-scale range thus limiting the comparator threshold from 0V to 1.5V in 256 steps. Comparator A can select either VTH0 or VTH1 as the threshold. Comparator B/C/D can also select between VTH0 and external threshold. VTH1 is also sent to a unity gain buffer for use a DAC output. The buffer can supply or sink up to 150uA. Individual comparators, when enabled, consume about 80uA/each, and the unity gain buffer consumes about 400uA/800uA under 3V/5V supply conditions.

The CPU can read the real-time outputs of the comparator directly through register access. The output is also sent to an edge-detector and any edge transition can be used to trigger an interrupt. The stabilization time from off state to enabled state of the comparator block is about 20usec. The Block Diagram of the analog comparator is shown in the following diagram.

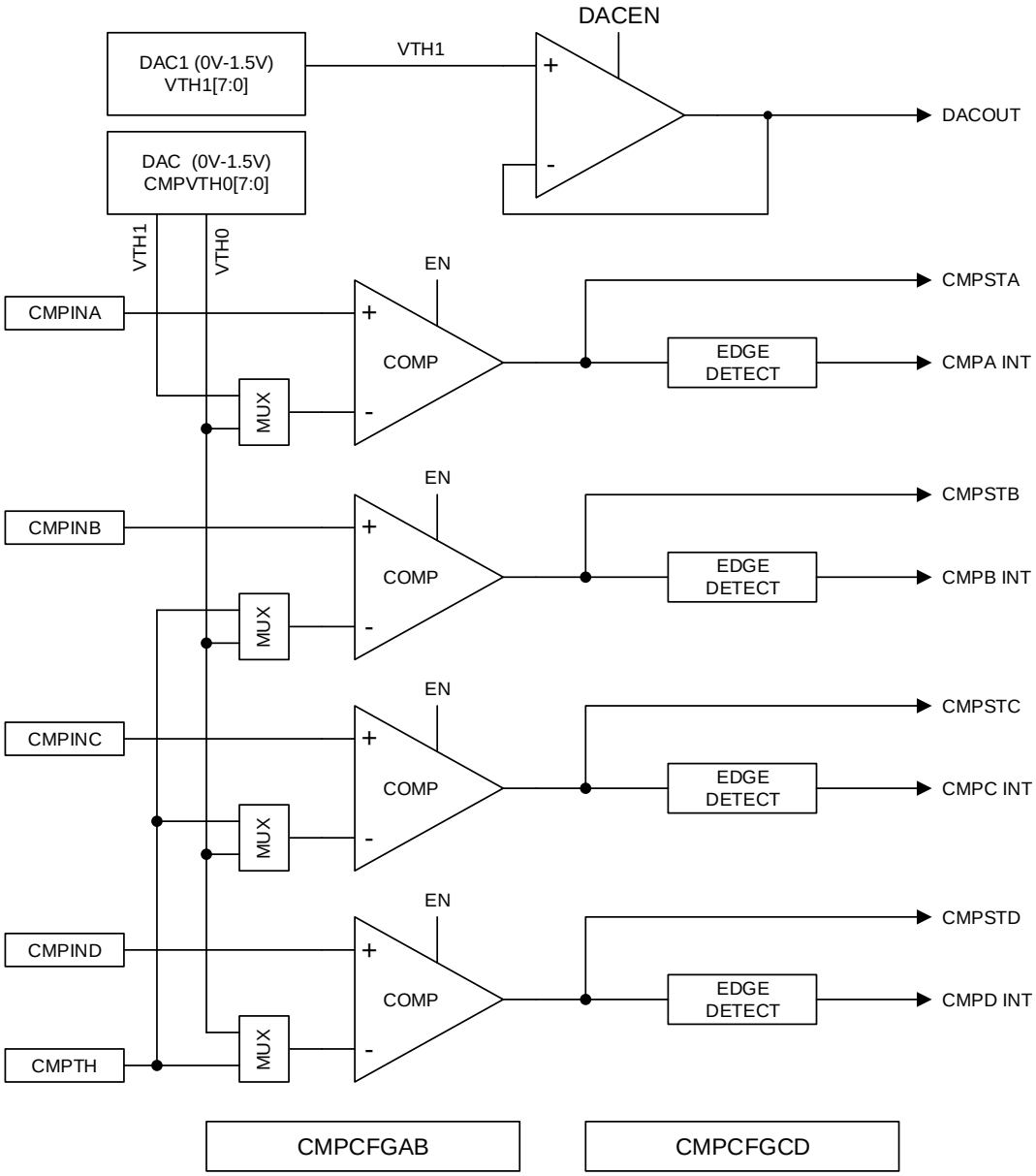


Figure 27-1 Comparator

CMPCFGAB (0x4000_D000) Analog Comparator A/B Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CMPENA	THSELA	INTENA	-	CMPENB	THSELB	INTENB	-
WR	CMPENA	THSELA	INTENA	-	CMPENB	THSELB	INTENB	-

CMPENA	Comparator A Enable bit. Set to enable the comparator. When CMPENA is set from 0 to 1, the program needs to wait at least 20us allowing analog bias to stabilize to ensure comparator A's proper functionality.
THSELA	Comparator A Threshold Select bit. THSELA = 0, the comparator A uses VTH0 as the threshold. THSELA = 1, the comparator A uses VTH1 as the threshold.
INTENA	Set to enable the comparator A's interrupt.
CMPENB	Comparator B Enable bit. Set to enable the comparator. When CMPENB is set from 0 to 1, the program needs to wait at least 20us allowing analog bias to stabilize to ensure comparator B's proper functionality.
THSELB	Comparator B Threshold Select Bit. THSELB = 0, the comparator B uses VTH0 as the threshold. THSELB = 1, the comparator B uses external threshold.
INTENB	Set to enable the comparator B's interrupt.

CMPCFGCD (0x4000_D001) Analog Comparator C/D Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CMPENC	THSELC	INTENC	-	CMPEND	THSELD	INTEND	-
WR	CMPENC	THSELC	INTENC	-	CMPEND	THSELD	INTEND	-

CMPENC	Comparator C Enable Bit. Set to enable the comparator. When CMPENC is set from 0 to 1, the program needs to wait at least 20us to allow analog bias to stabilize to ensure comparator C's proper functionality.
THSELC	Comparator C Threshold Select Bit. THSELC = 0, the comparator C uses VTH0 as the threshold. THSELC = 1, the comparator C uses external threshold.
INTENC	Set to enable the comparator C interrupt.
CMPEND	Comparator D Enable Bit. Set to enable the comparator. When CMPEND is set from 0 to 1, the program need to wait at least 20us to allow analog bias to stabilize to ensure comparator D's proper functionality.
THSELD	Comparator D Threshold Select Bit. THSELD = 0, the comparator D uses VTH0 as the threshold. THSELD = 1, the comparator D uses external threshold.
INTEND	Set to enable the comparator D interrupt.

CMPVTH0 (0x4000_D004) Analog Comparator Threshold Control Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	VTH0 Register							
WR	VTH0 Register							

CMPVTH0 register controls the comparator threshold VTH0 through 8-bit DAC. When set to 0x00h, the threshold is 0V. When set to 0xFFh, the threshold is at 1.5V (VDDC). When not used, it should be set to 0x00 to save power consumption.

CMPVTH1 (0x4000_D005) Analog Comparator Threshold Control Register R/W(0x00)

	7	6	5	4	3	2	1	0
RD	VTH1 Register							
WR	VTH1 Register							

CMPVTH1 register controls the comparator threshold VTH1 through 8-bit DAC. When set to 0x00h, the threshold is 0V. When set to 0xFFh, the threshold is at 1.5V (VDDC). When not used, it should be set to 0x00 to save power consumption.

CMPST (0x4000_D008) Analog Comparator Status Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CMPIFD	CMPIFC	CMPIFB	CMPIFA	CMPSTD	CMPSTC	CMPSTB	CMPSTA
WR	CMPIFD	CMPIFC	CMPIFB	CMPIFA	FILEND	FILENC	FILENB	FILENA

CMPIFD	Comparator D Interrupt Flag bit. This bit is set when CMPSTD is toggled and the comparator D setting is enabled. This bit must be cleared by software writing 1.
CMPIFC	Comparator C Interrupt Flag bit. This bit is set when CMPSTC is toggled and the comparator C setting is enabled. This bit must be cleared by software writing 1.

CMPIFB	Comparator B Interrupt Flag bit. This bit is set when CMPSTB is toggled and the comparator B setting is enabled. This bit must be cleared by software writing 1.
CMPIFA	Comparator A Interrupt Flag bit. This bit is set when CMPSTA is toggled and the comparator A setting is enabled. This bit must be cleared by software writing 1.
CMPSTD	Comparator D Real-time Output. If the comparator is disabled, this bit is forced low.
FILEND	Comparator D Digital Filter Enable. Filter is 16 SYSCLK.
CMPSTC	Comparator C Real-time Output. If the comparator is disabled, this bit is forced low.
FILENC	Comparator C Digital Filter Enable. Filter is 16 SYSCLK.
CMPSTB	Comparator B Real-time Output. If the comparator is disabled, this bit is forced low.
FILENB	Comparator B Digital Filter Enable. Filter is 16 SYSCLK.
CMPSTA	Comparator A Real-time Output. If the comparator is disabled, this bit is forced low.
FILENA	Comparator A Digital Filter Enable. Filter is 16 SYSCLK.

DACCFG (0x4000_D00C) Analog Comparator Status Register R/W(0x00)

	7	6	5	4	3	2	1	0
RD	DACEN	VDDCCMPA	DACTEST	-	CMPHYSD	CMPHYSC	CMPHYSB	CMPHYSA
WR	DACEN	VDDCCMPA	DACTEST	-	CMPHYSD	CMPHYSC	CMPHYSB	CMPHYSA

DACEN	DAC Enable DACEN=1 turns on the DAC output buffer. DACEN=0 turns off the output buffer
VDDCCMPA	Force CMPINA as VDDC. VDDCCMPA = 1, connect CMPINA to VDDC. This is for testing purpose only. By connecting VDDC to CMPINA and GPIO ANIO switch, VDDC is exposed on GPIO pin so testing and trimming of VDDC can be done.
DACTEST	DAC/ADC Test Mode DACTEST=1 connect DACOUT to ADC's CHB input internally. This needs software to perform DAC output and ADC conversion.
CMPHYSD	Comparator D Hysteresis Disable CMPHYSD = 0 disables the hysteresis of Comparator D CMPHYSD = 1 enables the hysteresis (typical 10mV) of Comparator D.
CMPHYSC	Comparator C Hysteresis Disable CMPHYSC = 0 disables the hysteresis of Comparator C CMPHYSC = 1 enables the hysteresis (typical 10mV) of Comparator C.
CMPHYSB	Comparator B Hysteresis Disable CMPHYSB = 0 disables the hysteresis of Comparator B CMPHYSB = 1 enables the hysteresis (typical 10mV) of Comparator B.
CMPHYSA	Comparator A Hysteresis Disable CMPHYSA = 0 disables the hysteresis of Comparator A CMPHYSA = 1 enables the hysteresis (typical 10mV) of Comparator A.

28. Touch Key Controller

TK3 is an enhanced TK2 implementation with differential dual slope operations. The capacitance to time conversion goes through two phases of charge transfer, one is charging up and one is discharging down using two thresholds equally spaced from ½ VDDC. Each charge transfer is obtained by subtraction of charge on internal reference capacitance and key capacitance. The difference of charge/discharge counting behavior is used to determine the key capacitance change in ratio of internal capacitance. Better noise immunity from power and ground noise and common-mode noise is achieved by dual slope operation. Better S/N can also be achieved because only differential charge is used for transfer, and the internal capacitance exhibits better temperature and environmental stability making the conversion result less sensitive to these changes.

CREF, the integration capacitor of the charge transfer, is connected to P10 through ANIO multiplexer and CKEY is connected to other GPIO through multiplexer. A replica signal of CKEY is provided through a buffer and routed out as SHIELD through GPIO. The shield signal can be used to cancel mutual capacitance effect from neighboring signal trace of the detected key and provides better noise immunity against moisture or water.

To detect a key press, the duty count value TKLDT[15-0] or TKHDT[15-0] can be processed by software and compare with an average non-press duty count. The hardware can also be configured to auto repeat accumulations of the duty cycle count to filter the sporadic noise effect. Since the comparator output should be a random duty with average equals to the capacitance ratio, for low frequency noise rejection, the hardware can be set to reject a continuous high or low comparator output that exceed long durations. For high frequency noise rejection, the hardware includes a pseudo-random sequence that randomizes the charge and discharge timing sequences. A slow moving average of the duty count value is stored in TKBASE[15-0] and software can use this for baseline calculation to auto compensate environment change.

Issuing a START command in the TK3CFGD register starts a conversion sequence that accumulates the comparator output into count value. The count value and the total number of the cycle of the sequence can then be calculated to obtain the capacitance of the key. The timing diagram of the TK3 in normal operation is shown in the following diagram. CREF is first equalized to VREFX that is in close range of VREF. When a START command is issued, first few edges of the comparator output is ignored to avoid any noise caused by the VREFX switching. And then the compactor output is accumulated into DTYL and DTYH registers. A sequence can consist of several conversion cycles depending on the RPT setting, and DTYL and DTYH maintains accumulation to obtain higher resolutions. After the sequence completed, CREF is also connected to VREFX to stay ready for next sequence to start.

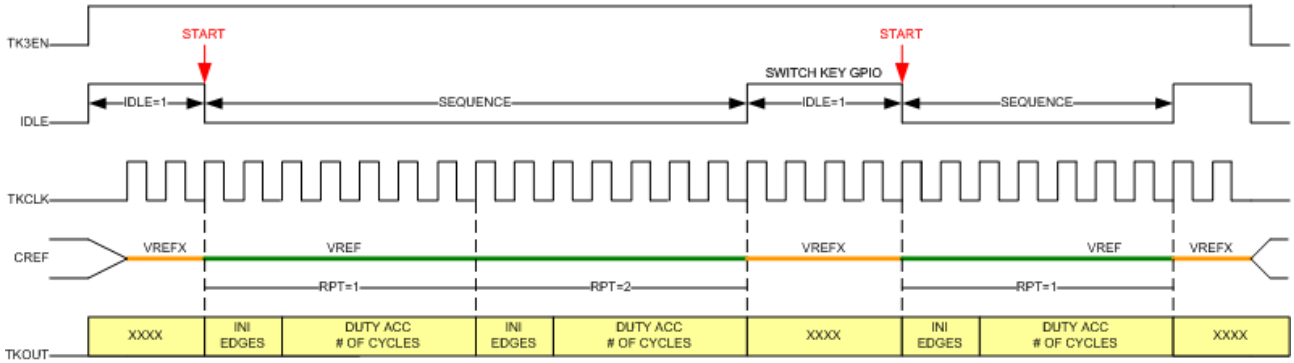


Figure 28-1 Touch Key Operation Sequence under Normal mode

TK3 can be set into low power auto detect mode by setting AUTO bit in TK3CFGA. In this mode, an ultra-low power comparator is used and the clock for TK3 should be set to SOSC (128KHz). This mode can be used specifically for touch key wakeup during the MCU sleep mode. The total power consumption of TK3 in this mode is less than 5uA. A threshold register can be set to determine the auto detect threshold either in absolute value or relative value versus the slow-moving baseline value. When the duty count value exceeds the threshold value, a wakeup and interrupt is generated to CPU. The timing diagram for auto mode detection and entering into SLEEP mode is shown in the following diagram. Note the actual start of the sequence is delayed by AUTO START DELAY setting. This allows the internal VDDC to stabilize from switching normal mode to sleep mode supply regulators.

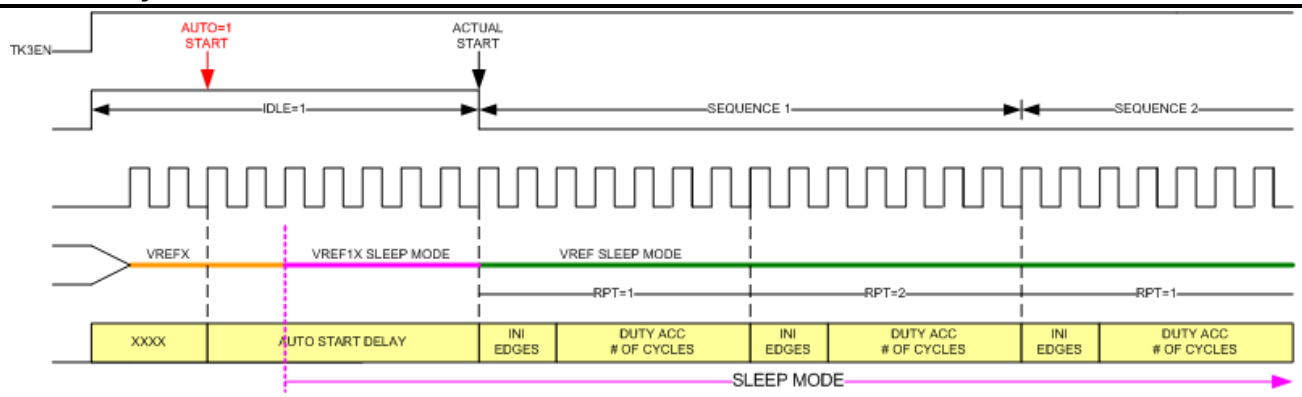


Figure 28-2 Touch Key Operation Sequence under Auto mode

TK3CFGA (0x4000_C000) TK3 Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TK3EN	CHOPEN	CMPHYS	REFSEL	SHIELDEN	TKIEN	TKLPM	AUTO
WR	TK3EN	CHOPEN	CMPHYS	REFSEL	SHIELDEN	TKIEN	TKLPM	AUTO

TK3EN — TK3 Enable
TK3EN=0 Disables the TK3 circuits and clear all states
TK3EN=1 for TK3 normal operations.

CHOPEN — Comparator Offset Cancellation Enable
CHOPEN=1 enables offset cancellation
CHOPEN=0 disables offset cancellation
Offset cancellation can reduce low frequency flicker noise.

CMPHYS — Comparator Hysteresis
CMPHYS=0 disables the comparator hysteresis.
CMPHYS=1 enables the comparator hysteresis.
Hysteresis helps immunity to external EMI disturbance, but it also increase the noise under normal condition.

REFSEL — Comparator Reference Select
REFSEL=0 uses ½ VDDC as the reference.
REFSEL=1 uses ¾ VDDC as the reference.
¾ VDDC has higher EMI immunity but with less dynamic range.

SHIELDEN — Shield Output Buffer Enable
SHIELDEN=1 enables the shield signal buffer. The buffer consumes about 200uA when enabled.

TKIEN — TK3 Interrupt Enable
TKIEN=1 enables the TK3 interrupt. TK3 interrupt is generated when a counting sequence is completed (including the repeat count if RPT[1-0] is not 00). Interrupt and wakeup is also generated when TKIEN=1 and AUTO=1 after auto detection threshold is met.
When TK3 interrupt is generated, TKIF is also set to 1 by hardware.

TKLPM — TK3 Low Power Mode
TKLPM=0 for normal mode operations.
TKLPM=1 put the comparator into ultra-low power mode and should be used in auto wakeup power saving mode. In this mode, TKCLK should use SOSC/2 slow clock.

AUTO — Auto Wake Up Mode
AUTO=1 enables auto detect mode. Writing START with “1” initiate continuous automatic key detect. In auto mode, the current duty count register value is compared with baseline plus threshold (either absolute or relative). If duty count value is higher, an interrupt and wakeup is generated.
AUTO=0 enable normal detect mode. In normal mode, writing START with “1” initiates a conversion sequence, and when the duty count is obtained, an interrupt is generated.

TK3CFGB (0x4000_C001) TK3 Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RPT[1-0]		INI[1-0]		ASTDLY[1-0]		LFNF[1-0]	
WR	RPT[1-0]		INI[1-0]		ASTDLY[1-0]		LFNF[1-0]	

RPT[1-0] Repeat Sequence Count
00 = No Repeat
01 = 4 times
10 = 8 times
11 = 16 times

INI[1-0] Discard Starting Comparator Edges
The number of initial comparator output edges is ignored when each sequence starts for better stable response. The number is (INI[1-0] + 1) *4. These edges are not included in the DTY count and the cycle count.

ASTDLY[1-0] Auto Mode Start Delay
STDLY[1-0] inserts an inter-sequence idle time of (ASTDLY[1-0]+1) * 256 TKCLK at each sequence start. Tis delay allows the stabilization time of VREFX from normal mode to sleep mode.

LFNF[1-0] Low Frequency Noise Filter Setting
00 = disables LFNF
Comparator output after staying continuous either “0” or “1” longer than LFNF[1-0]*8 is ignored in duty counter.

TK3CFGC (0x4000_C002) TK3 Configuration Registers C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SLOW[1-0]		CYCLE[2-0]			BASEINI	THDSEL	AUTOLFEN
WR	SLOW[1-0]		CYCLE[2-0]			BASEINI	THDSEL	AUTOLFEN

SLOW[1-0] Baseline Slow Moving Average setting
00 = 32 average
01 = 64 average
10 = 128 average
11 = 256 average
The duty value is averaged by SLOW[1-0] conversion and updated to BASELINE register through moving average.

CYCLE[2-0] Cycle Count of each conversion sequence
000 = 1024
001 = 2048
010 = 4096
011 = 8192
100 = 12288
101 = 16384
110 = 32768
111 = 65536
The cycle count is each sequence cycle count. And it is repeated if RPT is not 0.

BASEINI Baseline Initial Value
If BASEINI=1, then the first DTYL count after entering auto mode is loaded to BASELINE register as its initial value to start moving average.
If BASEINI=0, then the value written in BASELINE before entering auto mode is used as the initial value to start moving average.

THDSEL Threshold Value Setting
THDSEL=0 uses TKTHD[15-0] as the threshold to compare with DTYL to generate the interrupt and wakeup
THDSEL=1 uses TKTHD[15-0] + TKBASE[15-0] as the threshold.

AUTOLFEN Low Frequency Noise Filtering in Auto mode
If AUTOLFEN=0, then low frequency noise filtering in Auto mode is disabled.
If AUTOLFEN=1, then low frequency noise filtering in auto mode is enabled.

The low noise filtering status flag is still valid regardless of AUTOLFEN setting. Software can determine if to discard the current conversion result by checking LFNF flag.

TK3CFGD (0x4000_C003) TK3 Configuration Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	ASEN	TKCS[3-0]			
WR	-	-	-	ASEN	TKCS[3-0]			

ASEN — Asymmetric Clock Enable
ASEN=1 uses asymmetric clock for charge sharing
ASEN=0 uses symmetric clock for charge sharing
ASEN is only meaningful when SYSCLK is the clock source

TKCS[3-0] — TK3 Clock Select
TKCS[3-0]=0000 SYSCLK/4, ASEN controlled
TKCS[3-0]=0001 SYSCLK/8, ASEN controlled
TKCS[3-0]=0010 SYSCLK/16, ASEN controlled
TKCS[3-0]=0011 SYSCLK/20, ASEN controlled
TKCS[3-0]=0100 SYSCLK/24
TKCS[3-0]=0101 SYSCLK/32
TKCS[3-0]=0110 SYSCLK/64
TKCS[3-0]=0111 SYSCLK/128
TKCS[3-0]=Other Reserved
TKCS[3-0]=1110 SOSC128K/2 (64KHz)
TKCS[3-0]=1111 SOSC128K/4 (32KHz)
SOSC128K/2 should be used for sleep mode auto wakeup. Typical SOSC128K/2 is 64KHz.

TK3CRA (0x4000_C004) TK3 Control Registers A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CCHG[2-0]			ASTDLYEN	PSRDEN	LFNF	TKIF	BUSY
WR	CCHG[2-0]			ASTDLYEN	PSRDEN	LFNF	TKIF	START

CCHG[2-0] — Charge Capacitance Setting
CCHG[0] = 5pf
CCHG[1] = 10pF
CCHG[2] = 20pF
Charge capacitance is used to compensate the parasitic capacitance of the key. The setting can be key specific and should be set along with DC pull-up and CSEN[1-0] (Sensing Capacitance) to obtain a reasonable duty count in 25% to 80% of full count.

ASTDLYEN — Auto Start Delay Enable
ASTDLYEN=1 enables ASTDLY[1-0] delay start for auto mode.
ASTDLYEN=0 disables ASTDLY[1-0] delay.

PSRDEN — Pseudo Random Sequence Enable
PSRDEN=1 enables the random sequence in conversion
PSRDEN=0 disables

LFNF — Low Frequency Noise Detection Flag
LFNF is set by hardware if in the present conversion a Low Frequency Noise is detected. Software needs to write LFNF “1” to clear.

TKIF — TK3 Interrupt Flag
TKIF is set by hardware when a TK3 interrupt occurred by either conversion sequence completed or a valid detection in auto mode. Software needs to write TKIF “1” to clear.

BUSY — Conversion Status

START — Start Conversion
Writing “1” into START initiates the conversion sequence. It is cleared by hardware when conversion is complete. Please note writing AUTO “1” starts the conversion in auto mode.

BUSY is set to 1 by hardware indicating the conversion sequences are still running.

A DC pull-up can compensate the high equivalent pull-down current induced by large parasitic capacitance associated with sensing key due to long signal trace or size of the key. There are two means for providing DC pullup either through resistor or through current source. Typically, resistor option is used for normal mode because switch frequency is high and equivalent pulldown current is large. Current source can be used in auto detection because the switching frequency is low. DC pullup along with CCHG[2-0] should be used together for obtaining a duty count in reasonable range.

TK3CRB (0x4000_C005) TK3 Pull-Up Control Register H R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PUIEN	PUREN	CSEN[1-0]		PU[3-0]			
WR	PUIEN	PUREN	CSEN[1-0]		PU[3-0]			

PUIEN Pull-up DC Current Enable

PUREN Pull-up DC Resistor Enable

CSEN[1-0] Sensing Capacitance Setting

ΔC_{KEY}/CSEN is represented in the change of duty count. Therefore, the smaller CSEN is the higher the sensitivity of the touch key detection. An intrinsic CSEN of 5pF is always there. CSEN[1-0] adds CSEN to larger value to reduce the sensitivity in case of noisy environment.

CSEN[0] adds 5pF.

CSEN[1] adds 10pF.

The maximum CSEN is thus 20pF.

PU[3-0] Pull-Up Selection

For DC current, PU[3-0] enables 8uA/4uA/2uA/1uA current source.

For Resistor, PU[3-0] enables 5K/10K/20K/40K resistor.

To estimate the compensation effect using DC pull-up through resistors or current source, please use the following estimation. The equivalent pull-down current of $C_{KEY\ PARASITIC}$ is F_{TK} * $C_{KEY\ PARASITIC}$. The pull-up current using resistor is VDDC (1.5V) – VREF (either ¾ VDDC or ½ VDDC). For example, if $C_{KEY\ PARASITIC}$ is 100pF, then the equivalent current is 100uA for TCK of 1MHz. If VREF is selected as ¾ VDDC, the equivalent 3.75K is needed to compensate. This can be roughly achieved by putting 5K and 20K in parallel to form 4K.

TK3HDTYL (0x4000_C008) TK3 High Duty Count Register L RO (0x00)

	7	6	5	4	3	2	1	0
RD	TK3HDTY[7-0]							
WR	-							

TK3HDTYH (0x4000_C009) TK3 High Duty Count Register H RO (0x00)

	7	6	5	4	3	2	1	0
RD	TK3HDTY[15-8]							
WR	-							

TK3HDTY[15-0] TK Duty High value

TK3LDTYL (0x4000_C00C) TK3 Low Duty Count Register L RO (0x00)

	7	6	5	4	3	2	1	0
RD	TK3LDTY[7-0]							
WR	-							

TK3LDTYH (0x4000_C00D) TK3 Low Duty Count Register H RO (0x00)

	7	6	5	4	3	2	1	0
RD	TK3LDTY[15-8]							
WR	-							

TK3LDTY[15-0] TK Duty Low value

TK3BASEL (0x4000_C010) TK3 Baseline Register L R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TK3BASE[7-0]							
WR	TK3BASE[7-0]							

TK3BASEH (0x4000_C011) TK3 Baseline Register H R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TK3BASE[15-8]							
WR	TK3BASE[15-8]							

TK3BASE[15-0] TK Baseline value
When read, it is current baseline value
When write, it initialize the baseline with value written.

TK3THDL (0x4000_C014) TK3 Threshold Register L R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TK3THD[7-0]							
WR	TK3THD[7-0]							

TK3DTYH (0x4000_C015) TK3 Threshold Register H R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TK3THD[15-8]							
WR	TK3THD[15-8]							

TK3THD[15-0] TK Threshold value

29. Active Proximity Sensor

The active proximity sensor uses mutual capacitance sensing by driving a transmit electrode and sensing the electric field change at the receive electrode. This is shown as the following illustrations.

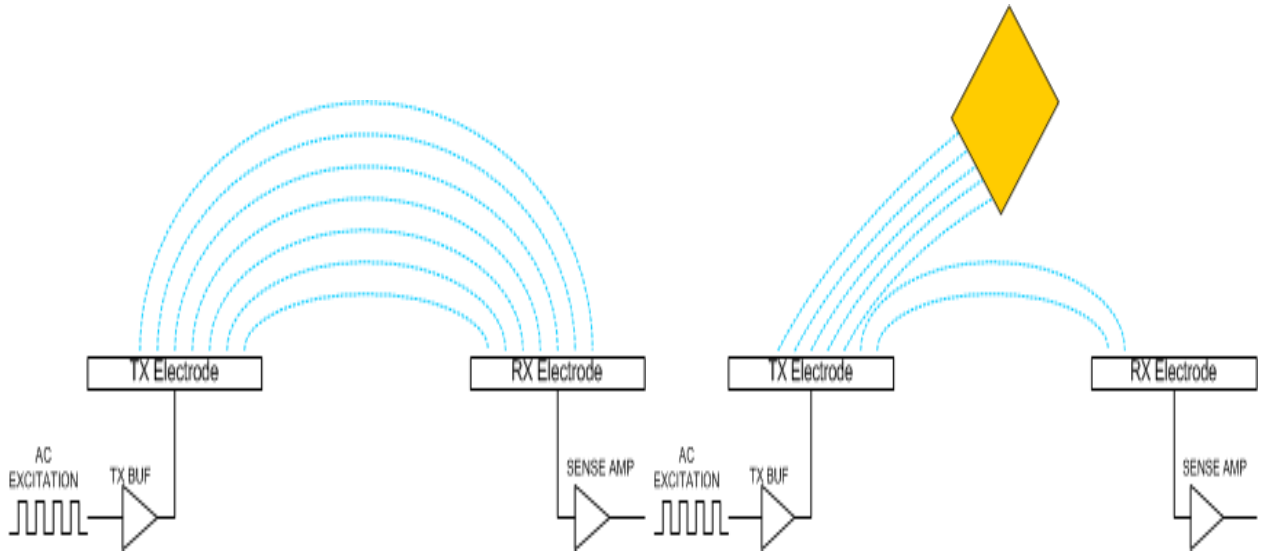


Figure 29-1 Active Proximity Sense

On the left, an AC excitation voltage is driving the TX electrode and leads to electric field established between the TX electrode and RX electrode. When a mass-conductive object such as finger approaches, the flux lines between the electrodes get disturbed. Using a charge sense amplifier, the change of flux lines can be amplified and thus accomplishes proximity sensing. In the diagram, we can see that if the distance between TX and RX electrode is farther, then the detection of proximity can be longer range. We can also see that larger amplitude of TX output can lead to easier proximity detection.

The proximity sensor is tightly coupled with Touch Key controller. It consists of an excitation waveform generator, and a synchronous charge amplifier followed by a programmable amplifier as the sense amplifier. The output of the sense amplifier is connected as an input to the Touch Key Controller and TK controller is used to detect the change of sense amplifier output as proximity detections. Typical excitation signal operates at frequency between 32KHz to 128KHz. Since PS is at the same clock domain with TK controller, setting TK clock will determine the excitation frequency. Typically, it should use 64KHz for TK clock.

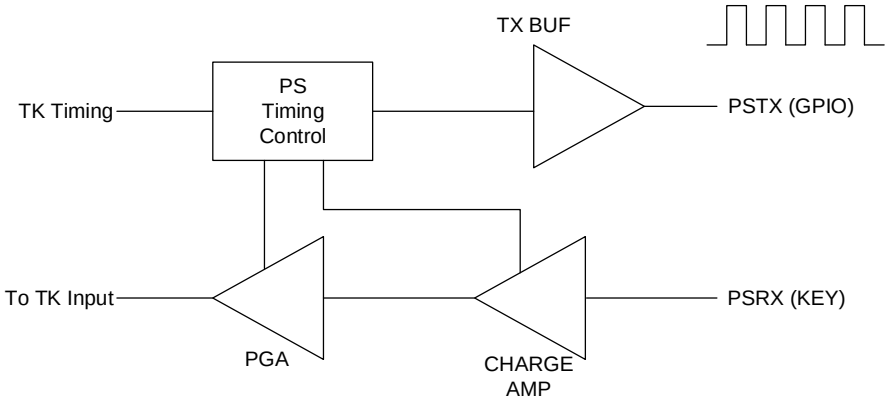


Figure 29-2 Active Proximity Sensing Block Diagram

Please note the output PSTX is routed to externally through multi-function select of the GPIO. Hence any GPIO pin can be used for PSTX purpose. The input PSRX share the ANIO multiplexer used for TK Key input therefore any pin can be configured as APS's input through ANIO1.

APSCFGA (0x4000_C100) Active Proximity Sensor Configuration Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	APSEN	RXCAL[6-0]						
WR	APSEN	RXCAL[6-0]						

APSEN — Active PS Enable
APSEN=1 enables the APS. If APS is enabled, the TK controller is connected to PS output.

RXCAL[6-0] — Receive Electrode Capacitance Calibration
RXCAL is used to adjust the cancelation of the parasitic capacitance on RX electrode. Each bit controls one of the binary weighted capacitance array.
RXCAL[0]=1, 32fF
RXCAL[1]=1, 64fF
RXCAL[2]=1, 128fF
RXCAL[3]=1, 256fF
RXCAL[4]=1, 512fF
RXCAL[5]=1, 1024fF
RXCAL[6]=1, 2048fF
The range is 32fF to 4pF.

APSCFGB (0x4000_C101) Active Proximity Sensor Charge Amplifier Configuration Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CREFSEL[3-0]				CAGAIN[3-0]			
WR	CREFSEL[3-0]				CAGAIN[3-0]			

CREFSEL[3-0] — Output Charge Capacitance Setting
This is equivalent of CCHG setting of TK controller.
Each bit of CREFSEL[3-0] selects a binary weighted capacitor array.
CREFSET[0]=1, 64fF
CREFSET[1]=1, 128fF
CREFSET[2]=1, 256fF
CREFSET[3]=1, 512fF
CREFSET[3-0]=0000 is not allowed.
Ideally, CREF should be set to between 400fF to 800fF.

CAGAIN[3-0] — Charge Amplifier Gain Setting
Charge Amplifier is always enabled when PS is enabled.
Each bit of CAGAIN[3-0] selects a binary weighted capacitor array for the feedback capacitor of the charge amplifier. The ratio of mutual capacitance between TX/RX electrodes and the feedback capacitor determines the gain of the charge amplifier. The smaller the feedback capacitance, the gain is higher however noise is higher too.
CAGAIN[0]=1, 64fF
CAGAIN[1]=1, 128fF
CAGAIN[2]=1, 256fF
CAGAIN[3]=1, 512fF
The range is from 64fF to 960fF.

APSCFGC (0x4000_C102) Active Proximity Sensor PGA Configuration Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PGAEN	PC[2-0]			PGASET[3-0]			
WR	PGAEN	PC[2-0]			PGASET[3-0]			

PGAEN — 2nd Stage PGA Enable

PC[2-0] — Power Control Setting
PC[2-0] sets the power consumption of the charge amplifier and PGA. Each bit turns on one of the binary weighted current sources. The higher the setting results higher power and faster speed when parasitic receive capacitance is high which requires faster settling time of the amplifiers.
PC[0]=1, 0.4uA
PC[1]=1, 0.8uA
PC[2]=1, 1.6uA
PC[2-0]=000 is not allowed.

PGASET[3-0] — Charge Amplifier Setting
GAIN = 8/(4*PGASET[3] + 2*PGASET[2] + PGASET[1] + PGASET[0])

Maximum gain is 8 when PGASET[3-0]=0001 or 0010.
Minimum gain is 1 when PGASET[3-0]=1111
PGASET[3-0]=0000 is not allowed.

APSCFGD (0x4000_C103) Active Proximity Sensor PGA Configuration Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TXPHASE	FRXCAL[2-0]			PSLOAD[3-0]			
WR	TXPHASE	FRXCAL[2-0]			PSLOAD[3-0]			

TXPHASE — APS TX Phase Select
TXPHASE=0 use normal phase as APS TX output. This is default.
TXPHASE=1 reverse the phase of APS TX output.

FRXCAL[2-0] — Receive Electrode Capacitance Calibration in fine step
FRXCAL[2-0] provides finer steps in RX capacitance calibration, Using FRXCAL[2-0] along with RXCAL[6-0] can make calibration easier. The additional capacitance provided by each bit of FRXCAL[2-0] is listed as following:
FRXCAL[0] = 8fF
FRXCAL[1] = 16fF
FRXVAL[2] = 32fF

PSLOAD[3-0] — Charge Amplifier Output Load Setting
PSLOAD[0] 100fF
PSLOAD[1] 200fF
PSLOAD[2] 400fF
PSLOAD[3] 800fF

30. GPIOs

Each GPIO port pin is configured by a 32-bit register which can be byte-accessed. The configuration includes its port data/flag manipulation, interrupt configuration, I/O drive configuration, and multi-function select.

GPIO00A (0x5000_2000) GPIO P00 Register A R/W (0x00)

	7	6	5	4	3	2	1	0
RD	RIF	FIF	-	IOERF	PINDB	PIN	-	POUT
WR	CLRRIF	CLRFIF	-	CLRIOERF	-	-	PS[1-0]	

RIF — Rising edge interrupt flag
CLRRIF — Clear RIF
Writing 1 to CLRRIF will clear RIF
FIF — Falling edge interrupt flag
CLRFIF — Clear FIF
Writing 1 to CLRFIF will clear FIF
IOERF — IO Compare error flag
CLRIOERF — Clear IOERF
Writing 1 to clear IOERF
PINDB — Debounced input state
PIN — Input state
POUT — Current registered output value
PS[1-0] — POUT Set/Clear
00 – unchanged
01 – set POUT=1
10 – clear POUT=0
11 – toggle POUT

GPIO00B (0x5000_2001) GPIO P00 Register B R/W (0x00)

	7	6	5	4	3	2	1	0
RD	INTREN	INTFEN	-	INTSEL	DBEN	FAST	IEN	IPOL
WR	INTREN	INTFEN	-	INTSEL	DBEN	FAST	IEN	IPOL

INTREN — Input Rising Edge Interrupt Enable (Debounced Input)
INTFEN — Input Falling Edge Interrupt Enable (Debounced Input)
INTSEL — Interrupt Select
0 - GPIO0 interrupt
1 - GPIO1 interrupt
DBEN — De-Bounce Enable
DBEN=1 enables de-bounce of 3 SOSC32K. This is automatically turned off during sleep/stop mode
DBEN=0 turns off de-bounce.
FAST — Output Buffer Speed Control
FAST=1 enables output buffer in fast mode. This allows up to 20MHz output. However, this will cause larger ground bounce thus creating more noise.
FAST=0 put output buffer in slow mode. This allows up to 8MHz.
IEN — Input Buffer Enable
INEN=1 enables the input buffer.
INEN=0 disables the input buffer.
In the disabled state, the output of input buffer is logic 0.
If input is floating or not solid 0 and 1 voltage level, DC current may flow in the input buffer. Disabling input buffer can remove DC leakage of input buffer due to this reason.
IPOL — Input Polarity
IPOL=1 reverse the input logic. IPOL=0 for normal logic polarity.

GPIO00C (0x5000_2002) GPIO P00 Register C R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PDRVEN	NDRVEN	IOCMPEN	OPOL	PUEN	PDEN	ANEN2	ANEN1
WR	PDRVEN	NDRVEN	IOCMPEN	OPOL	PUEN	PDEN	ANEN2	ANEN1

PDRVEN — Output PMOS driver enable. Set this bit to enable the PMOS of the output driver. DISABLE is the default value.

NDRVEN — Output NMOS driver enable. Set this bit to enable the NMOS of the output driver. DISABLE is the default value.

IOCMPEN — Input and Output Compare Enable
IOCMPEN=1 enables comparison of pad output state with input state. The compare status is sampled 2 SYSCLK after output change state. This can be used to detect output short or open. If the comparison is not matched, it generates an interrupt. The flag is shown in IOERF flag bit.

OPOL — Output Polarity Control
Output data polarity control.

PUEN — Pull up resistor control. Set this bit to enable pull-up resistor connection to the pin. The pull-up resistor is approximately 6K Ohm. DISABLE is the default value.

PDEN — Pull down resistor control. Set this bit to enable pull-down resistor connection to the pin. The pull-down resistor is approximately 6K Ohm. DISABLE is the default value

ANEN2 — Analog MUX 2 enables control. Set this bit to connect the pin to the internal analog peripheral. DISABLE is the default value.

ANEN1 — Analog MUX 1 enables control. Set this bit to connect the pin to the internal analog peripheral. DISABLE is the default value.

GPIO00D (0x5000_2003) GPIO P00 Register D R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	MFCFG[5-0]					
WR	-	-	MFCFG[5-0]					

MFCFG[5-0] defines the switch box of peripheral external connections to the GPIO port I/O.

MFCFG[5-0]	Function	Function Description
000000	LOW	This forces the output to logic low state.
000001	GPIO	GPIO port
000010	SCK	SPI1 SCK input or output depending on SPI MS setting.
000011	SDI	SPI1 SDI input corresponding to MI or SI depending on SPI MS setting.
000100	SDO	SPI1 SDO output corresponding to MO or SO depending on SPI MS setting.
000101	SSN	SPI1 SSN input or output depending on SPI MS setting.
000110	SSCL	I^2C Slave SCL I/O
000111	SSDA	I^2C Slave SDA I/O
001000	MSCL	I^2C Master SCL I/O
001001	MSDA	I^2C Master SDA I/O
001010	TX1	EUART1 TX output
001011	RX1	EUART1 RX input
001100	TX2	EUART2/LIN TX output
001101	RX2	EUART2/LIN RX input
001110	BZ/POW	Even port is Buzzer/Melody output. Odd port is POW output.
001111	XCLK	External system clock input
010000	IDX	QEC IDX Input
010001	PHA	QEC PHA Input

MFCFG[5-0]	Function	Function Description
010010	PHB	QEC PHB Input
010011	XCAPT2	TCC2 (Timer Compare/Capture) Capture Input
010100	TC2	TCC2 (Timer Compare/Capture) Terminal Count output
010101	CC2	TCC2 (Timer Compare/Capture) Compare Count output
010110	XCAPT1	TCC1 (Timer Compare/Capture) Capture Input
010111	TC1	TCC1 (Timer Compare/Capture) Terminal Count output
011000	CC1	TCC1 (Timer Compare/Capture) Compare Count output
011001	PWM01	PWM Channel 0/1. Even port is PWM0, odd port is PWM1.
011010	PWM23	PWM Channel 2/3. Even port is PWM2, odd port is PWM3.
011011	PWM45	PWM Channel 4/5. Even port is PWM4, odd port is PWM5.
011100	PWM67	PWM Channel 6/7. Even port is PWM6, odd port is PWM7.
011101	CLKO	Clock output
011110	PSTX	Proximity sensor TX output
011111	HIGH	This forces the output to logic high state.
1xxxxxx	Special Function	

Note:

1. Input for GPIO port, interrupt/wakeup is always enabled. For other functions, the inputs are multiplexed to the specific function blocks.

The following tables lists the GPIO configuration addresses.

GPIO Number	Register Address	GPIO Number	Register Address	GPIO Number	Register Address	GPIO Number	Register Address
GPIO00	5000_2000	GPIO20	5200_2040	GPIO40	5200_2080	GPIO60	5200_20C0
GPIO01	5000_2004	GPIO21	5200_2044	GPIO41	5200_2084	GPIO61	5200_20C4
GPIO02	5000_2008	GPIO22	5200_2048	GPIO42	5200_2088	GPIO62	5200_20C8
GPIO03	5000_200C	GPIO23	5200_204C	GPIO43	5200_208C	GPIO63	5200_20CC
GPIO04	5000_2010	GPIO24	5200_2050	GPIO44	5200_2090	GPIO64	5200_20D0
GPIO05	5000_2014	GPIO25	5200_2054	GPIO45	5200_2094	GPIO65	5200_20D4
GPIO06	5000_2018	GPIO26	5200_2058	GPIO46	5200_2098	GPIO66	5200_20D8
GPIO07	5000_201C	GPIO27	5200_205C	GPIO47	5200_209C	GPIO67	5200_20DC
GPIO10	5000_2020	GPIO30	5200_2060	GPIO50	5200_20A0	GPIO70	5200_20E0
GPIO11	5000_2024	GPIO31	5200_2064	GPIO51	5200_20A4	GPIO71	5200_20E4
GPIO12	5000_2028	GPIO32	5200_2068	GPIO52	5200_20A8	GPIO72	5200_20E8
GPIO13	5000_202C	GPIO33	5200_206C	GPIO53	5200_20AC	GPIO73	5200_20EC
GPIO14	5000_2030	GPIO34	5200_2070	GPIO54	5200_20B0	GPIO74	5200_20F0
GPIO15	5000_2034	GPIO35	5200_2074	GPIO55	5200_20B4	GPIO75	5200_20F4
GPIO16	5000_2038	GPIO36	5200_2078	GPIO56	5200_20B8	GPIO76	5200_20F8
GPIO17	5000_203C	GPIO37	5200_207C	GPIO57	5200_20BC	GPIO77	5200_20FC

The port information of GPIO is also grouped for 32-bit access for ease of access and data processing. The following register can be accessed as byte, word, and double word.

GP03OUT (0x5000_2100) GP00-GP07 ~ GP30-GP37 Output Register RW (0x00000000)

	31-0
RD	POUT[31-0]
WR	POUT[31-0]

POUT[31-0]

GP03IN (0x5000_2104) GP00-GP07 ~ GP30-GP37 Input Status Register RO (0x00000000)

	31-0
RD	PIN[31-0]
WR	-

PIN[31-0]

GP03INDB (0x5000_2108) GP00-GP07 ~ GP30-GP37 Debounced Input Status Register RO (0x00000000)

	31-0
RD	PINDB[31-0]
WR	-

PINDB[31-0]

GP03IF (0x5000_2110) GP00-GP07 ~ GP30-GP37 Edge Interrupt Flag Register RO (0x00000000)

	31-0
RD	RIF[31-0]+FIF[31-0]
WR	-

RIF[31-0]+FIF[31-0]

GP03ERF (0x5000_2124) GP00-GP07 ~ GP30-GP37 Compare Mismatch Flag Register RO (0x00000000)

	31-0
RD	IOERF[31-0]
WR	-

IOERF[31-0]

GP47OUT (0x5000_2200) GP40-GP47 ~ GP70-GP77 Output Register RW (0x00000000)

	31-0
RD	POUT[31-0]
WR	POUT[31-0]

POUT[31-0]

GP47IN (0x5000_2204) GP40-GP47 ~ GP70-GP77 Input Status Register RO (0x00000000)

	31-0
RD	PIN[31-0]
WR	-

PIN[31-0]

GP47INDB (0x5000_2208) GP40-GP47 ~ GP70-GP77 Debounced Input Status Register RO (0x00000000)

	31-0
RD	PINDB[31-0]
WR	-

PINDB[31-0]

GP47IF (0x5000_2210) GP40-GP47 ~ GP70-GP77 Edge Interrupt Flag Register RO (0x00000000)

	31-0
RD	RIF[31-0]+FIF[31-0]
WR	-

RIF[31-0]+FIF[31-0]

GP47ERF (0x5000_2224) GP40-GP47 ~ GP70-GP77 Compare Mismatch Flag Register RO (0x00000000)

	31-0
RD	IOERF[31-0]
WR	-

IOERF[31-0]

All GPIO can set as clock output. The clock output is enabled when MFCF[5-0]=0x1D. The output clock is defined by CLKOUT register.

CLKOUT (0x5000_2300) Clock Out Control Register RW (0x00000000)

	31-9	8	7-5	4-0
RD	-	CLKOEN	CLKSEL[2-0]	CLKDIV[4-0]
WR	-	CLKOEN	CLKSEL[2-0]	CLKDIV[4-0]

CLKOEN — Clock Out Enable
CLKOEN=0 will disable clock out function
CLKOEN=1 enables the divider

CLKSEL[2-0] — Clock Source Select
000 = SYSCLK
001 = IOSC
010 = SOSC32K
011 = PLL
100 = RTC
101 – 111 = Reserved and the output is zero

CLKDIV[4-0] — Clock Divider
The clock output is Clock Source divided by (CLKDIV[4-0]+1).

TESTSEL (0x5000_2304) Test Out Select Register RW (0x00000000)

	31-9	8	7-	4-0
RD	-	-	-	TESTSEL[4-0]
WR	-	-	-	TESTSEL[4-0]

TESTSEL[4-0] — TESTOUT Selection
TESTOUT is selected for special function for P00.

00011	LVD
00101	TBIT
00111	LINERESET
10001	IOSC/16
10111	SIOSC
10111	RTCCLK

31. Information Block (IFB)

There are four IFBs each is 1Kbyte size in 256 x 39 organization. The upper 7-bits are the ECC code, and lower 32-bits are data. IFB00 is for manufacturing, and IFB01, IFB02, IFB03 are for user.

31.1 IFB0 Address 0x1F00-0000~0x1F00-03FF

Word Address	Type	Description
00	M	IFB Version
01	M	Product Name
02	M	Package and Product Code
03	M	Product Version and Revision
04	M	Flash Memory Size
05	M	SRAM Size
06	M	Flash Segment Size
07	M	Flash Data Width
08-0E	M	Reserved
0F	M	Checksum for 0x00 – 0x0E
10	M	CP1 Information
11	M	CP2 Information
12	M	CP3 Version
13	M	CP3 BIN
14	M	FT Version
15	M	FT BIN
16	M	Last Test Date
17	M	Boot Code Version
18	M	Boot Code Segment
19	M	Customer Specific Code – Customer Name
1A	M	Customer Specific Code - Version
1B – 1E	M	Reserved
1F	M	Checksum for 0x10 – 0x1E
20	M	REGTRM value for 1.5V
21	M	OSC ITRM/VTRM value for 8MHz
22	M	OSC ITRM/VTRM value for 16MHz
23	M	OSC ITRM/VTRM value for 32MHz
24	M	SOSC TRM
25	M	PLL TRM
26	M	LVDTHD value for detection of 4.0V/3.0V/2.4V
27	M	ADC-12 Offset and Trim 1, voltage in 1mV (14-bit), ADC value in LSB (14-bit), offset[7-4]
28	M	ADC-12 Offset and Trim 2, voltage in 1mV (14-bit), ADC value in LSB (14-bit), offset[3-0]
29	M	Temperature Offset/Coefficient (Negative) [31-24] temperature coefficient LSB/K [23-16] FT temperature in Celsius [15-0] ADC measurement at FT

2A	M	Temperature Offset/Coefficient (Positive) [31-24] Ideal temperature coefficient LSB/K [23-16] FT temperature in Celsius [15-0] ADC measurement at FT
2B	M	Precision Timer Trim
2C	M	ADC-14 Offset and Trim 1
2D	M	ADC-14 Offset and Trim 2
2E	M	Reserved
2F	M	Checksum for 0x20 – 0x2E
30 – 3E	M	Reserved
3F	M	Checksum for 0x30 – 0x3E
40 – 4F	M	Retention Value 1
50 – 5F	M	Retention Value 2
60 – 6F	M	Endurance Value 1
70 – 7F	M	Endurance Value 2
80 – EF	M	Reserved
F0 – FB	M	Reserved
FC	M	MKEY0
FD	M	MKEY1
FE	M	MKEY2
FF	M	MKEY3

31.2 IFB1 Address 0x1FFF-0000~0x1FFF-03FF

Word Address	Type	Description
00 – EF	U	User programmable
F0	U	Wait time
F1	U	ISP Interface
F2 – FD	U	Reserved
FE	U	UKEY0
FF	U	UKEY1

31.3 IFB2 Address 0x1FFF-0400~0x1FFF-07FF

Word Address	Type	Description
00 – FF	U	User programmable

31.4 IFB3 Address 0x1FFF-0800~0x1FFF-0BFF

Word Address	Type	Description
00 – FF	U	User programmable

Note: The actual Byte address should be word address*4

32. ARM SWIO Controller

SWIO port is the standard ARM Cortex debug connection interface. SWIO can be used for debug as well as flash memory programming. CS9202 implementation is compatible with ARM standard ISP/debug environment. SWIO port consists of two pins, SWC and SWD. For CS9202, SWC maps to P13 and SWD maps to P14.

SWIO controller is clocked by the system clock (SYSCLK). The maximum SWC clock rate is limited to ¼ of SYSCLK. As results, if SYSCLK is set by a residing program to an abnormal frequency, or the CPU enters into sleep mode immediately after reset, SWIO controller will not function.

32.1 Connection by Line Reset

After any reset, an internal timer (default with 2 second time) enables SWIO controller to capture SWIO "line-reset". If the line-reset waveform is then the SWIO controller acquires control of the SWC and SWD pins and keep SWIO interface active. The line-reset waveform is at least 50 SWC clocks with SWD high and followed by at least 2 SWC clock with SWD low.

32.2 Read and Write Timing Waveform

SWIO allows the host to read and write the target register. The required timing waveform is compatible with ARM environment.

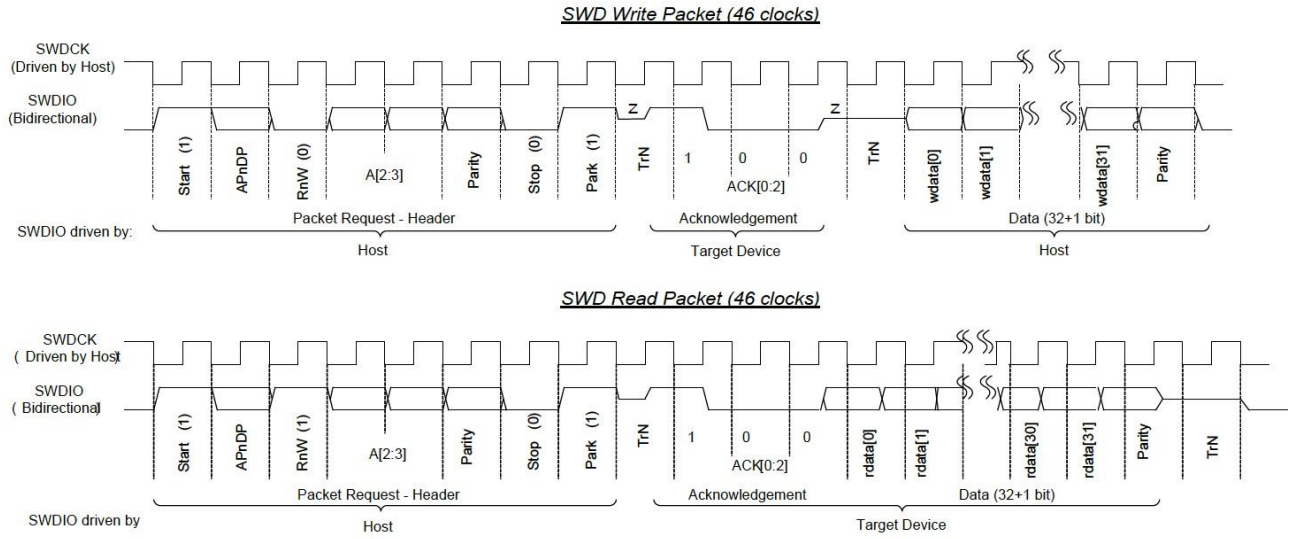


Figure 32-1 SWD Write/Read Packet (46 clocks)

32.3 DP Registers

A[3:2]:2'b00	ApnDp	RnW	DpBank	Name	Descriptions
0x0	0	1	x	DPIDR	Read DP Identification Register
0x0	0	0	x	ABORT	Write AP Abort Register
0x4	0	1	0	STAT	Read DP Status Register
0x4	0	0	0	CTRL	Write DP Control Register
0x4	0	1	1	DLCR	Read Data Link Control Register
0x4	0	0	1	DLCR	Write Data Link Control Register
0x8	0	1	x	-	Reserved
0x8	0	0	x	SELECT	Write AP Select Register
0xc	0	1	x	RDBUF	Read Buffer Register
0xc	0	0	x	-	Reserved

32.4 AP Registers

A[3:2]:2'b00	ApnDp	RnW	ApBank	Name	Descriptions
0x0	1	1	0	CSW	Read Control / Status Word

A[3:2]:2'b00	ApnDp	RnW	ApBank	Name	Descriptions
0x0	1	0	0	CSW	Write Control / Status Word
0x4	1	1	0	TAR	Read Transfer Address Register
0x4	1	0	0	TAR	Write Transfer Address Register
0x8	1	1	x	-	Reserved
0x8	1	0	x	-	Reserved
0xc	1	1	x	DRW	Read AP Data Register
0xc	1	0	x	DRW	Write AP Data Register

32.5 Typical Access

Typical access of SWIO interface from the host starts with the following steps to access the internal data and address bus of CS9202. Here TAR is used to specify the address and DRW is used for data buffer.

1. Line Reset
2. Read DPIDR; confirm access
3. Write TAR; set target address
4. Read/Write DRW; access data specified by TAR address

To perform e-Flash programming or debug, host needs to manipulate the Flash Controller register (FLSHCMD, FLSHCFG, FLSHDAT, etc.). The first step that needs to be performed is KEY unlock. When in locked state, TAR range is limited to 0x4000_0000 to 0x4FFF_FFFF, and only limited commands can be allowed in the Flash Controller as described in the e-Flash Controller sections

33. Special Modes

Lumissil 32-bit MCU family includes special mode functions. These cover modes such as test, debugging and e-Flash erase modes. The special modes are entered using waveform sequence on RSTN or SWC/JCK pins. The mode detection logic is illustrated as the following Block Diagram.

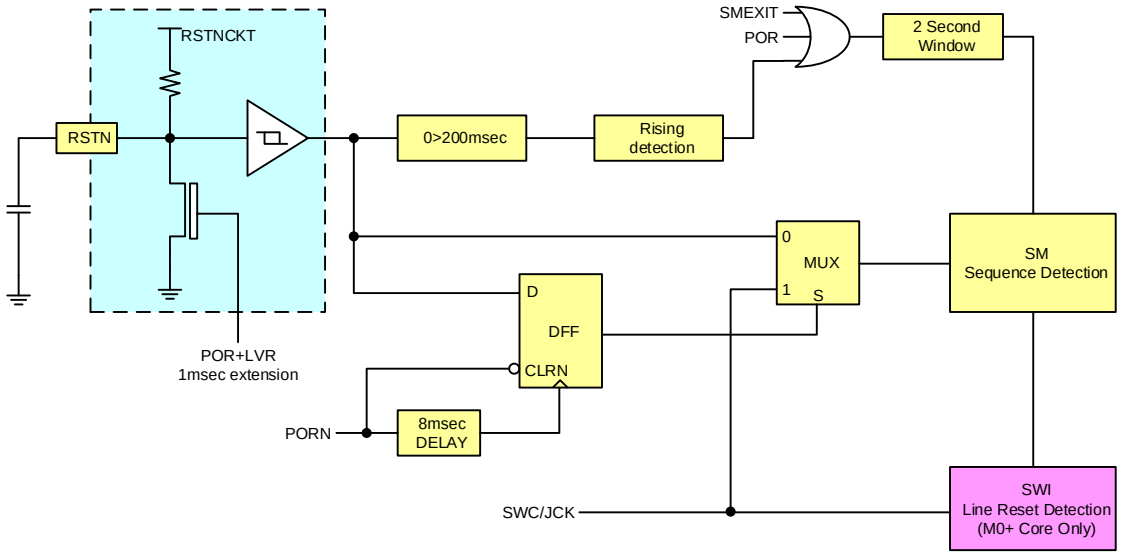


Figure 33-1 Special mode Block Diagram

*** Please DE update the Block Diagram

For special mode detection, first select which pin is used for sequence input. RSTN is sampled 8msec after power on. If RSTN is high, then SWC/JCK pin is used if RSTN is low then RSTN is used. For host using RSTN control, host should force RSTN low during power-on and release after 50msec.

Detection window of two seconds is opened after power-on, or RSTN=low longer than 200msec (host should pull RSTN low for 250msec to ensure proper detection), or SMEXIT is issued for exiting previous special mode.

Please also note that for M0+ MCU, the 2 second window is also used to allow M0+ SWIO debug controller for detecting initialization of debug interfacing using “line reset” signaling. After the initial connection of SWI interface is established, the debug mode with SWI interface is locked until exited.

The special mode sequence is built upon duty cycle bit encoding. The waveform follows a duty cycle encoded bit stream including synchronization bit and logic 1/0 bits. The coding mechanism is shown in the following diagram. Each bit time (BT) consists of 16 TQ with TQ range from 0.2msec to 2.0 msec. Using JLINK TQ can be set to 0.5msec minimum using 1msec steps.

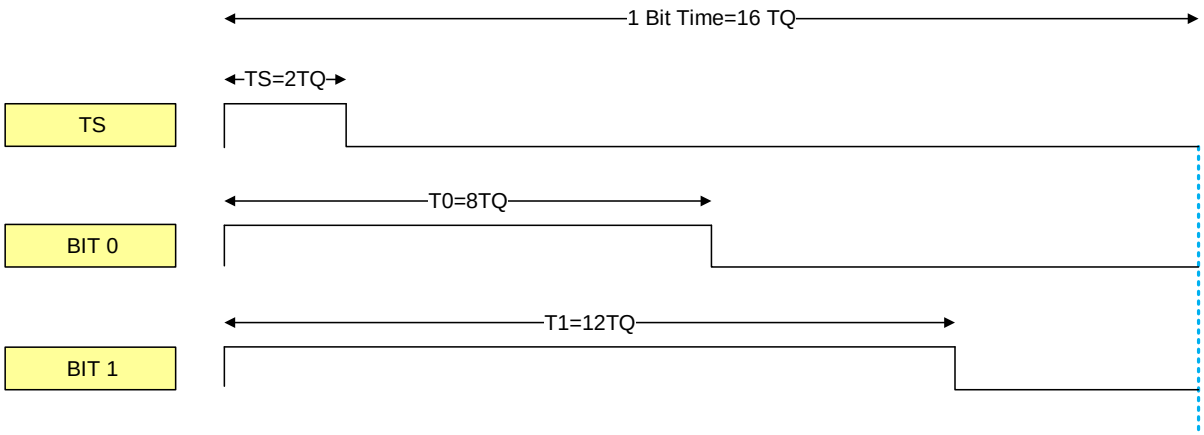


Figure 33-2 Special mode bit configuration

The Mode entry must follow the following waveform sequence and the time window for recognizing the sequence is limited to 2 seconds.

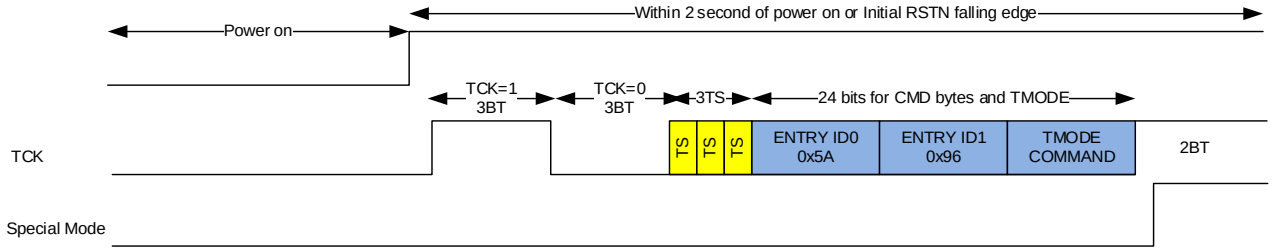


Figure 33-3 Enter special mode by Power on and TCK

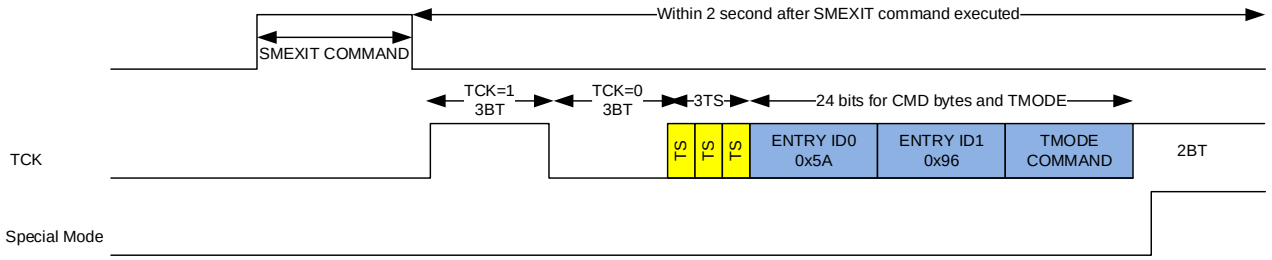


Figure 33-4 Enter special mode by SMEXIT command and TCK

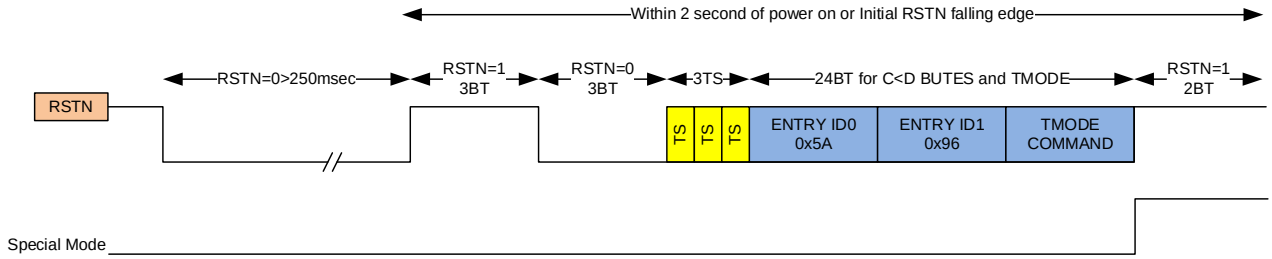


Figure 33-5 Enter special mode by RSTN

The following TMODE Command are allowed.

Function	TMODE[7-0]	Comments
Enable SWI interface	0x3C	This forces SWI interface pins configurations to allow debugging mode.
Enable CJTAG interface	0x4B	This forces CJTAG pins in correct IO configurations to allow CJTAG interface. To access debug, unlock procedure still needs to be carried out.
Enable JTAG interface	0x5A	This forces JTAG pins in correct IO configurations to allow JTAG interface. To access debug, unlock procedure still needs to be carried out.
Enable Scan Test	0x94	Scan test mode.
Force Erase Whole Chip	0xE5	This should be avoided as it also erases IFB0 and Boot Code
Force Erase MM and IFB1	0xE9	Restore the device to default blank state

Entry ID0 = 0x5A and Entry ID1=0x96. This should be confidential. (msb send first in the bit shift sequence)

To exit the special mode,

1. Power on/off.

2. Hold RSTN=0 for > 250msec then release RSTN=1.

3. Writing to the special mode exit register (SMEXIT 0x5000_0500) with data 0x55AA9669 followed by 0xAA556996. This clears the special mode status and performs an equivalent system reset, then opens 2 second of special mode detection window. Please note SMEXIT is not accessible from CPU and only accessible by debug interface when debug mode is on.

34. In System Programming (ISP)

ISP is accomplished by preprogrammed boot code. In default state (IFB1 F0=F1=FF), boot code will scan Lumibus, EUART1, EUART2/LIN, and I2CS for presence of ISP protocol within 6 seconds of power-on.

After user code is programmed, user code can also program IFB0 F0 and F1 for specifying specific power-on wait time and specific ISP interface. The boot code exits to user code after the wait time expires.

If a user code disables all the ISP interfaces, and puts wait time as 0, only way to restore the chip is to perform special mode whole chip erase operation.

35. Electrical Characteristics

35.1 Absolute Maximum Ratings

Symbol	Parameter	Rating	Unit	Note
VDD	Supply Voltage	5.5	V	
T_A	Ambient Operating Temperature	-40 – 85	°C	For IS31CS9202
		-40 – 125	°C	For IS32CS9202
T_{STG}	Storage Temperature	-65 – 150	°C	

35.2 Recommended Operating Condition

Symbol	Parameter	Rating	Unit	Note
VDD	Supply Voltage for IO and 1.5V regulator	2.5 – 5.5	V	
T_A	Ambient Operating Temperature	-40 – 85	°C	For IS31CS9202
		-40 – 125	°C	For IS32CS9202

35.3 DC Electrical Characteristics

VDD =2.5V to 5.5V, T_A=-40°C to 85°C for IS31CS9202, T_A=-40°C to 125°C for IS32CS9202

Symbol	Parameter	MIN	TYP	MAX	Unit	Note
Power Supply Current						
IDD_{16MHz}	Total IDD through VDD at 16MHz	-	3.5	-	mA	All peripherals off
IDD_{1MHz}	Total IDD through VDD at 1MHz	-	1.0	-	mA	All peripherals off
IDD versus Frequency	IDD Core Current versus Frequency	-	150	-	uA/ MHz	
$IDD_{Light\ SLP}$	IDD, light sleep mode	-	500	-	µA	Main regulator on Clocks off
$IDD_{Deep\ SLP}$	IDD, deep sleep mode, 25°C	-	2.5	5	µA	Main regulator off
	IDD, deep sleep mode, 85°C	-	4	10	µA	Main regulator off
	IDD, deep sleep mode, 125°C	-	25	40	µA	Main regulator off
RSTN Reset						
$V_{IH,\ RSTN}$	Input High Voltage	0.7	-	-	VDD	Scale with VDD
$V_{IL,\ RSTN}$	Input Low Voltage,	-	-	0.2	VDD	Scale with VDD
$V_{HYS,\ RSTN}$	RSTN Hysteresis	-	0.1	-	VDD	Scale with VDD
$R_{PU,\ RSTN}$	RSTN Pull-Up resistance	-	5K	-	Ohm	
GPIO DC Characteristics						
V_{OH}	Output High Voltage at 1mA, VDD=5V	-	-0.2	-0.5	V	Reference to VDD
	Output High Voltage at 1mA, VDD=3.3V	-	-0.3	-0.6	V	Reference to VDD
V_{OL}	Output Low Voltage at 8 mA, VDD=5V	-	0.3	0.5	V	Reference to VSS
	Output Low Voltage at 8 mA, VDD=3.3V	-	0.3	0.6	V	Reference to VSS
$I_{IO,\ TOTAL}$	Total IO Sink and Source Current	-100	-	100	mA	Observe this to avoid excessive ground or supply drops
V_{IH}	Input High Voltage	0.75	-	-	VDD	
V_{IL}	Input Low Voltage	-	-	0.25	VDD	
$V_{HYS,\ IN}$	Input Hysteresis	-	1	-	-	
$R_{PU,\ IO}$	Equivalent Pull-Up resistance	-	25K	-	Ohm	
$R_{PD,\ IO}$	Equivalent Pull-Down Resistance	-	25K	-	Ohm	

Symbol	Parameter	MIN	TYP	MAX	Unit	Note
REQ_{AN1}	Equivalent ANIO1 Switch Resistance VDD=3.3V and ½ VDD common mode	-	110	-	Ohm	
	Equivalent ANIO1 Switch Resistance VDD=5.0V and ½ VDD common mode	-	100	-	Ohm	ANIO1 Switch
REQ_{AN2}	Equivalent ANIO2 Switch Resistance @3.3V	-	450	-	Ohm	ANIO2 Switch
	Equivalent ANIO Switch Resistance @5V	-	350	-	Ohm	ANIO2 Switch
VDDC Characteristics						
$VDDC_{NRM}$	Core Voltage normal mode after calibration	1.45	1.5	1.55	V	Normal Mode
$VDDC_{SLP}$	Core voltage sleep mode	-	1.40	-	V	Sleep Mode
Low Supply (VDD) Voltage Detection						
VDET	Detection range	2.0	-	4.8	V	
VDETHYS	Detection hysteresis	-	20	-	mV	
SARADC12 Characteristics						
ADC_{LIN}	ADC linearity, Center range	-	+/- 2	-	LSB	
	ADC linearity, 0.2V to FS-0.2V	-	+/- 4	-	LSB	
ADCFQ	ADC frequency	-	0.5	1	MHz	
ADCRNG	ADC input range	0	-	VDD-2	V	
ADCCYC	ADC conversion cycle count	-	14	-	-	
SDADC14 Characteristics						
ADC_{LIN}	ADC linearity	-	+/-3	-	LSB	
ADCFQ	ADC frequency	-	2	4	MHz	
ADCRNG	ADC input range	-	1	-	V	
ADCCYC	ADC conversion cycle count	-	132	-	Cycle	

21.4 AC Electrical Characteristics

VDD =2.5V to 5.5V, T_A=-40°C to 85°C for IS31CS9202, T_A=-40°C to 125°C for IS32CS9202.

Typical values are given for 5V/25°C, unless otherwise specified.

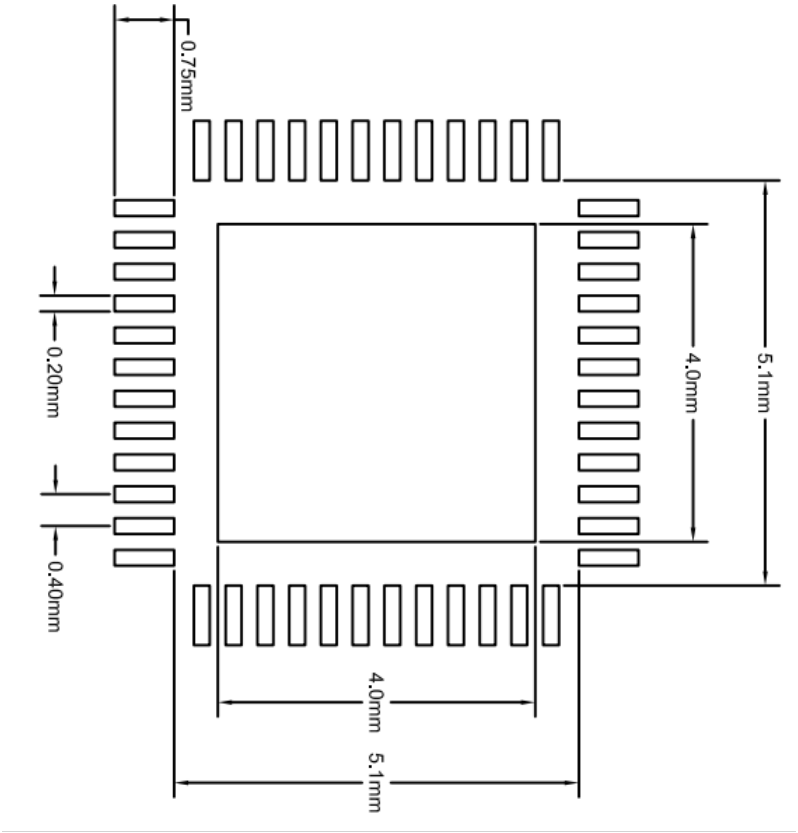
Symbol	Parameter	MIN	TYP	MAX	Unit	Note
Supply Timing						
T_{SUPRU}	VDD ramp up time	1	-	100	msec	WST = 0 for 16MHz
T_{SUPRD}	VDD ramp down time	-	-	500	msec	LVR should be on
T_{POR}	Power on reset delay	-	5	-	msec	
T_{STUP}	System startup delay	-	2	-	msec	After POR or RSTN
System Clock						
F_{SYS}	System Clock Frequency	-	16	50	MHz	
IOSC						
F_{IOSC}	IOSC Calibrated 16MHz/32MHz	-1	0	+1	%	
	IOSC startup time	-	-	10	µsec	
	Temperature and VDD variation 85°C	-2	0	+2	%	
	Temperature and VDD variation 125°C	-3	0	+3	%	

Symbol	Parameter	MIN	TYP	MAX	Unit	Note
SOSC						
F_{SOSC}	Slow oscillator frequency	-	128	-	KHz	
	Slow oscillator frequency accuracy	-	+/-30	-	%	
XOSC						
F_{XOSC}	Crystal oscillator frequency	5	16	25	MHz	
	Crystal oscillator startup time	-	2	-	msec	
PLL						
F_{PLL}	PLL frequency range	40	-	100	MHz	
$T_{STUP, PLL}$	PLL stabilization time	-	100	-	µsec	
IO Timing						
$T_{PD3\ ++}$	Propagation Delay 3.3V No load	-	15	-	nsec	
	Propagation Delay 3.3V 25pF load	-	20	-	nsec	
	Propagation Delay 3.3V 50pF load	-	25	-	nsec	
$T_{PD3\ --}$	Propagation Delay 3.3V No load	-	15	-	nsec	
	Propagation Delay 3.3V 25pF load	-	20	-	nsec	
	Propagation Delay 3.3V 50pF load	-	25	-	nsec	
TPD5 ++	Propagation Delay 5.0V No load	-	12	-	nsec	
TPD5 ++	Propagation Delay 5.0V 25pF load	-	15	-	nsec	
TPD5 ++	Propagation Delay 5.0V 50pF load	-	22	-	nsec	
TPD5 --	Propagation Delay 5.0V No load	-	12	-	nsec	
TPD5 --	Propagation Delay 5.0V 25pF load	-	15	-	nsec	
TPD5 --	Propagation Delay 5.0V 50pF load	-	22	-	nsec	
Flash Memory Timing						
TEMAC	Embedded Flash Access Time	-	40	45	ns	TWAIT must be larger than TEMAC
TEMWR	Embedded Flash Write Time	-	20	25	µsec	
TEMSER	Embedded Flash Sector Erase Time	-	2	2.5	msec	
TEMMER	Embedded Flash Mass Erase Time	-	10	12	msec	

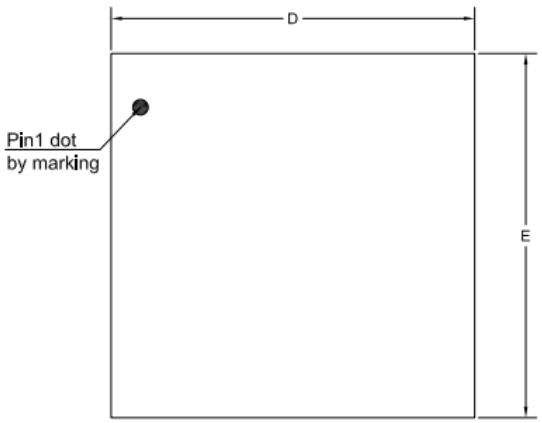
36. Packaging Outline

36.1 48-pin QFN

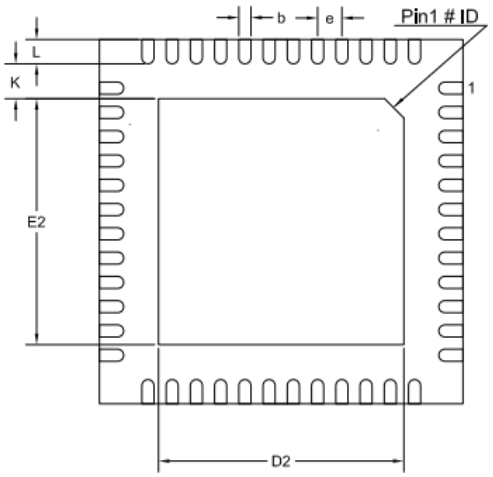
36.1.1 Recommended Land Pattern



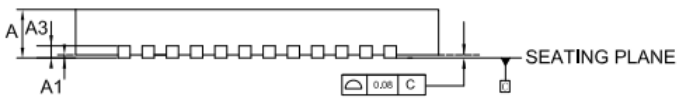
36.1.2POD



TOP VIEW



BOTTOM VIEW



SIDE VIEW

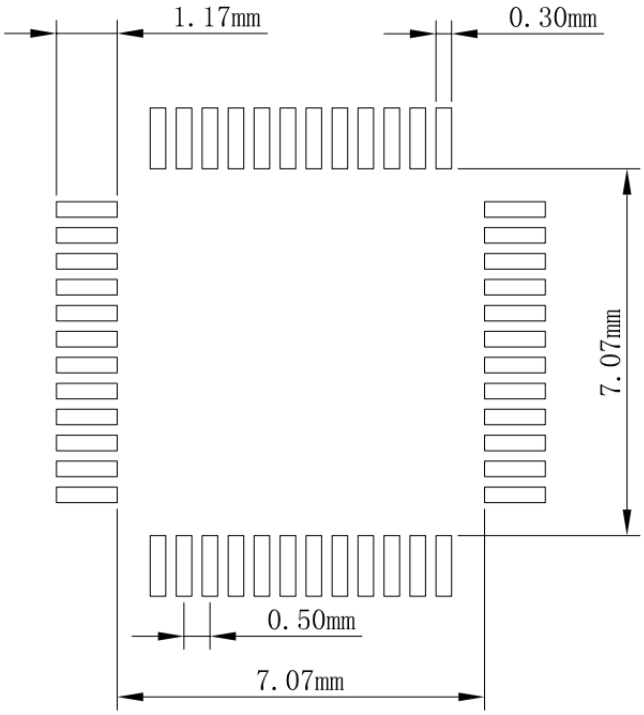
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0	0.02	0.05
A3	0.18	0.20	0.23
K	0.20	-	-
b	0.15	0.20	0.25
D	5.90	6.00	6.10
E	5.90	6.00	6.10
D2	3.95	-	4.65
E2	3.95	-	4.65
L	0.30	0.40	0.50
e	0.40BSC		

Notes:

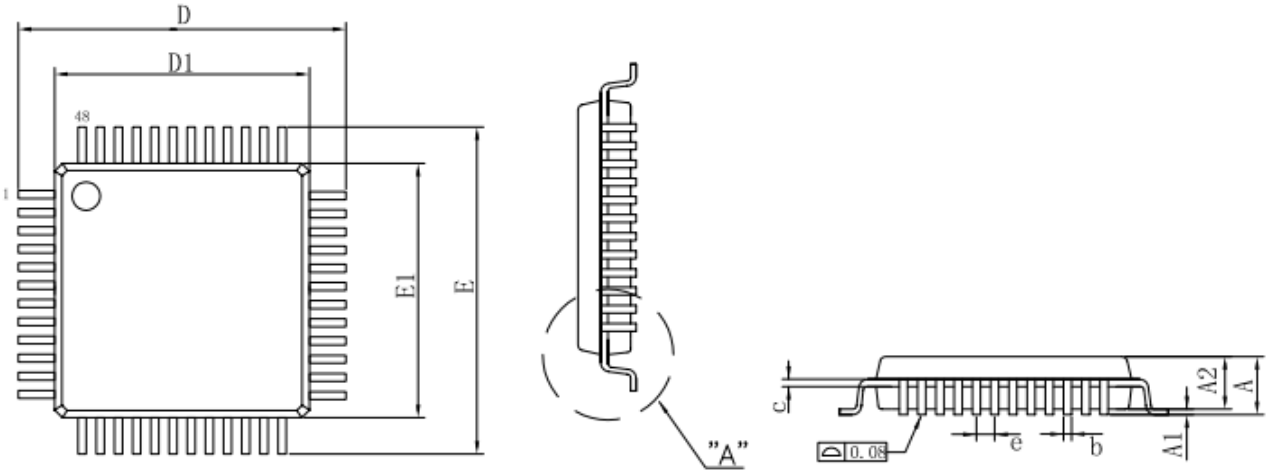
1. Controlling Dimension: MM
2. Reference Document: JEDEC MO-220
3. The Pin’s Sharp and Thermal Pad Shows Different Shape Among Different Factories.

36.2 48-pin LQFP

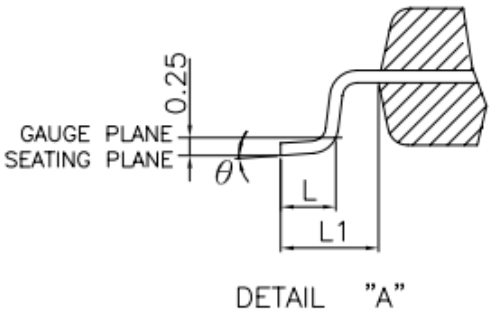
36.2.1 Recommended Landing Pattern



36.2.2POD



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
D	9.0BSC		
E	9.0BSC		
E1	7.0BSC		
D1	7.0BSC		
L1	1.0REF		
L	0.45	0.60	0.75
c	0.09	—	0.20
e	0.50BSC		
b	0.17	0.22	0.27
θ	0°	3.5°	7°

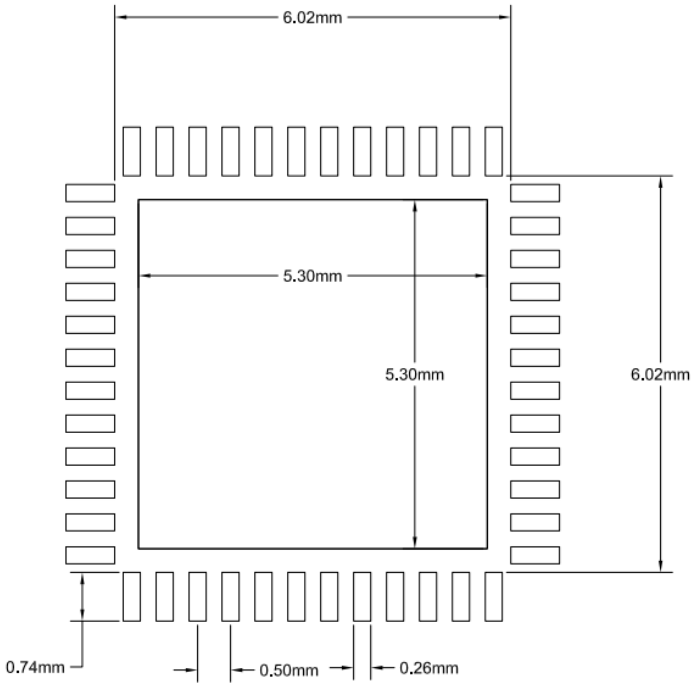


Notes:

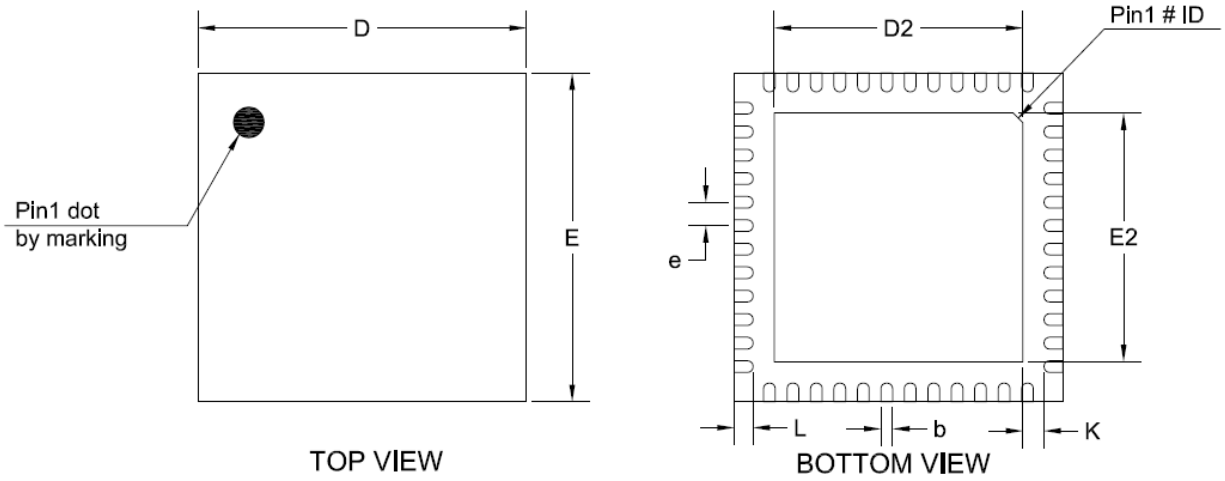
1. Controlling Dimension: MM
2. Reference Document: JEDEC MO-026

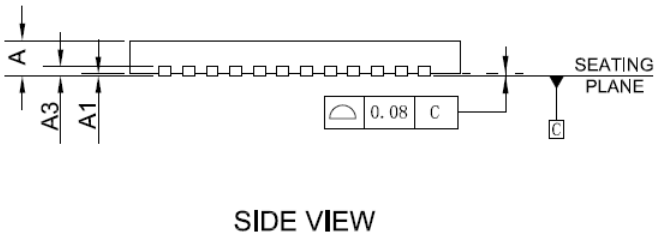
36.3 48-pin WFQFN

36.3.1 Recommended Land Pattern



36.3.2POD





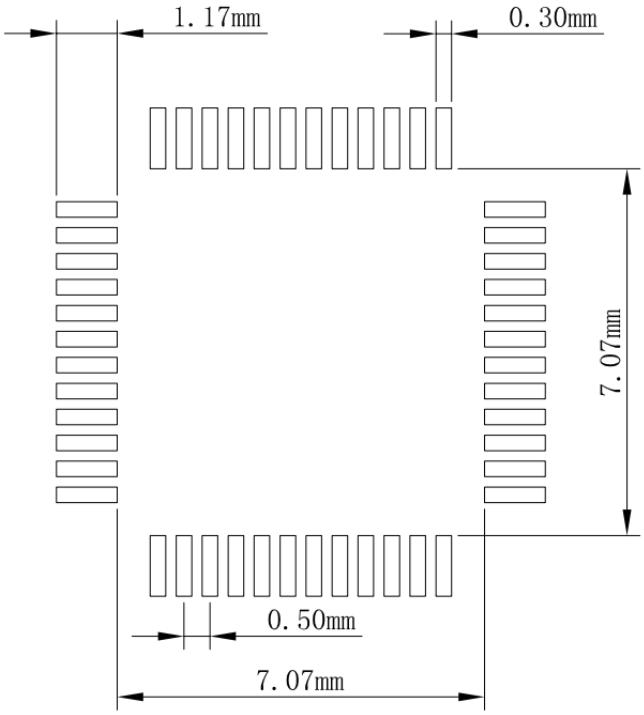
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	—	0.02	0.05
A3	0.203REF		
D	6.90	7.00	7.10
E	6.90	7.00	7.10
E2	5.20	5.30	5.40
D2	5.20	5.30	5.40
b	0.18	0.25	0.30
L	0.35	0.40	0.45
K	0.20	—	—
e	0.50BSC		

Notes:

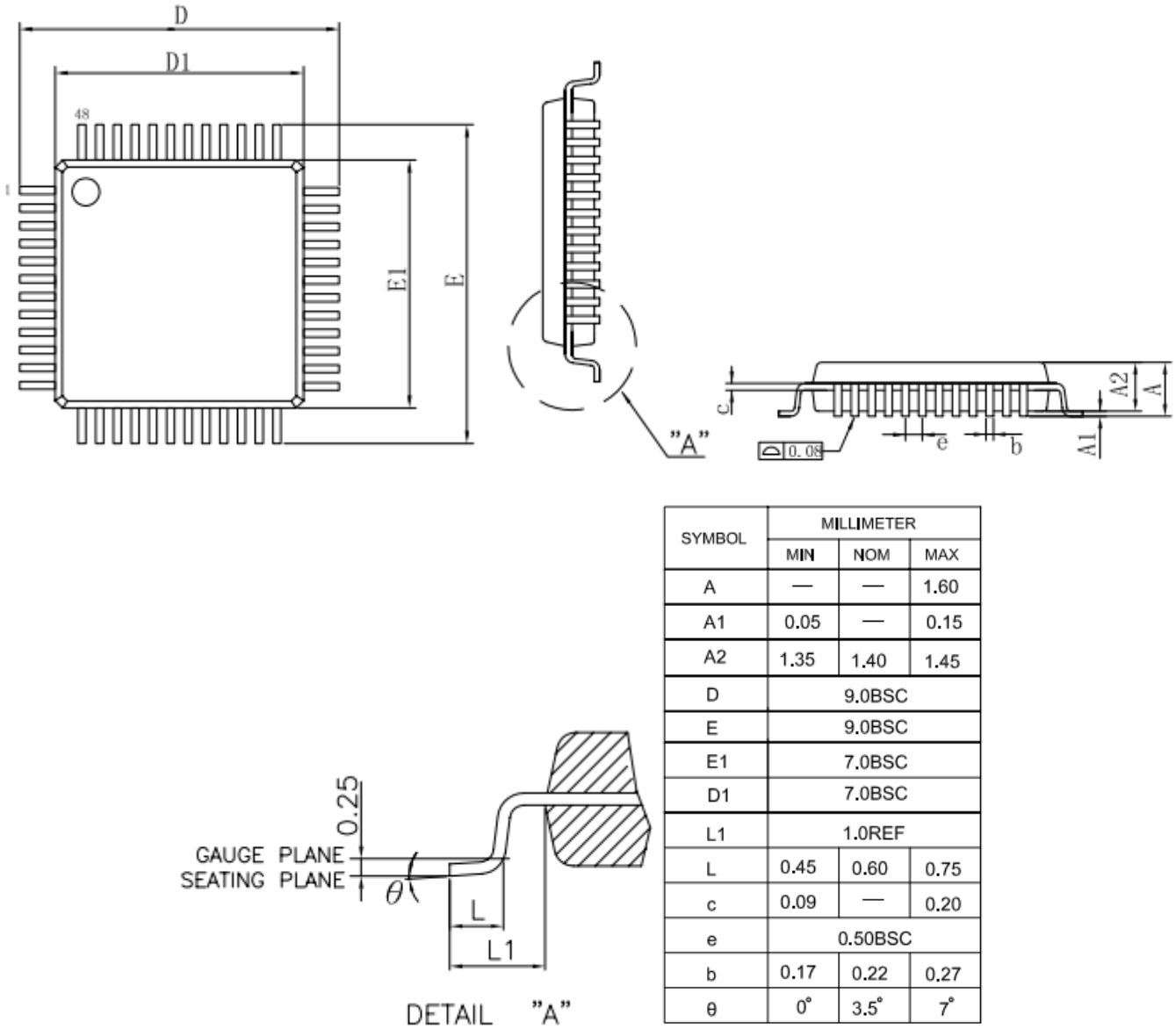
1. Controlling Dimension: MM
2. Reference Document: MO-220

36.4 48-pin LQFP

36.4.1 Recommended Landing Pattern



36.4.2POD



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
D		9.0BSC	
E		9.0BSC	
E1		7.0BSC	
D1		7.0BSC	
L1		1.0REF	
L	0.45	0.60	0.75
c	0.09	—	0.20
e		0.50BSC	
b	0.17	0.22	0.27
θ	0°	3.5°	7°

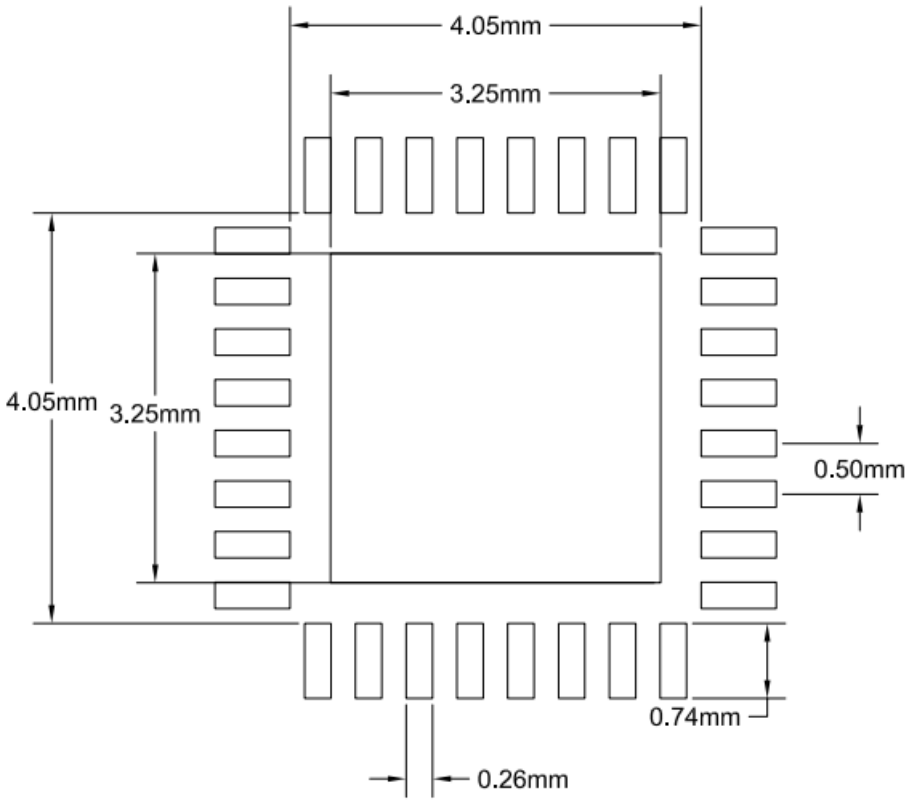
Notes:

1. Controlling Dimension: MM

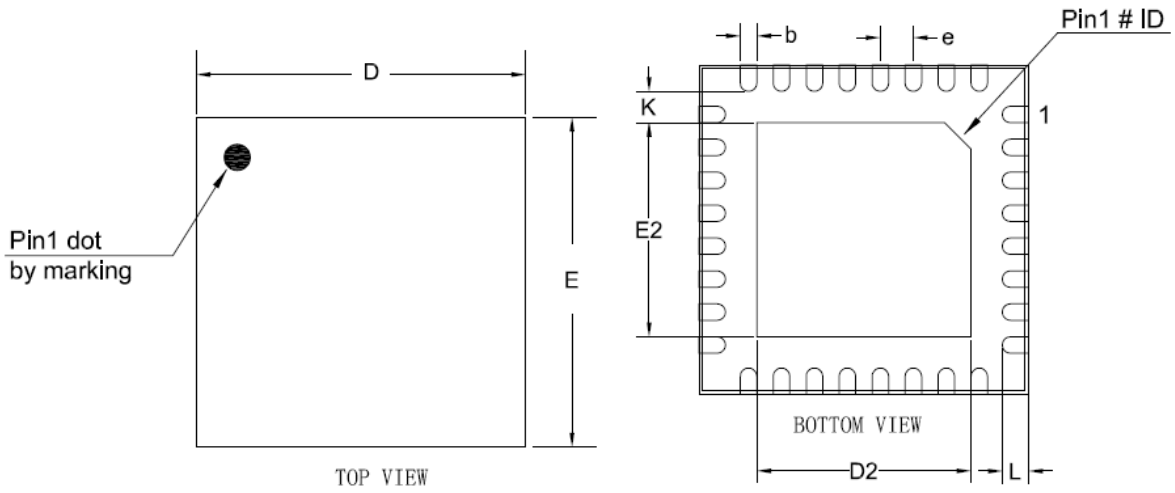
2. Reference Document: JEDEC MO-026

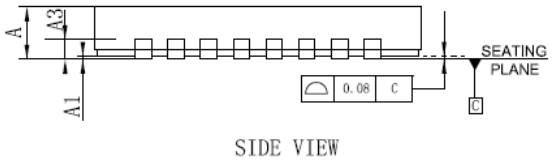
36.5 32-pin WFQFN

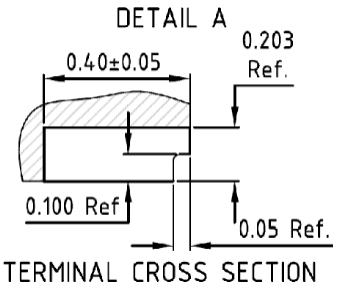
36.5.1 Recommended Landing Pattern



36.5.2POD







SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203REF		
D	4.90	5.00	5.10
E	4.90	5.00	5.10
E2	3.20	3.25	3.30
D2	3.20	3.25	3.30
b	0.18	0.25	0.30
L	0.35	0.40	0.45
K	0.20	–	–
e	0.50BSC		

Notes:

1. Controlling Dimension: MM
2. Reference Document: JEDEC MO-220

37. Ordering Information

Temperature Range: -40°C to 85°C

Order Part No.	Package	QTY/Reel	Remark
IS31CS9202A-QFLS2-TR	QFN-48, Lead-free	2500/Reel	
IS31CS9202-LQLS2	LQFP-48, Lead-free	250/Tray 2500/Box	

Temperature Range: -40°C to 125°C

Order Part No.	Package	QTY/Reel	Remark
IS32CS9202A-QWLA3-TR	Wettable Flank QFN-48, Lead-free	2500/Reel	
IS32CS9202B-QWLA3-TR	Wettable Flank QFN-32, Lead-free	2500/Reel	
IS32CS9202-LQLA3	LQFP-48, Lead-free	250/Tray 2500/Box	

38. Errata

39. Revision

Revision	Detailed Information	Date
0A	Preliminary Release	2022/09/26
0B	1. Unify Register description format 2. Update description for DMARXF of SPFCMDC 3. Correct Flash controller for IFB1 lock state 4. Update SPV waveform for various operation command 5. Update description of I2CMCR 6. Update TCC function block diagram 7. Add PLL setting and PLL VCO table 8. Update description for ADCEN of ADCCFG0 9. Add and update Special mode waveform 10. Fix minor spelling error and Register name 11. Update SDADC control due to metal fix 12. Modify Lumibus descriptions 13. Add DC/AC characteristics	2024/1/11
	CRC/CS update e-Flash controller update. CMD, refers ECC AP note, PRTDLY	2024/06/04
	Misc. feedback for correction. IFB add positive temperature entry FLSHCMD mass/chip erase only in debug mode	2024/6/18
	MBZINTF is set to 1 if play queue. Flash protection clarification.	2024/7/7
	IFB0 definition update	2024/8/24