Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	Automotive
Core Processor	S12
Program Memory Type	FLASH (32KB)
Controller Series	HCS12
RAM Size	2K x 8
Interface	LIN, SCI
Number of I/O	9
Voltage - Supply	2.25V ~ 5.5V
Operating Temperature	-40°C ~ 105°C
Mounting Type	Surface Mount
Package / Case	48-LQFP Exposed Pad
Supplier Device Package	48-HLQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mm912f634bv1ae

1 Ordering Information

Table 1. Ordering Information

Device [2]	Temperature Range (T_A)	Package	Max. Bus Frequency (MHz) (f_{BUSMAX})	Flash (kB)	RAM (kB)	Analog Option [1]
MM912F634DV1AE	-40 to 105 °C	98ASA00173D, 48-PIN LQFP-EP	20	32	2	1
MM912F634DV2AE						2
MM912F634DV2AP		98ASH00962A, 48-PIN LQFP	16	32		

Note:
1. See Table 2.
2. For Tape and Reel orders add R2 to the part suffix

Table 2. Analog Options[3]

Feature	Option 1	Option 2
Current Sense Module	YES	NO
Wake-up Inputs (Lx)	L0…L5	L0…L3

Note:
3. This table only highlights the analog die differences between the derivatives. See Section 4.2.3, “Analog Die Options" for detailed information.

The device part number follows the standard scheme below:

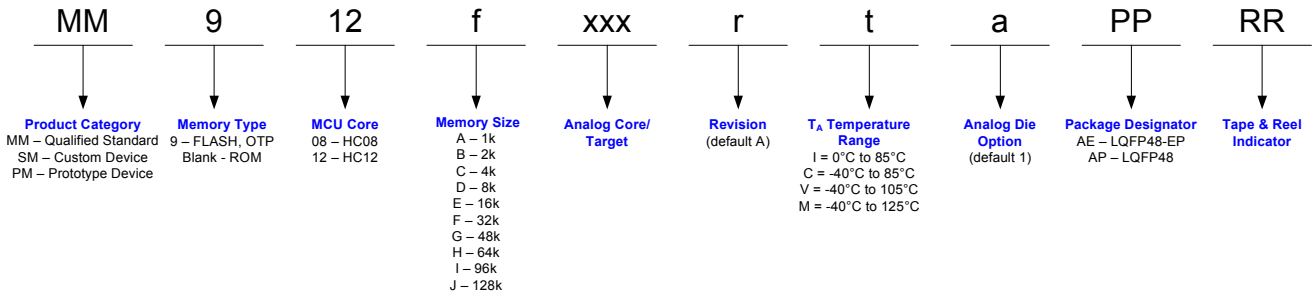


Figure 2. Part Number Scheme

3.5.2 Static Electrical Characteristics MCU Die

3.5.2.1 I/O Characteristics

This section describes the characteristics of all I/O pins except EXTAL, XTAL, TEST and supply pins.

Table 25. 5.0 V I/O Characteristics for PTA, $\overline{\text{RESET}}$ and BKGD Pins

Ratings	Symbol	Min	Typ	Max	Unit
Input high voltage	V_{IH}	0.65*V_{DD}	-	-	V
Input high voltage	V_{IH}	-	-	V_{DD} + 0.3	V
Input low voltage	V_{IL}	-	-	0.35*V_{DD}	V
Input low voltage	V_{IL}	V_{SS} - 0.3	-	-	V
Input hysteresis	V_{HYS}	-	250	-	mV
Input leakage current (pins in high-impedance input mode) V_{in} = V_{DDX} or V_{SSX}	I_{IN}	-1.0	-	1.0	µA
Output high voltage (pins in output mode) Partial Drive I_{OH} = -2.0 mA	V_{OH}	V_{DD} – 0.8	-	-	V
Output high voltage (pins in output mode) Full Drive I_{OH} = -10 mA	V_{OH}	V_{DD} – 0.8	-	-	V
Output low voltage (pins in output mode) Partial drive I_{OL} = +2.0 mA	V_{OL}	-	-	0.8	V
Output low voltage (pins in output mode) Full Drive I_{OL} = +10 mA	V_{OL}	-	-	0.8	V
Internal pull-up resistance (V_{IH}min > input voltage > V_{IL}max)	R_{PUL}	25	-	50	kΩ
Internal pull-down resistance (V_{IH}min > input voltage > V_{IL}max)	R_{PDH}	25	-	50	kΩ
Input capacitance	C_{in}	-	6.0	-	pF
Injection current[29] Single pin limit Total device Limit, sum of all injected currents	 I_{ICS} I_{ICP}	 -2.5 -25	 - -	 2.5 25	mA

Note:
29. Refer to Section 3.8, “ESD Protection and Latch-up Immunity" for more details.

3.6 Dynamic Electrical Characteristics

Dynamic electrical characteristics noted under conditions 5.5V ≤ VSUP ≤ 18 V, -40 °C ≤ T_A ≤ 105 °C, unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25 °C under nominal conditions unless otherwise noted.

3.6.1 Dynamic Electrical Characteristics Analog Die

Table 26. Dynamic Electrical Characteristics - Modes of Operation

Ratings	Symbol	Min	Typ	Max	Unit
VDD Short Timeout	t_{VTO}	110	150	205	ms
Analog Base Clock	f_{BASE}	-	100	-	kHz
Reset Delay	t_{RST}	140	200	280	µs

Table 27. Dynamic Electrical Characteristics - Power Supply

Ratings	Symbol	Min	Typ	Max	Unit
Glitch Filter Low Battery Warning (LBI)[30]	t_{LB}	-	2.0	-	µs
Glitch Filter Low Voltage Warning (LVI)[30]	t_{LV}	-	2.0	-	µs

Table 58. 0x00E0–0x00E7 Reserved Register Space

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00E0-0x00E7	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 59. 0x00E8–0x00EF Serial Peripheral Interface (SPI)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00E8	SPI0CR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
		W								
0x00E9	SPI0CR2	R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
		W								
0x00EA	SPI0BR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
		W								
0x00EB	SPI0SR	R	SPIF	0	SPTEF	MODF	0	0	0	0
		W								
0x00EC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00ED	SPI0DR	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		W								
0x00EE	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00EF	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 60. 0x00F0–0x00FF Reserved Register Space

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00F0-0x00FF	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 61. 0x0100–0x0113 Flash Control & Status Register FTSR

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0100	FCLKDIV	R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
		W								
0x0101	FSEC	R	KEYEN1	KEYEN0	0	0	0	0	SEC1	SEC0
		W								
0x0102	FRSV0	R	0	0	0	0	0	0	0	0
		W								
0x0103	FCNFG	R	CBEIE	CCIE	KEYACC	0	0	0	0	0
		W								

Table 65. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3 (continued)

Offset	Name		7	6	5	4	3	2	1	0
0x15	RSR	R	0	0	WDR	EXR	WUR	LVRX	LVR	POR
	Reset Status Register	W								
0x16	MCR	R	0	0	0	0	0	0	MODE	
	Mode Control Register	W								
0x18	LINR	R	LINOTIE	LINOTC	RX	TX	LVSD	LINEN	LINSR	
	LIN Register	W								
0x20	PTBC1	R	0	PUEB2	PUEB1	PUEB0	0	DDRB2	DDRB1	DDRB0
	Port B Configuration Register 1	W								
0x21	PTBC2	R	0	0	0	0	PWMCS	PWMEN	SERMOD	
	Port B Config Register 2	W								
0x22	PTB	R	0	0	0	0	0	PTB2	PTB1	PTB0
	Port B Data Register	W								
0x28	HSCR	R	HSOTIE	HSHVSDE	PWMCS2	PWMCS1	PWMHS2	PWMHS1	HS2	HS1
	High-side Control Register	W								
0x29	HSSR	R	HSOTC	0	0	0	HS2CL	HS1CL	HS2OL	HS1OL
	High-side Status Register	W								
0x30	LSCR	R	LSOTIE	0	PWMCS2	PWMCS1	PWMLS2	PWMLS1	LS2	LS1
	Low-side Control Register	W								
0x31	LSSR	R	LSOTC	0	0	0	LS2CL	LS1CL	LS2OL	LS1OL
	Low-side Status Register	W								
0x32	LSCEN	R	0	0	0	0	LSCEN			
	Low-Side Control Enable Register	W								
0x38	HSR	R	HOTIE	HOTC	0	0	0	0	0	HSUPON
	Hall Supply Register	W								
0x3C	CSR	R	CSE	0	0	0	CCD	CSGS		
	Current Sense Register	W								
0x40	SCIBD (hi)	R	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
	SCI Baud Rate Register	W								
0x41	SCIBD (lo)	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	SCI Baud Rate Register	W								
0x42	SCIC1	R	LOOPS	0	RSRC	M	0	ILT	PE	PT
	SCI Control Register 1	W								
0x43	SCIC2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	SCI Control Register 2	W								
0x44	SCIS1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	SCI Status Register 1	W								
0x45	SCIS2	R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
	SCI Status Register 2	W								

4.4.1 Voltage Regulators VDD (2.5 V) & VDDX (5.0 V)

To supply the MCU die and minor additional loads two cascaded voltage regulators have been implemented, VDDX (5.0 V) and VDD (2.5 V). External capacitors (C_{VDD}) and (C_{VDDX}) are required for proper regulation.

4.4.2 Power Up Behavior / Power Down Behavior

To guarantee safe power up and down behavior, special dependencies are implemented to prevent unwanted MCU execution.

Figure 18 shows a standard power up and power down sequence.

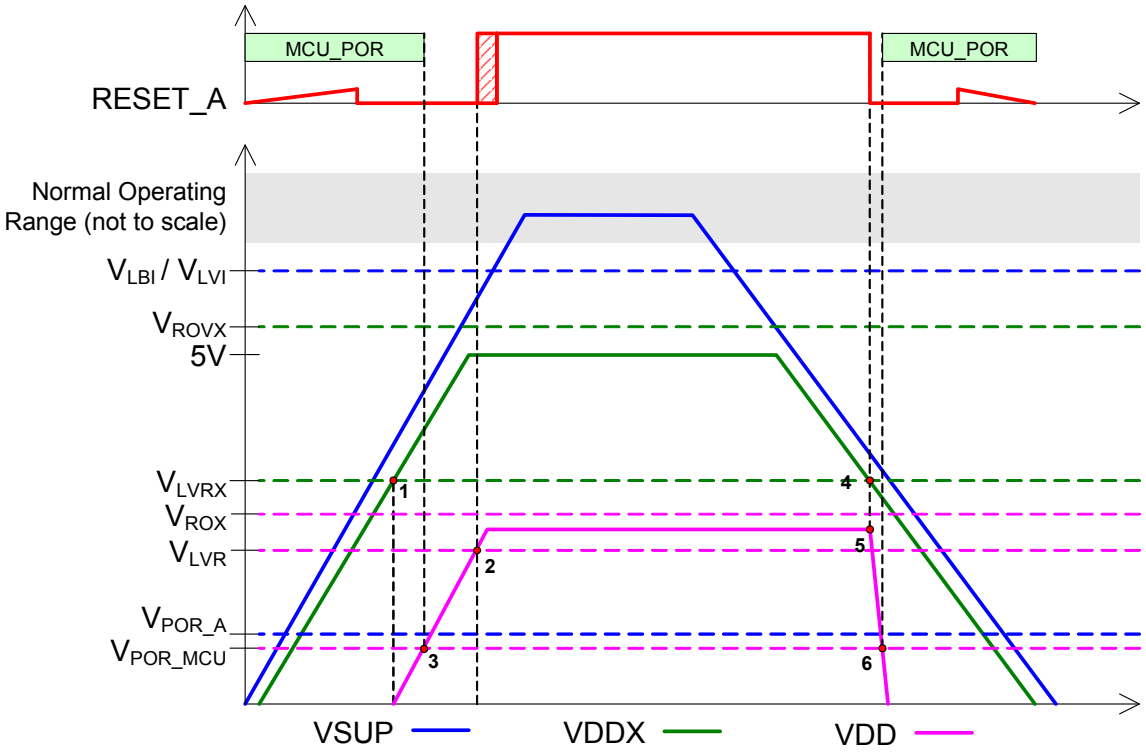


Figure 18. Power-up / Down Sequence

To avoid any abnormal device behavior, it is essential to have the MCU Power on Reset (POR) block complete its start-up sequence before the analog die reset signal ($\overline{\text{RESET A}}$) is asserted. Since the $\overline{\text{RESET A}}$ circuitry is supplied by V_{DDX}, the voltage on the 2.5 V supply (V_{DD}) needs to remain below the POR threshold whenever V_{DDX} is too low to guarantee $\overline{\text{RESET A}}$ can be properly asserted (3;6). This is achieved with the following implementation.

Power-up:

- The VDD regulator is enabled after VDDX has reached the V_{LVRX} threshold (1).
- Once VDD reaches V_{LVR}, the $\overline{\text{RESET\_A}}$ is released (2).

Power-down:

- Once VDDX has reached the V_{LVRX} threshold (4), the VDD regulator is disabled and the regulator output is actively pulled down to discharge any VDD capacitance (5). $\overline{\text{RESET\_A}}$ is activated as well.
- The active discharge guarantees VDD to be below POR level before VDDX discharges below critical level for the reset circuity.

Table 79. ISR - Register Field Descriptions

Field	Description
0 - VSI	VSI - Voltage Status Interrupt combining the following sources: • Low Battery Interrupt • Low Voltage Interrupt • High Voltage Interrupt • Voltage Regulator Overvoltage Interrupt • Voltage Regulator High Temperature Interrupt
1 - CH0	CH0 - TIM Channel 0 Interrupt
2 - CH1	CH1 - TIM Channel 1 Interrupt
3 - CH2	CH2 - TIM Channel 2 Interrupt
4 - CH3	CH3 - TIM Channel 3 Interrupt
5 - TOV	TOV - Timer Overflow Interrupt
6 - ERR	ERR - SCI Error Interrupt
7 - TX	TX - SCI Transmit Interrupt
8 - RX	RX - SCI Receive Interrupt
9 - SCI	SCI - ADC Sequence Complete Interrupt
10 - LINOT	LINOT - LIN Driver Overtemperature Interrupt
11 - HSOT	HSOT - High-side Overtemperature Interrupt
12 - LSOT	LSOT - Low-side Overtemperature Interrupt
13 - HOT	HOT - HSUP Overtemperature Interrupt

4.6.1.2 Interrupt Vector Emulation by Priority

To allow a vector based interrupt handling by the MCU, the number of the highest prioritized interrupt pending is returned in the Interrupt Vector Register. To allow an offset based vector table, the result is pre-shifted (multiple of 2). Reading this register does not acknowledge an interrupt. An additional D2D access is necessary to serve the specific module.

4.6.1.2.1 Interrupt Vector Register (IVR)

Table 80. Interrupt Vector Register (IVR)

Offset[72] 0x02 Access: User read

	7	6	5	4	3	2	1	0
R	0	0	IRQ					
W								

Note:
72. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 81. IVR - Register Field Descriptions

Field	Description
5:0 IRQ	Represents the highest prioritized interrupt pending. See Table 82 In case no interrupt is pending, the result is 0.

The following table is listing all MM912F634 analog die interrupt sources with the corresponding priority.

4.11.7.2 High-side Status Register (HSSR)

Table 99. High-side Status Register (HSSR)

Offset[83] 0x29 Access: User read

	7	6	5	4	3	2	1	0
R	HSOTC	0	0	0	HS2CL	HS1CL	HS2OL	HS1OL
W								
Reset	0	0	0	0	0	0	0	0

Note:
83. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 100. HSSR - Register Field Descriptions

Field	Description
7 - HSOTC	High-side Overtemperature Condition present. Both drivers are turned off. Reading the register clears the HSOT interrupt flag if present. See Section 4.6, "Interrupts" for details.
3 - HS2CL	High-side 2 Current Limitation
2 - HS1CL	High-side 1 Current Limitation
1 - HS2OL	High-side 2 Open Load [84]
0 - HS1OL	High-side 1 Open Load [84]

Note:
84. When the High-side is in OFF state, the Open Load Detection function is not operating. When reading the HSSR register while the High-side is operating in PWM and is in the OFF state, the HS1OL and HS2OL bits does not indicate Open Load.

4.12 Low-side Drivers - LSx

4.12.1 Introduction / Features

These outputs are two Low-side drivers intended to drive relays (inductive loads) incorporating the following features:

- PWM capability
- Open load detection
- Current limitation
- Overtemperature shutdown (with maskable interrupt)
- Active clamp
- Independent VREG - High Voltage Shutdown

To calculate the output period, take the selected clock source period for the channel of interest (A, B, SA, or SB) and multiply it by the value in the period register for this channel:

- Left aligned output (CAEx = 0)
- PWMx Period = Channel Clock Period * PWMPERx Center Aligned Output (CAEx = 1)
- PWMx Period = Channel Clock Period * (2 * PWMPERx)

For boundary case programming values, refer to Section 4.13.4.2.7, “PWM Boundary Cases".

Table 119. PWM Channel Period Registers (PWMPERx)

Offset[95] 0x66/0x67 — Access: User read/write

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0

Note:
95. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.13.3.7 PWM Channel Duty Registers (PWMDTYx)

NOTE

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active duty due to the double buffering scheme.

There is a dedicated duty register for each channel. The value in this register determines the duty of the associated PWM channel. The duty value is compared to the counter and if it is equal to the counter value a match occurs and the output changes state. The duty registers for each channel are double buffered, so if they change while the channel is enabled, the change does NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM is always either the old duty waveform or the new duty waveform, not some variation in between. If the channel is not enabled, then writes to the duty register goes directly to the latches as well as the buffer. See Section 4.13.4.2.3, “PWM Period and Duty" for more information.

NOTE

Depending on the polarity bit, the duty registers contains the count of either the high time or the low time. If the polarity bit is one, the output starts high and then goes low when the duty count is reached, so the duty registers contain a count of the high time. If the polarity bit is zero, the output starts low and then goes high when the duty count is reached, so the duty registers contain a count of the low time.

To calculate the output duty cycle (high time as a% of period) for a particular channel:

- Polarity = 0 (PPOL x =0)
  - Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
- Polarity = 1 (PPOLx = 1)
  - Duty Cycle = [PWMDTYx / PWMPERx] * 100%

For boundary case programming values, refer to Section 4.13.4.2.7, “PWM Boundary Cases".

4.18 Basic Timer Module - TIM (TIM16B4C)

4.18.1 Introduction

4.18.1.1 Overview

The basic timer consists of a 16-bit, software-programmable counter driven by a seven-stage programmable prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from microseconds to many seconds.

This timer contains 4 complete input capture/output compare channels [IOC 3:2]. The input capture function is used to detect a selected transition edge and record the time. The output compare function is used for generating output signals or for timer software delays.

A full access for the counter registers or the input capture/output compare registers should take place in 16bit word access. Accessing high byte and low byte separately for all of these registers may not yield the same result as accessing them in one word.

4.18.1.2 Features

The TIM16B4C includes these distinctive features:

- Four input capture/output compare channels.
- Clock prescaler
- 16-bit counter

4.18.1.3 Modes of Operation

The TIM16B4C is only active during Normal mode.

4.18.1.4 Block Diagram

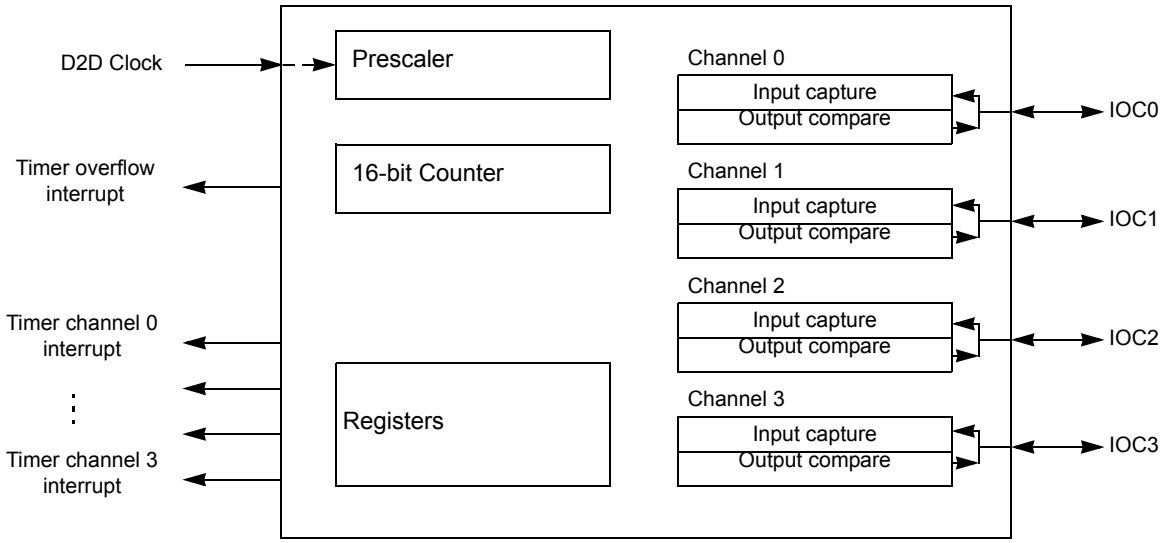


Figure 36. Timer Block Diagram

For more information see the respective functional descriptions see Section 4.18.4, “Functional Description" of this chapter.

4.18.2 Signal Description

4.18.2.1 Overview

The TIM16B4C module is internally connected to the PTB (IOC0, IOC1, IOC2) and to the Rx signal as specified in Section 4.17, “General Purpose I/O - PTB[0…2]" (IOC3).

Table 253. BDM Register Summary (continued)

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x3_FF08	BDMPPR	R	BPAE	0	0	0	BPP3	BPP2	BPP1	BPP0
		W								
0x3_FF09	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3_FF0A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3_FF0B	Reserved	R	0	0	0	0	0	0	0	0
		W								

4.30.3.2.1 BDM Status Register (BDMSTS)

Table 254. Register Global Address 0x3_FF01

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	0	SDV	TRACE	0	UNSEC	0
W								
Reset								
Special Single-Chip Mode	0[176]	1	0	0	0	0	0[177]	0
All Other Modes	0	0	0	0	0	0	0	0

Note:
176. ENBDM is read as 1 by a debugging environment in special single chip mode when the device is not secured or secured but fully erased (Flash). This is because the ENBDM bit is set by the standard firmware before a BDM command can be fully transmitted and executed.
177. UNSEC is read as 1 by a debugging environment in special single chip mode when the device is secured and fully erased, else it is 0 and can only be read if not secure (see also bit description).

4.30.3.2.2 BDM Status Register (BDMSTS)

NOTE

When BDM is made active, the CPU stores the content of its CCR register in the BDMCCR register. However, out of special single-chip reset, the BDMCCR is set to 0xD8 and not 0xD0, which is the reset value of the CCR register in this CPU mode. Out of reset in all other modes, the BDMCCR register is read zero.

Read: All modes through BDM operation when not secured
Write: All modes through BDM operation when not secured, but subject to the following:

- ENBDM should only be set via a BDM hardware command if the BDM firmware commands are needed. (This does not apply in special single chip mode).
- BDMACT can only be set by BDM hardware upon entry into BDM. It can only be cleared by the standard BDM firmware lookup table upon exit from BDM active mode.
- All other bits, while writable via BDM hardware or standard BDM firmware write commands, should only be altered by the BDM hardware or standard firmware lookup table as part of BDM command execution.

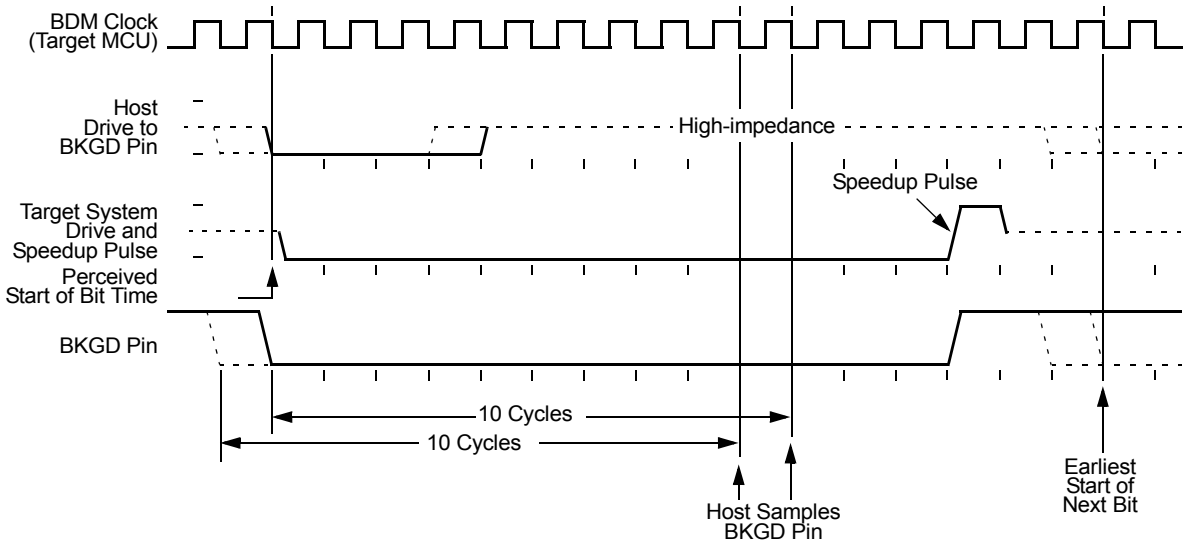


Figure 62. BDM Target-to-Host Serial Bit Timing (Logic 0)

4.30.4.7 Serial Interface Hardware Handshake Protocol

NOTE

If the ACK pulse was issued by the target, the host assumes the previous command was executed. If the CPU enters wait or stop prior to executing a hardware command, the ACK pulse is not issued meaning the BDM command was not executed. After entering wait or stop mode, the BDM command is no longer pending.

BDM commands which require CPU execution are ultimately treated at the MCU bus rate. Since the BDM clock source can be modified relative to the bus clock, when modifying DCO clock or the bus clock divider, it is very helpful to provide a handshake protocol in which the host could determine when an issued command is executed by the CPU. The alternative is to always wait the amount of time equal to the appropriate number of cycles at the slowest possible rate the clock could be running. This sub-section describes the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when an issued command was successfully executed by the target. This protocol is implemented by a 16 serial clock cycle low pulse followed by a brief speedup pulse in the BKGD pin. This pulse is generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 63). This pulse is referred to as the ACK pulse. After the ACK pulse has finished: the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command (BACKGROUND, GO, GO_UNTIL or TRACE1). The ACK pulse is not issued earlier than 32 serial clock cycles after the BDM command was issued. The end of the BDM command is assumed to be the 16th tick of the last bit. This minimum delay assures enough time for the host to perceive the ACK pulse. Note also that, there is no upper limit for the delay between the command and the related ACK pulse, since the command execution depends upon the CPU bus, which in some cases could be very slow due to long accesses taking place.This protocol allows a great flexibility for the POD designers, since it does not rely on any accurate time measurement or short response time to any event in the serial communication.

Figure 66 shows a conflict between the ACK pulse and the SYNC request pulse. This conflict could occur if a POD device is connected to the target BKGD pin and the target is already in debug active mode. Consider that the target CPU is executing a pending BDM command at the exact moment the POD is being connected to the BKGD pin. In this case, an ACK pulse is issued along with the SYNC command. In this case, there is an electrical conflict between the ACK speedup pulse and the SYNC pulse. Since this is not a probable situation, the protocol does not prevent this conflict from happening.

NOTE

This information is being provided so the MCU integrator is aware such a conflict could occur.

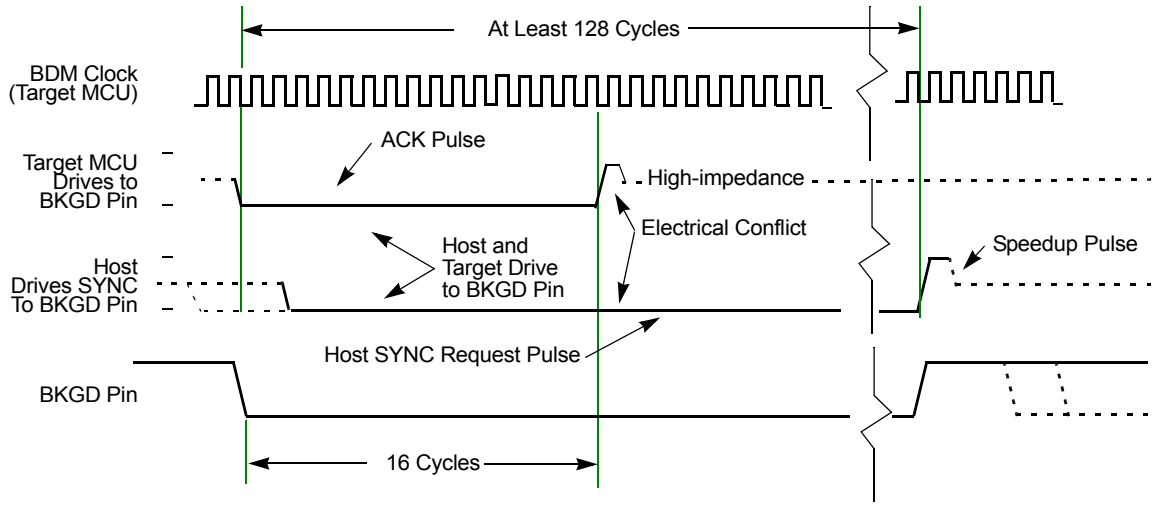


Figure 66. ACK Pulse and SYNC Request Conflict

The hardware handshake protocol is enabled by the ACK_ENABLE and disabled by the ACK_DISABLE BDM commands. This provides backwards compatibility with the existing POD devices which are not able to execute the hardware handshake protocol. It also allows for new POD devices, supporting the hardware handshake protocol, freely communicate with the target device. If desired, without the need for waiting for the ACK pulse.

The commands are described as follows:

- ACK_ENABLE — enables the hardware handshake protocol. The target issues the ACK pulse when a CPU command is executed by the CPU. The ACK_ENABLE command itself also has the ACK pulse as a response.
- ACK_DISABLE — disables the ACK pulse protocol. In this case, the host needs to use the worst case delay time at the appropriate places in the protocol.

The default state of the BDM after reset is hardware handshake protocol disabled. All the read commands ACK (if enabled) when the data bus cycle has completed, and the data is then ready for reading out by the BKGD serial pin. All the write commands ACK (if enabled) after the data has been received by the BDM through the BKGD serial pin, and when the data bus cycle is complete. See Section 4.30.4.3, “BDM Hardware Commands" and Section 4.30.4.4, “Standard BDM Firmware Commands" for more information on the BDM commands.

The ACK_ENABLE sends an ACK pulse when the command has been completed. This feature could be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows the target supports the hardware handshake protocol. If the target does not support the hardware handshake protocol the ACK pulse is not issued. In this case, the ACK_ENABLE command is ignored by the target since it is not recognized as a valid command.

The BACKGROUND command issues an ACK pulse when the CPU changes from normal to background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO command issues an ACK pulse when the CPU exits from background mode. The ACK pulse related to this command could be aborted using the SYNC command.

NOTE

When a COF instruction with destination address is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously, then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets executed after the interrupt service routine.

In the following example, an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The BRN at the destination (SUB_1) is not executed until after the IRQ service routine, but the destination address is entered into the trace buffer to indicate the indexed JMP COF has taken place.

```
        LDX     #SUB_1
MARK1   JMP     0,X                     ; IRQ interrupt occurs during execution of this
MARK2   NOP                             ;

SUB_1   BRN     *                       ; JMP Destination address TRACE BUFFER ENTRY 1
                                        ; RTI Destination address TRACE BUFFER ENTRY 3
        NOP                             ;
ADDR1   DBNE    A,PART5                 ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR LDAB    #$F0                    ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
        STAB    VAR_C1
        RTI                             ;
```

The execution flow taking into account the IRQ is as follows

```
        LDX     #SUB_1
MARK1   JMP     0,X                     ;
IRQ_ISR LDAB    #$F0                    ;
        STAB    VAR_C1
        RTI                             ;
SUB_1   BRN     *
        NOP                             ;
ADDR1   DBNE    A,PART5                 ;
```

4.31.4.5.3.1 Loop1 Mode

NOTE

In certain very tight loops, the source address has already been fetched again before the background comparator is updated. This results in the source address being stored twice before further duplicate entries are suppressed. This condition occurs with branch-on-bit instructions when the branch is fetched by the first P-cycle of the branch or with loop-construct instructions in which the branch is fetched with the first or second P cycle. See examples below:

Loop1 mode, similarly to Normal mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the DBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 mode only inhibits consecutive duplicate source address entries which would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user's code which the DBG module is designed to help find.

```
LOOP    INX                             ; 1-byte instruction fetched by 1st P-cycle of BRCLR
        BRCLR           CMPTMP,#$0c,LOOP ; the BRCLR instruction also is fetched by 1st
                                        ; P-cycle of BRCLR

LOOP2   BRN     *                       ; 2-byte instruction fetched by 1st P-cycle of DBNE
        NOP                             ; 1-byte instruction fetched by 2nd P-cycle of DBNE
        DBNE            A,LOOP2         ; this instruction also fetched by 2nd P-cycle of DBNE
```

Table 317. Field Descriptions

Field	Description
3 CSZ	**Access Type Indicator**— This bit indicates if the access was a byte or word size when tracing in Detail mode 0 Word Access 1 Byte Access
2 CRW	**Read Write Indicator** — This bit indicates if the corresponding stored address corresponds to a read or write access when tracing in Detail Mode. 0 Write Access 1 Read Access
1 ADRH17	**Address Bus bit 17**— Corresponds to system address bus bit 17.
0 ADRH16	**Address Bus bit 16**— Corresponds to system address bus bit 16.

4.31.4.5.4.3 Field2 Bits in Normal, Pure PC and Loop1 Modes

Table 318. Information Bits PCH

Bit 3	Bit 2	Bit 1	Bit 0
CSD	CVA	PC17	PC16

Table 319. PCH Field Descriptions

Field	Description
3 CSD	**Source Destination Indicator** — In Normal and Loop1 mode this bit indicates if the corresponding stored address is a source or destination address. This bit has no meaning in Pure PC mode. 0 Source Address 1 Destination Address
2 CVA	**Vector Indicator** — In Normal and Loop1 mode this bit indicates if the corresponding stored address is a vector address. Vector addresses are destination addresses, thus if CVA is set, then the corresponding CSD is also set. This bit has no meaning in Pure PC mode. 0 Non-Vector Destination Address 1 Vector Destination Address
1 PC17	**Program Counter bit 17**— In Normal, Pure PC, and Loop1 mode this bit corresponds to program counter bit 17.
0 PC16	**Program Counter bit 16**— In Normal, Pure PC, and Loop1 mode this bit corresponds to program counter bit 16.

4.31.4.5.5 Reading Data from Trace Buffer

The data stored in the Trace Buffer can be read, provided the DBG module is not armed, is configured for tracing (TSOURCE bit is set) and the system not secured. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by a single aligned word write to DBGTB when the module is disarmed.

The Trace Buffer can only be read through the DBGTB register using aligned word reads, any byte or misaligned reads return 0 and do not cause the trace buffer pointer to increment to the next trace buffer address. The Trace Buffer data is read out first-in first-out. By reading CNT in DBGCNT, the number of valid lines can be determined. DBGCNT does not decrement as data is read.

Whilst reading, an internal pointer is used to determine the next line to be read. After a tracing session, the pointer points to the oldest data entry, thus if no overflow has occurred, the pointer points to line0, otherwise it points to the line with the oldest entry. The pointer is initialized by each aligned write to DBGTBH to point to the oldest data again. This enables an interrupted trace buffer read sequence to be easily restarted from the oldest data entry.

4.32.1.2.2 Wait Mode

For 9S12I32PIMV1 Wait mode is same as Run mode.

4.32.1.2.3 Stop Mode

- This mode is entered by executing the CPU STOP instruction.
- The Bus Clock is turned off.
- The Oscillator Clock and the Oscillator Monitor is turned off.
- The DCO Clock is turned off.
- The Internal Reference Clock can be kept enabled by peripherals like e.g. the Real Time Interrupt module (RTI). See device and other block descriptions for details.

4.32.1.3 Block Diagram

Figure 70 shows a block diagram of the 9S12I32PIMV1.

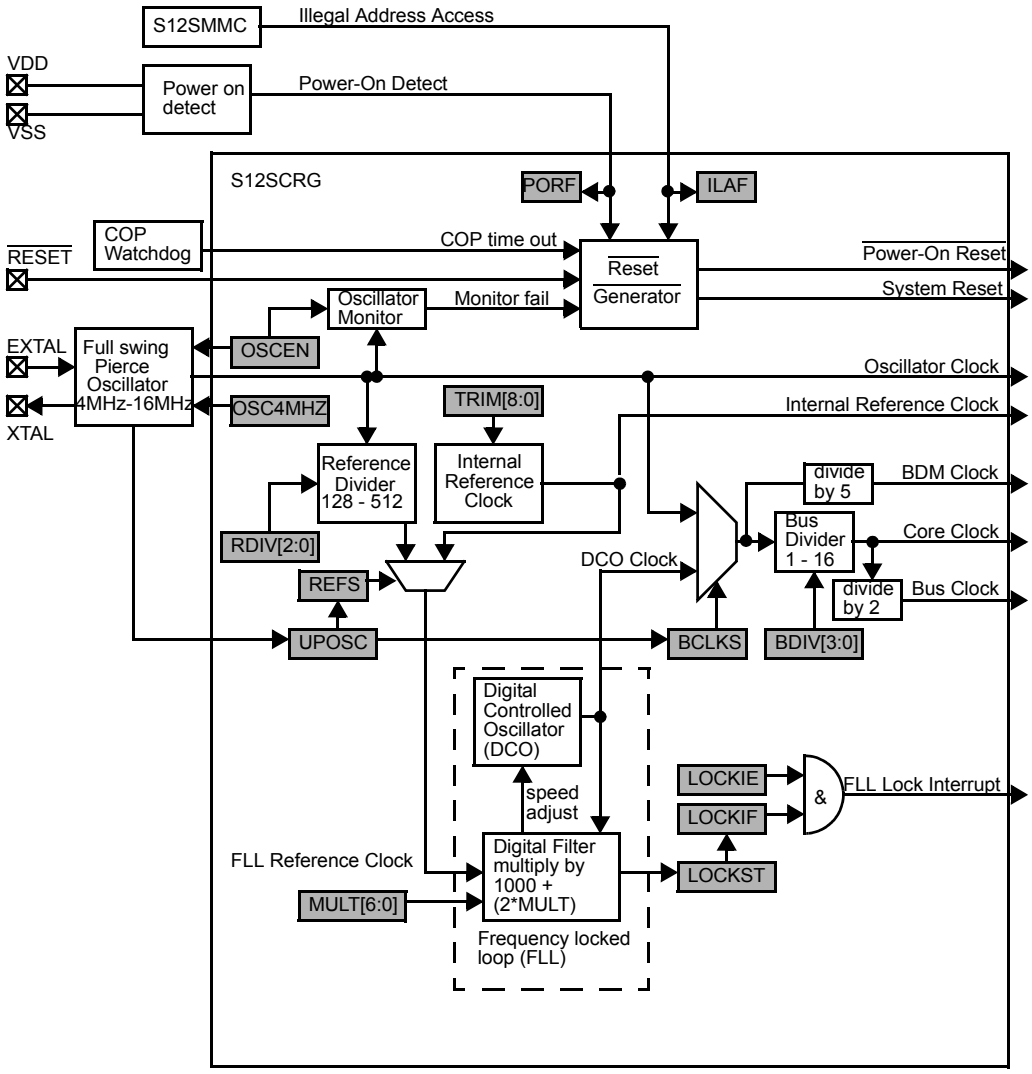


Figure 70. Block diagram of 9S12I32PIMV1

4.32.3.2.1 9S12I32PIMV1 Control Register 0 (CRGCTL0)

Table 323. 9S12I32PIMV1 Control Register 0 (CRGCTL0)

0x0034

	7	6	5	4	3	2	1	0
R	OSCEN	RDIV[2:0]			BCLKS	REFS	OSC4MHZ	0
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime
Write: See individual bit descriptions.

Writing the CRGCTL0 register clears the LOCKST bit, but does not set the LOCKIF bit in the CRGFLG register.

Table 324. CRGCTL0 Field Descriptions

Field	Description
7 OSCEN	Oscillator Enable Bit 0 Oscillator Clock and Oscillator Monitor are disabled. 1 Oscillator Clock and Oscillator Monitor are enabled.
6, 5, 4 RDIV[2:0]	Reference Divider Bits These bits divide the Oscillator Clock down in frequency. Divided down frequency must be in the allowed range for f_{FLLREF}. See device electrical characteristics for details. 000 divide by 128 001 divide by 160 010 divide by 192 011 divide by 256 100 divide by 320 101 divide by 384 110 divide by 512 111 Reserved
3 BCLKS	Bus Clock Source Select Bit Writing BCLKS = 1 is only possible if oscillator startup flag is set (UPOSC = 1). BCLKS is cleared with disabling the Oscillator, which is either OSCEN = 0 or entering Stop Mode. 0 DCO Clock is selected as basis for the Bus Clock. 1 Oscillator Clock is selected as basis for the Bus Clock. DCO is disabled.
2 REFS	Reference Select Bit Writing REFS = 1 is only possible if oscillator startup flag is set (UPOSC = 1). REFS is cleared with disabling the Oscillator, which is either OSCEN = 0 or entering Stop Mode. 0 Internal Reference Clock is selected as FLL Reference Clock. 1 Divided down Oscillator Clock is selected as FLL Reference Clock.
1 OSC4MHZ	4.0 MHz Oscillator low pass filter select Bit The Oscillator contains a noise filter in its signal path from EXTAL/XTAL to chip internal Oscillator Clock. This is to improve high frequency noise immunity. Writing OSC4MHZ is only possible if OSCEN was zero before. 0 Oscillator uses noise filter with high bandwidth. To be used with crystals/resonators > 4.0 Mhz. 1 Oscillator uses noise filter with low bandwidth. To be used with crystals/resonators = 4.0 Mhz. Choosing a low bandwidth in case of a 4.0 MHz crystal/resonator further improves noise immunity at lower frequencies.

4.35.1.2 Features

The COP includes these distinctive features:

- Watchdog timer with a timeout clear window.
- Default maximum COP rate and no Window COP in Special Single Chip mode after system reset.
- Auto COP rate load after system reset in SoC Normal mode. (For source of COP rate bits refer to the Device User Guide)
- Software selectable COP operation in WAIT and STOP mode.
- Customer selectable COP off while BDM active (debugging session).

4.35.1.3 Modes of Operation

- Run mode
  If COP functionality is required, the individual bits of the associated rate select registers (COPCTL) have to be set to a non-zero value. The COP is stopped if all rate select bits are zero.
- Wait mode
  If the respective enable bit (COPSWAI) is cleared, the COP continues to run, else COP remains frozen.
- Stop mode
  If the respective enable bit (COPRSTP) is set, the COP continues to run, else COP remains frozen.

4.35.2 External Signal Description

There are no external signals associated with this module.

4.35.3 Memory Map and Register

4.35.3.1 Module Memory Map

A summary of the registers associated with the COP module is shown in Table 339.

Table 344. COP Register Summary

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x003E	COPCTL	R	WCOP	RSBCK	0	COPSWAI	COPRSTP	CR2	CR1	CR0
		W			WRTMASK					
0x003F	ARMCOP	R	0	0	0	0	0	0	0	0
		W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

4.35.3.2 Register Descriptions

This section describes in address order all the S12SCOP registers and their individual bits

Table 350. Initial WCOP Configuration

NV[3] in FCTL Register	WCOP in COPCTL Register
1	0
0	1

4.36 32 kbyte Flash Module (S12SFTSR32KV1)

4.36.1 Introduction

This document describes the S12SFTSR32K module, which includes a 32 kbyte Flash (nonvolatile) memory.

CAUTION

A Flash block address must be in the erased state before being programmed. Cumulative programming of bits within a Flash block address is not allowed, except for status field updates required in EEPROM emulation applications.

The Flash memory is ideal for single-supply applications, allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller which executes commands to modify Flash memory contents.

Array read access time is one bus cycle for bytes and aligned words, and two bus cycles for misaligned words. For Flash memory, an erased bit reads 1 and a programmed bit reads 0. It is not possible to read from a Flash block while any command is executing on a specific Flash block.

4.36.1.1 Glossary

Command Write Sequence — A three step MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.

Flash Array — The Flash array constitutes the main memory portion of a Flash block.

Flash Block — An analog block consisting of the Flash array and Flash IFR with supporting high voltage and parametric test circuitry.

Flash IFR — Nonvolatile information memory, consisting of 128 bytes, located in the Flash block outside of Flash main memory. Refer to the SoC Guide on how to make the Flash IFR visible in the global memory map.

4.36.1.2 Features

- 32 kbytes of Flash memory comprised of one 32 kbyte block divided into 64 sectors of 512 bytes
- Nonvolatile information memory (Flash IFR) comprised of one 128 byte block
- Automated program and erase algorithm
- Interrupt on Flash command completion, command buffer empty
- Fast program and sector erase operation
- Burst program command for faster Flash array program times
- Flexible protection scheme to prevent accidental program or erase
- Single power supply for all Flash operations including program and erase
- Security feature to prevent unauthorized access to the Flash memory

4.38.3.2.4 SPI Status Register (SPISR)

Table 415. SPI Status Register (SPISR)

0x00EB

	7	6	5	4	3	2	1	0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W								
Reset	0	0	1	0	0	0	0	0

Read: Anytime
Write: Has no effect

Table 416. SPISR Field Descriptions

Field	Description
7 SPIF	**SPIF Interrupt Flag** — This bit is set after a received data byte has been transferred into the SPI data register. This bit is cleared by reading the SPISR register (with SPIF set) followed by a read access to the SPI data register. 0 Transfer not yet complete. 1 New data copied to SPIDR.
5 SPTEF	**SPI Transmit Empty Interrupt Flag** — If set, this bit indicates the transmit data register is empty. To clear this bit and place data into the transmit data register, SPISR must be read with SPTEF = 1, followed by a write to SPIDR. Any write to the SPI data register without reading SPTEF = 1, is effectively ignored. 0 SPI data register not empty. 1 SPI data register empty.
4 MODF	**Mode Fault Flag** — This bit is set if the $\overline{SS}$ input becomes low, while the SPI is configured as a master and mode fault detection is enabled, the MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in Section 4.38.3.2.2, “SPI Control Register 2 (SPICR2)". The flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to the SPI control register 1. 0 Mode fault has not occurred. 1 Mode fault has occurred.

4.38.3.2.5 SPI Data Register (SPIDR)

Table 417. SPI Data Register (SPIDR)

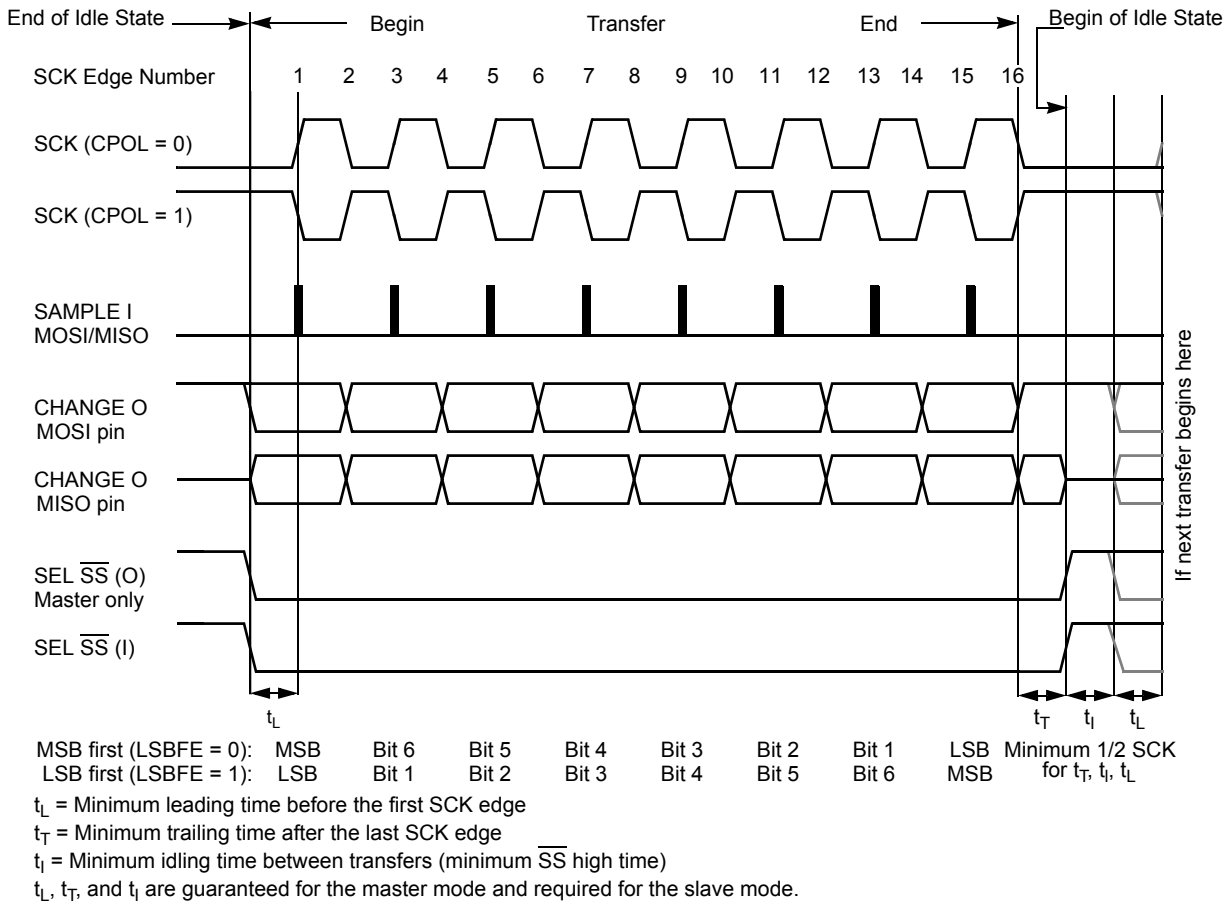
0x00ED

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	2	Bit 0
Reset	0	0	0	0	0	0	0	0

Read: Anytime; normally read only when SPIF is set
Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows a data byte to be queued and transmitted. For an SPI configured as a master, a queued data byte is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data.

Received data in the SPIDR is valid when SPIF is set. If SPIF is cleared and a byte has been received, the received byte is transferred from the receive shift register to the SPIDR and SPIF is set. If SPIF is set and not serviced, and a second byte has been received, the second received byte is kept as valid byte in the receive shift register until the start of another transmission. The byte in the SPIDR does not change.



t_L = Minimum leading time before the first SCK edge
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)
t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 101. SPI Clock Format 0 (CPHA = 0)

In slave mode, if the $\overline{SS}$ line is not de-asserted between the successive transmissions, then the content of the SPI data register is not transmitted. Instead, the last received byte is transmitted. If the $\overline{SS}$ line is de-asserted for at least minimum idle time (half SCK cycle) between successive transmissions, then the content of the SPI data register is transmitted.

In master mode, with slave select output enabled the $\overline{SS}$ line is always de-asserted and reasserted between successive transfers for at least minimum idle time.

4.38.4.3.3 CPHA = 1 Transfer Format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the 8-cycle transfer operation. The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master. A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the SPI shift register, depending on LSBFE bit. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

This process continues for a total of 16 edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges. Data reception is double buffered, data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in. After the 16th SCK edge:

- Data previously in the SPI data register of the master is now in the data register of the slave, and data in the data register of the slave is in the master.
- The SPIF flag bit in SPISR is set indicating the transfer is complete.