Un.THR Transmit Data Hold Register

The UART Transmit Data Hold register is an 8-bit write-only register. The data for transmit can be stored in this register. But the write data cannot be read from this register. The data which is written in the Un.THR register will be transferred into the Transmit Shifter register whenever the Transmit Shifter register is empty.

U0.THR=0x4000_8000, U1.THR=0x4000_8100

7	6	5	4	3	2	1	0
THR							
-							
WO							

7 0	THR	Transmit Data Hold Register

Un.IER UART Interrupt Enable Register

The UART Interrupt Enable register is an 8-bit register.

U0.IER=0x4000_8004, U1.IER=0x4000_8104

7	6	5	4	3	2	1	0
-	-			-	RLSIE	THREIE	DRIE
0	0	0	0	0	0	0	0
					RW	RW	RW

2	RLSIE	Receiver line status interrupt enable	
		0	Receive line status interrupt is disabled.
		1	Receive line status interrupt is enabled
1	THREIE	Transmit holding register empty interrupt enable	
		0	Transmit holding register empty interrupt is disabled
		1	Transmit holding register empty interrupt is enabled
0	DRIE	Data receive interrupt enable	
		0	Data receive interrupt is disabled
		1	Data receive interrupt is enabled

Un.IIR UART Interrupt ID Register

The UART Interrupt ID register is an 8-bit register.

U0.IIR=0x4000_8008, U1.IIR=0x4000_8108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													RLS	THRE	DR												TXE	IID			IPEN
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
													R	R	R												R		R		R

18	RLS	Receiver line status flag (Error)	
17	THRE	Transmit holding register empty flag	
16	DR	Data receive interrupt flag	
4	TXE	Interrupt source ID See interrupt source ID table	
3 1	IID	Interrupt source ID See interrupt source ID table	
0	IPEN	Interrupt pending bit	
		0	Interrupt is pending
		1	No interrupt is pending.

The UART supports 3-priority interrupt generation and the Interrupt Source ID register shows one interrupt source which has the highest priority among pending interrupts. The priority is defined as:

- Receive line status interrupt
- Receive data ready interrupt/ Character timeout interrupt
- Transmit hold register empty interrupt

Table 12-4 Interrupt ID and Control

Priority	TXE	IID		IPEN	Interrupt Sources		
	Bit 4	Bit 2	Bit 1	Bit 0	Interrupt	Interrupt Condition	Interrupt Clear
-	0	0	0	1	None	-	-
1	0	1	1	0	Receiver Line Status	Overrun, Parity, Framing or Break Error	Read LSR register
2	0	1	0	0	Receiver Data Available	Receive data is available.	Read receive register or read IIR register
3	0	0	1	0	Transmitter Holding Register Empty	Transmit buffer empty	Write transmit hold register or read IIR register
4	1	X	X	X	Transmitter Register Empty	Transmit register empty	Write transmit hold register or read IIR register

Un.LCR UART Line Control Register

The UART Line Control register is an 8-bit register.

U0.LCR=0x4000_800C, U1.LCR=0x4000_810C

7	6	5	4	3	2	1	0
	BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN[1:0]	
0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW

6	BREAK	When this bit is set, TxD pin will be driven at low state in order to notice the alert to the receiver.
	0	Normal transfer mode
	1	Break transmit mode
5	STICKP	Force parity and it will be effective when PEN bit is set.
	0	Parity stuck is disabled
	1	Parity stuck is enabled and parity always the bit of PARITY.
4	PARITY	Parity mode selection bit and stuck parity select bit
	0	Odd parity mode
	1	Even parity mode
3	PEN	Parity bit transfer enable
	0	The parity bit disabled
	1	The parity bit enabled
2	STOPBIT	The number of stop bit followed by data bits.
	0	1 stop bit
	1	1.5 / 2 stop bit In case of 5 bit data case, 1.5 stop bit is added. In case of 6,7 or 8 bit data, 2 stop bit is added
1 0	DLEN	The data length in one transfer word.
	00	5 bit data
	01	6 bit data
	10	7 bit data
	11	8 bit data

Parity bit will be generated according to bit 3,4,5 of Un.LCR register. Table 12-5 shows the variation of parity bit generation.

Table 12-5 Interrupt ID and Control

STICKP	PARITY	PEN	Parity
X	X	0	No Parity
0	0	1	Odd Parity
0	1	1	Even Parity
1	0	1	Force parity as “1”
1	1	1	Force parity as “0”

Un.DCR UART Data Control Register

The UART Data Control register is an 8-bit register. The inversion function of Tx or Rx data line, is controlled by this Un.DCR register. When the corresponding bit is set to 1, the data line of Tx or RX signal will be inverted.

U0.DCR=0x4000_8010, U1.DCR=0x4000_8110

7	6	5	4	3	2	1	0
			LBON	RXINV	TXINV		
0	0	0	0	0	0	0	0
			RW	RW	RW		

4	LBON	Local loopback test mode enable	
		0	Normal mode
		1	Local loopback mode (TxD connected to RxD internally)
3	RXINV	Rx Data Inversion Selection	
		0	Normal RxData Input
		1	Inverted RxData Input
2	TXINV	Tx Data Inversion Selection	
		0	Normal TxData Output
		1	Inverted TxData Output

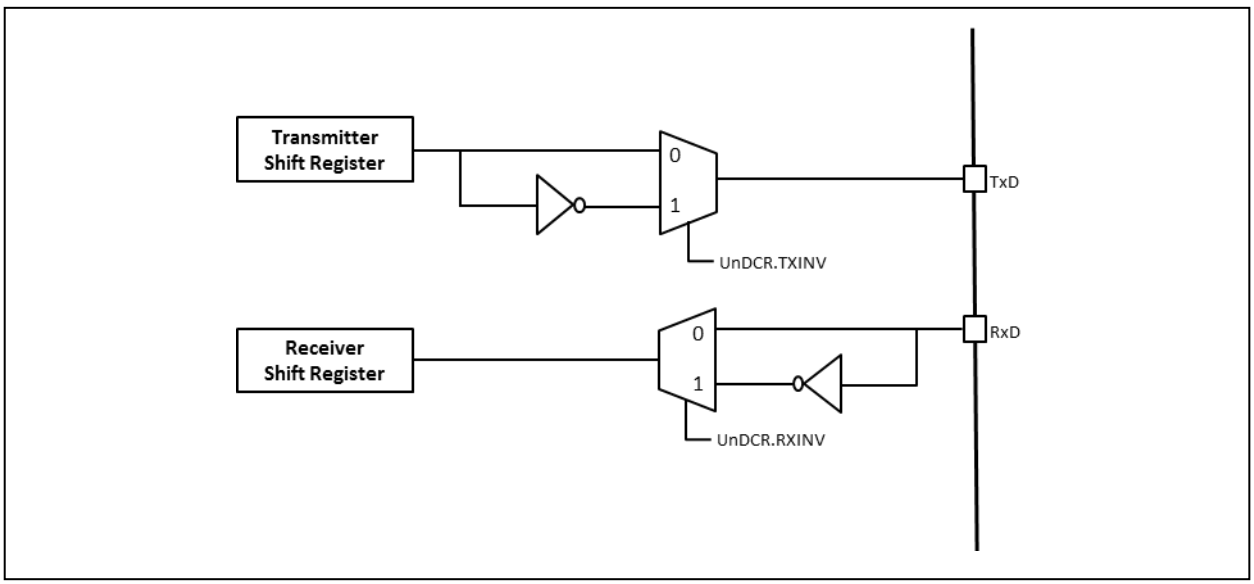


Figure 12-2 Data Inversion Control Diagram

Un.LSR UART Line Status Register

The UART Line Status register is an 8-bit register.

U0.LSR=0x4000_8014, U1.LSR=0x4000_8114

7	6	5	4	3	2	1	0
-	TEMT	THRE	BI	FE	PE	OE	DR
0	1	1	0	0	0	0	0
	R	R	R	R	R	R	R

6	TEMT	Transmit empty.
	0	Transmit register has the data is now transferring
	1	Transmit register is empty.
5	THRE	Transmit holding empty.
	0	Transmit holding register is not empty.
	1	Transmit holding register empty
4	BI	Break condition indication bit
	0	Normal status
	1	Break condition is detected
3	FE	Frame Error.
	0	No framing error.
	1	Framing error. The receive character did not have a valid stop bit
2	PE	Parity Error
	0	No parity error
	1	Parity error. The receive character does not have correct parity information.
1	OE	Overrun error
	0	No overrun error
	1	Overrun error. Additional data arrives while the RHR is full
0	DR	Data received
	0	No data in receive holding register.
	1	Data has been received and is saved in the receive holding register

This register provides the status of data transfers between Transmitter and Receiver. Users can get the line status information from this register and can handle the next process. Bits 1,2,3,4 will raise the line status interrupt when the RLSIE bit in the Un.IEN register is set. Other bits can generate its interrupt when its interrupt enable bit in the Un.IEN register is set.

Un.BDR Baud Rate Divisor Latch Register

The UART Baud Rate Divisor Latch register is a 16-bit register.

Note: Make sure the UART clock is set in MCCR4.

U0.BDR=0x4000_8020, U1.BDR=0x4000_8120

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDR															
0x0000															
RW															

15 0	BDR	Baud rate Divider latch value

To establish communication with the UART channel, the baud rate should be set properly. The programmable baud rate generator is provided to give from 1 to 65535 divider number. The 16-bit divider register (UnBDR) should be written for the expected baud rate $UART_{clock}$ gets from MCCR4.

The baud rate calculation formula is shown in the following equation:

$$\text{BDR} = \frac{UART_{clock}}{16 \times BaudRate}$$

For a $UART_{clock}$ speed of 40 MHz, the divider value and error rate is listed in Table 12-6.

Table 12-6 Example of Baud Rate Calculation (without BFR)

$UART_{clock}$=40 MHz		
Baud rate	**Divider**	**Error (%)**
1200	2083	0.02%
2400	1041	0.06%
4800	520	0.16%
9600	260	0.16%
19200	130	0.16%
38400	65	0.16%
57600	43	0.94%
115200	21	3.34%

Un.BFR Baud Rate Fraction Counter Register

The Baud Rate Fraction Counter register is an 8-bit register.

U0.BFR=0x4000_8024, U1.BFR=0x4000_8124

7	6	5	4	3	2	1	0
BFR							
0x00							
RW							

7 0	BFR	Fractions counter value.	
		0	Fraction counter is disabled
		N	Fraction counter enabled. Fraction compensation mode is operating. Fraction counter is incremented by FCNT.

Table 12-7 Example of Baud Rate Calculation

$UART_{clock}$ =40 MHz			
Baud rate	**Divider**	**FCNT**	**Error (%)**
1200	2083	85	0.00%
2400	1041	170	0.00%
4800	520	213	0.00%
9600	260	106	0.00%
19200	130	53	0.00%
38400	65	262	0.00%
57600	43	103	0.00%
115200	21	179	0.01%

$$\mathrm{FCNT} = \mathrm{Float} * 256$$

The FCNT value is calculated using the equation above. For example, when the target baud rate is 4800 bps and $UART_{clock}$ is 40MHz, the BDR value is 520.8333. The integer number 520 should be the BDR value and the floating number 0.8333 will result in the FCNT value of 213, as shown below:

FCNT = 0.8333 * 256 = 213.3333, so the FCNT value is 213.

The 8-bit fractional counter will count up by FCNT value every (baud rate)/16 periods and when the fractional counter overflows, the divisor value increments by 1. Therefore, this period will be compensated. In the next period, the divisor value will return to the original set value.

Un.IDTR Inter-frame Delay Time Register

The UART Inter-frame Time register is an 8-bit register. A dummy delay can be inserted between two continuous transmits.

U0.IDTR=0x4000_8030, U1.IDTR=0x4000_8130

7	6	5	4	3	2	1	0
SMS	DMS				WAITVAL		
1	0	0	0	0	000		
RW	RW				RW		

7	SMS	Start Bit Multi sampling enable
	0	Multi sampling is disable for start bit, Single sample will be done at 8/16 baud rate for the start bit
	1	Multi sampling is enabled for start bit. Sampling is done 3 times at 7/16, 8/16 and 9/16 baud rate. Dominant value of 3 samples will be selected for the start bit
6	DMS	Data Bit Multi sampling enable
	0	Multi sampling is disable for data bit, Single sample will be done at 8/16 baud rate for the data bit
	1	Multi sampling is enabled for data bit. Sampling is done 3 times at 7/16, 8/16 and 9/16 baud rate. Dominant value of 3 samples will be selected for the data bit
2 0	WAITVAL	Wait time is decided by this value $\text{Wait Time} = \frac{WAITVAL}{BAUDRATE}$

Functional Description

The UART module is compatible with 16450 UART. Additionally, fractional baud rate compensation logic is provided. This module does not have an internal FIFO block. Therefore, data transfer will establish interactive support.

Receiver Sampling Timing

The UART operates per the following timing:

If the falling edge is on the receive line, UART judges it as the start bit. From the start timing, UART oversamples 16 times of 1-bit and detects the bit value at the 7th sample of 16 samples.

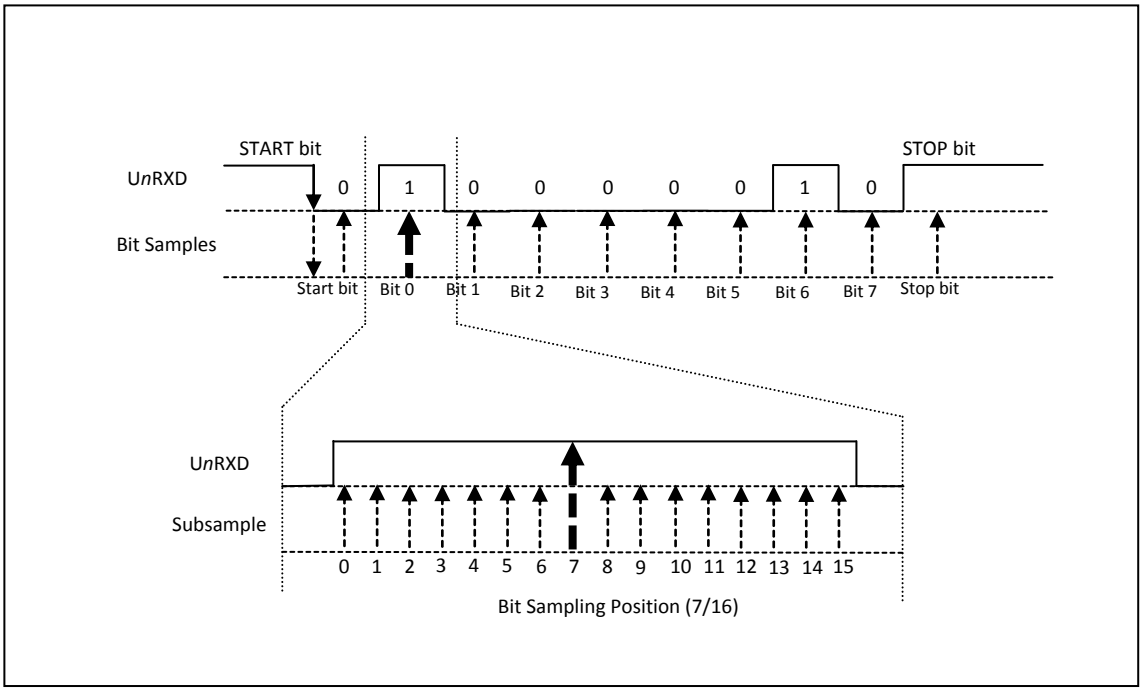


Figure 12-3 Sampling Timing of UART Receiver

Note: It is recommended to enable debounce settings in the PCU block to reinforce the immunity of external glitch noise.

Transmitter

The transmitter's function is to transmit data. The start bit, data bits, optional parity bit, and stop bit are serially shifted, with the least significant bit first. The number of data bits is selected in the DLAN[1:0] field in the Un.LCR register.

The parity bit is set according to the PARITY and PEN bit field in the Un.LCR register. If the parity type is even, then the parity bit depends on the one bit sum of all data bits. For odd parity, the parity bit is the inverted sum of all data bits.

The number of stop bits is selected in the STOPBIT field in the Un.LCR register.

An example of transmit data format is shown in Figure 12-4.

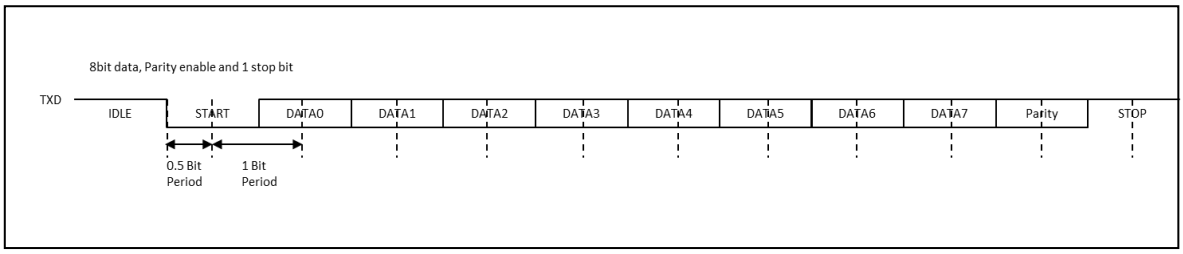


Figure 12-4 Transmit Data Format Example

Inter-frame Delay Transmission

The inter-frame delay function allows the transmitter to insert an Idle state on the TXD line between two characters. The width of the Idle state is defined in the WAITVAL field in the Un.IDTR register. When this field is set to 0, no time-delay is generated. Otherwise, the transmitter holds a high level on TXD after each transmitted character during the number of bit periods defined in the WATIVAL field.

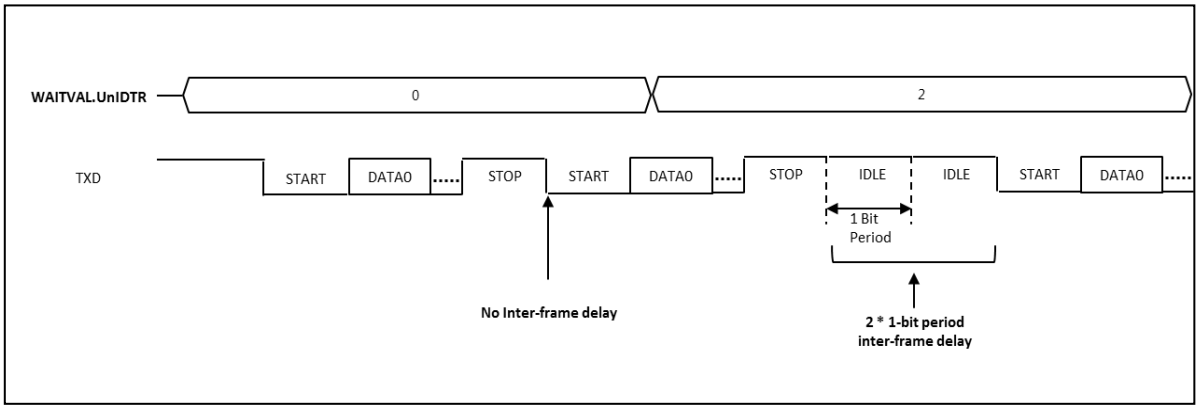


Figure 12-5 Inter-frame Delay Timing Diagram

Transmit Interrupt

The transmit operation creates interrupt flags. When the Transmitter Holding register is empty, the THRE interrupt flag will be set. When the Transmitter Shifter register is empty, the TXE interrupt flag will be set. Users can select which interrupt timing is best for the application.

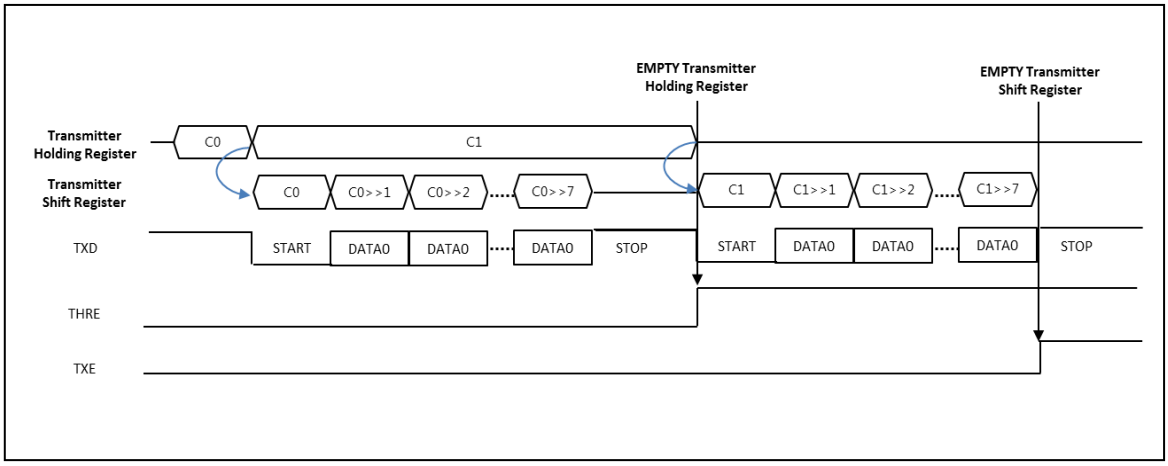


Figure 12-6 Transmit Interrupt Timing Diagram

13. Serial Peripheral Interface (SPI)

Overview

One-channel serial interface is provided for synchronous serial communications with external peripherals. The SPI block supports Master and Slave modes. Four signals are used for SPI communication – SS, SCK, MOSI, and MISO.

- Master or Slave operation
- Programmable clock polarity and phase
- 8, 9, 16, 17-bit wide transmit/receive register
- 8, 9, 16, 17-bit wide data frame
- Loop-back mode
- Programmable start, burst, and stop delay time

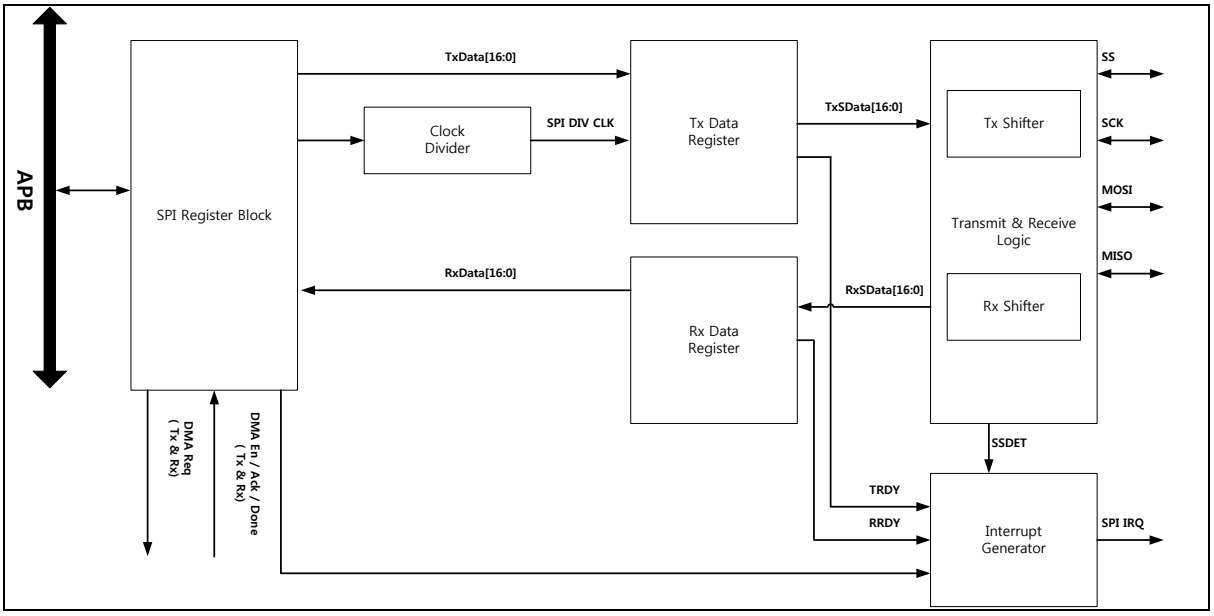


Figure 13-1 Block Diagram

Pin Description

Table 13-1 External Pins

PIN NAME	TYPE	DESCRIPTION
SS	I/O	SPI Slave select input / output
SCK	I/O	SPI Serial clock input / output
MOSI	I/O	SPI Serial data (Master output, Slave input)
MISO	I/O	SPI Serial data (Master input, Slave output)

Registers

The base address of SPI is 0x4000_9000 and the register map is described in Table 13-2 and Table 13-3.

Table 13-2 SPI Base Address

NAME	BASE ADDRESS
SPI	0x4000_9000

Table 13-3 SPI Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
SP.TDR	0x00	W	SPI Transmit Data Register	-
SP.RDR	0x00	R	SPI Receive Data Register	0x000000
SP.CR	0x04	RW	SPI Control Register	0x001020
SP.SR	0x08	RW	SPI Status Register	0x000006
SP.BR	0x0C	RW	SPI Baud rate Register	0x0000FF
SP.EN	0x10	RW	SPI Enable register	0x000000
SP.LR	0x14	RW	SPI delay Length Register	0x010101

SP.TDR SPI Transmit Data Register

SP.TDR is a 17-bit read/write register. It contains serial transmit data.

SP.TDR=0x4000_9000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TDR															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000															
																RW															

16 0	TDR	Transmit Data Register

SP.RDR SPI Receive Data Register

SP.RDR is a 17-bit read/write register. It contains serial receive data.

SP.RDR=0x4000_9000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RDR															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000															
																RW															

16 0	RDR	Receive Data Register

SP.CR SPI Control Register

SP.CR is a 20-bit read/write register and can be set to configure SPI operation mode.

SP.CR=0x4000_9004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											TXBC	RXBC			SSCIE	TXIE	RXIE	SSMOD	SSOUT	LBE	SSMARK	SSMO	SSPOL			MS	MSBF	CPHA	CPOL	BITSZ	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	00	
											RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	

20	TXBC	Tx buffer clear bit.
		0 No action
		1 Clear Tx buffer
19	RXBC	Rx buffer clear bit
		0 No action
		1 Clear Rx buffer
16	SSCIE	SS Edge Change Interrupt Enable bit.
		0 nSS interrupt is disabled.
		1 nSS interrupt is enabled for both edges (L→H, H→L)
15	TXIE	Transmit Interrupt Enable bit.
		0 Transmit Interrupt is disabled.
		1 Transmit Interrupt is enabled.
14	RXIE	Receive Interrupt Enable bit..
		0 Receive Interrupt is disabled.
		1 Receive Interrupt is enabled.
13	SSMOD	SS Auto/Manual output select bit.
		0 SS output is not set by SSOUT (SP.CR[12]). - SS signal is in normal operation mode.
		1 SS output signal is set by SSOUT.
12	SSOUT	SS output signal select bit.
		0 SS output is ‘L.’
		1 SS output is ‘H’.
11	LBE	Loop-back mode select bit in master mode.
		0 Loop-back mode is disabled.
		1 Loop-back mode is enabled.
10	SSMASK	SS signal masking bit in slave mode.
		0 SS signal masking is disabled. - Receive data when SS signal is active.
		1 SS signal masking is enabled. - Receive data at SCLK edges. SS signal is ignored.
9	SSMO	SS output signal select bit.
		0 SS output signal is disabled.
		1 SS output signal is enabled.
8	SSPOL	SS signal Polarity select bit.
		0 SS signal is Active-Low.
		1 SS signal is Active-High.
5	MS	Master/Slave select bit.
		0 SPI is in Slave mode.
		1 SPI is in Master mode.
4	MSBF	MSB/LSB Transmit select bit.
		0 LSB is transferred first.
		1 MSB is transferred first.
3	CPHA	SPI Clock Phase bit.

		0	Sampling of data occurs at odd edges (1,3,5,…,15).
		1	Sampling of data occurs at even edges (2,4,6,…,16).
2	CPOL	SPI Clock Polarity bit.	
		0	Active-high clocks selected.
		1	Active-low clocks selected.
1	BITSZ	Transmit/Receive Data Bits select bit.	
		00	8 bits
		01	9 bits
		10	16 bits
0		11	17 bits

CPOL=0, CPHA=0 : data sampling at rising edge, data changing at falling edge
CPOL=0, CPHA=1 : data sampling at falling edge, data changing at rising edge
CPOL=1, CPHA=0 : data sampling at falling edge, data changing at rising edge
CPOL=1, CPHA=1 : data sampling at rising edge, data changing at falling edge

SP.SR SPI Status Register

SP.SR is a 10-bit read/write register. It contains the status of SPI interface.

SP.SR=0x4000_9008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SSDET	SSON	OVRF	UDRF	TXIDLE	TRDY	RRDY
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
									RC1	RC1	RC1	RC1	R	R	R

6	SSDET	The rising or falling edge of SS signal Detect flag.
		0 SS edge is not detected.
		1 SS edge is detected.
		- The bit is cleared when it is written as “0”.
5	SSON	SS signal Status flag.
		0 SS signal is inactive.
		1 SS signal is active.
4	OVRF	Receive Overrun Error flag.
		0 Receive Overrun error is not detected.
		1 Receive Overrun error is detected.
		- This bit is cleared by writing or reading SP.RDR.
3	UDRF	Transmit Underrun Error flag.
		0 Transmit Underrun is not occurred.
		1 Transmit Underrun is occurred.
		- This bit is cleared by writing or reading SP.TDR.
2	TXIDLE	Transmit/Receive Operation flag.
		0 SPI is transmitting data
		1 SPI is in IDLE state.
1	TRDY	Transmit buffer Empty flag.
		0 Transmit buffer is busy.
		1 Transmit buffer is ready.
		- This bit is cleared by writing data to SP.TDR.
0	RRDY	Receive buffer Ready flag.
		0 Receive buffer has no data.
		1 Receive buffer has data.
		- This bit is cleared by writing data to SP.RDR.

SP.BR SPI Baud Rate Register

SP.BR is a 16-bit read/write register. The baud rate can be set by writing to the register.

SP.BR=0x4000_900C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR															
0x00FF															
RW															

15	BR	Baud rate setting bits
		Baud Rate = PCLK / (BR + 1)
0		(BR must be bigger than “0”, BR >= 2)

SP.EN SPI Enable Register

SP.EN is a bit read/write register. It contains the SPI enable bit.

SP.EN=0x4000_9010

7	6	5	4	3	2	1	0
							ENABLE
0	0	0	0	0	0	0	0
							RW

0	ENABLE	SPI Enable bit
		0 SPI is disabled. - SP.SR is initialized by writing "0" to this bit but other registers aren't initialized.
		1 SPI is enabled. - When this bit is written as "1", the dummy data of transmit buffer will be shifted. To prevent this, write data to SP.TDR before this bit is active.

SP.LR SPI Delay Length Register

SP.LR is a 24-bit read/write register. It contains start, burst, and stop length values.

SP.LR=0x4000_9014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SPL								BTL								STL							
0	0	0	0	0	0	0	0	0x01								0x01								0x01							
								RW								RW								RW							

23	SPL	StoP Length value
16		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (SPL ≥ 1)
15	BTL	BursT Length value
8		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (BTL ≥ 1)
7	STL	STart Length value
0		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (STL ≥ 1)

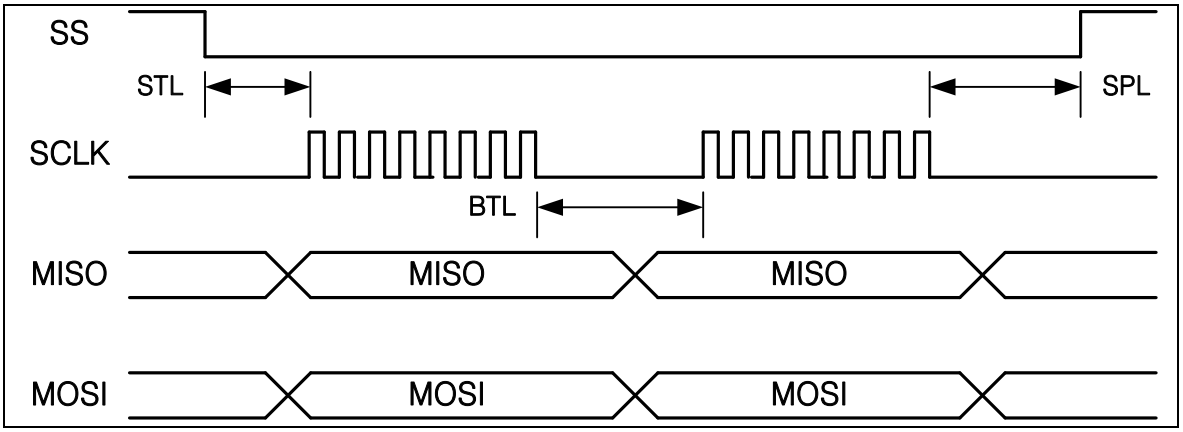


Figure 13-2 SPI Wave form (STL, BTL and SPL)

Functional Description

The SPI Transmit block and Receive block share the Clock Gen block but they are independent of each other. The Transmit and Receive blocks have double buffers and SPI is available for back to back transfer operation.

SPI Timing

The SPI has four modes of operation. These modes essentially control the way data is clocked in or out of an SPI device. The configuration is done by two bits in the SPI control register (SP.CR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock. The clock phase (CPHA) control bit selects one of the two fundamentally different transfer formats. To ensure proper communication between master and slave, both devices have to run in the same mode. This can require a reconfiguration of the master to match the requirements of different peripheral slaves.

The clock polarity has no significant effect on the transfer format. Switching this bit causes the clock signal to be inverted (active high becomes active low and idle low becomes idle high). The settings of the clock phase, however, select one of two different transfer timings, which are described in further detail in the next two chapters. Because the MOSI and MISO lines of the master and the slave are directly connected to each other, the diagrams show the timing of both devices – master and slave. The nSS line is the slave select input of the slave. The nSS pin of the master is not shown in the diagrams. It has to be inactive by a high level on this pin (if configured as input pin) or by configuring it as an output pin.

The timing of an SPI transfer where CPHA is zero is shown in Figure 13-3 and Figure 13-4. Two wave forms are shown for the SCK signal – one for CPOL equals zero and another for CPOL equals one.

When the SPI is configured as a slave, the transmission starts with the falling edge of the /SS line. This activates the SPI of the slave and the MSB of the byte stored in its data register (SP.TDR) is output on the MISO line. The actual transfer is started by a software write to the SP.TDR of the master. This causes the clock signal to be generated. In cases where the CPHA equals zero, the SCLK signal remains zero for the first half of the first SCLK cycle. This ensures that the data is stable on the input lines of both the master and the slave. The data on the input lines is read with the edge of the SCLK line from inactive to active state. The edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one) causes the data to be shifted one bit further so that the next bit is output on the MOSI and MISO lines.

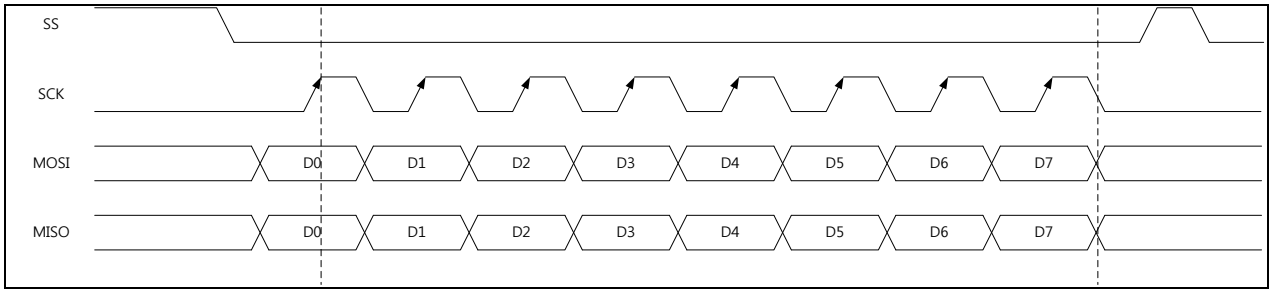


Figure 13-3 SPI Transfer Timing 1/4 (CPHA=0, CPOL=0, MSBF=0)

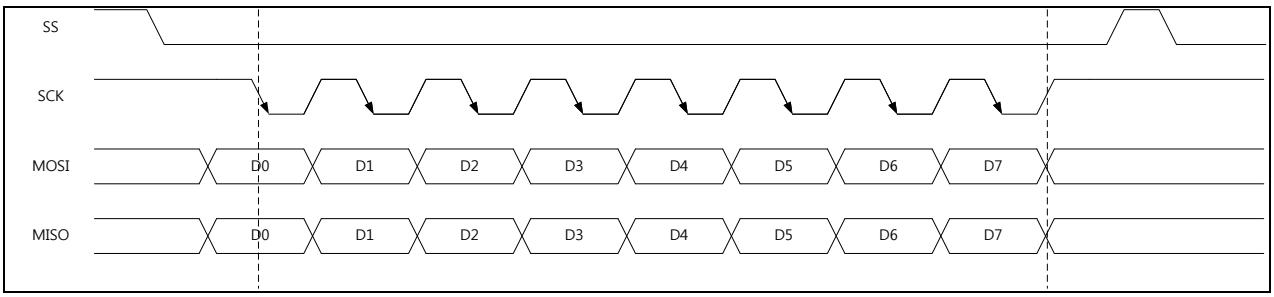


Figure 13-4 SPI Transfer Timing 2/4 (CPHA=0, CPOL=1, MSBF=1)

The timing of an SPI transfer where CPHA is 1, is shown in Figure 13-5 and Figure 13-6.Two wave forms are shown for the SCLK signal – one for CPOL equals zero and another for CPOL equals one.

Similar to the previous cases, the falling edge of the nSS lines selects and activates the slave. Compared to the previous cases, where CPHA equals zero, the transmission is not started and the MSB is not output by the slave at this stage. The actual transfer is started by a software write to the SP.TDR of the master which causes the clock signal to be generated. The first edge of the SCLK signal from its inactive to its active state (rising edge if CPOL equals zero and falling edge if CPOL equals one) causes both the master and the slave to output the MSB of the byte in the SP.TDR.

As shown in Figure 13-3 and Figure 13-4, there is no delay of half a SCLK-cycle. The SCLK line changes its level immediately at the beginning of the first SCLK-cycle. The data on the input lines is read with the edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one). After eight clock pulses, the transmission is completed.

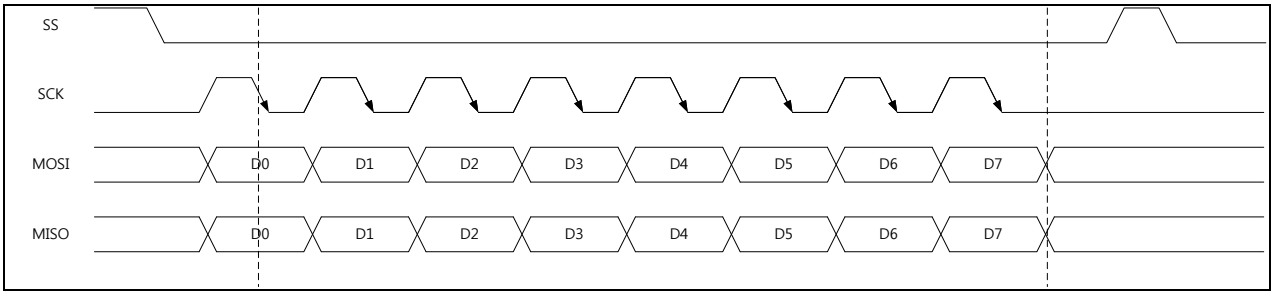


Figure 13-5 SPI Transfer Timing 3/4 (CPHA=1, CPOL=0, MSBF=0)

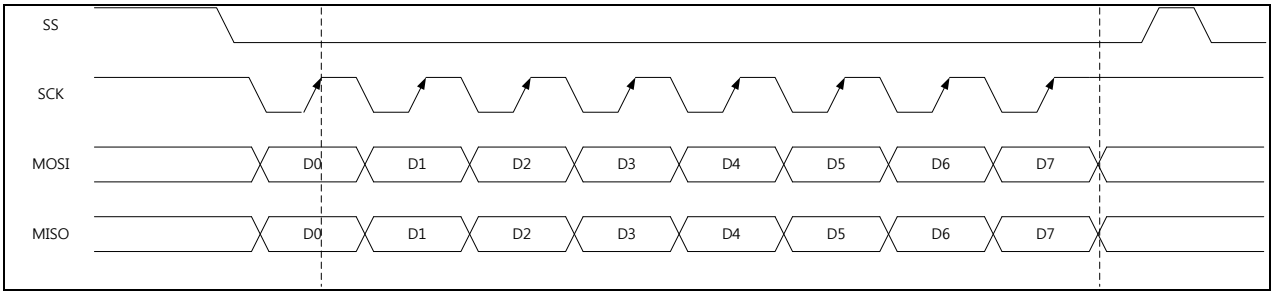


Figure 13-6 SPI Transfer Timing 4/4 (CPHA=1, CPOL=1, MSBF=1)

14. I²C Interface

Overview

The Inter-Integrated Circuit (I²C) bus serves as an interface between the microcontroller and the serial I²C bus. It provides two wires, serial bus interface to a large number of popular devices and allows parallel-bus systems to communicate bidirectionally with the I²C-bus. Features include:

- Master and slave operation
- Programmable communication speed
- Multi-master bus configuration
- 7-bit addressing mode
- Standard data rate of 100/400 KBps
- STOP signal generation and detection
- START signal generation
- ACK bit generation and detection

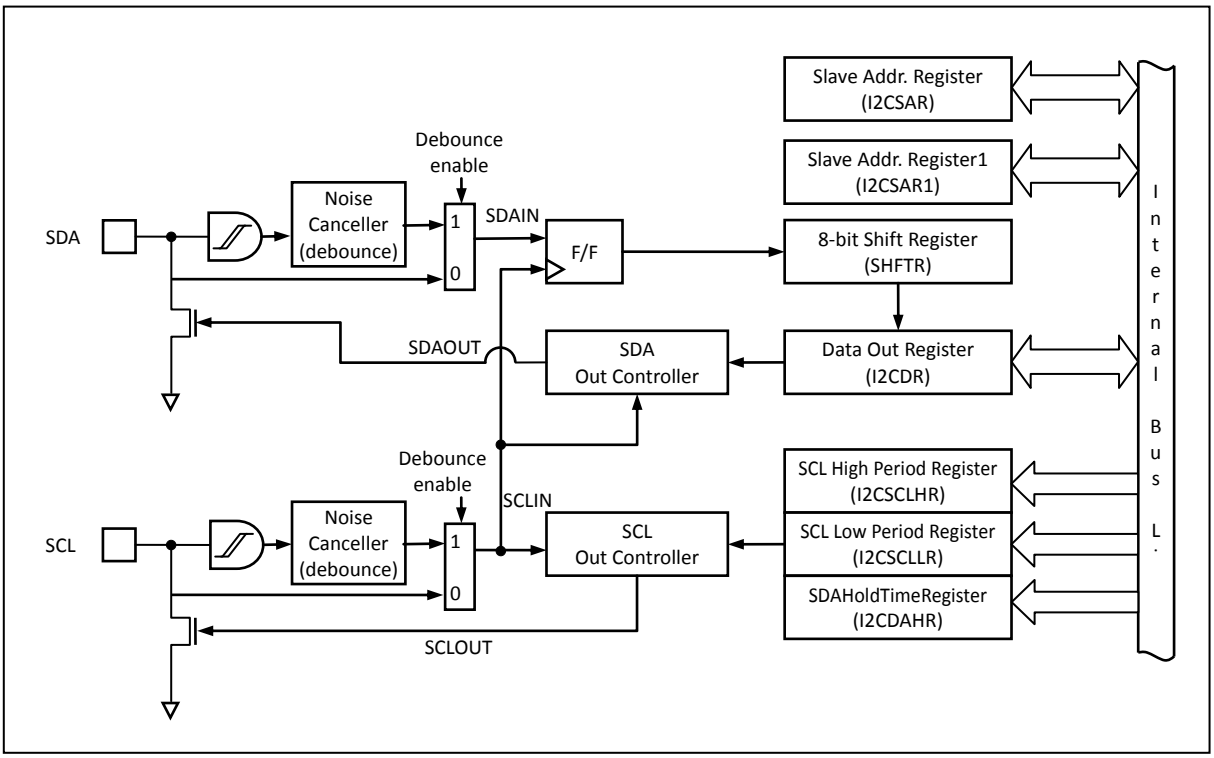


Figure 14-1. I²C Block Diagram

Pin Description

Table 14-1 I²C Interface External Pins

PIN NAME	TYPE	DESCRIPTION
SCL	I/O	I²C channel Serial clock bus line (open-drain)
SDA	I/O	I²C channel Serial data bus line (open-drain)

Registers

The base address of I²C is 0x4000_A000. The register map is described in Table 14-2 and Table 14-3.

Table 14-2 I²C Interface Base Address

NAME	BASE ADDRESS
I²C	0x4000_A000

Table 14-3 I²C Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
IC.DR	0x00	RW	I²C Data Register	0xFF
IC.SR	0x08	R, RW	I²C Status Register	0x00
IC.SAR	0x0C	RW	I²C Slave Address Register	0x00
IC.CR	0x14	RW	I²C Control Register	0x00
IC.SCLL	0x18	RW	I²C SCL LOW duration Register	0xFFFF
IC.SCLH	0x1C	RW	I²C SCL HIGH duration Register	0xFFFF
IC.SDH	0x20	RW	I²C SDA Hold Register	0x7F

IC.DR I²C Data Register

IC.DR is an 8-bit read/write register. It contains a byte of serial data to be transmitted or a byte which has just been received.

IC.DR=0x4000_A000

7	6	5	4	3	2	1	0
DR							
0xFF							
RW							

7 0	DR	The most recently received data or data to be transmitted.

IC.SR I²C Status Register

IC.SR is an 8-bit read/write register. It contains the status of I²C bus interface. Writing to the register clears the status bits.

IC.SR=0x4000_A008

7	6	5	4	3	2	1	0
GCALL	TEND	STOP	SSEL	MLOST	BUSY	TMODE	RXACK
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

7	GCALL	General call flag	
		0	General call is not detected.
		1	General call detected or slave address (ID byte) was sent.
6	TEND	1 Byte transmission complete flag	
		0	The transmission is working or not completed.
		1	The transmission is completed.
5	STOP	STOP flag	
		0	STOP is not detected.
		1	STOP is detected.
4	SSEL	Slave flag	
		0	Slave is not selected.
		1	Slave is selected.
3	MLOST	Mastership lost flag	
		0	Mastership is not lost.
		1	Mastership is lost.
2	BUSY	BUSY flag	
		0	I²C bus is in IDLE state.
		1	I²C bus is busy.
1	TMODE	Transmitter/Receiver mode flag	
		0	Receiver mode.
		1	Transmitter mode.
0	RXACK	Rx ACK flag	
		0	Rx ACK is not received.
		1	Rx ACK is received.

IC.SAR I²C Slave Address Register

IC.SAR is an 8-bits read/write register. It shows the address in Slave mode.

IC.SAR=0x4000_A00C

7	6	5	4	3	2	1	0
SVAD							GCEN
0x00							0
RW							RW

7 1	SVAD	7-bit Slave Address
0	GCEN	General call enable bit
	0	General call is disabled.
	1	General call is enabled.

IC.CR I²C Control Register

IC.CR is a 16-bit read/write register. This register can be set to configure I²C operation mode and simultaneously allowed for I²C transactions to be kicked off.

IC.CR=0x4000_A014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						INTDEL		IIF		SOFTRST	INTEN	ACKEN		STOP	START
0	0	0	0	0	0	00		0	0	0	0	0	0	0	0
						RW		R		RW	RW	RW		RW	RW

9 8	INTDEL	Interval delay value between address and data transfer (or DATA and DATA)
	0	1 * ICnSCLL
	1	2 * ICnSCLL
	2	4 * ICnSCLL
	3	8 * ICnSCLL
7	IIF	Interrupt status bit
	0	Interrupt is inactive
	1	Interrupt is active
5	SOFTRST	Soft Reset enable bit.
	0	Soft Reset is disabled.
	1	Soft Reset is enabled..
4	INTEN	Interrupt enabled bit.
	0	Interrupt is disabled.
	1	Interrupt is enabled.
3	ACKEN	ACK enable bit in Receiver mode.
	0	ACK is not sent after receiving data.
	1	ACK is sent after receiving data.
1	STOP	Stop enable bit. When this bit is set as “1” in transmitter mode, next transmission will be stopped even though ACK signal has been received.
	0	Stop is disabled.
	1	Stop is enabled. When this bit is set, transmission will be stopped.
0	START	Transmission start bit in master mode.
	0	Waits in slave mode.
	1	Starts transmission in master mode.

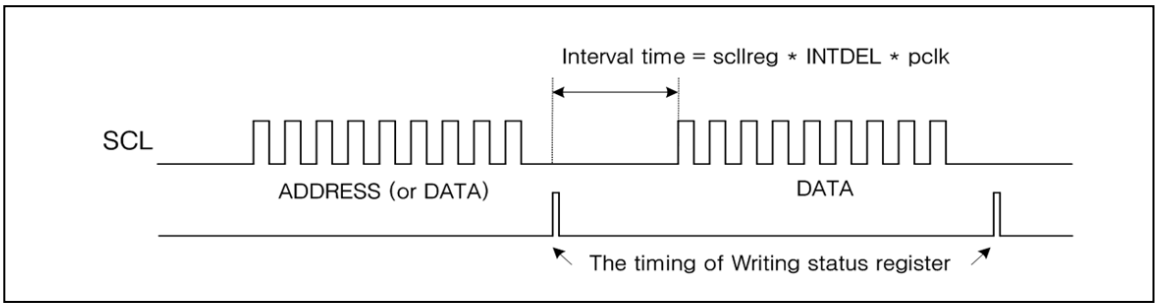


Figure 14-2 INTDEL in Master Mode

IC.SCLL I²C SCL LOW Duration Register

IC.SCLL is a 16-bit read/write register. SCL LOW time can be set by writing this register in Master mode.

IC.SCLL=0x4000_A018

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLL															
0xFFFF															
RW															

15	SCLL	SCL LOW duration value. SCLL = (PCLK * SCLL[15:0]) + 2*PCLKs
0		Default value is 0xFFFF.

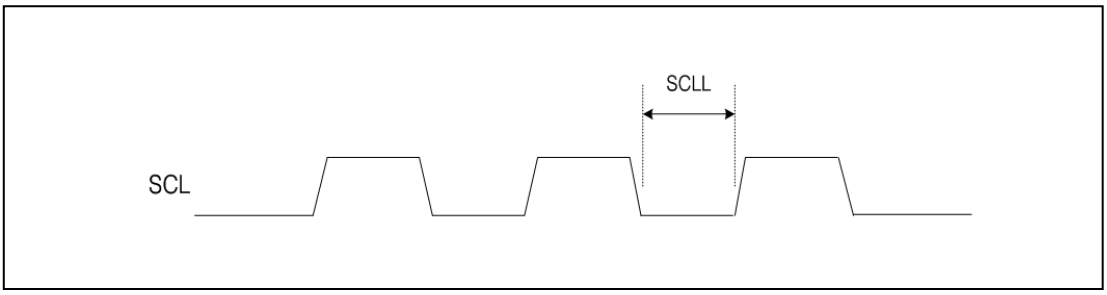


Figure 14-3 SCL LOW Timing

IC.SCLHI²C SCL HIGH Duration Register

IC.SCLH is a 16-bit read/write register. SCL HIGH time can be set by writing this register in Master mode.

IC.SCLH=0x4000_A01C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH															
0xFFFF															
RW															

15	SCLH	SCL HIGH duration value. SCLH = (PCLK * SCLH[15:0]) + 3 PCLKs
0		Default value is 0xFFFF.

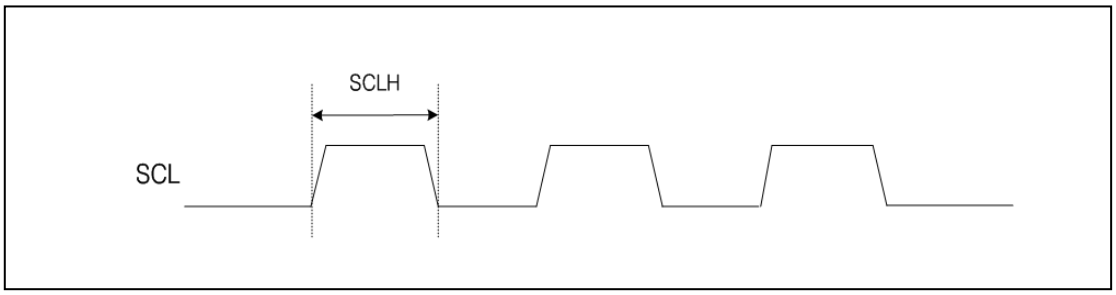


Figure 14-4 SCL HIGH Timing

IC.SDH SDA Hold Register

IC.SDH is a 15-bit read/write register. SDA HOLD time can be set by writing this register in Master mode.

IC.SDH=0x4000_A020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDH														
0x7FFF															
RW															

14	SDH	SDA HOLD time setting value. SDH = (PCLK * SDH[14:0]) + 4 PCLKs
0		Default value is 0x7FFF.

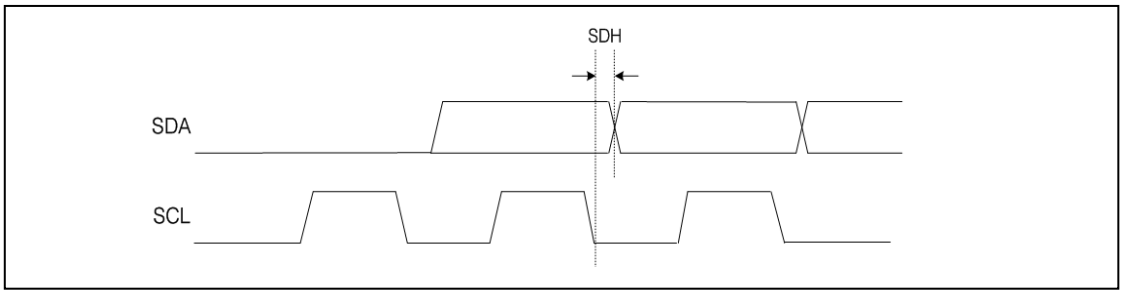


Figure 14-5 SDA HOLD Timing

Functional Description

I²C Bit Transfer

The data on the SDA line must be stable during the “H” period of the clock. The “H” or “L” state of the data line can only change when the clock signal on the SCL line is “L” (see Figure 14-6).

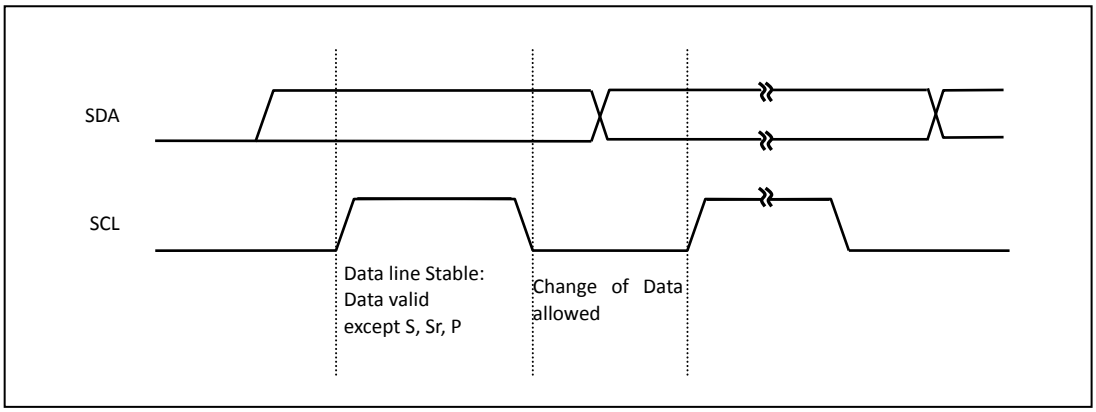


Figure 14-6 I²C Bus Bit Transfer

START/Repeated START/STOP

Within the procedure of the I²C-bus, unique situations arise which are defined as START(S) and STOP(P) conditions (see Figure 14-7).

An “H” to “L” transition on the SDA line while SCL is “H” is one such unique case. This situation indicates a START condition. An “L” to “H” transition on the SDA line while SCL is “H” defines a STOP condition.

START and STOP conditions are always generated by the Master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition.

The bus is busy if a repeated START(Sr) is generated instead of a STOP condition. In this respect, the START(S) and repeated START(Sr) conditions are functionally identical. For the remainder of this document therefore, the S symbol will be used as a generic term to represent both the START and repeated START conditions, unless Sr is particularly relevant.

Detection of START and STOP conditions by devices connected to the bus is easy if they incorporate the necessary interfacing hardware. However, microcontrollers with no such interface have to sample the SDA line at least twice per clock period to sense the transition.

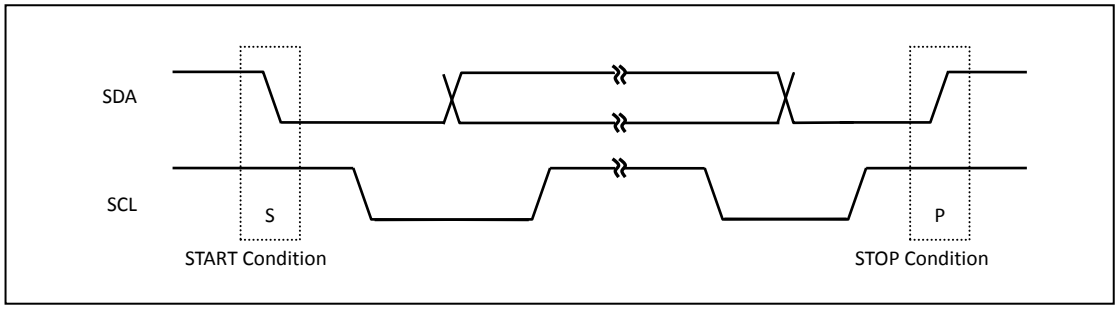


Figure 14-7 START and STOP Condition

Data Transfer

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first (see Figure 14-8). If a slave can't receive or transmit another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL "L" to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL.

A message which starts with such an address can be terminated by generation of a STOP condition, even during the transmission of a byte. In this case, no acknowledge is generated.

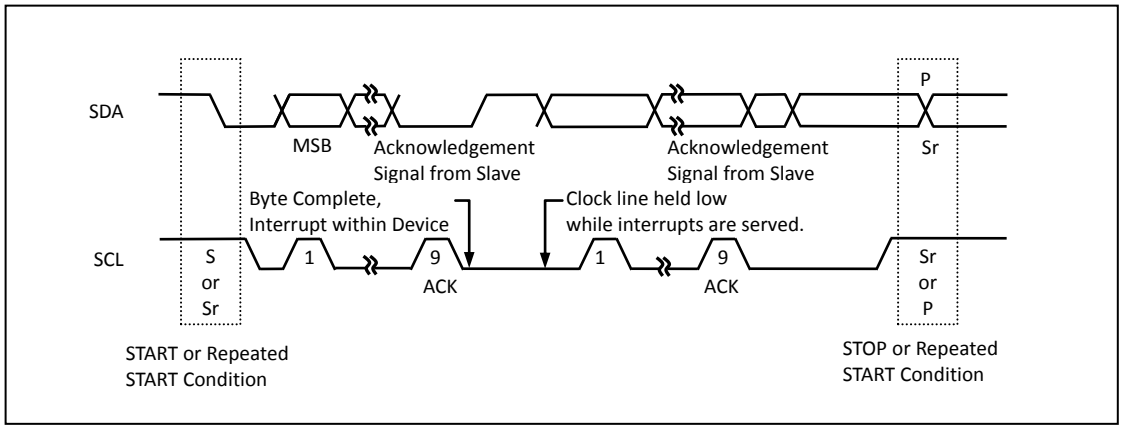


Figure 14-8 I^2C Bus Data Transfer

Acknowledge

Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse.

The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable “L” during the “H” period of this clock pulse (see Figure 14-9). Set-up and hold times must also be taken into account.

When a slave doesn’t acknowledge the slave address (for example, it is unable to receive or transmit because it is performing some real-time function), the data line must be left “H” by the slave. The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a slave-receiver does acknowledge the slave address but, sometime later in the transfer cannot receive any more data bytes, the master must again abort the transfer. This is indicated by the slave generating the not-acknowledge on the first byte to follow. The slave leaves the data line “H” and the master generates a STOP or a repeated START condition.

If a master-receiver is involved in a transfer, it must signal the end of data to the slave-transmitter by not generating acknowledge on the last byte that was clocked out of the slave. The slave-transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

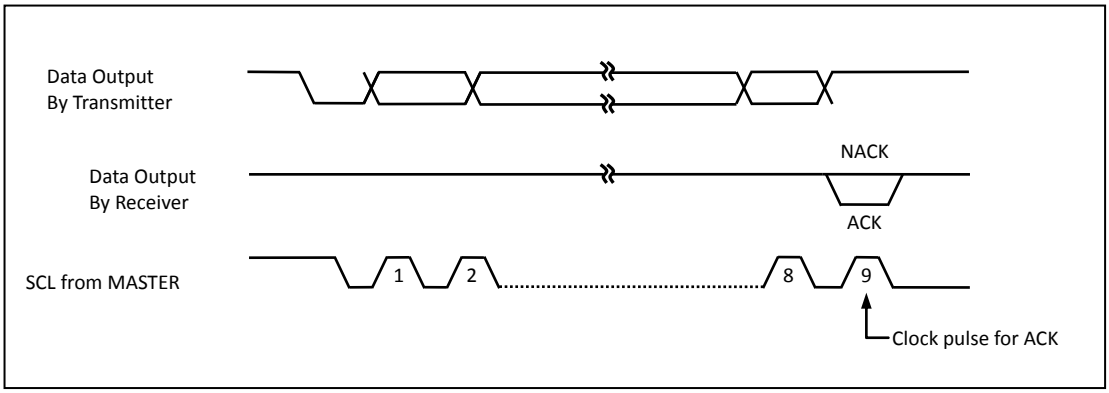


Figure 14-9 I²C Bus Acknowledge

Synchronization

All masters generate their own clock on the SCL line to transfer messages on the I^2C-bus. Data is only valid during the “H” period of the clock. Therefore, a defined clock is required for the bit-by-bit arbitration procedure to take place.

Clock synchronization is performed using the wired-AND connection of I^2C interfaces to the SCL line. This means that an “H” to “L” transition on the SCL line will cause the devices to start counting off their “L” period and, once a device clock has gone “L”, it will hold the SCL line in that state until the clock “H” state is reached (see Figure 14-10). However, the “L” to “H” transition of this clock may not change the state of the SCL line if another clock is still within its “L” by the device with the longest “L” period. Devices with shorter “L” periods enter an “H” wait-state during this time.

When all devices concerned have counted off their “L” period, the clock line will be released and go “H”. There will then be no difference between the device clocks and the state of the SCL line, and the devices will start counting their “H” periods. The first device to complete its “H” period will again pull the SCL line “L”.

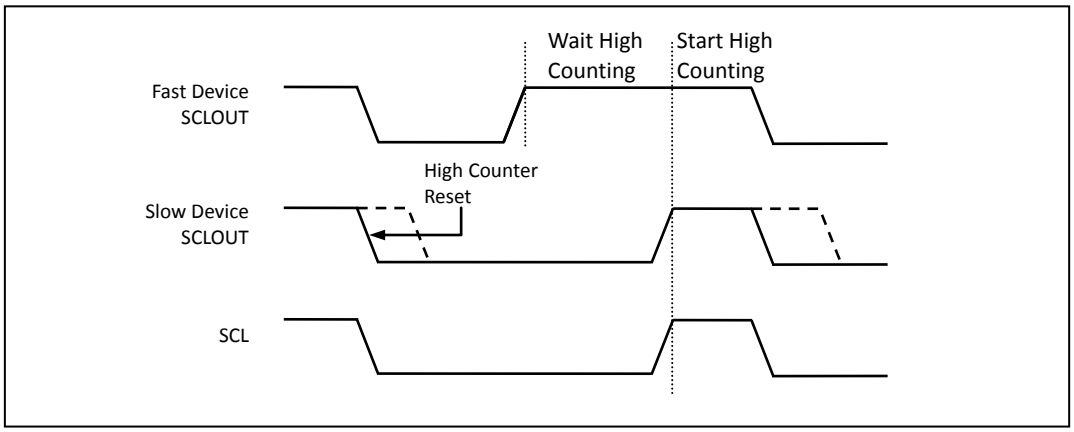


Figure 14-10 Clock Synchronization During the Arbitration Procedure

Arbitration

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition within the minimum hold time of the START condition which results in a defined START condition to the bus.

Arbitration takes place on the SDA line, while the SCL line is at the “H” level, in such a way that the master which transmits “H” level, while another master is transmitting “L” level will switch off its DATA output stage because the level on the bus doesn’t correspond to its own level.

Arbitration can continue for many bits. Its first stage is comparison of the address bits. If the masters are each trying to address the same device, arbitration continues with comparison of the data-bits if they are master-transmitter or acknowledge-bits if they are master-receiver. Because address and data information on the I^2C-bus is determined by the winning master, no information is lost during the arbitration process.

A master that loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration.

If a master also incorporates a slave function and it loses arbitration during the addressing stage, it’s possible that the winning master is trying to address it. The losing master must therefore switch over immediately to its slave mode.

Figure 14-11 shows the arbitration procedure for two masters. Of course, more may be involved (depending on how many masters are connected to the bus). As soon as there is a difference between the internal data level of the master generating Device1 data out and the actual level on the SDA line, its data output is switched off, which means that a “H” output level is then connected to the bus. This will not affect the data transfer initiated by the winning master.

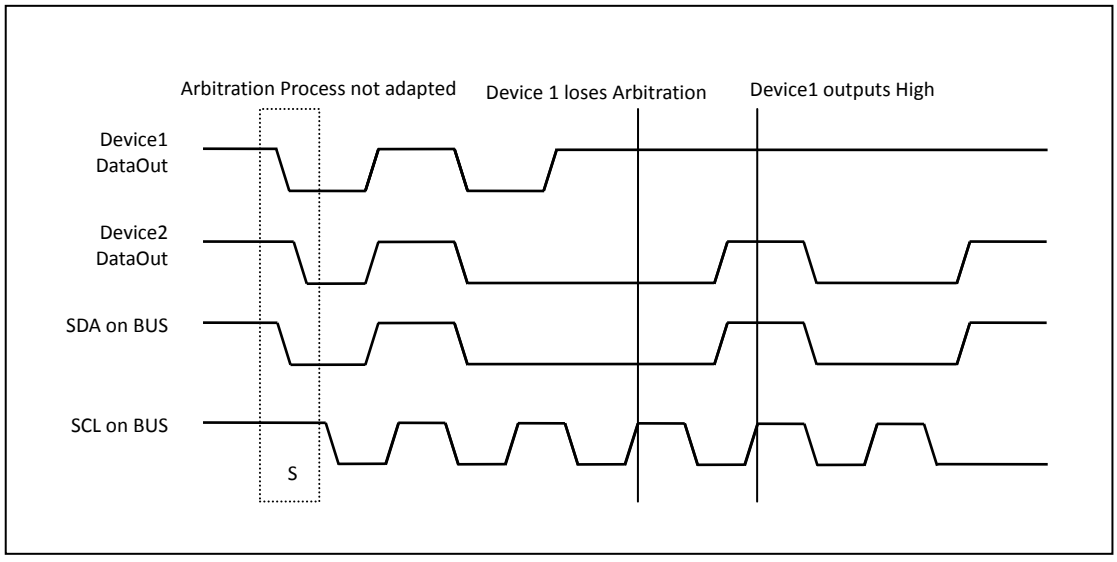


Figure 14-11 Arbitration Procedure Between Two Masters

I^2C Operation

I^2C supports the interrupt operation. Once interrupt is serviced, the IIF (IC.CR[7]) flag is set. ICnSR shows I^2C-bus status information and the SCL line stays “L” before the register is written as a certain value. The status register can be cleared by writing to the status register.

Master Transmitter

The master transmitter shows the flow of transmitter in Master mode (see Figure 14-12).

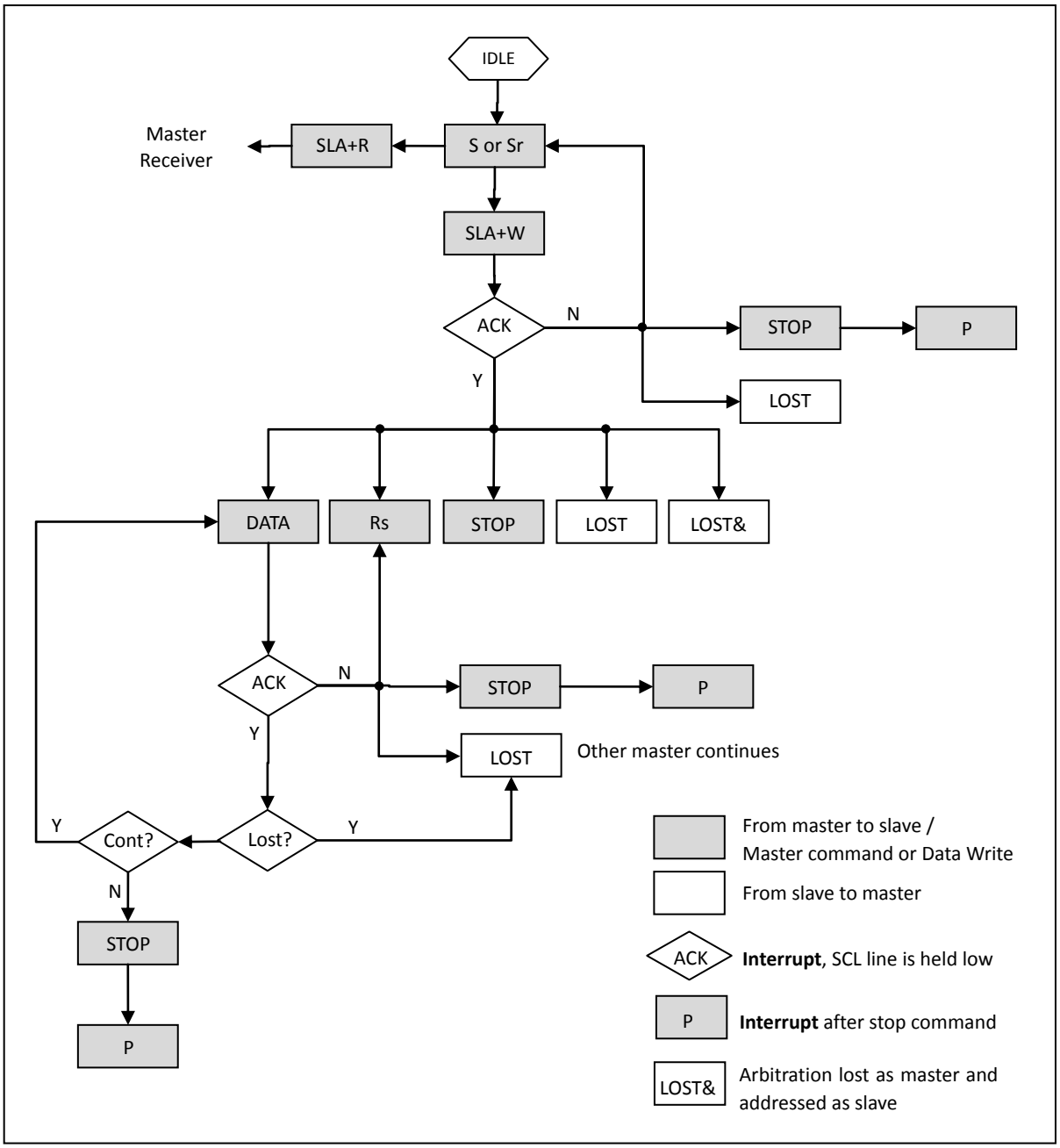


Figure 14-12 Transmitter Flowchart in Master Mode

Master Receiver

The master receiver shows the flow of receiver in Master mode (see Figure 14-13).

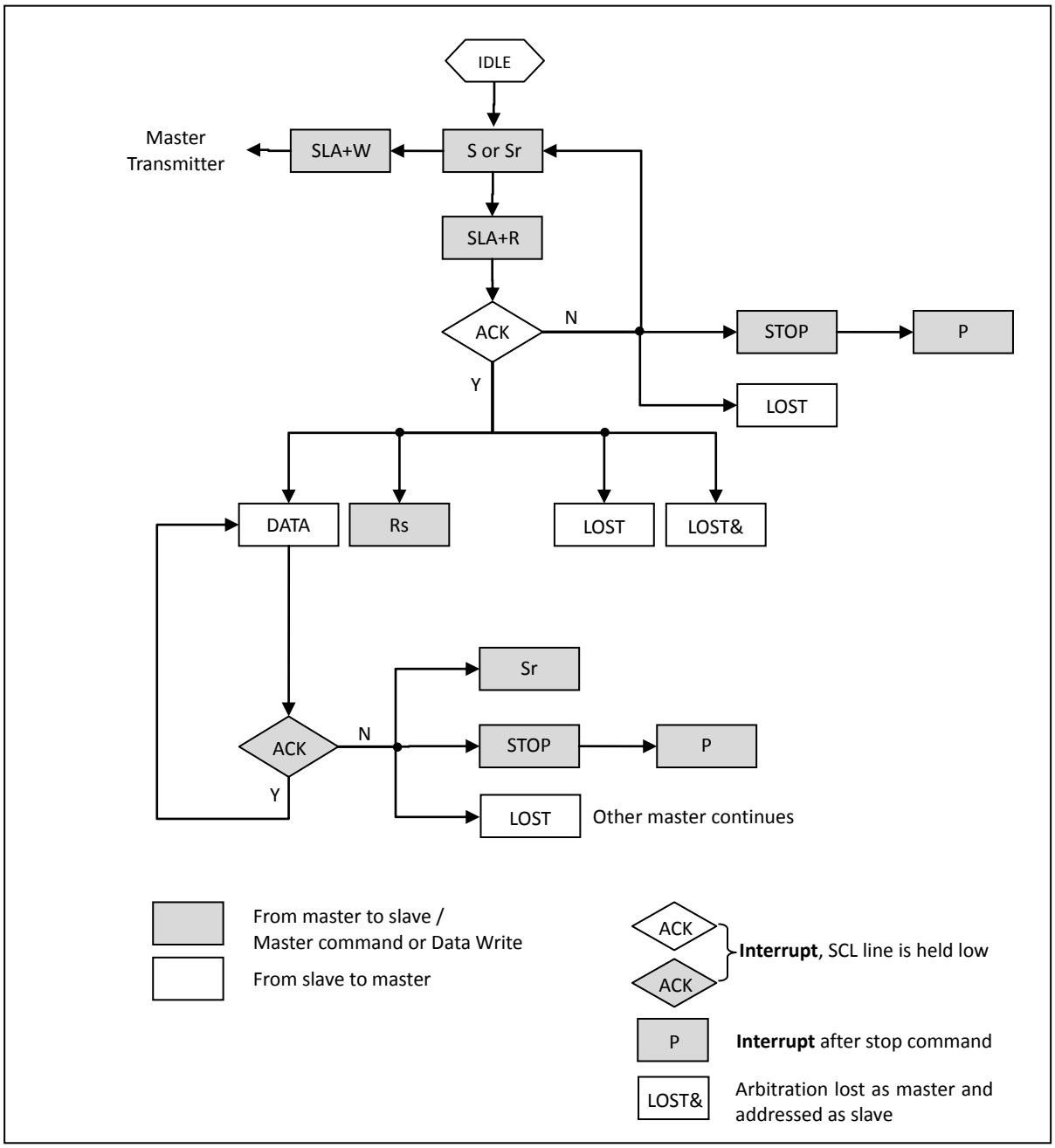


Figure 14-13 Receiver Flowchart in Master Mode

Slave Transmitter

The slave transmitter shows the flow of transmitter in Slave mode (see Figure 14-14).

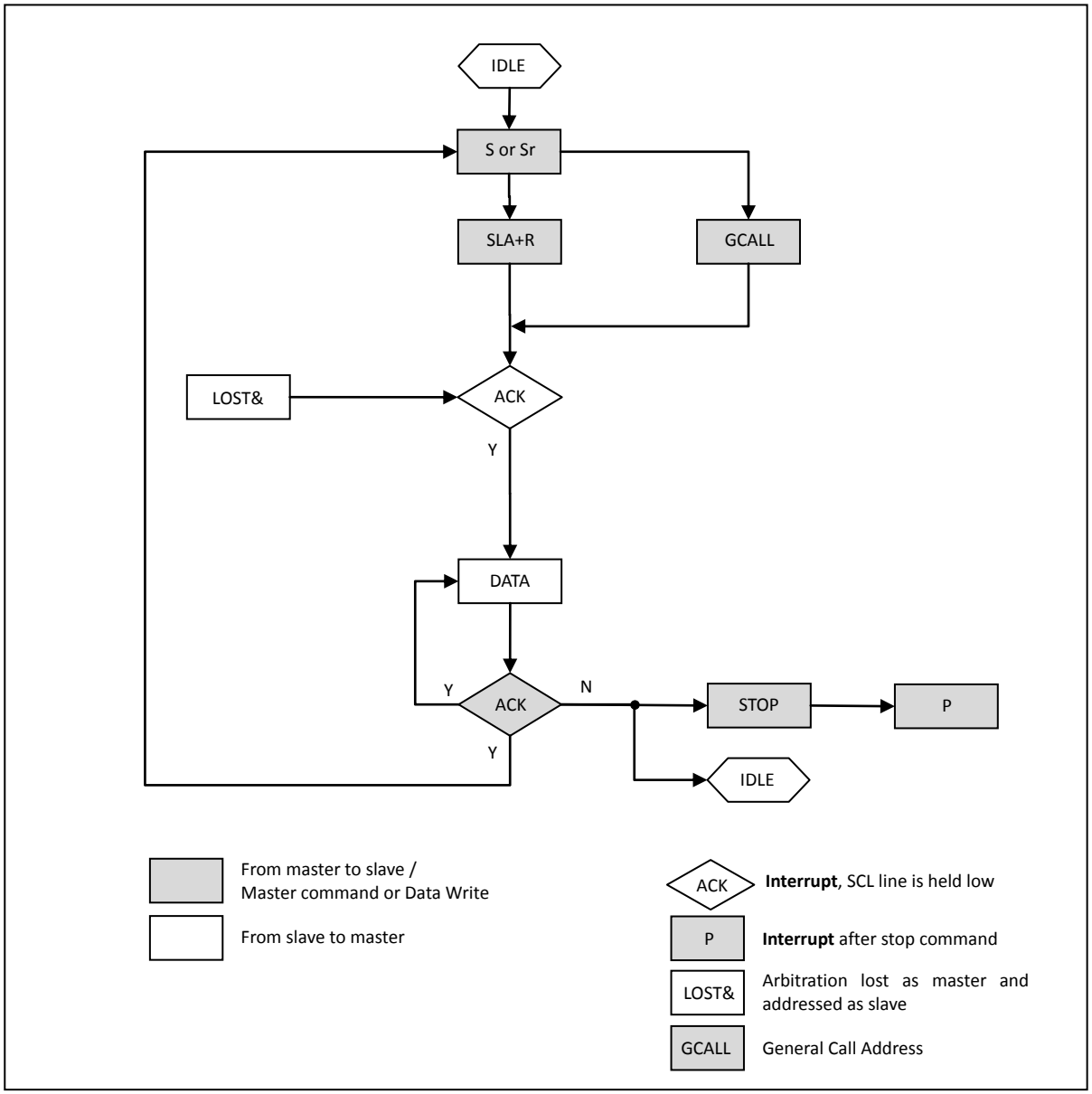


Figure 14-14 Transmitter Flowchart in Slave Mode

Slave Receiver

The slave receiver shows the flow of receiver in Slave mode (see Figure 14-15).

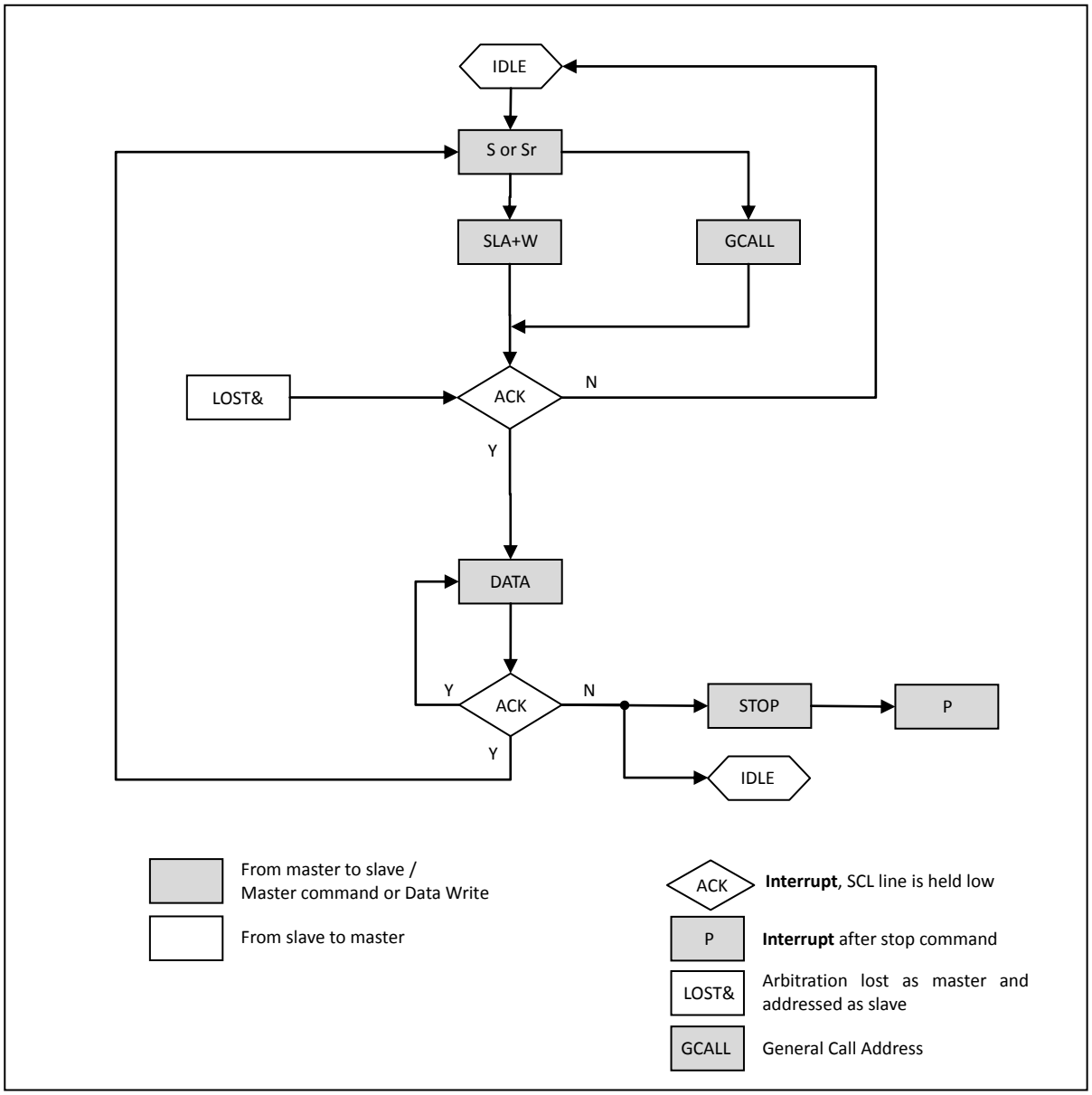


Figure 14-15 Receiver Flowchart in Slave Mode

15. Motor Pulse Width Modulator (MPWM)

Overview

Motor Pulse Width Modulator (MPWM) is a programmable motor controller which is optimized for 3-phase AC and DC motor control applications. It can be used in many other applications that require timing, counting, and comparison features.

MPWM includes 3 channels, each of which controls a pair of outputs that can control a motor.

- 16-bit counter
- 6-channel outputs for motor control
- Dead-time support
- Protection event and over voltage event handling
- 6 ADC trigger outputs
- Interval interrupt mode (period interrupt only)
- Up-down count mode

The MPWM clock source which is MPWM counter clock source will be provided from the SCU block. The MPWM resolution and period will be defined by this MPWM clock configuration. The default MPWM clock is the same as the RINGOSC clock. Prior to enabling the MPWM module, proper MPWM clock selection is required.

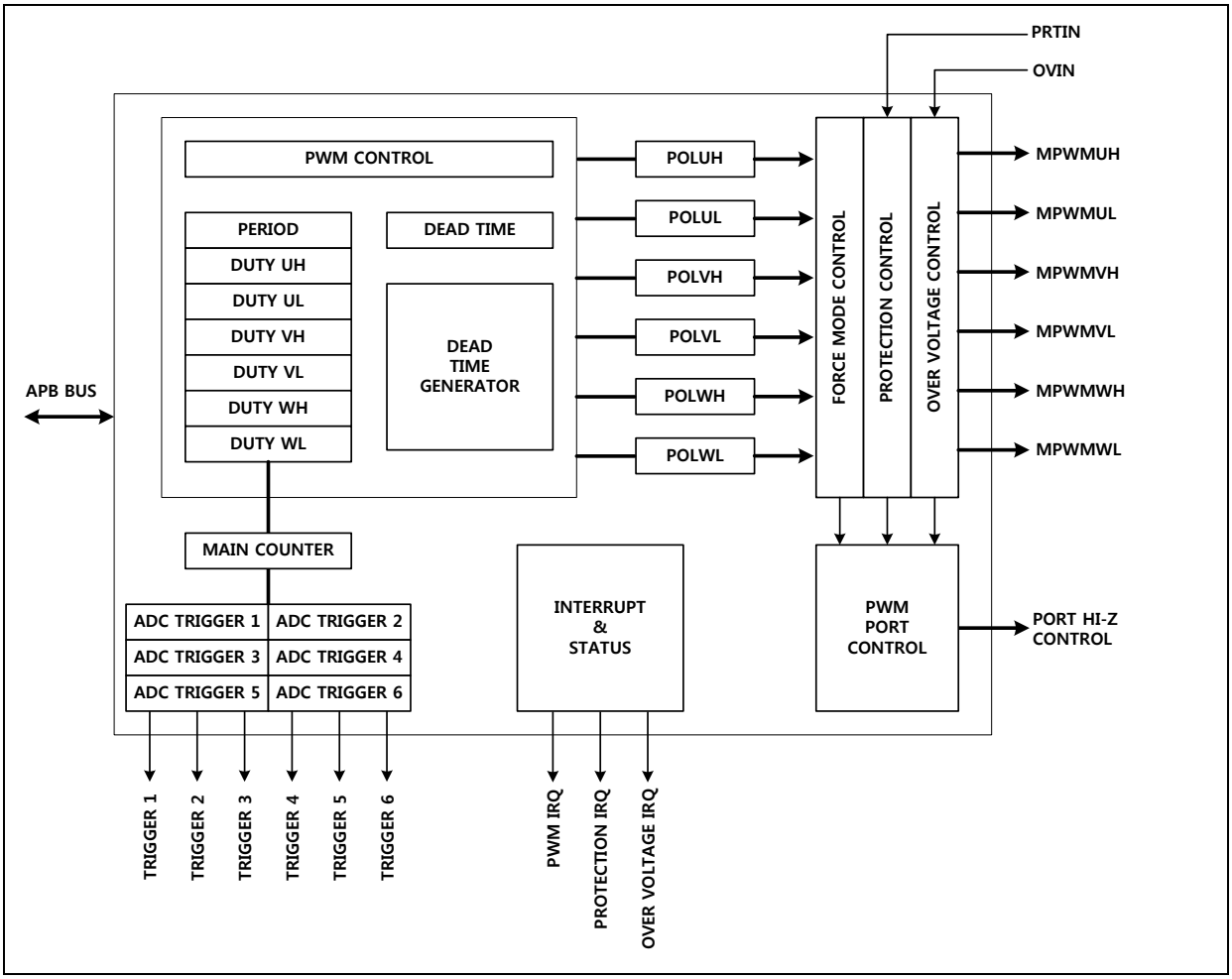


Figure 15-1 Block Diagram

Pin Description

Table 15-1 External Signals

PIN NAME	TYPE	DESCRIPTION
MPWMUH	O	MPWM Phase-U H-side output
MPWMUL	O	MPWM Phase-U L-side output
MPWMVH	O	MPWM Phase-V H-side output
MPWMVL	O	MPWM Phase-V L-side output
MPWMWH	O	MPWM Phase-W H-side output
MPWMWL	O	MPWM Phase-W L-side output
PRTIN	I	MPWM Protection Input
OVIN	I	MPWM Over-voltage Input

Registers

The base address of MPWM is shown in Table 15-2.

Table 15-2 MPWM Base Address

NAME	BASE ADDRESS
MPWM	0x4000_4000

Table 15-3 shows the register memory map.

Table 15-3 MPWM Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
MP.MR	0x0000	RW	MPWM Mode register	0x0000_0000
MP.OLR	0x0004	RW	MPWM Output Level register	0x0000_0000
MP.FOR	0x0008	RW	MPWM Force Output register	0x0000_0000
MP.PRD	0x000C	RW	MPWM Period register	0x0000_0002
MP.DUH	0x0010	RW	MPWM Duty UH register	0x0000_0001
MP.DVH	0x0014	RW	MPWM Duty VH register	0x0000_0001
MP.DWH	0x0018	RW	MPWM Duty WH register	0x0000_0001
MP.DUL	0x001C	RW	MPWM Duty UL register	0x0000_0001
MP.DVL	0x0020	RW	MPWM Duty VL register	0x0000_0001
MP.DWL	0x0024	RW	MPWM Duty WL register	0x0000_0001
MP.CR1	0x0028	RW	MPWM Control register 1	0x0000_0000
MP.CR2	0x002C	RW	MPWM Control register 2	0x0000_0000
MP.SR	0x0030	R	MPWM Status register	0x0000_0000
MP.IER	0x0034	RW	MPWM Interrupt Enable	0x0000_0000
MP.CNT	0x0038	R	MPWM counter register	0x0000_0001
MP.DTR	0x003C	RW	MPWM dead time control	0x0000_0000
MP.PCR0	0x0040	RW	MPWM protection 0 control register	0x0000_0000
MP.PSR0	0x0044	RW	MPWM protection 0 status register	0x0000_0080
MP.PCR1	0x0048	RW	MPWM protection 1 control register	0x0000_0000
MP.PSR1	0x004C	RW	MPWM protection 1 status register	0x0000_0000
-	0x0054	-	Reserved	-
MP.ATR1	0x0058	RW	MPWM ADC Trigger reg1	0x0000_0000
MP.ATR2	0x005C	RW	MPWM ADC Trigger reg2	0x0000_0000
MP.ATR3	0x0060	RW	MPWM ADC Trigger reg3	0x0000_0000
MP.ATR4	0x0064	RW	MPWM ADC Trigger reg4	0x0000_0000
MP.ATR5	0x0068	RW	MPWM ADC Trigger reg5	0x0000_0000
MP.ATR6	0x006C	RW	MPWM ADC Trigger reg6	0x0000_0000

MP.MR MPWM Mode Register

The Motor PWM operation mode register is a 16-bit register.

MP.MR=0x4000_4000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOTORB								UAO		TUP	BUP		MCHMOD		UPDOWN
0								0		0	0		00		0
RW								RW		RW	RW		RW		RW

15	MOTORB	0	Motor mode
		1	Normal mode
7	UAO	0	Update will be executed at designated timing.
		1	Update all duty, period register at once. When UPDATE set, Duty and Period registers are updated after two PWM clocks
5	TUP	0	Period, duty values are not updated at every period match.
		1	Period, duty values are updated at every period match.
4	BUP	0	Period, duty values are not updated at every bottom match
		1	Period, duty values are updated at every bottom match
2 1	MCHMOD	00	2 channels symmetric mode Duty H decides toggle high/low time of H-ch Duty L decides toggle high/low time of L-ch
		01	1 channel asymmetric mode Duty H decides toggle high time of H-ch Duty L decides toggle low time of H-ch L channel become the inversion of H channel
		10	1 channel symmetric mode Duty H decides toggle high/low time of H-ch L channel become the inversion of H channel
		11	Not valid (same with 00)
0	UPDOWN	0	PWM Up count mode (only available when MOTORB='1')
		1	PWM Up/Down count mode (This bit should be '1' if MOTORB='0')

After the initial PWM period and duty is set, the UAO bit should be set once for updating the setting value into internal operating registers. This action will help to transfer the setting data from the user interface register to the internal operating register. The UAO bit should stay at the set state for at least 2 PWM clock periods. If this does not occur, the update command can be missed and internal registers will retain the previous data.

The MCHMOD in the MP.MR field is only effective when MOTORB in MP.MR is a clear "0". Otherwise, the MCHMOD field value will be ignored internally and will retain the "00" value.

The UPDOWN in the MP.MR field is only effective when MOTORB in MP.MR is set to "1". Otherwise, the UPDOWN field value will be ignored internally and will retain the "1" value. In the Motor mode, the counter is always updown count operation.

MP.OLR MPWM Output Level Register

The PWM output level register is an 8-bit register. This register controls the active level of each PWM output port. The default active level is negated when the corresponding bit is set.

The normal level is defined in each operating mode.

MP.OLR=0x4000_4004

7	6	5	4	3	2	1	0
		WHL	VHL	UHL	WLL	VLL	ULL
0	0	0	0	0	0	0	0
		RW	RW	RW	RW	RW	RW

WHL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
VHL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
UHL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
WLL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
VLL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
ULL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L

The normal level is defined in each operating mode as shown in Table 15-4.

Table 15-4 MPWM Output Level Setting

PWM Output	Level	NORMAL mode		MOTOR Mode
		UP Mode	UPDOWN Mode	
WH	Default level	LOW	HIGH	LOW
	Active level	HIGH	LOW	HIGH
WL	Default level	LOW	LOW	HIGH
	Active level	HIGH	HIGH	LOW
VH	Default level	LOW	HIGH	LOW
	Active level	HIGH	LOW	HIGH
VL	Default level	LOW	LOW	HIGH
	Active level	HIGH	HIGH	LOW
UH	Default level	LOW	HIGH	LOW
	Active level	HIGH	LOW	HIGH
UL	Default level	LOW	LOW	HIGH
	Active level	HIGH	HIGH	LOW

The Polarity Control block is shown in Figure 15-2 using the WH signal polarity control example.

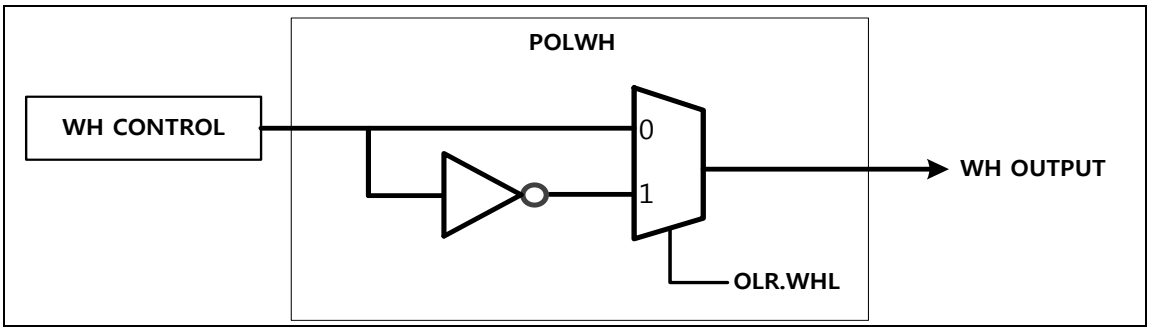


Figure 15-2 Polarity Control Block

MP.FOR MPWM Force Output Register

The PWM force output register is an 8-bit register. The PWM output level can be forced by an abnormal event externally or user-intended condition. When the forced condition occurs, each PWM output level which is programmed in the Force Output register will be forced.

MP.FOR=0x4000_4008

7	6	5	4	3	2	1	0
		WHFL	VHFL	UHFL	WLFL	VLFL	ULFL
0	0	0	0	0	0	0	0
		RW	RW	RW	RW	RW	RW

5	WHFL	Select WH Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
4	VHFL	Select VH Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
3	UHFL	Select UH Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
2	WLFL	Select WL Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
1	VLFL	Select VL Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
0	ULFL	Select UL Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’

MP.CR1 MPWM Control Register 1

The PWM Control Register 1 is a 16-bit register.

MP.CR1=0x4000_4028

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					IRQN										PWMEN
					000				0	0	0	0	0	0	0
					RW										RW

10 8	IRQN	IRQ interval number (Every 1~8th PRDIRQ,BOTIRQ,ATRn)
0	PWMEN	PWM enable When this bit set 0, the PWM block stay in reset state but user interface can be accessed. To operate the PWM block, this bit should be set 1.

Basically, PRDIRQ and BOTIRQ are generated every period. However, the interrupt interval can be controlled from 0 to 8 periods. When IRQN.CR1 = 0, the interrupt is requested every period, otherwise the interrupt is requested every (IRQN+1) times of period.

MP.CR2 MPWM Control Register 2

The PWM Control Register 2 is an 8-bit register.

MP.CR2=0x4000_402C

7	6	5	4	3	2	1	0
HALT							PSTART
0	0	0	0	0	0	0	0
RW							RW

7	HALT	PWM HALT (PWM counter stop but not reset) PWM outputs keep previous state	
0	PSTART	0	PWM counter stop and clear
		1	PWM counter start (will be resynced @PWM clock twice) PWMEN should be “1” to start PWM counter

MP.PRD MPWM Period Register

The PWM Period register is a 16-bit register.

MP.PRD=0x4000400C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERIOD															
0x0002															
RW															

15 0	PERIOD	16-bit PWM period. It should be larger than 0x0010 (if Duty is 0x0000, PWM will not work)

MP.DUH MPWM Duty UH Register

The PWM UH channel duty register is a 16-bit register.

MP.DUH=0x4000_4010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15 0	DUTY	16-bit PWM Duty for UH output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MP.DVH MPWM Duty VH Register

The PWM VH channel duty register is a 16-bit register.

MP.DVH=0x4000_4014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15 0	DUTY	16-bit PWM Duty for VH output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MP.DWHMPWM Duty WH Register

The PWM WH channel duty register is a 16-bit register.

MP.DWH=0x4000_4018

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15	DUTY	16-bit PWM Duty for WH output.
0		It should be larger than 0x0001
		(if Duty is 0x0000, PWM will not work)

MP.DUL MPWM Duty UL Register

The PWM UL channel duty register is a 16-bit register.

MP.DUL=0x4000_401C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15	DUTY	16-bit PWM Duty for UL output.
		It should be larger than 0x0001
0		(if Duty is 0x0000, PWM will not work)

MP.DVL MPWM Duty VL Register

The PWM VL channel duty register is a 16-bit register.

MP.DVL=0x4000_4020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15	DUTY	16-bit PWM Duty for VL output.
		It should be larger than 0x0001
0		(if Duty is 0x0000, PWM will not work)

MP.DWL MPWM Duty WL Register

The PWM WL channel duty register is a 16-bit register.

MP.DWL=0x4000_4024

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15	DUTY	16-bit PWM Duty for WL output. It should be larger than 0x0001
0		(if Duty is 0x0000, PWM will not work)

MP.IER MPWM Interrupt Enable Register

The PWM Interrupt Enable Register is an 8-bit register.

MP.IER=0x4000_4034

7	6	5	4	3	2	1	0
PRDIEN	BOTIEN	WHIE	VHIE	UHIE	WLIE	VLIE	ULIE
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

7	PRDIEN	PWM Counter Period Interrupt enable
		0 interrupt disable
		1 interrupt enable
6	BOTIEN	PWM Counter Bottom Interrupt enable
		0 interrupt disable
		1 interrupt enable
5	WHIE ATR6IE	WH Duty or ATR6 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
4	VHIE ATR5IE	VH Duty or ATR5 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
3	UHIE ATR4IE	UH Duty or ATR4 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
2	WLIE ATR3IE	WL Duty or ATR3 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
1	VLIE ATR2IE	VL Duty or ATR2 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
0	ULIE ATR1IE	UL Duty or ATR1 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable

MP.IER[5:0] control bits are shared by the duty match interrupt event and ADC trigger match interrupt event. When ADC trigger mode is disabled, the interrupt is generated by the duty match condition. In other instances, the interrupt is generated by the ADC trigger counter match condition. The ADC trigger mode is selected by the ATMOD bit field in the ATRm register.

MP.SR MPWM Status Register

The PWM Status Register is a 16-bit register.

MP.SR=0x4000_4030

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOWN	IRQCNT							PRDIF	BOTIF	DWHIF ATR6F	DVHIF ATR5F	DUHIF ATR4F	DWLIF ATR3F	DVLIF ATR2F	DULIF ATR1F
0	000			0	0	0	0	0	0	0	0	0	0	0	0
RW	RW							RW	RW	RW	RW	RW	RW	RW	RW

15	DOWN	0	PWM Count Up
		1	PWM Count Down
14 12	IRQCNT[2:0]		Interrupt count number of period match (Interval PRDIRQ mode)
7	PRDIF		PWM Period Interrupt flag(write “1” to clear flag)
		0	No interrupt occurred
		1	Interrupt occurred
6	BOTIF		PWM Bottom Interrupt flag(write “1” to clear flag)
		0	No interrupt occurred
		1	Interrupt occurred
5	DWHIF ATR6F		PWM duty WH interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR6 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
4	DVHIF ATR5F		PWM duty VH interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR5 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
3	DUHIF ATR4F		PWM duty UH interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR4 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
2	DWLIF ATR3F		PWM duty WL interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR3 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
1	DVLIF ATR2F		PWM duty VL interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR2 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
0	DULIF ATR1F		PWM duty UL interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR1 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred

MP.SR[5:0] status bits are shared by the duty match interrupt event and ADC trigger match interrupt event. When the ADC trigger mode is disabled, the interrupt is generated by the duty match condition. In other instances, the interrupt is generated by the ADC trigger counter match condition. The ADC trigger mode is selected by the ATMOD bit field in the ATRm register.

MP.CNT MPWM Counter Register

The PWM Counter Register is a 16-bit read-only register.

MP.CNT=0x4000_4038

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
0x0000															
RW															

	CNT	PWM Counter Value

MP.DTR MPWM Dead Time Register

The PWM Dead Time register is a 16-bit register.

MP.DTR=0x4000_403C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEN	PSHRT						DTCLK	DT							
0	0	0	0	0	0	0	0	0x00							
RW							RW	RW							

15	DTEN	Dead-time function enable 2 channel symmetric mode does not support dead time function. It should be disabled in 2 channel symmetric mode. 0 Disable Dead-time function 1 Enable Dead-time function
14	PSHRT	Protect short condition This function is effective only for 2 channel symmetric mode. For 1 channel mode, never activated on both H-side and L-side at same time. L-side is always opposite of H-side. 0 Enable output short protection function. (Turn off both output when both H-side and L-side are active.) 1 Disable output short protection function.
8	DTCLK	Dead-time prescaler 0 Dead time counter uses PWM CLK/4 1 Dead time counter uses PWM CLK/16
7 0	DT	Dead Time value (Dead time setting makes output delay of 'low to high transition' in normal polarity) 0x01 ~0xFF : Dead time

The Protect Short condition is only for internal PWM level, not for external PWM level. When the internal signal of H-side and L-side are the same high level, the protection short function works to force both H-side and L-side to low level.

MP.PCRn MPWM Protection 0,1 Control Register

The PWM Protection Control register is a 16-bit register.

MP.PCR0=0x4000_4040, MP.PCR1=0x4000_4048

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROTEN	PROTPOL				PROTD			PROTIE		WHPROTM	VHPROTM	UHPROTM	WLPROTM	VLPROTM	ULPROTM
0	0				000			0		0	0	0	0	0	0
RW	RW				RW			RW		RW	RW	RW	RW	RW	RW

15	PROT0EN	Enable Protection Input 0
14	PROT0POL	Select Protection Input Polarity 0: Low-Active 1: High-Active
10 8	PROTD	Protection Input debounce 0 – no debounce 1~7 – debounce by (MPWMCLK * PROTD[2:0])
7	PROTIE	Protection Interrupt enable 0 Disable protection interrupt 1 Enable protection interrupt
5	WHPROTM	Activate W-phase H-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
4	VHPROTM	Activate V-phase H-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
3	UHPROTM	Activate U-phase H-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
2	WLPROTM	Activate W-phase L-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
1	VLPROTM	Activate V-phase L-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
0	ULPROTM	Activate U-phase L-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value

Note: MP.PCR0 is related to the PRTIN pin and MP.PCR1 is related to OVIN.

MP.PSRn MPWM Protection 0,1 Status Register

The PWM Protection Status Register is a 16-bit register.

This register indicates which outputs are disabled. Users have the ability to set the output masks manually.

If PROTKEY is not written when writing any value, the written values are ignored.

MP.PSR0=0x4000_4044, MP.PSR1=0x4000_404C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROTKEY								PROTIF		WHPROTF	VHPROTF	UHPROTF	WLPROTF	VLPROTF	ULPROTF
-								0		0	0	0	0	0	0
WO								RC		RW	RW	RW	RW	RW	RW

15 8	PROTKEY	Protection Clear Access Key To clear flags, write the key with protection flag (PSR0 key is 0xCA and PSR1 key is 0xAC) Writing without PROTKEY prohibited.
7	PROTIF	Protection Interrupt status 0 No Protection Interrupt 1 Protection Interrupt occurred
5	WHPROT	Activate W-phase H-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
4	VHPROT	Activate V-phase H-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
3	UHPROT	Activate U-phase H-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
2	WLPROT	Activate W-phase L-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
1	VLPROT	Activate V-phase L-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
0	ULPROT	Activate U-phase L-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled

If the PROTEN bit in the MP.PCRn register is enabled, on any asserting signal on the external protection pins, the PWM output will be prohibited when output values are defined in the MP.FOLR register.

Users can prohibit the output manually by writing the designated value into the MP.PSRn register.

Note: MP.PSR0 is related to the PRTIN pin and MP.PSR1 is related to OVIN.

MP.ATRm MPWM ADC Trigger Counter m Register

MP.ATR1	MPWM ADC Trigger Counter 1 Register
MP.ATR2	MPWM ADC Trigger Counter 2 Register
MP.ATR3	MPWM ADC Trigger Counter 3 Register
MP.ATR4	MPWM ADC Trigger Counter 4 Register
MP.ATR5	MPWM ADC Trigger Counter 5 Register
MP.ATR6	MPWM ADC Trigger Counter 6 Register

The PWM ADC Trigger Counter Register is a 32-bit register.

MP.ATR1=0x4000_4058
MP.ATR2=0x4000_405C
MP.ATR3=0x4000_4060
MP.ATR4=0x4000_4064
MP.ATR5=0x4000_4068
MP.ATR6=0x4000_406C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ATUDT		ATMOD		ATCNT																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0x0000															
												RW		RW		RW															

19	ATUDT	Trigger register update mode
		0 ADC trigger value applied at period match event (at the same time with period and duty registers update)
		1 Trigger register update mode When this bit set, written Trigger register values are sent to trigger compare block after two PWM clocks (through synchronization logic)
17	ATMOD	ADC trigger Mode register
16		00 ADC trigger Disable
		01 Trigger out when up count match
		10 Trigger out when down count match
		00 Trigger out when up-down count match
15	ATCNT	ADC Trigger counter
0		(it should be less than PWM period)

Functional Description

The MPWM includes three channels, each of which controls a pair of outputs that in turn can control an off-chip component. In normal PWM mode, each channel runs independently. 6 PWM outputs can be generated.

Each PWM output is built with various settings. Figure 15-3 shows the flow for generating PWM output signal.

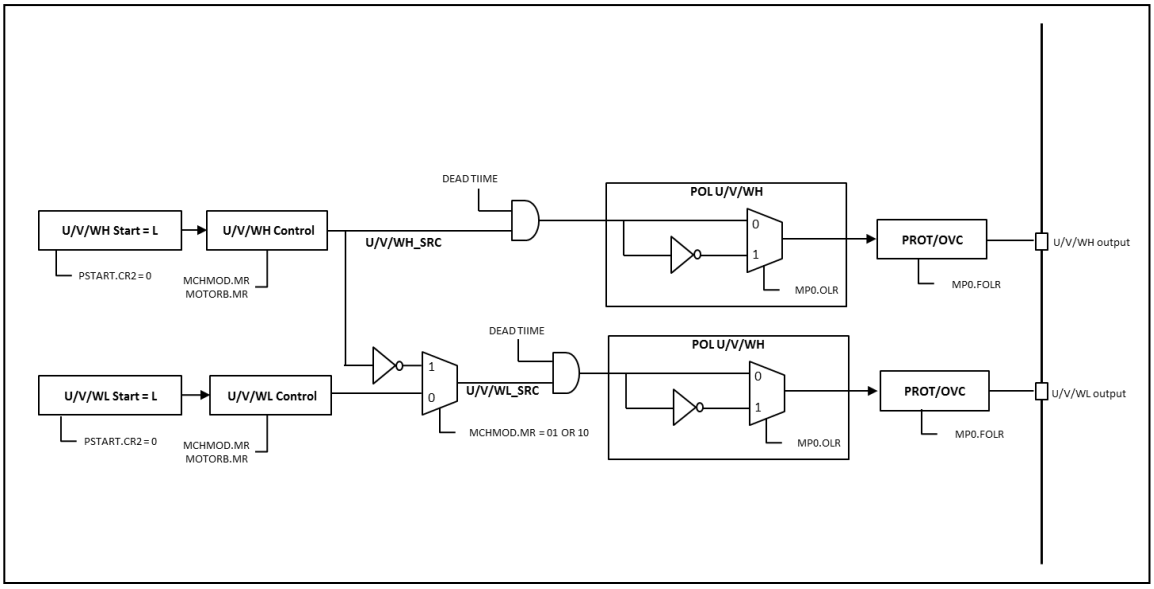


Figure 15-3 PWM Output Generation Chain

Normal PWM UP Count Mode Timing

In normal PWM mode, each channel runs independently. Six PWM outputs can be generated. The example waveform is shown in Figure 15-4. Before PSTART is activated, the PWM output will stay at default value L. When PSTART is enabled, the period counter starts up count until the MP.PRD count value. In the first period, the MPWM does not generate a PWM pulse.

The PWM pulse is generated from the second period. The active level is driven at the start of the counter value during duty value time.

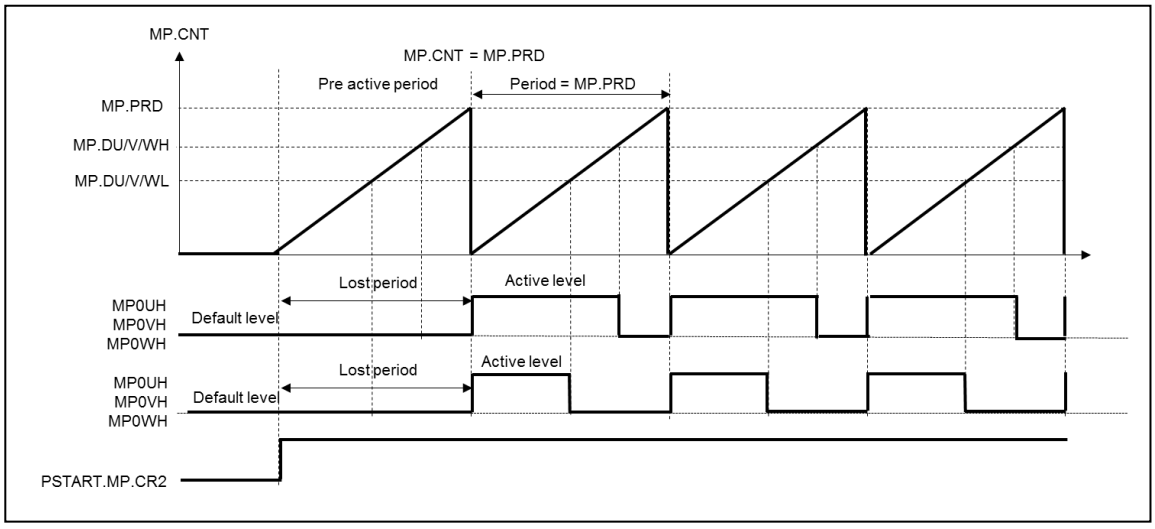


Figure 15-4 UP Count Mode Waveform (MOTORB=1, UPDOWN=0)

Normal PWM UP/DOWN Count Mode Timing

The basic operation of UP/DOWN count mode is the same as UP count mode except the one period is twice the UP count mode. The default active level is opposite in a pair PWM output. This output polarity can be controlled by the MP.OLR register.

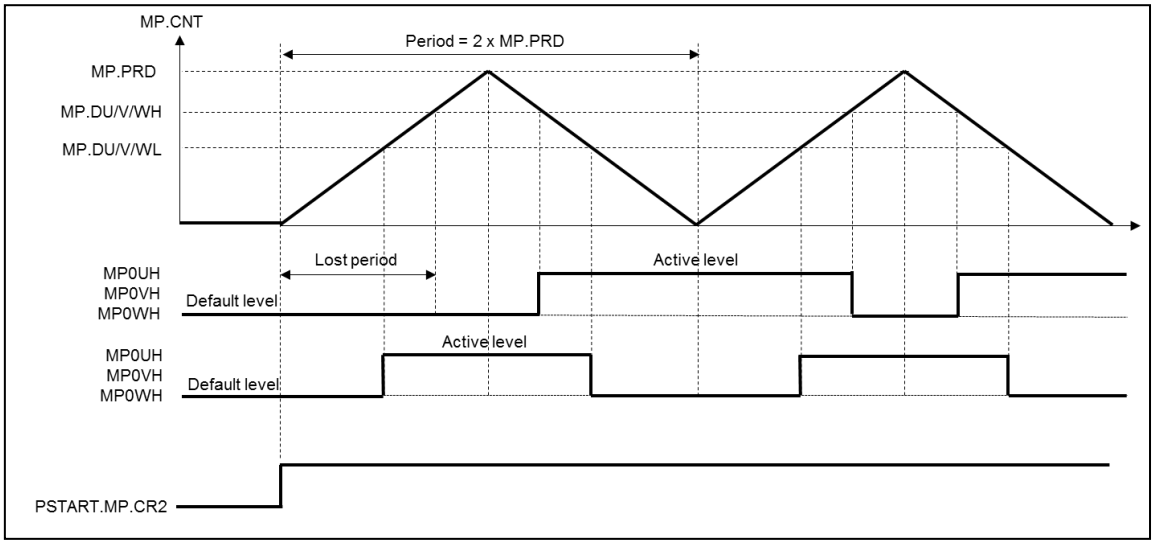


Figure 15.1 UP/DOWN Count Mode Waveform (MOTORB=0, MCHMOD=0, UPDOWN=1)

Motor PWM 2-Channel Symmetric Mode Timing

The motor PWM operation has three types of operating modes: 2-Channel Symmetric mode, 1-Channel Symmetric mode, and 1-Channel Asymmetric mode.

Figure 15-5 is for 2 channel symmetric mode waveform.

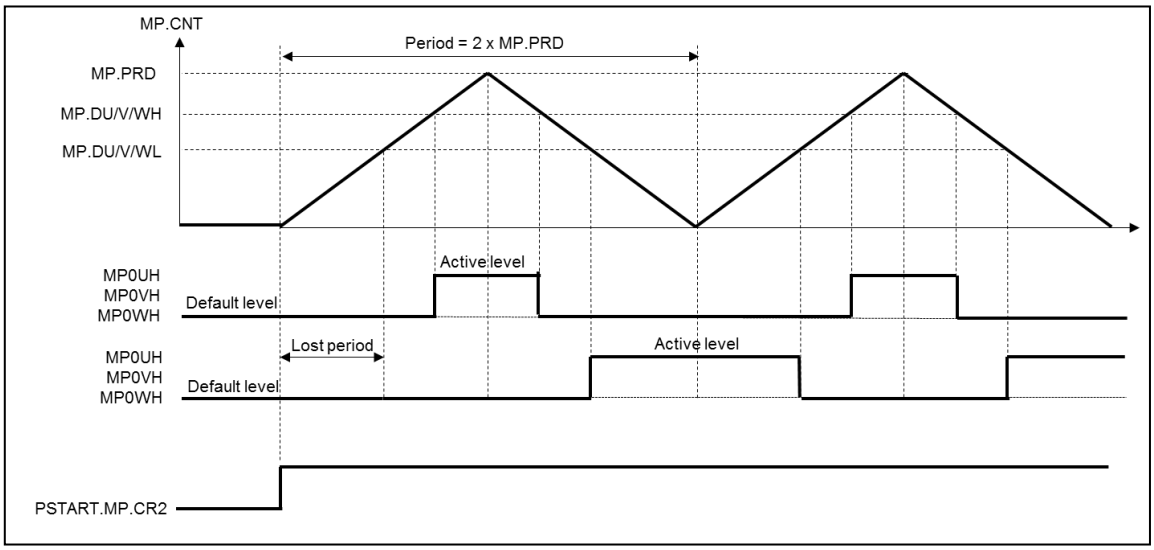


Figure 15-5 2-Channel Symmetric Mode Waveform (MOTORB=0, MCHMOD=00)

The default start level of both H-side and L-side is low. For the H-side, the PWM output level is changed to active level when the duty level is matched in up count period and is returned to the default level when the duty level is matched in down count period.

The symmetrical feature appears in each channel that is controlled by the corresponding duty register value.

Motor PWM 1-Channel Asymmetric Mode Timing

The 1-Channel Asymmetric mode makes asymmetric duration pulses which are defined by the H-side and L-side duty register. Therefore, the L-side signal is always the negative signal of H-side. During up count period, the H-side duty register matching condition makes the active level pulse and during down count period, the L-side duty register matching condition makes the default level pulse.

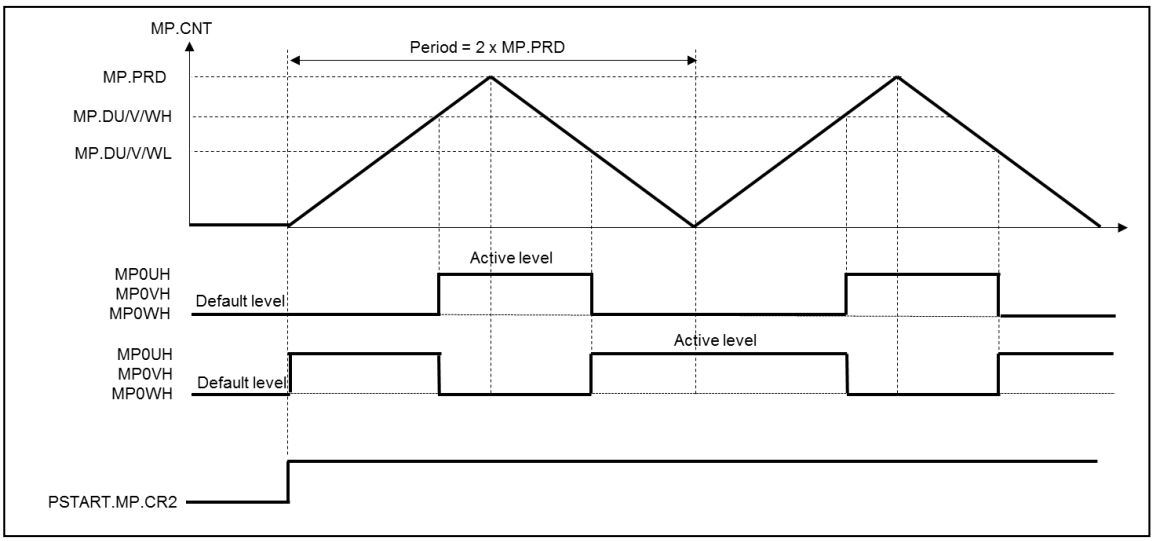


Figure 15.2 1-Channel Asymmetric Mode Waveform (MOTORB=0, MCHMOD=01)

The default start level of both H-side and L-side is low. For the H-side, the PWM output level is changed to active level when the H-side duty level is matched in up count period and is returned to the default level when the L-side duty level is matched in down count period.

When the PSTART is set, the L-side PWM output is changed to the active level, then the L-side PWM output is the inverse output of H-side output.

Motor PWM 1-Channel Symmetric Mode Timing

The 1-channel symmetric mode makes symmetric duration pulses which are defined by the H-side DUTY register. Therefore, the L-side signal is always the negative signal of H-side. During up count period, the H-side DUTY register matching condition makes the active level pulse and during down count period, the H-side DUTY register matching condition also makes the default level pulse.

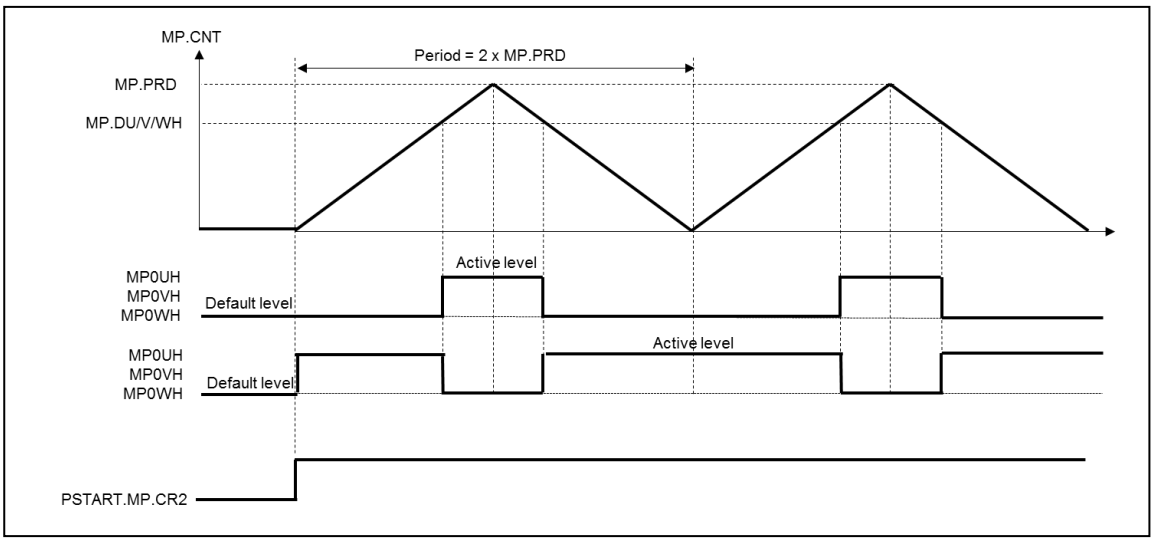


Figure 15.3 1-Channel Symmetric Mode Waveform (MOTORB=0, MCHMOD=10)

The default start level of both H-side and L-side is low. For the H-side, the PWM output level is changed to active level when the H-side duty level is matched in up count period and is returned to the default level when the H-side duty level is matched again in down count period.

When the PSTART is set, the L-side PWM output is changed to the active level, then the L-side PWM output is the inverse output of H-side output.

PWM Dead-time Operation

To prevent an external short condition, the MPWM provides dead time functionality. This function is only available for Motor PWM mode. When either H-side or L-side output changes to active level, dead time will be inserted if the DTEN.MP.DTR bit is enabled.

The duration of dead time is determined by the value in the DT.MP.DTR[7:0] field.

When DTCLK = 0, the dead time duration = DT[7:0] * (PWM clock period * 4)

When DTCLK = 1, the dead time duration = DT[7:0] * (PWM clock period * 16)

When the PWM counter reaches duty value, the PWM output is masked and the dead time counter starts to run. When the dead time counter reaches the value in the DT[7:0] register, the output mask is disabled.

Figure 15-6 is an example of dead time operation in 1-Channel Symmetric mode.

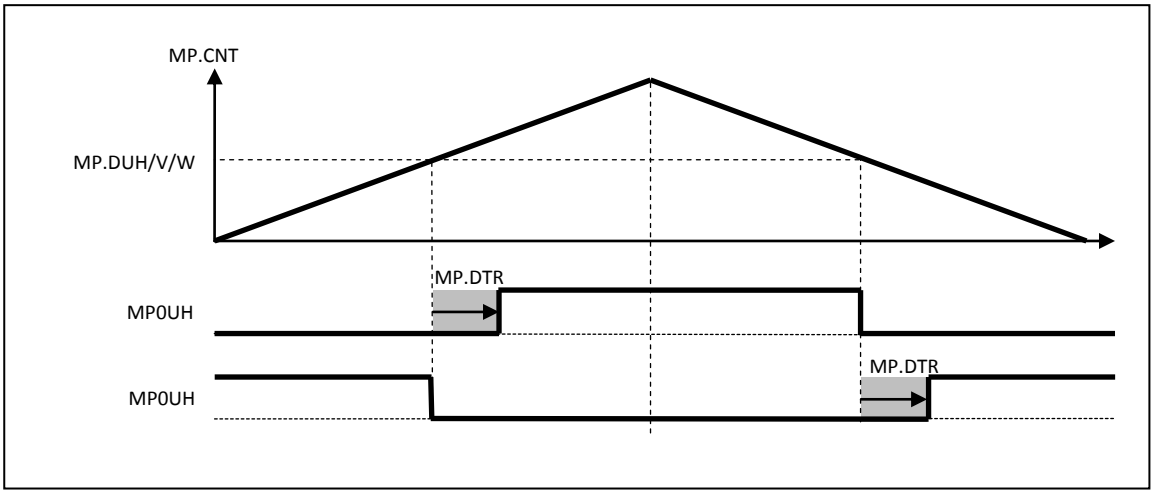


Figure 15-6 PWM Dead-time Operation Timing Diagram (Symmetric Mode)

Figure 15-7 shows an example of 1-Channel Asymmetric mode operation.

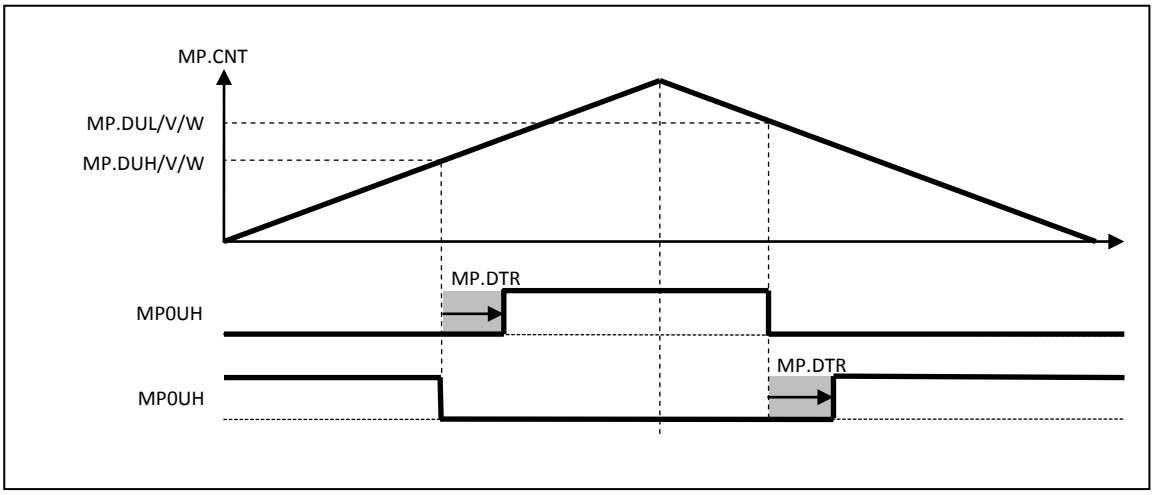


Figure 15-7 PWM Dead-time Operation Timing Diagram (Asymmetric Mode)

For 2-Channel Symmetric mode, the dead time function is not available. Therefore, the dead condition is generated by each channel's duty control.

MPWM Dead-time Timing Examples in Special Case

The following figures show how dead-time operates.

An example of normal dead time is explained. Dead time masking is activated at duty match time and the dead time counter runs. When the dead time counter reaches the dead time value, the mask is disabled.

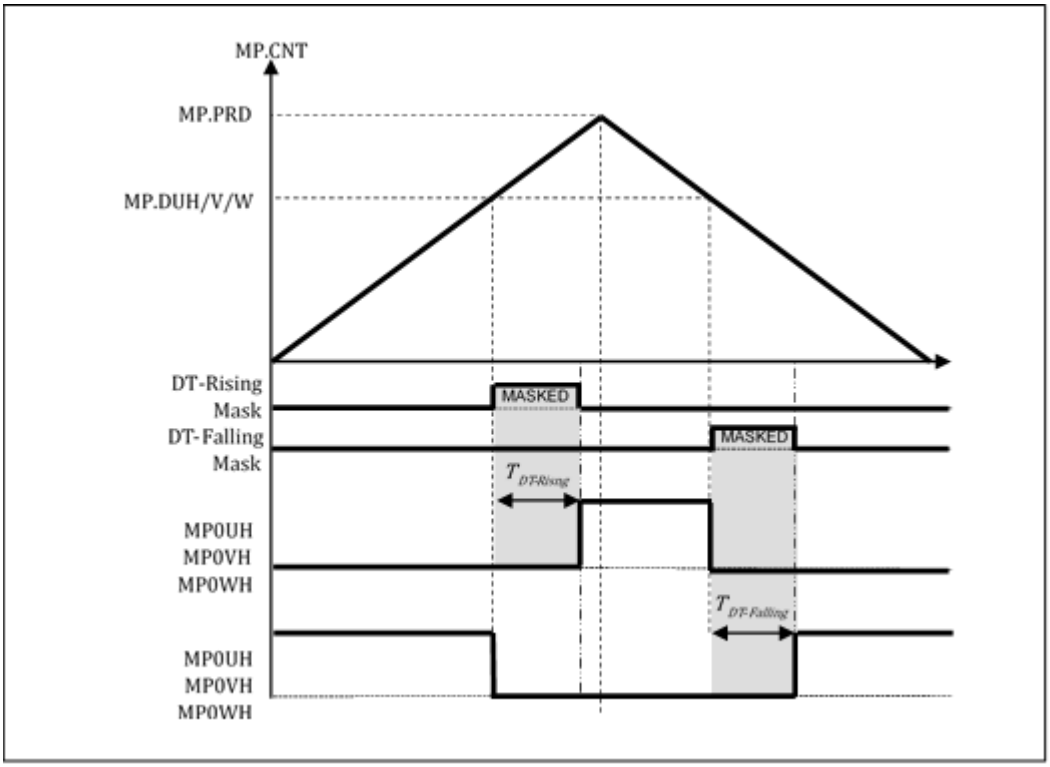


Figure 15.4 Normal Dead-time Operation (T_{DUTY}>T_{DT})

The following images show special instances of dead time configuration.

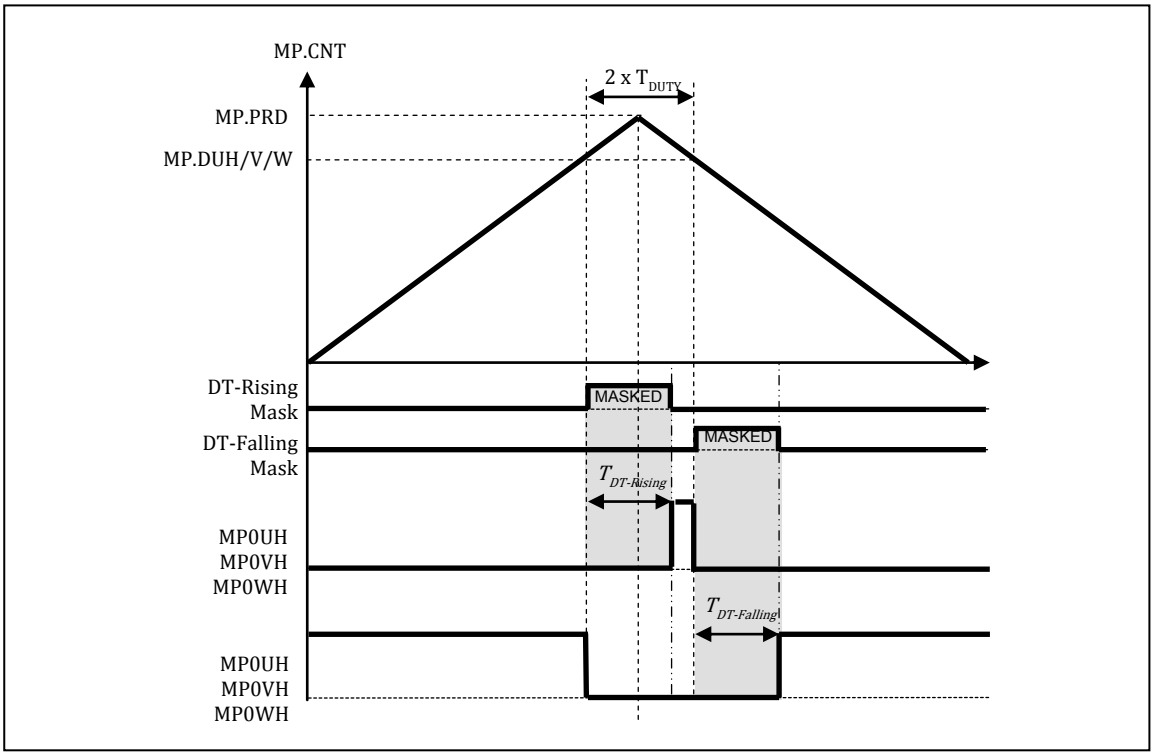


Figure 15-8 Minimum H-side Pulse Timing (T_{DUTY}<T_{DT}<2xT_{DUTY})

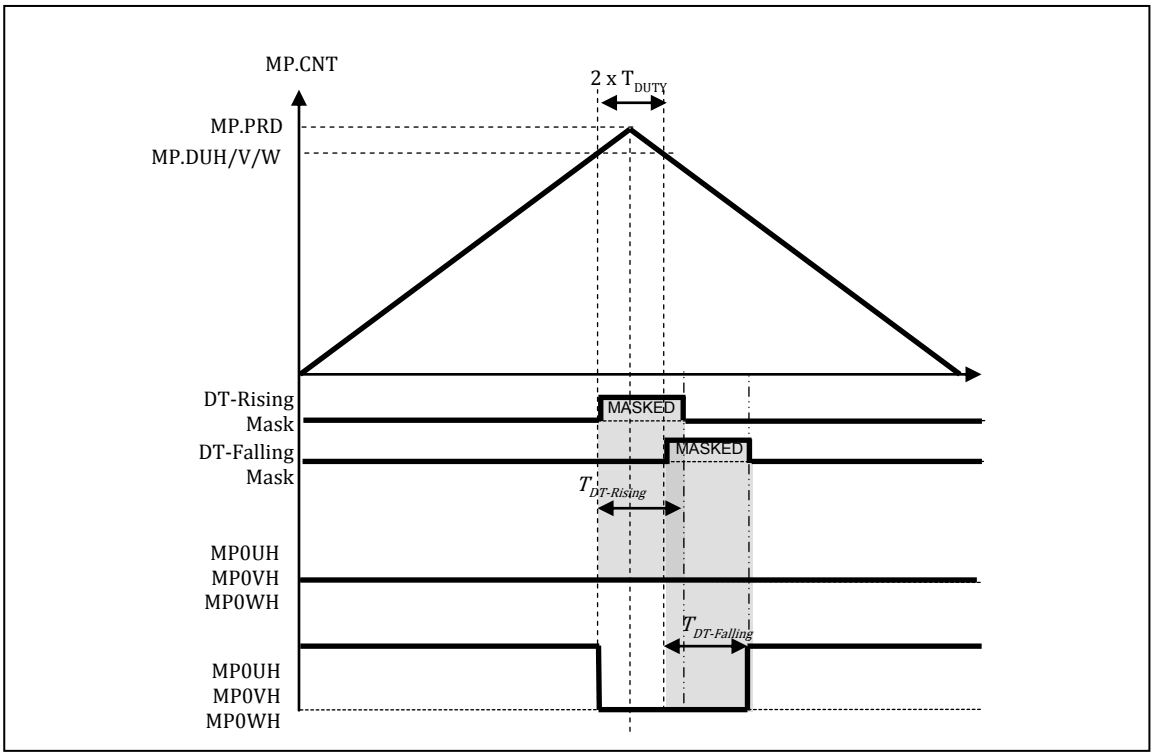


Figure 15-9 Zero H-side Pulse Timing (T_{DT}>2xT_{DUTY})

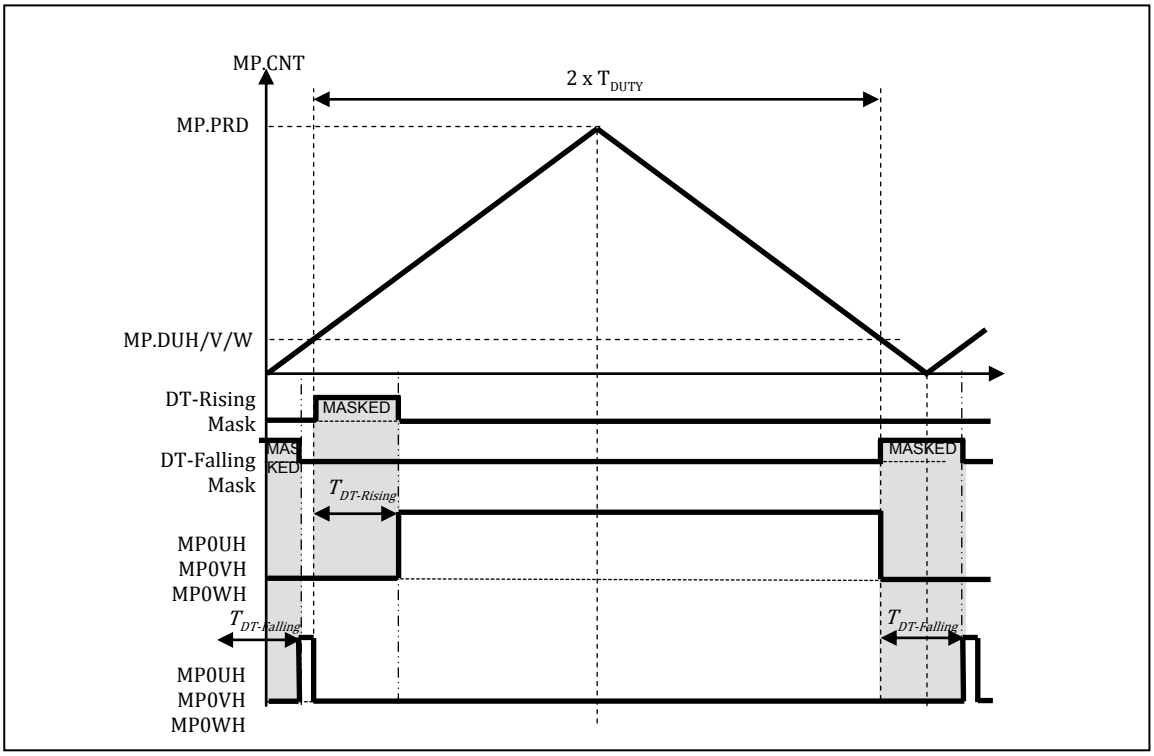


Figure 15-10 Minimum L-side Pulse Timing (T_{DT}<Period-T_{DUTY})

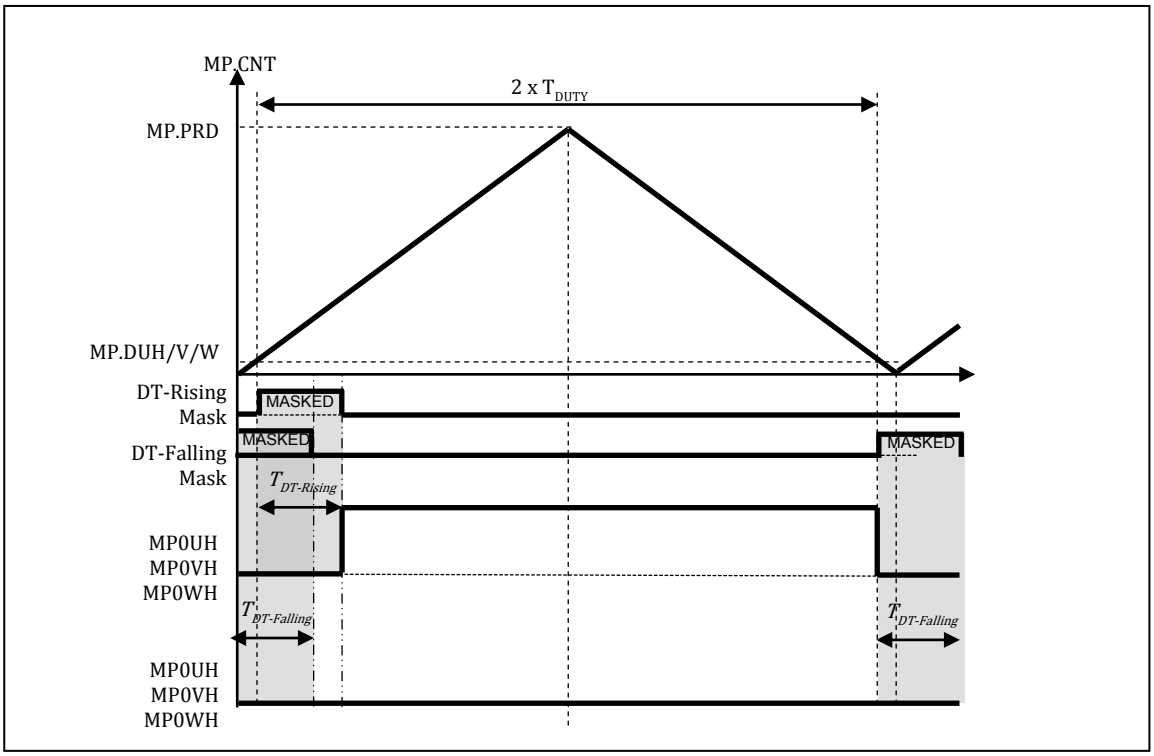


Figure 15-11 Zero L-side Pulse Timing (T_{DT}>Period-T_{DUTY})

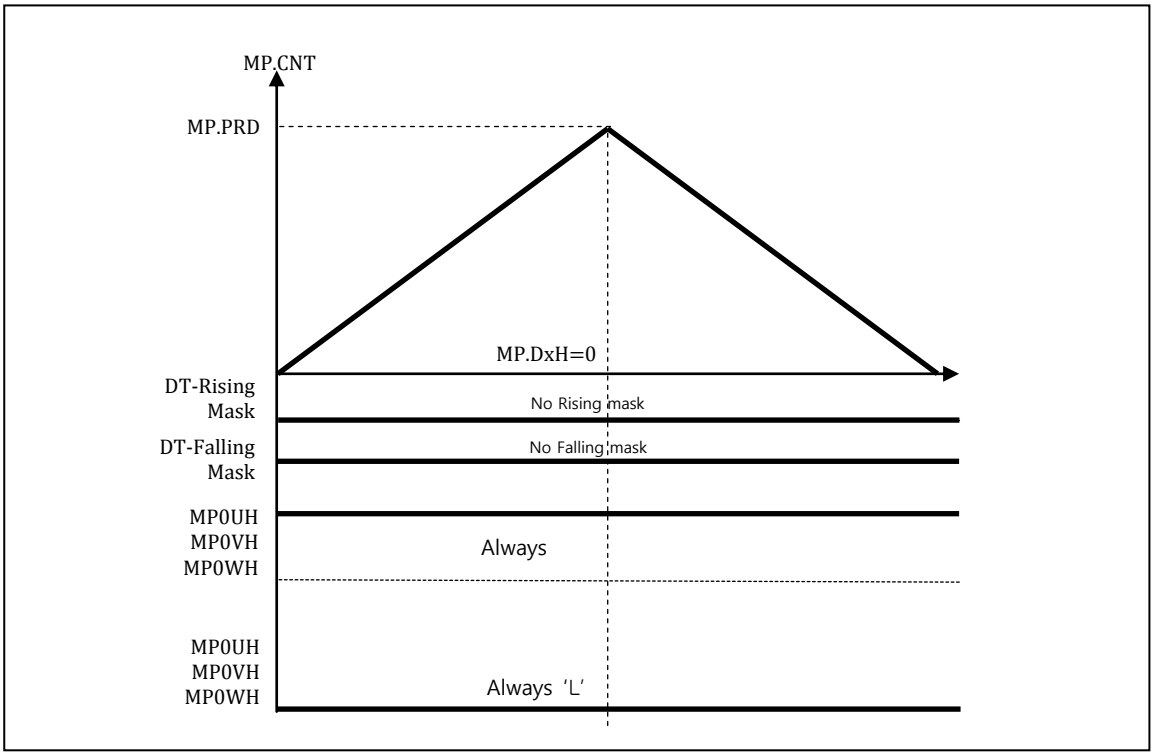


Figure 15-12 H-side Always On (T_{DUTY}=Period: Dead-time Disabled)

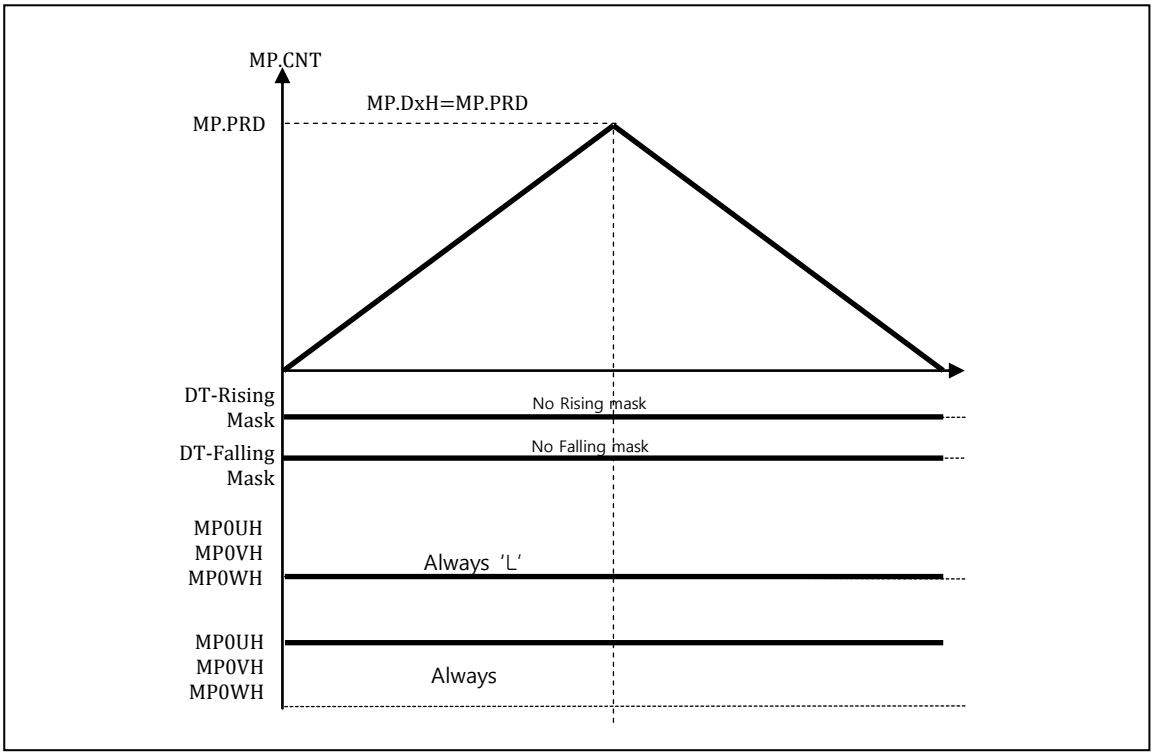


Figure 15-13 L-side Always On (T_{DUTY}='0': Dead-time Disabled)

Symmetrical Mode vs Asymmetrical Mode

In Symmetrical mode, the wave form is between the up and down counters. The same duty value is used for both the up and down counter matches. The on time and off time is the same between the up and down counters. The end result is that in a period, the duty time is centered in the period.

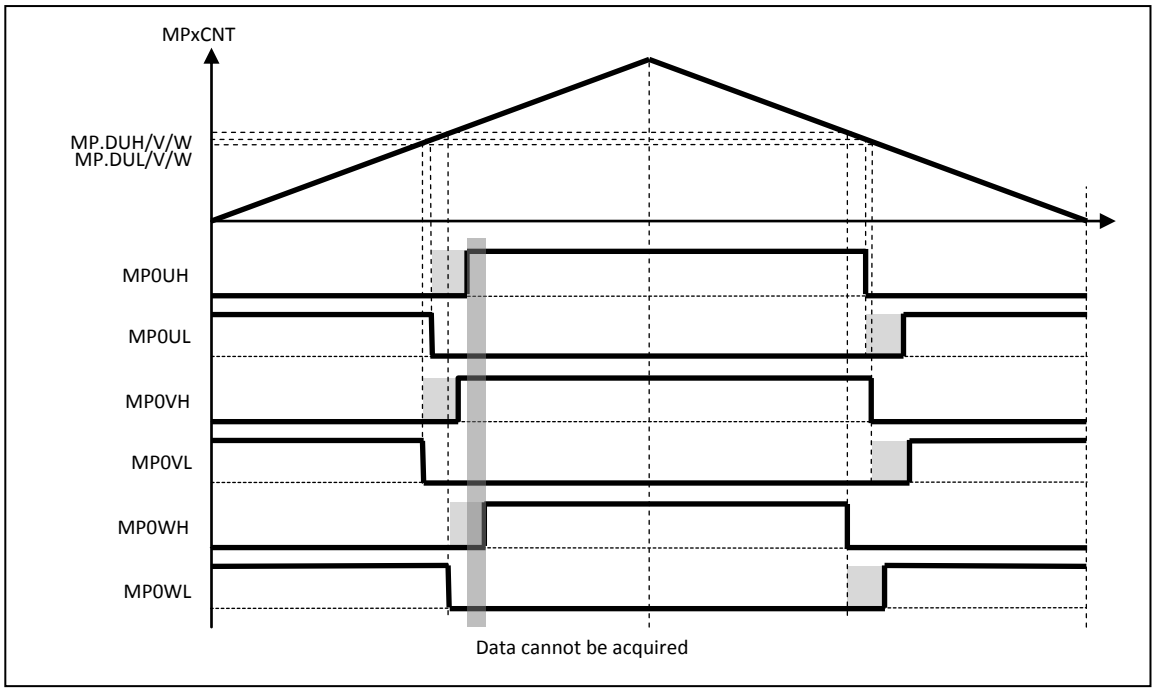


Figure 15-14 Symmetrical PWM Timing

In Asymmetrical mode, the wave from is not symmetric between the up and down counters. The Duty High is used to match on the up counter and the Duty Low is used to match on the down counter.

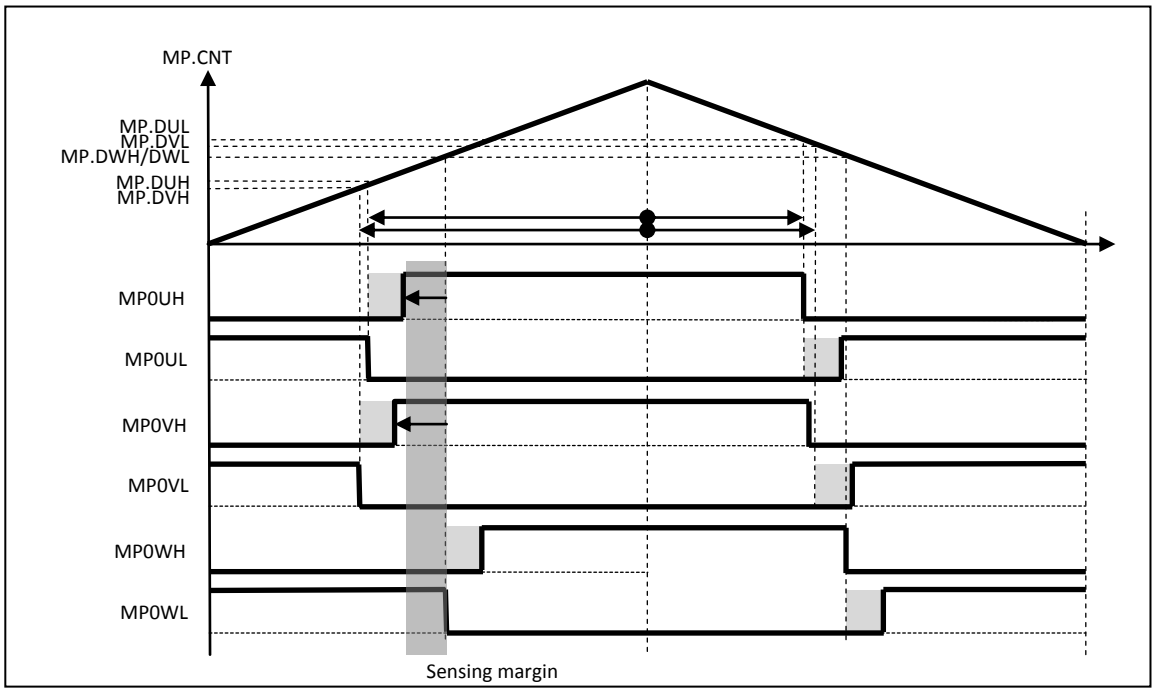


Figure 15-15 Asymmetrical PWM Timing and Sensing Margin

Description of ADC Triggering Function

A total of six ADC trigger timing registers are provided. This dedicated register triggers a signal to start ADC conversion. The conversion channel of ADC is defined in the ADC Control register.

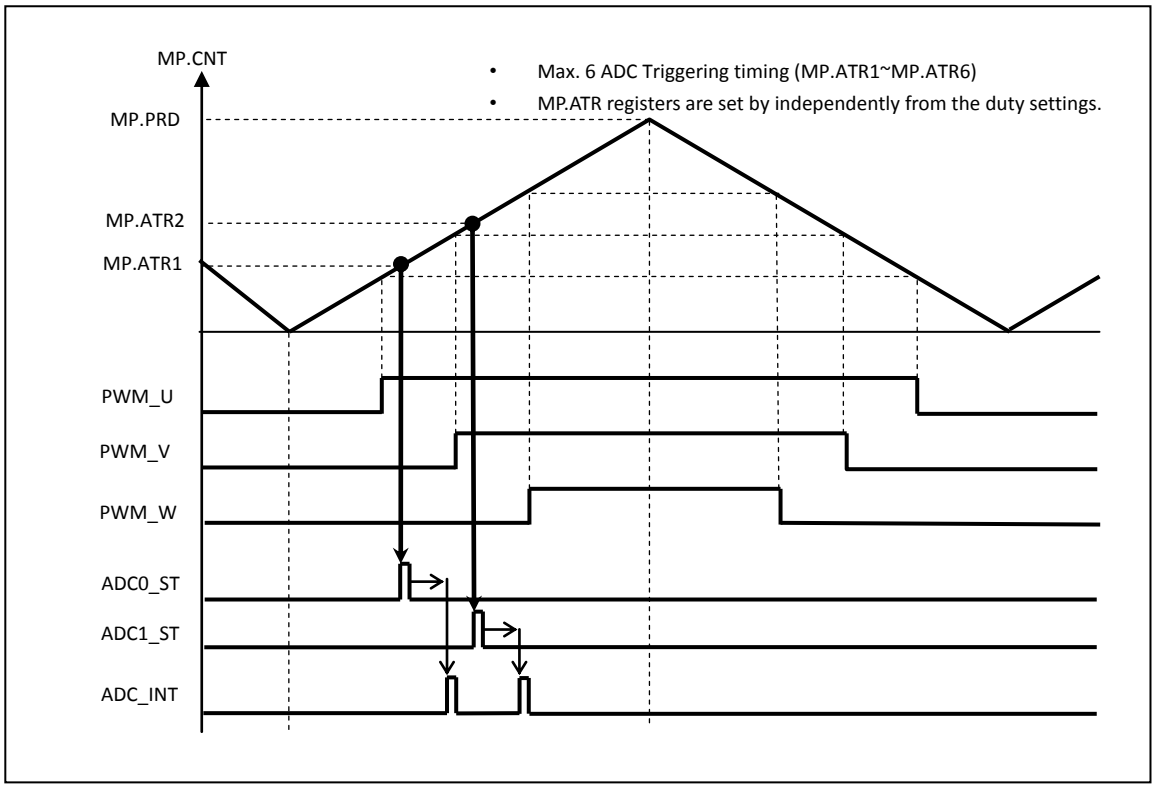


Figure 15-16 ADC Triggering Function Timing Diagram

Figure 15-17 shows an example of ADC Data acquisition.

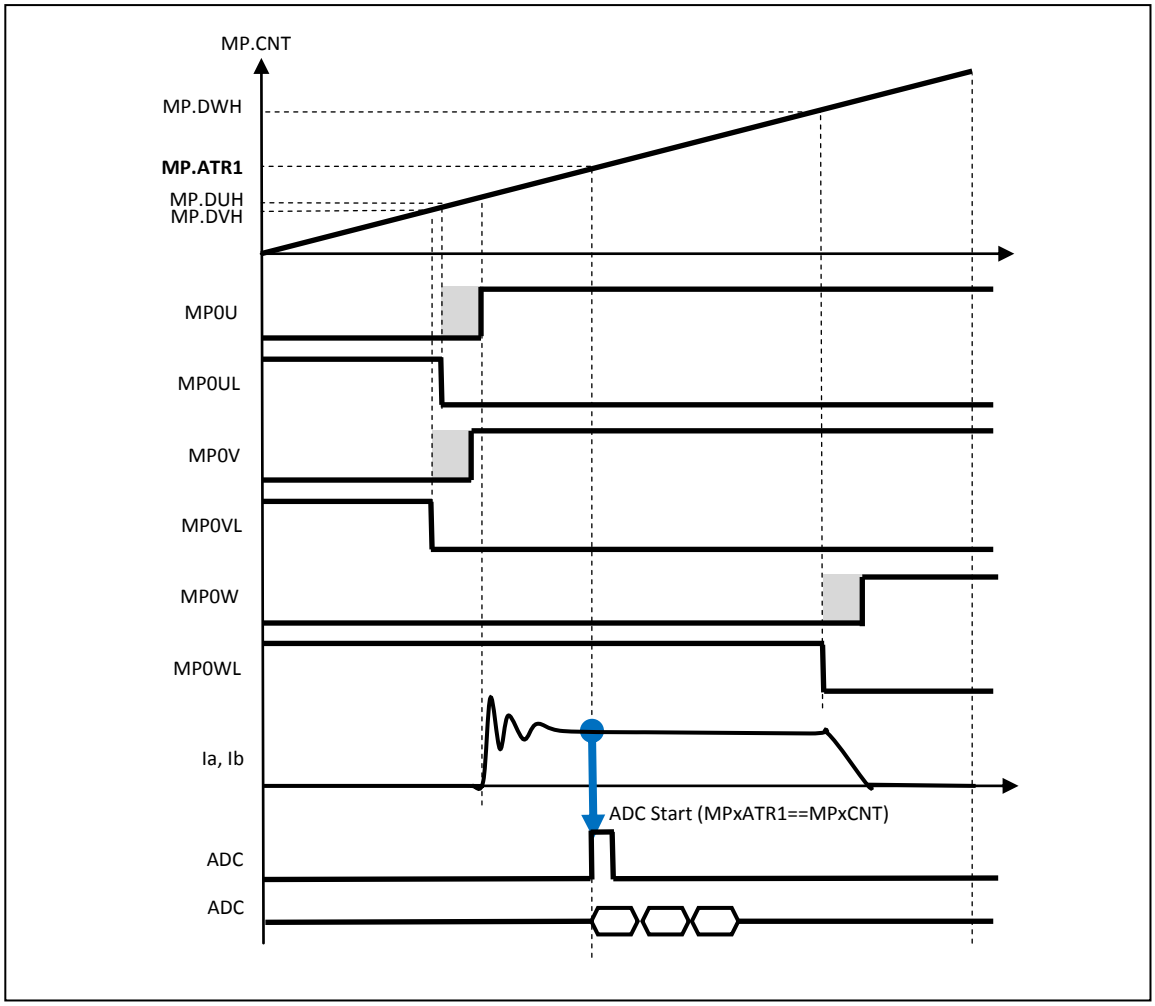


Figure 15-17 An Example of ADC Acquisition Timing by Event from MPWM

Interrupt Generation Timing

Each timing event can make an interrupt request to the CPU.

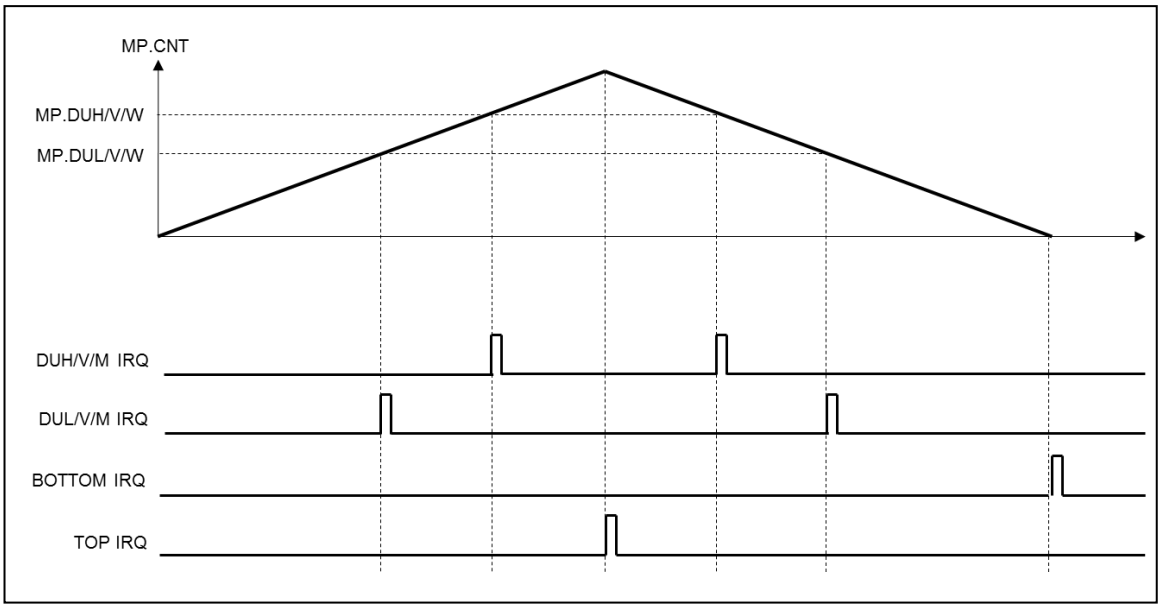


Figure 15-18 Interrupt Generation Timing

16. Divider (DIV64)

Overview

The divider module provides the hardware divider the ability to accelerate complicated calculations. This divider is a sequential 64-bit/32-bit divider and requires 32 clock cycles for one operation.

The equation for this operation is:

(AREGH,AREGL)/BREG = (QREGH,QREGL)

- Unsigned 64-bit dividend
- Unsigned 32-bit divisor
- Unsigned 64-bit quotient
- Unsigned 32-bit remainder
- Unsigned 32-cycle operating time

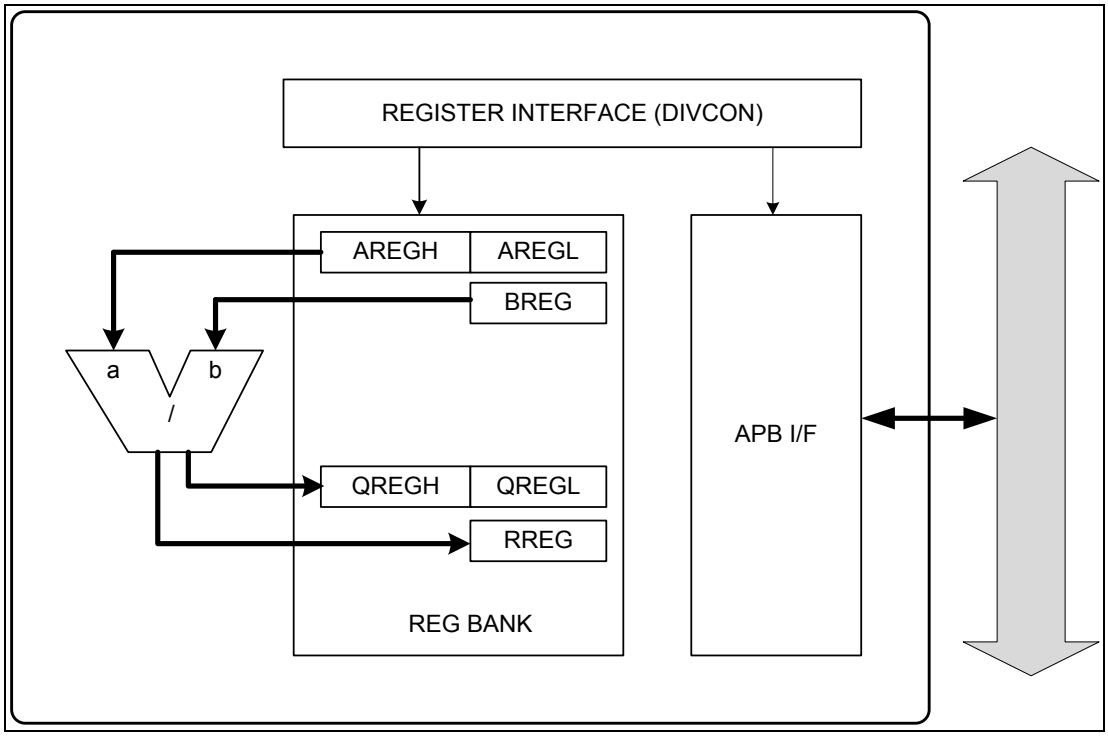


Figure 16-1 Block Diagram

Registers

The base address of the divider is 0x4000_0500 and the register map is described in Table 16-1.

Table 16-1 DIV64 Base Address

NAME	BASE ADDRESS
DIV64	0x4000_0500

Table 16-2 DIV64 Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
CR	0x0000	RW	DIV control register	0x00000000
AREGL	0x0004	RW	Most 32bit data register for dividend	0x00000000
AREGH	0x0008	RW	Least 32bit data register for dividend	0x00000000
BREG	0x000C	RW	32bit data register for divisor	0x00000000
QREGL	0x0010	R	Most 32bit data register for quotient	0x00000000
QREGH	0x0014	R	Least 32bit data register for quotient	0x00000000
RREG	0x0018	R	32bit data register for remainder	0x00000000

CR Divider Control Register

The DIVCON register controls the hardware divider module.

CR=0x4000_0500

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					I_ERROR	BUSY	DONE				MODE				START
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
					RO	RO	RO				RW				RW

10	I_ERROR	Divide by zero flag	
		0	Not divide by zero
		1	Divide by zero
9	BUSY	Divider is now under operating	
		0	Divider is not busy
		1	Divider is busy
8	DONE	Divider operation done flag	
		0	Divider is now operating
		1	Divider operation is done
4	MODE	Start operation mode	
		0	START bit write operation will trigger the divide operation
		1	BREG register write operation will trigger the divide operation
0	START	Divide operation start command. This bit is effective when MODE bit is 0	
		0	No effect
		1	Start divider

AREGL AREG (Dividend) Lower 32-bit Register

The lower 32-bit value of dividend should be written to this register.

AREGL=0x4000_0504

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AREGL[31:0]																															
0x0000_0000																															
RW																															

31 0	AREGL	Lower 32 bit value for dividend A.

AREGH AREG (Dividend) High 32-bit Register

The high 32-bit value of dividend should be written to this register.

AREGH=0x4000_0508

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AREGH[63:32]																															
0x0000_0000																															
RW																															

31 0	AREGH	High 32 bit value for dividend A.

BREG BREG (Divisor) Register

The 32-bit value of the divisor should be written to this register.

When the MODE bit is set to 1, the divide operation is started automatically as soon as the value is written to this register.

BREG=0x4000_050C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BREG[31:0]																															
0x0000_0000																															
RW																															

31 0	BREG	32 bit value for divisor B.

QREGL QREG (Quotient) Lower 32-bit Register

The divider stores the lower 32-bit value of the quotient in this register.

QREGL=0x4000_0510

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QREGL[31:0]																															
0x0000_0000																															
R																															

31 0	QREGL	Lower 32 bit value for quotient.

QREGH QREG (Quotient) High 32-bit Register

The divider stores the high 32-bit value of the quotient in this register.

QREGH=0x4000_0514

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QREGH[63:32]																															
0x0000_0000																															
R																															

31 0	QREGH	High 32 bit value for quotient.

RREG RREG (Remainter) Register

The divider stores the 32-bit value of the remainder in this register.

RREG=0x4000_0518

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RREG[31:0]																															
0x0000_0000																															
R																															

31 0	RREG	32 bit value for remainder.

17. 12-Bit A/D Converter

Introduction

The ADC block consists of 1 ADC unit, with the following features:

- 12 channels of analog inputs
- Single and Continuous conversion mode
- Up to 8 times sequential conversion support
- Software trigger support
- 3 internal trigger sources support (Soft-trig, MPWM, Timers)
- Adjustable sample and hold time

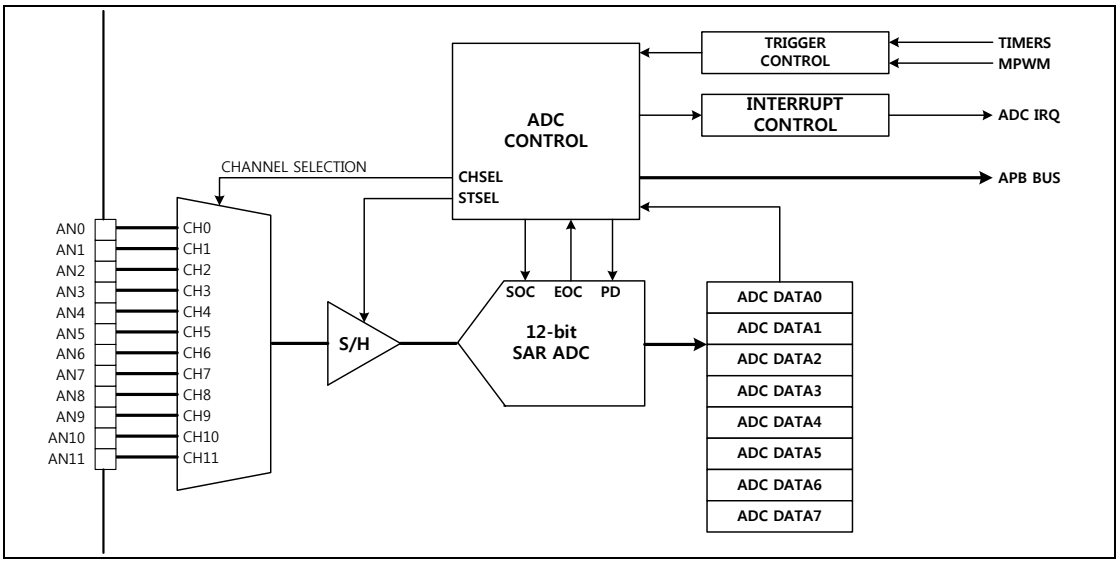


Figure 17-1 Block Diagram

Pin Description

Table 17-1 External Signal

PIN NAME	TYPE	DESCRIPTION
VDD	P	Analog Power(2.4V~5V)
VSS	P	Analog GND
AN0	A	ADC Input 0
AN1	A	ADC Input 1
AN2	A	ADC Input 2
AN3	A	ADC Input 3
AN4	A	ADC Input 4
AN5	A	ADC Input 5
AN6	A	ADC Input 6
AN7	A	ADC Input 7
AN8	A	ADC Input 8
AN9	A	ADC Input 9
AN10	A	ADC Input 10
AN11	A	ADC Input 11

Registers

The base address of the ADC unit is shown in Table 17-2.

Table 17-2 ADC Base Address

NAME	BASE ADDRESS
ADC	0x4000_B000

Table 17-3 ADC Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
AD.MR	0x0000	RW	ADC Mode register	0x00
AD.CSCR	0x0004	RW	ADC Current Sequence/Channel register	0x00
AD.CCR	0x0008	RW	ADC Clock Control register	0x80
AD.TRG	0x000C	RW	ADC Trigger Selection register	0x00
-	0x0010	-	Reserved	-
-	0x0014	-	Reserved	-
AD.SCSR	0x0018	RW	ADC Burst mode channel select	0x00
AD.CR	0x0020	RW	ADC Control register	0x00
AD.SR	0x0024	RW	ADC Status register	0x00
AD.IER	0x0028	RW	ADC Interrupt Enable register	0x00
-	0x002C	-	Reserved	-
AD.DR0	0x0030	R	ADCn Sequence 0 Data register	0x00
AD.DR1	0x0034	R	ADCn Sequence 1 Data register	0x00
AD.DR2	0x0038	R	ADCn Sequence 2 Data register	0x00
AD.DR3	0x003C	R	ADCn Sequence 3 Data register	0x00
AD.DR4	0x0040	R	ADCn Sequence 4 Data register	0x00
AD.DR5	0x0044	R	ADCn Sequence 5 Data register	0x00
AD.DR6	0x0048	R	ADCn Sequence 6 Data register	0x00
AD.DR7	0x004C	R	ADCn Sequence 7 Data register	0x00

AD.MR ADC Mode Register

The ADC Mode registers are 32-bit registers.

This register configures the ADC operation mode. This register should be writen first before the other registers.

AD.MR=0x4000_B000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																STSEL					SEQCNT			ADEN	ARST	ADMOD				TRGSEL	
																0x0					0x0			0x0	0x0	0x0				0x0	
																RW					RW			RW	RW	RW				RW	

16 12	STSEL	Sampling Time Selection ADC Sample & Hold circuit sampling time become (2 + STSEL[4:0]) MCLK cycles Minimum sampling time is 2 MCLK cycles			
10 8	SEQCNT	Number of coversion in a sequence If ADMOD is 2'h0 and SEQCNT is not 3'h0, CSEQN will be increased up to SEQCNT by trigger event. SEQCNT is burst count In burst mode. See 14.4.1~3 to know between Triggers and SEQCNT in burst conversion mode and in single sequential convertion mode			
		000	1st single sequential conversion or 1 burst count	100	5st single sequential conversion or 5 burst count
		001	2nd single sequential conversion or 2 burst counts	101	6st single sequential conversion or 6 burst counts
		010	3rd single sequential conversion or 3 burst counts	110	7st single sequential conversion or 7 burst counts
		011	4st single sequential conversion or 4 burst counts	111	8st single sequential conversion or 8 burst counts
7	ADEN	0	ADC disable		
		1	ADC enable		
6	ARST	0	Stop at the end of sequence. Should set ASTART as 1 to restart again		
		1	Restart at the end of sequence.		
5 4	ADMOD	00	Single conversion mode (single sequential conversion mode when SEQCNT is not 0x0)		
		01	Burst conversion mode		
		10	Reserved		
		11	Reserved		
1 0	TRGSEL	00	Event Trigger Disabled/Soft-Trigger Only		
		01	Timer Event Trigger		
		10	MPWM Event Trigger		
		11	Reserved		

If ADCMOD was set for Burst Mode, ADC channels are controlled by SEQ0CH ~ SEQ7CH. Sequential mode always start from SEQ0CH. (In 3 sequential mode, Analog inputs of channels which assigned at SEQ0CH, SEQ1CH and SEQ2CH are converted sequentially).

AD.CSCR ADC Current Sequence/Channel Register

ADC Current Sequence/Channel registers are 7-bit registers. This register consists of Current Sequence Numbers and Current Active Channel values. A Current Sequence Number (CSEQN) can be written to change the next sequence number. When you write CSEQN as 0x7 when CSEQN is 0x3 and AD.MR.SEQCNT is 0x7, the next sequence number is 0x7. AD converts the AD.SCSR.SEQ7CH channel and the 4,5,6 sequences are skipped. This register should be written first, before AD.SCSR.

AD.CSCR=0x4000_B004

7	6	5	4	3	2	1	0
-	CSEQN			CACH			
-	0x0			0x0			
-	RW			RO			

6 4	CSEQN	Current Sequence Number, can write when not abusy AD starts conversion the AD.SCSR.SEQ*CH's channel by AD.TRG.SEQTRG* in Single sequential mode. AD starts conversion the AD.SCSR.SEQ*CH's channel by AD.TRG.BSTTRG in Burst mode
	0000	Current Sequence is 0 the AD.SCSR.SEQ0CH's channel is converted by AD.TRG.SEQTRG0 in Single sequential mode or by AD.TRG.BSTTRG in Burst mode
	0001	Current Sequence is 1
	0010	Current Sequence is 2
	0011	Current Sequence is 3
	0100	Current Sequence is 4
	0101	Current Sequence is 5
	0110	Current Sequence is 6
	0111	Current Sequence is 7
3 0	CACH	Current Active Channel
	0000	ADC channel 0 is active
	0001	ADC channel 1 is active
	0010	ADC channel 2 is active
	0011	ADC channel 3 is active
	0100	ADC channel 4 is active
	0101	ADC channel 5 is active
	0110	ADC channel 6 is active
	0111	ADC channel 7 is active
	1000	ADC channel 8 is active
	1001	ADC channel 9 is active
	1010	ADC channel 10 is active
	1011	ADC channel 11 is active
	1100	reserved
	1101	reserved
	1110	reserved
	1111	reserved

AD.CCR ADC Clock Control Register

The ADC Control registers are 16-bit registers. The ADC Clock Control Register sets the ADC clock for determining the period to execute a conversion.

AD.CCR=0x4000_B008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPDA	CLKDIV							ADCPD	EXTCLK	CLKINVT	-				
0	0x00							1	0	0					
RW	RW							RW	RW	RW					

15	ADCPDA	ADC R-DAC disable to save power Don't set "1" here(it's optional bit)
14 8	CLKDIV[6:0]	ADC clock divider when EXTCLK is '0'. ADC clock = system clock/CLKDIV CKDIV=0 : ADC clock=system clock CKDIV=1 : ADC clock=stop
7	ADCPD	ADC Power Down 0 – ADC normal mode 1 – ADC Power Down mode
6	EXTCLK	Select if ADC uses external clock. 0 – internal clock(CKDIV enabled) 1 – external clock(SCU clock-MCCR4)
5	CLKINVT	Divided clock inversion(optional bit) 0 – duty ratio of divided clock is larger than 50% 1 – duty ratio of divided clock is less than 50%

AD.TRG ADC Trigger Selection Register

ADC Trigger registers are 32-bit registers.

For the ADC Trigger channel register, in Single/Burst mode, all the bit fields are used.

In Burst Conversion mode, only the BSTTRG bit field (bit3~bit0) is used.

AD.TRG=0x4000_B00C

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
SEQTRG7	SEQTRG6	SEQTRG5	SEQTRG4	SEQTRG3	SEQTRG2	SEQTRG1	SEQTRG0 BSTTRG
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
RW RW	RW RW	RW RW	RW RW	RW RW	RW RW	RW RW	RW RW

31 28	SEQTRG7	8th Sequence Trigger Source
27 24	SEQTRG6	7th Sequence Trigger Source
23 20	SEQTRG5	6th Sequence Trigger Source
19 16	SEQTRG4	5th Sequence Trigger Source
15 12	SEQTRG3	4th Sequence Trigger Source
11 8	SEQTRG2	3rd Sequence Trigger Source
7 4	SEQTRG1	2nd Sequence Trigger Source
3	SEQTRG0	1st Sequence Trigger Source
0	BSTTRG	Burst conversion Trigger Source

Value	Timer (TRGSEL '2'h1)	MPWM (TRGSEL '2'h2)
0	Timer 0	MP.ATR1
1	Timer 1	MP.ATR2
2	Timer 2	MP.ATR3
3	Timer 3	MP.ATR4
4		MP.ATR5
5		MP.ATR6
6	-	BOTTOM
7	-	PERIOD

AD.SCSR ADC Sequence Channel Selection Register

The ADC Burst Mode Channel Select register is a 32-bit register. For ADC single mode, it uses SEQ0CH to select the channel.

AD.SCSR=0x4000_B018

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
SEQ7CH	SEQ6CH	SEQ5CH	SEQ4CH	SEQ3CH	SEQ2CH	SEQ1CH	SEQ0CH
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
RW	RW	RW	RW	RW	RW	RW	RW

31 28	SEQ7CH	8th conversion sequence channel selection
27 24	SEQ6CH	7th conversion sequence channel selection
23 20	SEQ5CH	6th conversion sequence channel selection
19 16	SEQ4CH	5th conversion sequence channel selection
15 12	SEQ3CH	4th conversion sequence channel selection
11 8	SEQ2CH	3rd conversion sequence channel selection
7 4	SEQ1CH	2nd conversion sequence channel selection
3 0	SEQ0CH	1st conversion sequence channel selection This channel should be used for Single mode

AD.CR ADC Control Register

The ADC Control register is an 8-bit register.

AD.CR=0x4000_B020

7	6	5	4	3	2	1	0
ASTOP							ASTART
0							0
WO							RW

7	ASTOP	0	No operation
		1	ADC conversion stop (will be clear next @ADC clock) If ASTOP is set after a conversion starts, the conversion is completed and AD stops.
0	ASTART	0	No ADC conversion
		1	ADC conversion start when single mode (AD.MR.ADMOD and AD.MR.SEQCNT are 0x0. this bit will be cleared by coming @AD clock. If ASTART is set as 0 when ARST is 0 in Timer/MPWM trigger event mode, AD converts to AD.MR.SEQCNT once and AD stops. ASTART should be written to start the conversion sequence again

AD.SR ADC Status Register

The ADC Status register is an 8-bit register.

AD.SR=0x4000_B024

7	6	5	4	3	2	1	0
EOC	ABUSY	-	-	TRGIRQ	EOSIRQ	-	EOCIRQ
0	0	-	-	0	0	-	0
RO	RO	-	-	RC	RC	-	RC

7	EOC	ADC End-of-Conversion flag (Start-of-Conversion made by ADC_CLK clears this bit, not ASTART)
6	ABUSY	ADC conversion busy flag
-	-	Reserved.
-	-	Reserved.
3	TRGIRQ	ADC Trigger interrupt flag (Write “1” to clear flag) (0: no int / 1: int occurred)
2	EOSIRQ	This flag will be set at the end of a burst conversion or a sequence convrersion set (Write “1” to clear flag). *Sequence conversion set is the operation that AD converts to AD.MR.SEQCNT.
		0 None.
		1 End-of-Sequence Interrupt occurred in burst or single sequential mode
0	EOCIRQ	This flag will be set upon each conversion in a single is occurred (Write “1” to clear flag)
		0 None.
		1 End-of-Conversion Interrupt occurred

AD.IER Interrupt Enable Register

AD.IER=0x4000_B028

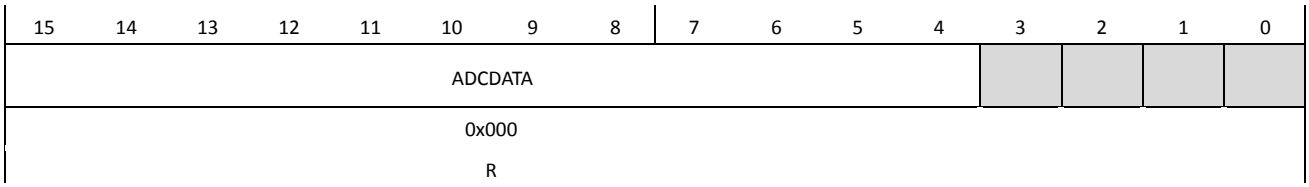
7	6	5	4	3	2	1	0
				TRGIRQE	EOSIRQE		EOCIRQE
0	0	0	0	0	0		0
				RW	RW		RW

3	TRGIRQE	ADC trigger conversion interrupt enable
2	EOSIRQE	ADC sequence conversion interrupt enable
1	-	Reserved.
0	EOCIRQE	ADC single conversion interrupt enable

AD.DRmADC Sequence Data Register 0~7

The ADC Data registers are 16-bit registers. The ADC Data registers contain the latest conversion results for each of the 8 sequence conversions.

AD.DR0=0x4000_B030, AD.DR1=0x4000_B034, AD.DR2=0x4000_B038, AD.DR3=0x4000_B03C
AD.DR4=0x4000_B040, AD.DR5=0x4000_B044, AD.DR6=0x4000_B048, AD.DR7=0x4000_B04C



15 4	ADCDATA	ADC channel 0~7 data (12-bit)

Functional Description

AD Conversion Timing Diagram

When AD.MR.ADMOD is 0x0 and AD.MR.SEQCNT is 0x0, ADC conversion is started by writing AD.CR.ASTART as ‘1’. After AD.CR.ASTART is set, Start of Conversion (SOC) is activated in 3 ADC clocks and AD.SR.EOCIRQ is set in 2 ADC clocks and 2 PCLKs after End of Conversion.

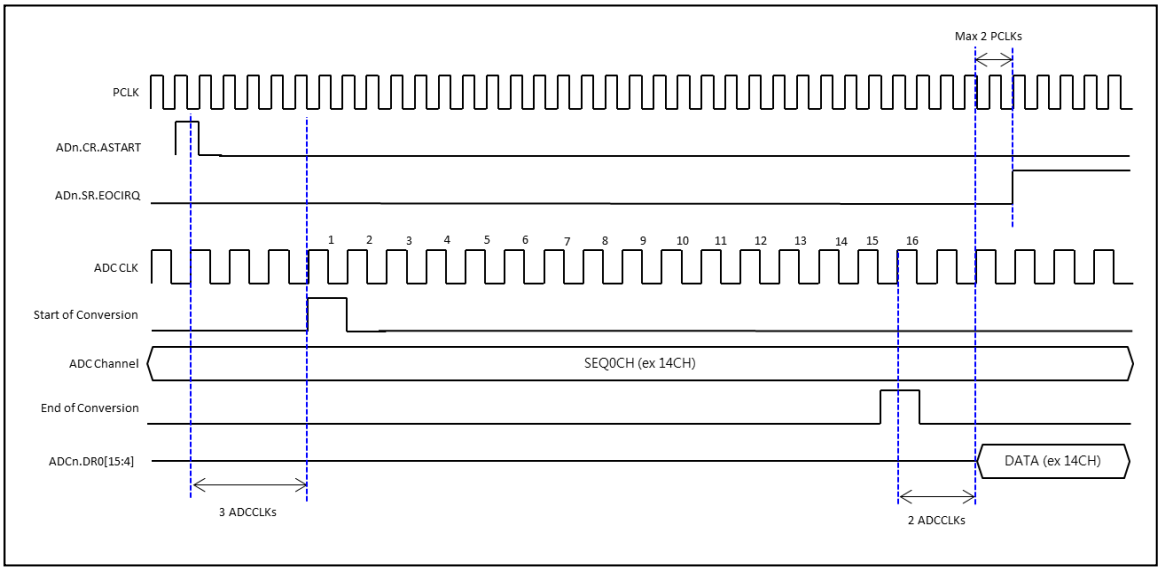


Figure 17-2 ADC Single Mode Timing (When ADCn.MR.AMOD = ‘0’)

ADC Burst Conversion Mode Timing Diagram

The Burst Conversion mode (Burst mode) occurs when AD.MR.ADMOD is 0x1. When there are two sources to make SOC in Burst mode, one is the TRG event (TIMER and MPWM) and the other is AD.CR.ASTART. When AD.MR.TRGSEL is set as timer event trigger or MPWM event trigger, SOC is made by the trigger of AD.TRG.BSTTRG (AD.TRG[3:0]). For example, ADC conversion is started by the trigger of TIMER3 if AD.TRG.BSTTRG is set as TIMER3. Once the BSTTRG triggers events, ADC converts ADC channels per the values set in AD.MR.SEQCNT. See Figure 17-3.

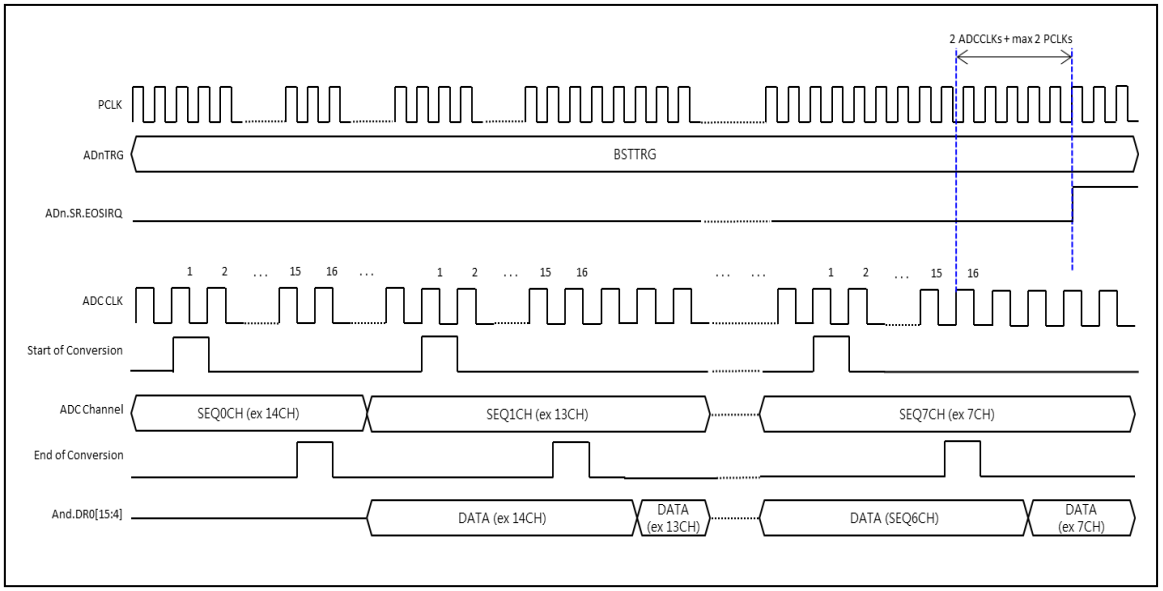


Figure 17-3 ADC Burst Mode Timing (When AD.MR.AMOD = '1')

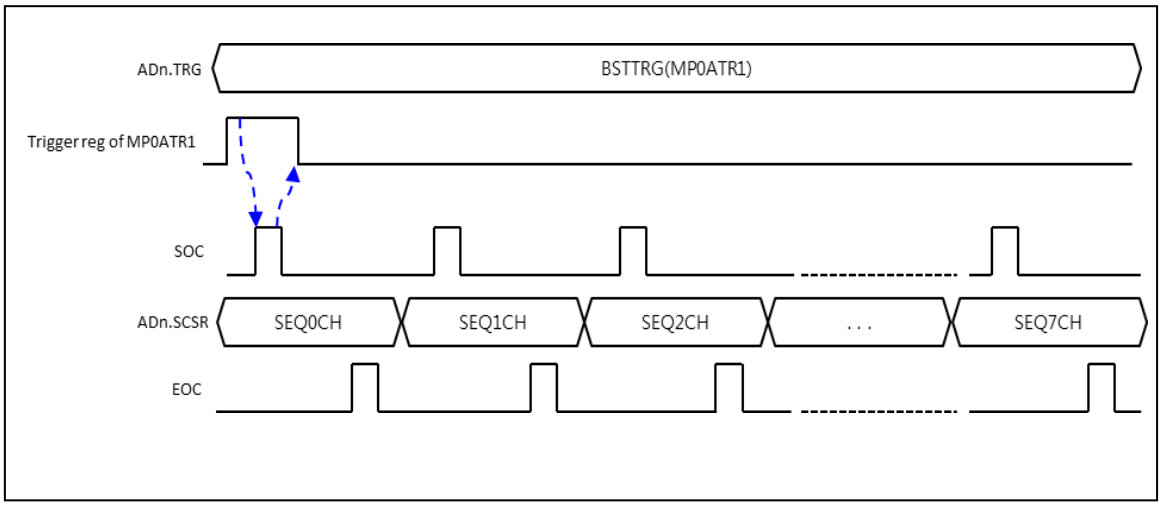


Figure 17-4 ADC Trigger Timing in Burst Mode (SEQCNT = 3'b111, 8 Sequence Coversion)

ADC Sequential Conversion Mode Timing Diagram

Single Sequential Conversion mode (Single Sequential mode) occurs when AD.MR.ADMOD is 0x0 and AD.MR.SEQCNT is not 0x0. To set Sequential Conversion mode, AD.MR.AMOD is 2'b00 and AD.MR.SEQCNT is not 2'b00.

The operation of Sequential mode is almost the same as the Burst mode. The difference is the source of SOC. Each SOC is made by the trigger of SEQTRGx as each SEQCNT. See Figure 17-5.

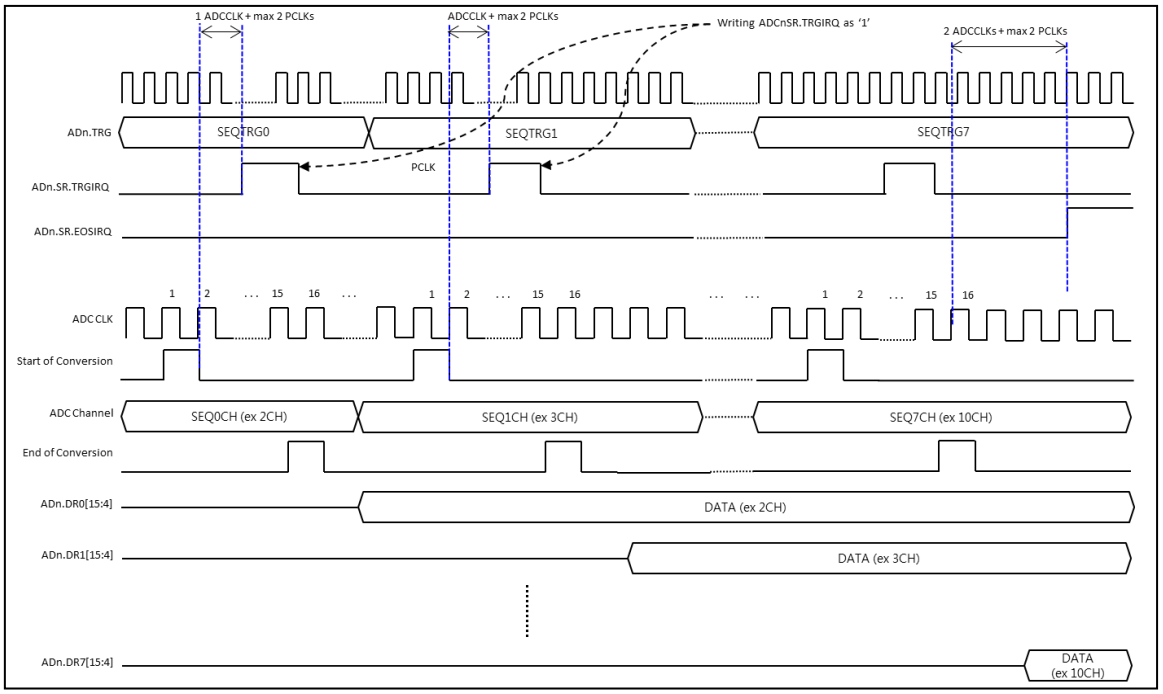


Figure 17-5 ADC Sequential Mode Timing (When AD.MR.AMOD = '0 and AD.MR.SEQCNT ≠ '0')

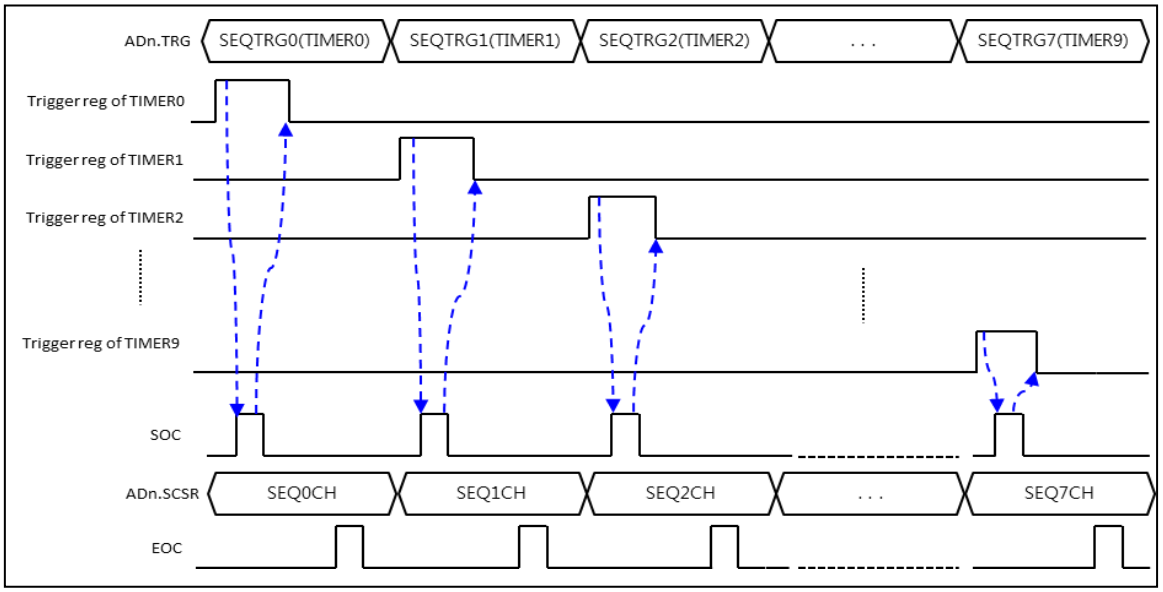


Figure 17-6 ADC Trigger Timing in Sequential Mode (SEQCNT = 3'b111, 8 Sequence Coversion)

18. Electrical Characteristics

DC Characteristics

Absolute Maximum Ratings

Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions.

Table 18-1 Absolute Maximum Rating

Parameter	Symbol	Min	Max	Unit
Power Supply (VDD)	VDD	-0.5	+6	V
Analog Power Supply (AVDD)	AVDD	-0.5	+6	V
VDC Output Voltage	VDD18			V
Input High Voltage		-	VDD+0.5	V
Input Low Voltage		VSS – 0.5	-	V
Output Low Current per pin	I_{OL}		5	mA
Output Low Current Total	$\sum I_{OL}$		40	mA
Output High Current per pin	I_{OH}		5	mA
Output Low Current Total	$\sum I_{OH}$		40	mA
Power consumption				mW
Input Main Clock Range		4	16	MHz
Operating Frequency		-	40	MHz
Storage Temperature	Tst	-55	+125	℃
Operating Temperature	Top	-40	+105	℃

DC Characteristics

Table 18-2 Recommended Operating Condition

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Supply Voltage	VDD		2.2	-	5.5	V
Supply Voltage	AVDD		2.2	-	5.5	V
Operating Frequency	FREQ	MOSC	4	-	16	MHz
		SOSC	-	32.768	-	kHz
		HSI	38.8	40	41.2	MHz
		LSI	32	40	48	kHz
Operating Temperature	Top	Top	-40	-	+105	℃

Table 18-3 DC Electrical Characteristics (VDD = +5V, Ta = 25°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Input Low Voltage	V_{IL}	Schmitt input	-	-	0.2VDD	V
Input High Voltage	V_{IH}	Schmitt input	0.8VDD	-	-	V
Output Low Voltage	V_{OL}	I_{OL} = 3mA	-	-	VSS+1.0	V
Output High Voltage	V_{OH}	I_{OH} = -3mA	VDD–1.0	-	-	V
Input High Leakage	I_{IH}				4	uA
Input Low Leakage	I_{IL}		-4			
Pull-up Resister	R_{PU}	VDD=5V	30	-	90	kΩ

Current Consumption

Table 18-4 describes the current consumption in Normal, Sleep, and Power Down modes under various conditions.

Table 18-4 Current Consumption in Each Mode (Temperature: +25°C Only)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Normal Operation	IDD_{NORMAL}	LSIOSC=RUN HSIOSC=RUN MXOSC=RUN SXOSC=RUN HCLK= LSIOSC	–	2.6	–	mA
		LSIOSC=RUN HSIOSC=OFF MXOSC=OFF SXOSC=OFF HCLK=LSIOSC	–	0.7	–	mA
		LSIOSC=RUN HSIOSC=RUN MXOSC=RUN SXOSC=RUN HCLK=HSIOSC	–	10.3	–	mA
		LSIOSC=OFF HSIOSC=RUN MXOSC=OFF SXOSC=OFF HCLK=HSIOSC	–	9.4	–	mA
		LSIOSC=RUN HSIOSC=RUN MXOSC=RUN SXOSC=RUN HCLK=MXOSC	–	4.2	–	mA
		LSIOSC=OFF HSIOSC=OFF MXOSC=RUN SXOSC=OFF HCLK=MXOSC	–	3.2	–	mA
		LSIOSC=RUN HSIOSC=RUN MXOSC=RUN SXOSC=RUN HCLK=SXOSC	–	2.6	–	mA
		LSIOSC=OFF HSIOSC=OFF MXOSC=OFF SXOSC=RUN HCLK=SXOSC	–	0.7	–	mA

Sleep Mode	IDD_{SLEEP}	LSIOSC=RUN HSIOSC=RUN SXOSC=RUN MXOSC=RUN HCLK=LSIOSC	–	2.5	–	mA
		LSIOSC=RUN HSIOSC=OFF SXOSC=OFF MXOSC=OFF HCLK=LSIOSC	–	0.6	–	mA
		LSIOSC=RUN HSIOSC=RUN SXOSC=RUN MXOSC=RUN HCLK=HSIOSC		7.6		mA
		LSIOSC=OFF HSIOSC=RUN SXOSC=OFF MXOSC=OFF HCLK=HSIOSC	–	6.8	–	mA
		LSIOSC=RUN HSIOSC=RUN SXOSC=RUN MXOSC=RUN HCLK=MXOSC	–	3.5	–	mA
		LSIOSC=OFF HSIOSC=OFF SXOSC=OFF MXOSC=RUN HCLK=MXOSC	–	2.5	–	mA
		LSIOSC=RUN HSIOSC=RUN SXOSC=RUN MXOSC=RUN HCLK=SXOSC	–	2.5	–	mA
		LSIOSC=OFF HSIOSC=OFF SXOSC=RUN MXOSC=OFF HCLK=SXOSC	–	0.6	–	mA
PowerDown Mode	IDD_{STOP}	LSIOSC=STOP HSIOSC=STOP SXOSC=STOP MXOSC=STOP HCLK=STOP	–	5	10	uA

Note:

UART en, 1 port toggle @5V

LSIOSC (40KHz), HSIOSC (40MHz), MXOSC (8MHz), SXOSC (32.768KHz)

POR Electrical Characteristics

Table 18-5 POR Electrical Characteristics (Temperature: -40 ~ +105°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	VDD18		1.6	1.8	2.0	V
Operating Current	IDD_{PoR}	Typ. <6uA If always on	-	60	-	nA
POR Set Level	VR_{PoR}	VDD rising (slow)	1.3	1.4	1.55	V
POR Reset Level	VF_{PoR}	VDD falling (slow)	1.1	1.2	1.4	V

LVD Electrical Characteristics

Table 18-6 LVD Electrical Characteristics (Temperature: -40 ~ +105°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	VDD		1.7		5	V
Operating Current	IDDLVD	Typ. <6uA when always on	-	1	-	mA
LVD Set Level 0	VLVD0	VDD falling (slow)	1.58	1.73	2.2	V
LVD Set Level 1	VLVD1	VDD falling (slow)	2.4	2.65	3.1	V
LVD Set Level 2	VLVD2	VDD falling (slow)	3.55	3.7	4.15	V
LVD Set Level 3(1)	VLVD3	VDD falling (slow)	4.2	4.35	4.8	V

Caution: (1) This LVD Voltage level is not recommended, because it sometimes can change LVD detection level at high temperature.

VDC Electrical Characteristics

Table 18-7 VDC Electrical Characteristics (Temperature: -40 ~ +105°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	VDD_{VDC}		2.2	-	5.5	V
Current Consumption	IDD_{NORM}	@RUN	-	100	150	uA
	IDD_{STOP}	@STOP	-	1	2	uA

External OSC Characteristics

Table 18-8 External OSC Characteristics (Temperature: -40 ~ +105°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Voltage	VDD		2.2	-	5.5	V
IDD		@4MHz/5V	-	240		uA
Frequency	$OSCF_{req}$		4	-	16	MHz
Output Voltage	OSC_{VOUT}		1.2	2.4	-	V
Load Capacitance	$LOAD_{CAP}$		5	22	35	pF

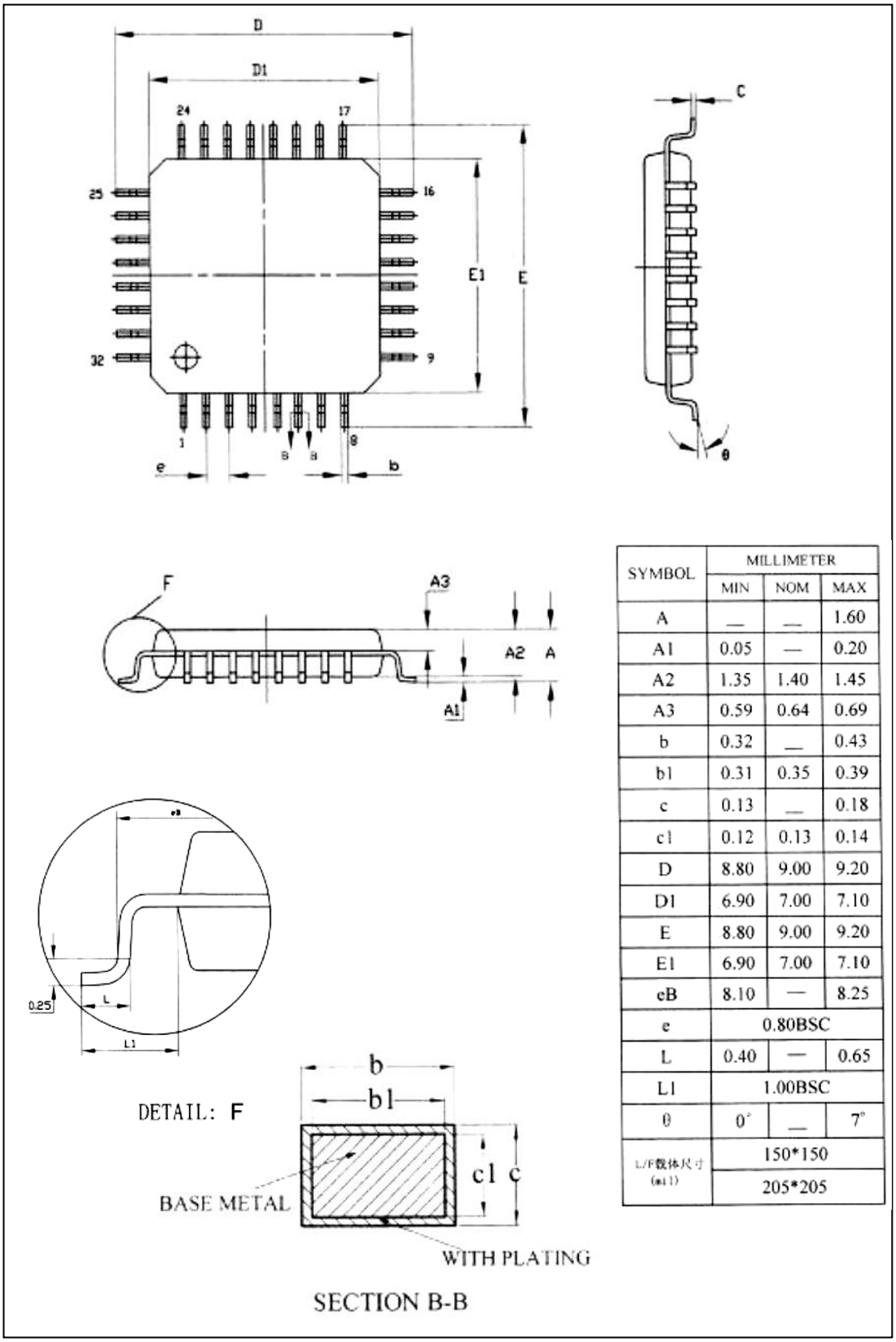
ADC Electrical Characteristics

Table 18-9. ADC Electrical Characteristics (Temperature: -40 ~ +105°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	AVDD		2.4	5	5.5	V
Resolution				12		Bit
Operating Current	IDDA				2.8	mA
Analog Input Range			0		AVDD	V
Conversion Rate				-	1.0	MSPS
Operating Frequency	ACLK				16	MHz
DC Accuracy	INL			±3.5		LSB
	DNL			±2.5		LSB
Offset Error				±1.5		LSB
Full Scale Error				±1.5		LSB
SNDR	SNDR			68		dB
THD				-70		dB

19. Package

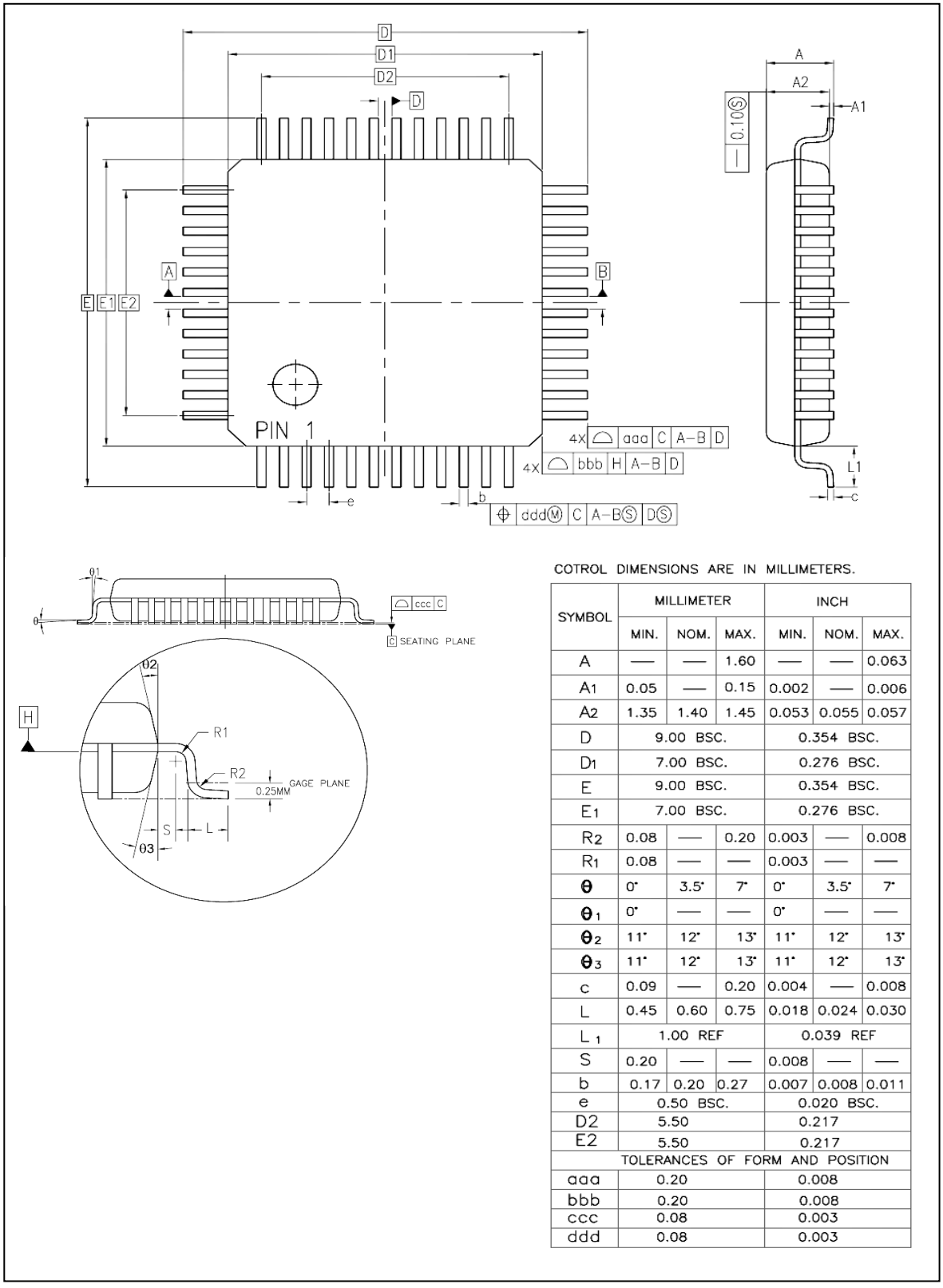
LQFP-32 Package Dimension



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.32	—	0.43
b1	0.31	0.35	0.39
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	—	8.25
e	0.80BSC		
L	0.40	—	0.65
L1	1.00BSC		
θ	0°	—	7°
L/F载体尺寸 (mil)	150*150		
	205*205		

Figure 19-1 Package Dimension (LQFP-32)

LQFP-48 Package Dimension



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D	9.00 BSC.			0.354 BSC.		
D1	7.00 BSC.			0.276 BSC.		
E	9.00 BSC.			0.354 BSC.		
E1	7.00 BSC.			0.276 BSC.		
R2	0.08	—	0.20	0.003	—	0.008
R1	0.08	—	—	0.003	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ1	0°	—	—	0°	—	—
θ2	11°	12°	13°	11°	12°	13°
θ3	11°	12°	13°	11°	12°	13°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		
S	0.20	—	—	0.008	—	—
b	0.17	0.20	0.27	0.007	0.008	0.011
e	0.50 BSC.			0.020 BSC.		
D2	5.50			0.217		
E2	5.50			0.217		
TOLERANCES OF FORM AND POSITION						
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.08			0.003		

Figure 19-2 Package Dimension (LQFP-48)

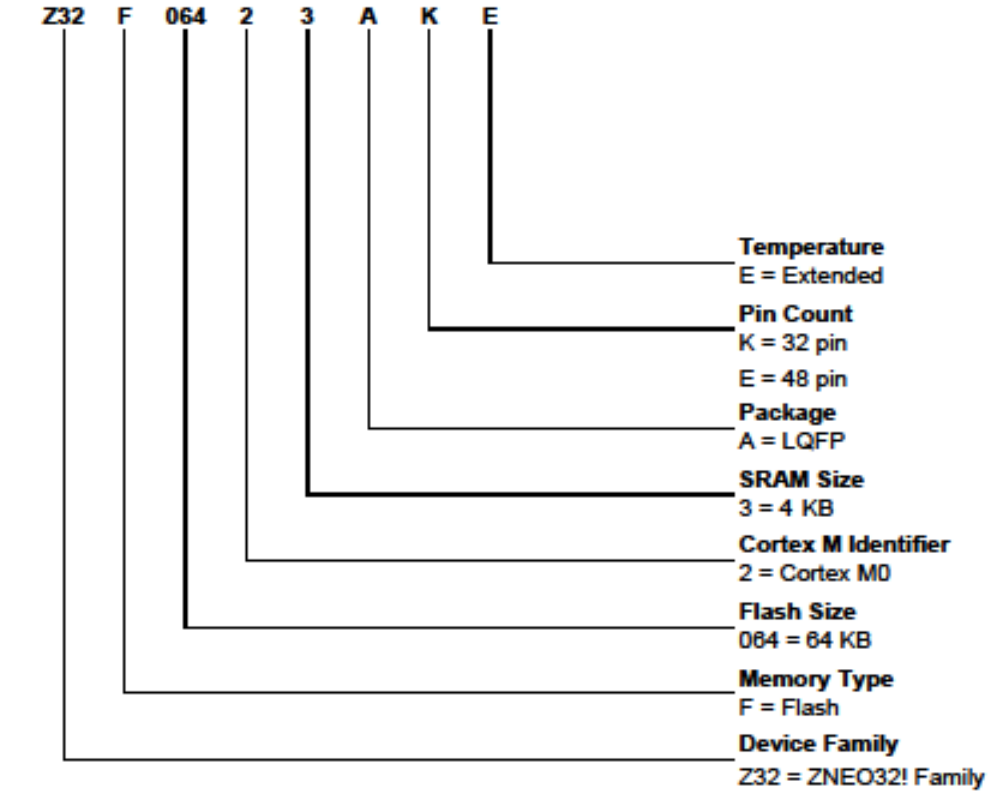
20. Ordering Information

Table 20-1 identifies the basic features and package styles available for the Z32F0642 MCU.

Table 20-1 Ordering Information

Part Number	Flash	SRAM	UART	SPI	I2C	MPWM	ADC	I/O Ports	Package
Z32F06423AKE	64KB	4KB	2	1	1	1	1-unit 10 ch	30	LQFP-32
Z32F06423AEE	64KB	4KB	2	1	1	1	1-unit 12 ch	44	LQFP-48

Zilog part numbers consist of a number of components, which are described below using part number Z32F06423AKE as an example.



Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.