Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	54
Program Memory Size	192KB (192K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	11K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12g192f0vlh

Table 1-24. 64-Pin LQFP Pinout for S12GA96 and S12GA128

	Function <----lowest-----PRIORITY-----highest---->					Power Supply	Internal Pull Resistor	
Package Pin	Pin	2nd Func.	3rd Func.	4th Func	5th Func		CTRL	Reset State
1	PJ6	KWJ6	SCK2	—	—	V_{DDX}	PERJ/PPSJ	Up
2	PJ5	KWJ5	MOSI2	—	—	V_{DDX}	PERJ/PPSJ	Up
3	PJ4	KWJ4	MISO2	—	—	V_{DDX}	PERJ/PPSJ	Up
4	RESET	—	—	—	—	V_{DDX}	PULLUP	
5	VDDX	—	—	—	—	—	—	—
6	VDDR	—	—	—	—	—	—	—
7	VSSX	—	—	—	—	—	—	—
8	PE0[1]	EXTAL	—	—	—	V_{DDX}	PUCR/PDPEE	Down
9	VSS	—	—	—	—	—	—	—
10	PE1[1]	XTAL	—	—	—	V_{DDX}	PUCR/PDPEE	Down
11	TEST	—	—	—	—	N.A.	$\overline{\text{RESET}}$ pin	Down
12	PJ0	KWJ0	MISO1	—	—	V_{DDX}	PERJ/PPSJ	Up
13	PJ1	KWJ1	MOSI1	—	—	V_{DDX}	PERJ/PPSJ	Up
14	PJ2	KWJ2	SCK1	—	—	V_{DDX}	PERJ/PPSJ	Up
15	PJ3	KWJ3	$\overline{\text{SS1}}$	—	—	V_{DDX}	PERJ/PPSJ	Up
16	BKGD	MODC	—	—	—	V_{DDX}	PUCR/BKPUE	Up
17	PP0	KWP0	ETRIG0	API_EXTCLK	PWM0	V_{DDX}	PERP/PPSP	Disabled
18	PP1	KWP1	ETRIG1	ECLKX2	PWM1	V_{DDX}	PERP/PPSP	Disabled
19	PP2	KWP2	ETRIG2	PWM2	—	V_{DDX}	PERP/PPSP	Disabled
20	PP3	KWP3	ETRIG3	PWM3	—	V_{DDX}	PERP/PPSP	Disabled
21	PP4	KWP4	PWM4	—	—	V_{DDX}	PERP/PPSP	Disabled
22	PP5	KWP5	PWM5	—	—	V_{DDX}	PERP/PPSP	Disabled
23	PP6	KWP6	PWM6	—	—	V_{DDX}	PERP/PPSP	Disabled
24	PP7	KWP7	PWM7	—	—	V_{DDX}	PERP/PPSP	Disabled
25	PT7	IOC7	—	—	—	V_{DDX}	PERT/PPST	Disabled
26	PT6	IOC6	—	—	—	V_{DDX}	PERT/PPST	Disabled
27	PT5	IOC5	—	—	—	V_{DDX}	PERT/PPST	Disabled

Table 1-31. 100-Pin LQFP Pinout for S12GA192 and S12GA240

	Function <----lowest-----PRIORITY-----highest---->				Power Supply	Internal Pull Resistor	
Package Pin	Pin	2nd Func.	3rd Func.	4th Func.		CTRL	Reset State
86	PS4	MISO0	—	—	V_{DDX}	PERS/PPSS	Up
87	PS5	MOSI0	—	—	V_{DDX}	PERS/PPSS	Up
88	PS6	SCK0	—	—	V_{DDX}	PERS/PPSS	Up
89	PS7	API_EXTCLK	$\overline{SS0}$	—	V_{DDX}	PERS/PPSS	Up
90	VSSX2	—	—	—	—	—	—
91	VDDX2	—	—	—	—	—	—
92	PM0	RXCAN	—	—	V_{DDX}	PERM/PPSM	Disabled
93	PM1	TXCAN	—	—	V_{DDX}	PERM/PPSM	Disabled
94	PD4	—	—	—	V_{DDX}	PUCR/PUPDE	Disabled
95	PD5	—	—	—	V_{DDX}	PUCR/PUPDE	Disabled
96	PD6	—	—	—	V_{DDX}	PUCR/PUPDE	Disabled
97	PD7	—	—	—	V_{DDX}	PUCR/PUPDE	Disabled
98	PM2	RXD2	—	—	V_{DDX}	PERM/PPSM	Disabled
99	PM3	TXD2	—	—	V_{DDX}	PERM/PPSM	Disabled
100	PJ7	KWJ7	$\overline{SS2}$	—	V_{DDX}	PERJ/PPSJ	Up

[1] The regular I/O characteristics (see Section A.2, “I/O Characteristics”) apply if the EXTAL/XTAL function is disabled

NOTE

The ACK pulse does not provide a time out. This means for the GO_UNTIL command that it can not be distinguished if a stop or wait has been executed (command discarded and ACK not issued) or if the “UNTIL” condition (BDM active) is just not reached yet. Hence in any case where the ACK pulse of a command is not issued the possible pending command should be aborted before issuing a new command. See the handshake abort procedure described in Section 7.4.8, “Hardware Handshake Abort Procedure”.

7.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. In order to abort a command, which had not issued the corresponding ACK pulse, the host controller should generate a low pulse in the BKGD pin by driving it low for at least 128 serial clock cycles and then driving it high for one serial clock cycle, providing a speedup pulse. By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see Section 7.4.9, “SYNC — Request Timed Reference Pulse”, and assumes that the pending command and therefore the related ACK pulse, are being aborted. Therefore, after the SYNC protocol has been completed the host is free to issue new BDM commands. For BDM firmware READ or WRITE commands it can not be guaranteed that the pending command is aborted when issuing a SYNC before the corresponding ACK pulse. There is a short latency time from the time the READ or WRITE access begins until it is finished and the corresponding ACK pulse is issued. The latency time depends on the firmware READ or WRITE command that is issued and on the selected bus clock rate. When the SYNC command starts during this latency time the READ or WRITE command will not be aborted, but the corresponding ACK pulse will be aborted. A pending GO, TRACE1 or GO_UNTIL command can not be aborted. Only the corresponding ACK pulse can be aborted by the SYNC command.

Although it is not recommended, the host could abort a pending BDM command by issuing a low pulse in the BKGD pin shorter than 128 serial clock cycles, which will not be interpreted as the SYNC command. The ACK is actually aborted when a negative edge is perceived by the target in the BKGD pin. The short abort pulse should have at least 4 clock cycles keeping the BKGD pin low, in order to allow the negative edge to be detected by the target. In this case, the target will not execute the SYNC protocol but the pending command will be aborted along with the ACK pulse. The potential problem with this abort procedure is when there is a conflict between the ACK pulse and the short abort pulse. In this case, the target may not perceive the abort pulse. The worst case is when the pending command is a read command (i.e., READ_BYTE). If the abort pulse is not perceived by the target the host will attempt to send a new command after the abort pulse was issued, while the target expects the host to retrieve the accessed memory byte. In this case, host and target will run out of synchronism. However, if the command to be aborted is not a read command the short abort pulse could be used. After a command is aborted the target assumes the next negative edge, after the abort pulse, is the first bit of a new BDM command.

NOTE

The details about the short abort pulse are being provided only as a reference for the reader to better understand the BDM internal behavior. It is not recommended that this procedure be used in a real application.

Since the host knows the target serial clock frequency, the SYNC command (used to abort a command) does not need to consider the lower possible target frequency. In this case, the host could issue a SYNC very close to the 128 serial clock cycles length. Providing a small overhead on the pulse length in order to assure the SYNC pulse will not be misinterpreted by the target. See Section 7.4.9, “SYNC — Request Timed Reference Pulse”.

Figure 7-12 shows a SYNC command being issued after a READ_BYTE, which aborts the READ_BYTE command. Note that, after the command is aborted a new command could be issued by the host computer.

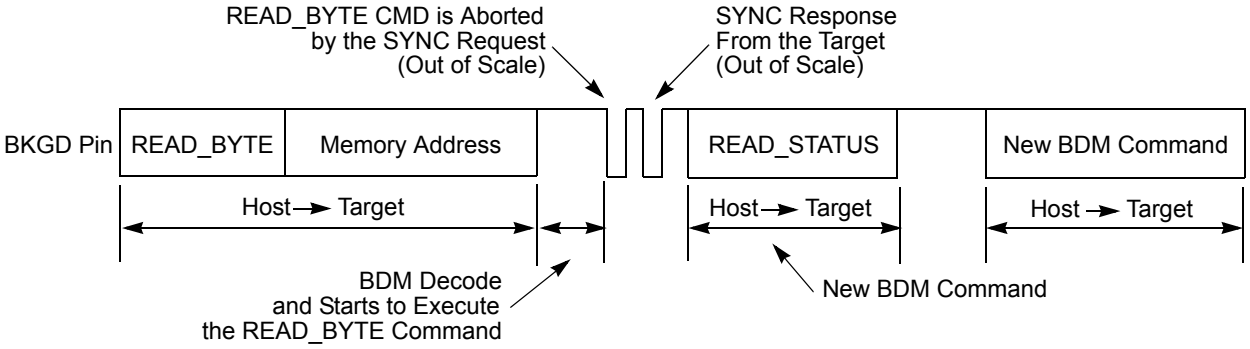


Figure 7-12. ACK Abort Procedure at the Command Level

NOTE

Figure 7-12 does not represent the signals in a true timing scale

Figure 7-13 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.

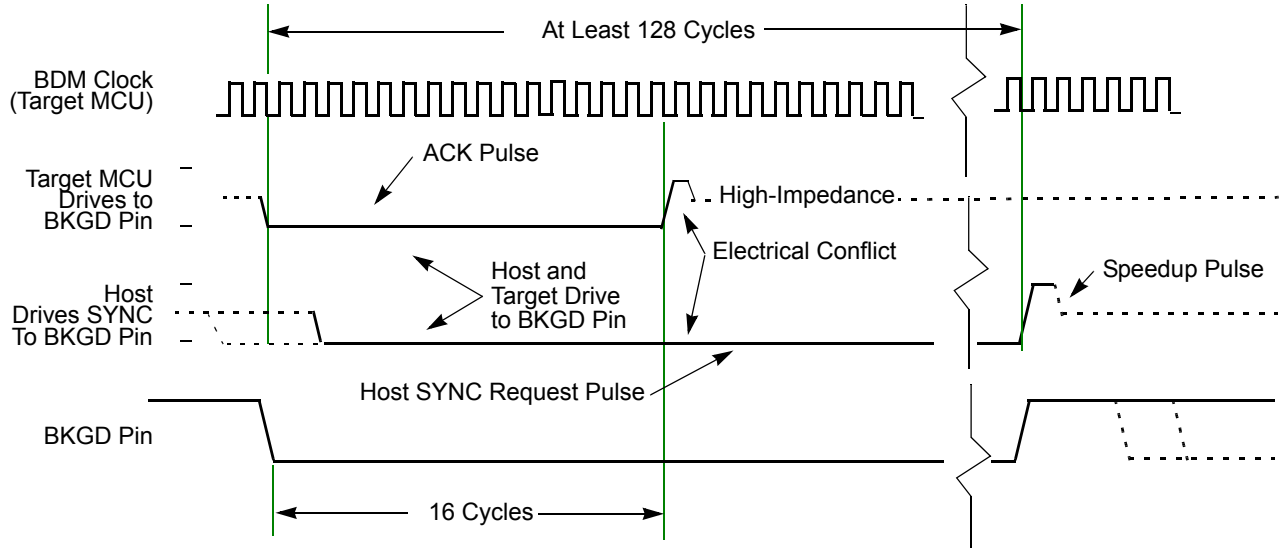


Figure 7-13. ACK Pulse and SYNC Request Conflict

NOTE

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

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, that support the hardware handshake protocol, to 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 will issue 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 will 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 will 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 7.4.3, “BDM Hardware Commands” and Section 7.4.4, “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 that 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 will issue 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 will issue an ACK pulse when the CPU exits from background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO_UNTIL command is equivalent to a GO command with exception that the ACK pulse, in this case, is issued when the CPU enters into background mode. This command is an alternative to the GO command and should be used when the host wants to trace if a breakpoint match occurs and causes the CPU to enter active background mode. Note that the ACK is issued whenever the CPU enters BDM, which could be caused by a breakpoint match or by a BGND instruction being executed. The ACK pulse related to this command could be aborted using the SYNC command.

The TRACE1 command has the related ACK pulse issued when the CPU enters background active mode after one instruction of the application program is executed. The ACK pulse related to this command could be aborted using the SYNC command.

Table 12-16. ATDSTAT0 Field Descriptions (continued)

Field	Description
3–0 CC[3:0]	**Conversion Counter** — These 4 read-only bits are the binary value of the conversion counter. The conversion counter points to the result register that will receive the result of the current conversion. E.g. CC3=0, CC2=1, CC1=1, CC0=0 indicates that the result of the current conversion will be in ATD Result Register 6. If in non-FIFO mode (FIFO=0) the conversion counter is initialized to zero at the beginning and end of the conversion sequence. If in FIFO mode (FIFO=1) the register counter is not initialized. The conversion counter wraps around when its maximum value is reached. Aborting a conversion or starting a new conversion clears the conversion counter even if FIFO=1.

12.3.2.8 ATD Compare Enable Register (ATDCMPE)

Writes to this register will abort current conversion sequence.

Read: Anytime

Write: Anytime

Module Base + 0x0008

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	CMPE[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 12-10. ATD Compare Enable Register (ATDCMPE)

Table 12-17. ATDCMPE Field Descriptions

Field	Description
7–0 CMPE[7:0]	**Compare Enable for Conversion Number *n* (*n*= 7, 6, 5, 4, 3, 2, 1, 0) of a Sequence (*n conversion number, NOT channel number!)*** — These bits enable automatic compare of conversion results individually for conversions of a sequence. The sense of each comparison is determined by the CMPHT[*n*] bit in the ATDCMPHT register. For each conversion number with CMPE[*n*]=1 do the following: 1) Write compare value to ATDDR*n* result register 2) Write compare operator with CMPHT[*n*] in ATDCPMHT register CCF[*n*] in ATDSTAT2 register will flag individual success of any comparison. 0 No automatic compare 1 Automatic compare of results for conversion *n* of a sequence is enabled.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0024	ATDDR10	R W	See Section 13.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 13.3.2.12.2, “Right Justified Result Data (DJM=1)”							
0x0026	ATDDR11	R W	See Section 13.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 13.3.2.12.2, “Right Justified Result Data (DJM=1)”							
0x0028 - 0x002F	Unimple-mented	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented or Reserved

Figure 13-2. ADC10B12C Register Summary (Sheet 3 of 3)

14.3.2.7 ATD Status Register 0 (ATDSTAT0)

This register contains the Sequence Complete Flag, overrun flags for external trigger and FIFO mode, and the conversion counter.

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-9. ATD Status Register 0 (ATDSTAT0)

Read: Anytime

Write: Anytime (No effect on (CC3, CC2, CC1, CC0))

Table 14-16. ATDSTAT0 Field Descriptions

Field	Description
7 SCF	**Sequence Complete Flag** — This flag is set upon completion of a conversion sequence. If conversion sequences are continuously performed (SCAN=1), the flag is set after each one is completed. This flag is cleared when one of the following occurs: A) Write “1” to SCF B) Write to ATDCTL5 (a new conversion sequence is started) C) If AFFC=1 and a result register is read 0 Conversion sequence not completed 1 Conversion sequence has completed
5 ETORF	**External Trigger Overrun Flag** — While in edge sensitive mode (ETRIGLE=0), if additional active edges are detected while a conversion sequence is in process the overrun flag is set. This flag is cleared when one of the following occurs: A) Write “1” to ETORF B) Write to ATDCTL0,1,2,3,4, ATDCMPE or ATDCMPHT (a conversion sequence is aborted) C) Write to ATDCTL5 (a new conversion sequence is started) 0 No External trigger overrun error has occurred 1 External trigger overrun error has occurred
4 FIFOR	**Result Register Overrun Flag** — This bit indicates that a result register has been written to before its associated conversion complete flag (CCF) has been cleared. This flag is most useful when using the FIFO mode because the flag potentially indicates that result registers are out of sync with the input channels. However, it is also practical for non-FIFO modes, and indicates that a result register has been overwritten before it has been read (i.e. the old data has been lost). This flag is cleared when one of the following occurs: A) Write “1” to FIFOR B) Write to ATDCTL0,1,2,3,4, ATDCMPE or ATDCMPHT (a conversion sequence is aborted) C) Write to ATDCTL5 (a new conversion sequence is started) 0 No overrun has occurred 1 Overrun condition exists (result register has been written while associated CCFx flag was still set)

Table 15-6. ATDCTL2 Field Descriptions

Field	Description
6 AFFC	**ATD Fast Flag Clear All** 0 ATD flag clearing done by write 1 to respective CCF[*n*] flag. 1 Changes all ATD conversion complete flags to a fast clear sequence. For compare disabled (CMPE[*n*]=0) a read access to the result register will cause the associated CCF[*n*] flag to clear automatically. For compare enabled (CMPE[*n*]=1) a write access to the result register will cause the associated CCF[*n*] flag to clear automatically.
5 Reserved	Do not alter this bit from its reset value.It is for Manufacturer use only and can change the ATD behavior.
4 ETRIGLE	**External Trigger Level/Edge Control** — This bit controls the sensitivity of the external trigger signal. See Table 15-7 for details.
3 ETRIGP	**External Trigger Polarity** — This bit controls the polarity of the external trigger signal. See Table 15-7 for details.
2 ETRIGE	**External Trigger Mode Enable** — This bit enables the external trigger on one of the AD channels or one of the ETRIG3-0 inputs as described in Table 15-5. If the external trigger source is one of the AD channels, the digital input buffer of this channel is enabled. The external trigger allows to synchronize the start of conversion with external events. 0 Disable external trigger 1 Enable external trigger
1 ASCIE	**ATD Sequence Complete Interrupt Enable** 0 ATD Sequence Complete interrupt requests are disabled. 1 ATD Sequence Complete interrupt will be requested whenever SCF=1 is set.
0 ACMPIE	**ATD Compare Interrupt Enable** — If automatic compare is enabled for conversion *n* (CMPE[*n*]=1 in ATDCMPE register) this bit enables the compare interrupt. If the CCF[*n*] flag is set (showing a successful compare for conversion *n*), the compare interrupt is triggered. 0 ATD Compare interrupt requests are disabled. 1 For the conversions in a sequence for which automatic compare is enabled (CMPE[*n*]=1), an ATD Compare Interrupt will be requested whenever any of the respective CCF flags is set.

Table 15-7. External Trigger Configurations

ETRIGLE	ETRIGP	External Trigger Sensitivity
0	0	Falling edge
0	1	Rising edge
1	0	Low level
1	1	High level

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0003	ATDCTL3	R	DJM	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
		W								
0x0004	ATDCTL4	R	SMP2	SMP1	SMP0	PRS[4:0]				
		W								
0x0005	ATDCTL5	R	0	SC	SCAN	MULT	CD	CC	CB	CA
		W								
0x0006	ATDSTAT0	R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
		W								
0x0007	Unimple-mented	R	0	0	0	0	0	0	0	0
		W								
0x0008	ATDCMPEH	R	CMPE[15:8]							
		W								
0x0009	ATDCMPEL	R	CMPE[7:0]							
		W								
0x000A	ATDSTAT2H	R	CCF[15:8]							
		W								
0x000B	ATDSTAT2L	R	CCF[7:0]							
		W								
0x000C	ATDDIENH	R	IEN[15:8]							
		W								
0x000D	ATDDIENL	R	IEN[7:0]							
		W								
0x000E	ATDCMPHTH	R	CMPHT[15:8]							
		W								
0x000F	ATDCMPHTL	R	CMPHT[7:0]							
		W								
0x0010	ATDDR0	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0012	ATDDR1	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0014	ATDDR2	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0016	ATDDR3	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0018	ATDDR4	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001A	ATDDR5	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001C	ATDDR6	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001E	ATDDR7	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0020	ATDDR8	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0022	ATDDR9	R	See Section 16.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 16.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								

= Unimplemented or Reserved

Figure 16-2. ADC12B16C Register Summary (Sheet 2 of 3)

16.3.2.10 ATD Input Enable Register (ATDDIEN)

Module Base + 0x000C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IEN[15:0]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-12. ATD Input Enable Register (ATDDIEN)

Read: Anytime

Write: Anytime

Table 16-19. ATDDIEN Field Descriptions

Field	Description
15–0 IEN[15:0]	**ATD Digital Input Enable on channel *x* (*x*= 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)** — This bit controls the digital input buffer from the analog input pin (AN*x*) to the digital data register. 0 Disable digital input buffer to AN*x* pin 1 Enable digital input buffer on AN*x* pin. **Note:** Setting this bit will enable the corresponding digital input buffer continuously. If this bit is set while simultaneously using it as an analog port, there is potentially increased power consumption because the digital input buffer maybe in the linear region.

16.3.2.11 ATD Compare Higher Than Register (ATDCMPHT)

Writes to this register will abort current conversion sequence.

Read: Anytime

Write: Anytime

Module Base + 0x000E

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMPHT[15:0]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-13. ATD Compare Higher Than Register (ATDCMPHT)

Table 16-20. ATDCMPHT Field Descriptions

Field	Description
15–0 CMPHT[15:0]	**Compare Operation Higher Than Enable for conversion number *n* (*n*= 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0) of a Sequence (*n conversion number, NOT channel number!)*** — This bit selects the operator for comparison of conversion results. 0 If result of conversion *n* is **lower or same than** compare value in ATDDR*n*, this is flagged in ATDSTAT2 1 If result of conversion *n* is **higher than** compare value in ATDDR*n*, this is flagged in ATDSTAT2

Table 20-10. SCICR2 Field Descriptions (continued)

Field	Description
1 RWU	**Receiver Wakeup Bit** — Standby state 0 Normal operation. 1 RWU enables the wakeup function and inhibits further receiver interrupt requests. Normally, hardware wakes the receiver by automatically clearing RWU.
0 SBK	**Send Break Bit** — Toggling SBK sends one break character (10 or 11 logic 0s, respectively 13 or 14 logics 0s if BRK13 is set). Toggling implies clearing the SBK bit before the break character has finished transmitting. As long as SBK is set, the transmitter continues to send complete break characters (10 or 11 bits, respectively 13 or 14 bits). 0 No break characters 1 Transmit break characters

20.3.2.7 SCI Status Register 1 (SCISR1)

The SCISR1 and SCISR2 registers provides inputs to the MCU for generation of SCI interrupts. Also, these registers can be polled by the MCU to check the status of these bits. The flag-clearing procedures require that the status register be read followed by a read or write to the SCI data register.It is permissible to execute other instructions between the two steps as long as it does not compromise the handling of I/O, but the order of operations is important for flag clearing.

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W								
Reset	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-10. SCI Status Register 1 (SCISR1)

Read: Anytime

Write: Has no meaning or effect

Table 22-11. TSCR2 Field Descriptions

Field	Description
7 TOI	Timer Overflow Interrupt Enable 0 Interrupt inhibited. 1 Hardware interrupt requested when TOF flag set.
2:0 PR[2:0]	**Timer Prescaler Select** — These three bits select the frequency of the timer prescaler clock derived from the Bus Clock as shown in Table 22-12.

Table 22-12. Timer Clock Selection

PR2	PR1	PR0	Timer Clock
0	0	0	Bus Clock / 1
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

NOTE

The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

22.3.2.10 Main Timer Interrupt Flag 1 (TFLG1)

Module Base + 0x000E

	7	6	5	4	3	2	1	0
R W	RESERVED	RESERVED	C5F	C4F	C3F	C2F	C1F	C0F
Reset	0	0	0	0	0	0	0	0

Figure 22-16. Main Timer Interrupt Flag 1 (TFLG1)

Read: Anytime

Write: Used in the clearing mechanism (set bits cause corresponding bits to be cleared). Writing a zero will not affect current status of the bit.

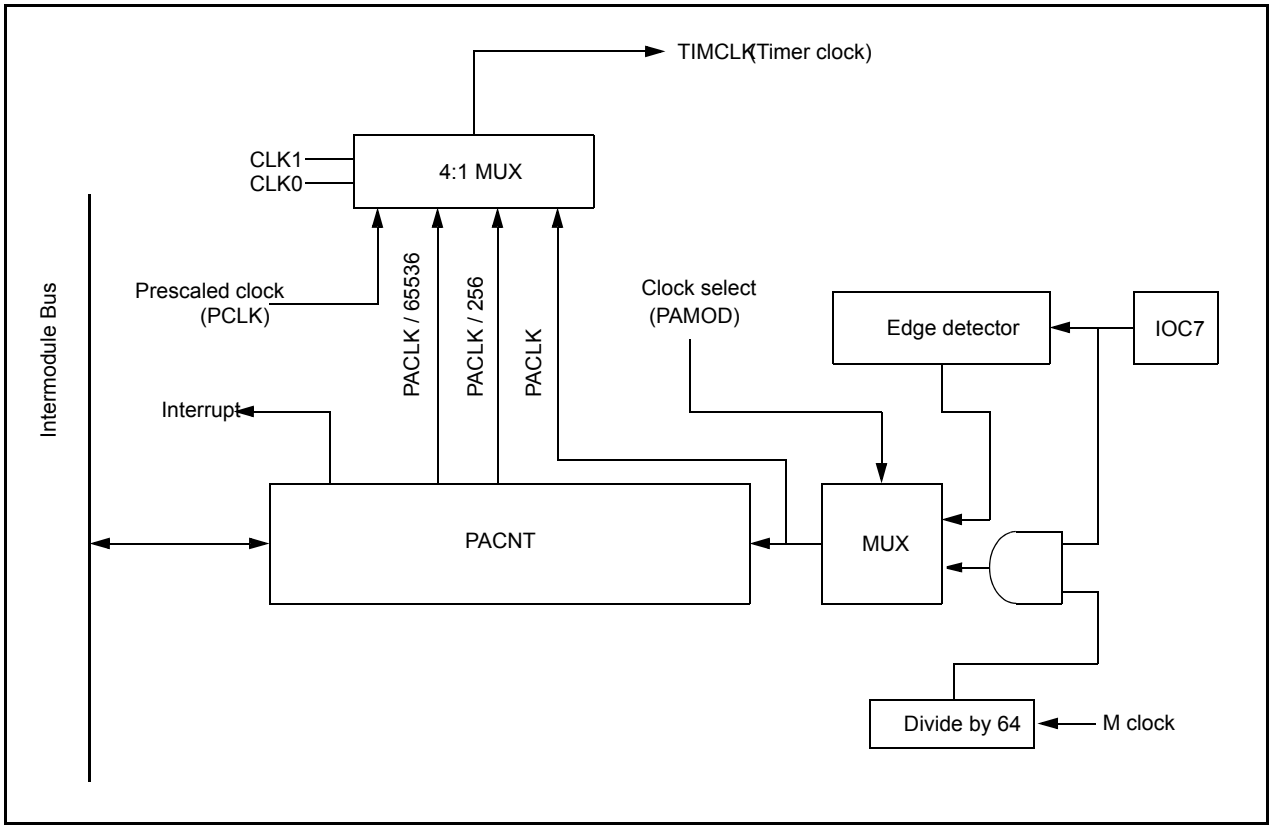


Figure 23-2. 16-Bit Pulse Accumulator Block Diagram

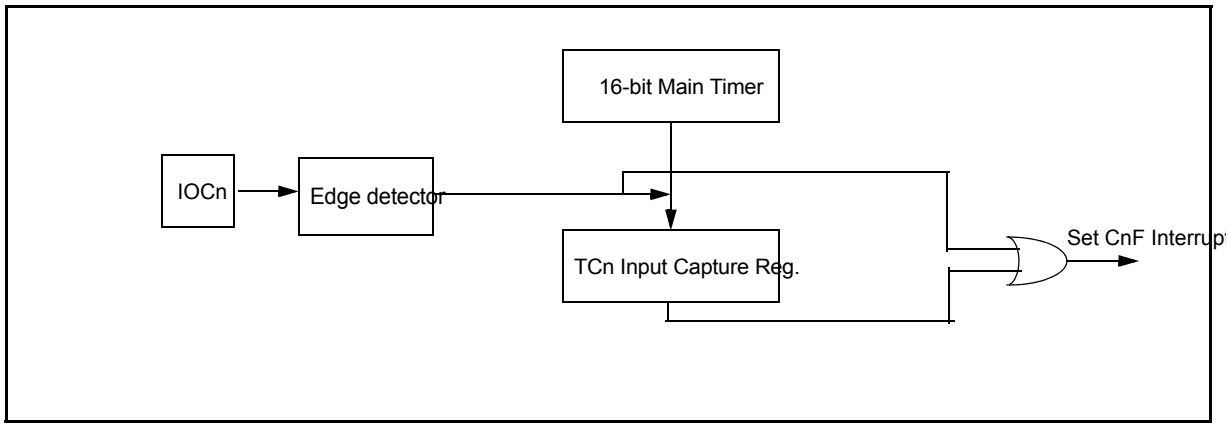


Figure 23-3. Interrupt Flag Setting

Any access to TCNT will clear TFLG2 register if the TFFCA bit in TSCR register is set.

Table 23-17. TRLG2 Field Descriptions

Field	Description
7 TOF	**Timer Overflow Flag** — Set when 16-bit free-running timer overflows from 0xFFFF to 0x0000. Clearing this bit requires writing a one to bit 7 of TFLG2 register while the TEN bit of TSCR1 or PAEN bit of PACTL is set to one (See also TCRE control bit explanation) .

23.3.2.14 Timer Input Capture/Output Compare Registers High and Low 0–7(TCxH and TCxL)

Module Base + 0x0010 = TC0H 0x0018=TC4H
0x0012 = TC1H 0x001A=TC5H
0x0014=TC2H 0x001C=TC6H
0x0016=TC3H 0x001E=TC7H

	15	14	13	12	11	10	9	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 23-22. Timer Input Capture/Output Compare Register x High (TCxH)

Module Base + 0x0011 = TC0L 0x0019 =TC4L
0x0013 = TC1L 0x001B=TC5L
0x0015 =TC2L 0x001D=TC6L
0x0017=TC3L 0x001F=TC7L

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

Figure 23-23. Timer Input Capture/Output Compare Register x Low (TCxL)

[1] This register is available only when the corresponding channel exists and is reserved if that channel does not exist. Writes to a reserved register have no functional effect. Reads from a reserved register return zeroes.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read: Anytime

Write: Anytime for output compare function.Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to 0x0000.

NOTE

Read/Write access in byte mode for high byte should take place before low byte otherwise it will give a different result.

Table 24-26. P-Flash Commands

FCMD	Command	Function on P-Flash Memory
0x02	Erase Verify Block	Verify that a P-Flash block is erased.
0x03	Erase Verify P-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x04	Read Once	Read a dedicated 64 byte field in the nonvolatile information register in P-Flash block that was previously programmed using the Program Once command.
0x06	Program P-Flash	Program a phrase in a P-Flash block.
0x07	Program Once	Program a dedicated 64 byte field in the nonvolatile information register in P-Flash block that is allowed to be programmed only once.
0x08	Erase All Blocks	Erase all P-Flash (and EEPROM) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the EEPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a P-Flash (or EEPROM) block. An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command.
0x0A	Erase P-Flash Sector	Erase all bytes in a P-Flash sector.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all P-Flash (and EEPROM) blocks and verifying that all P-Flash (and EEPROM) blocks are erased.
0x0C	Verify Backdoor Access Key	Supports a method of releasing MCU security by verifying a set of security keys.
0x0D	Set User Margin Level	Specifies a user margin read level for all P-Flash blocks.
0x0E	Set Field Margin Level	Specifies a field margin read level for all P-Flash blocks (special modes only).

24.4.4.5 EEPROM Commands

Table 24-27 summarizes the valid EEPROM commands along with the effects of the commands on the EEPROM block.

Table 24-27. EEPROM Commands

FCMD	Command	Function on EEPROM Memory
0x01	Erase Verify All Blocks	Verify that all EEPROM (and P-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that the EEPROM block is erased.

24.4.7 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed or when a Flash command operation has detected an ECC fault.

Table 24-64. Flash Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Command Complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit
ECC Double Bit Fault on Flash Read	DFDIF (FERSTAT register)	DFDIE (FERCNFG register)	I Bit
ECC Single Bit Fault on Flash Read	SFDIF (FERSTAT register)	SFDIE (FERCNFG register)	I Bit

NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

24.4.7.1 Description of Flash Interrupt Operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the DFDIF and SFDIF flags in combination with the DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Section 24.3.2.5, “Flash Configuration Register (FCNFG)”, Section 24.3.2.6, “Flash Error Configuration Register (FERCNFG)”, Section 24.3.2.7, “Flash Status Register (FSTAT)”, and Section 24.3.2.8, “Flash Error Status Register (FERSTAT)”.

The logic used for generating the Flash module interrupts is shown in Figure 24-26.

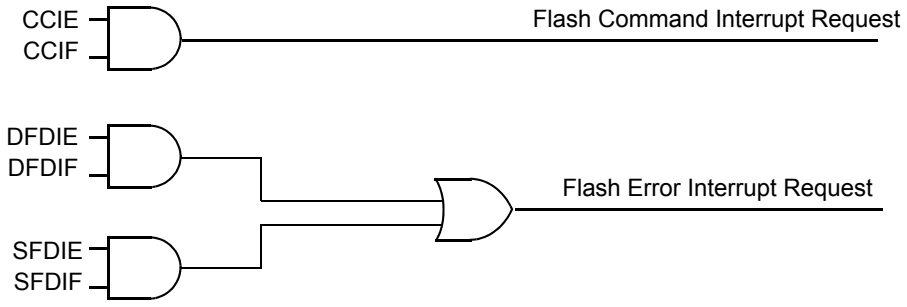


Figure 24-26. Flash Module Interrupts Implementation

Table 25-46. Erase Flash Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x09	Global address [17:16] to identify Flash block
001	Global address [15:0] in Flash block to be erased	

Upon clearing CCIF to launch the Erase Flash Block command, the Memory Controller will erase the selected Flash block and verify that it is erased. The CCIF flag will set after the Erase Flash Block operation has completed.

Table 25-47. Erase Flash Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 25-27)
		Set if an invalid global address [17:16] is supplied
		Set if the supplied P-Flash address is not phrase-aligned or if the EEPROM address is not word-aligned
	FPVIOL	Set if an area of the selected Flash block is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

25.4.6.9 Erase P-Flash Sector Command

The Erase P-Flash Sector operation will erase all addresses in a P-Flash sector.

Table 25-48. Erase P-Flash Sector Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0A	Global address [17:16] to identify P-Flash block to be erased
001	Global address [15:0] anywhere within the sector to be erased. Refer to Section 25.1.2.1 for the P-Flash sector size.	

Upon clearing CCIF to launch the Erase P-Flash Sector command, the Memory Controller will erase the selected Flash sector and then verify that it is erased. The CCIF flag will be set after the Erase P-Flash Sector operation has completed.

Each command timing is given by:

$$t_{command} = \left(f_{NVMOP(cycle)} \cdot \frac{1}{f_{NVMOP}} + f_{NVMBUS(cycle)} \cdot \frac{1}{f_{NVMBUS}} \right)$$

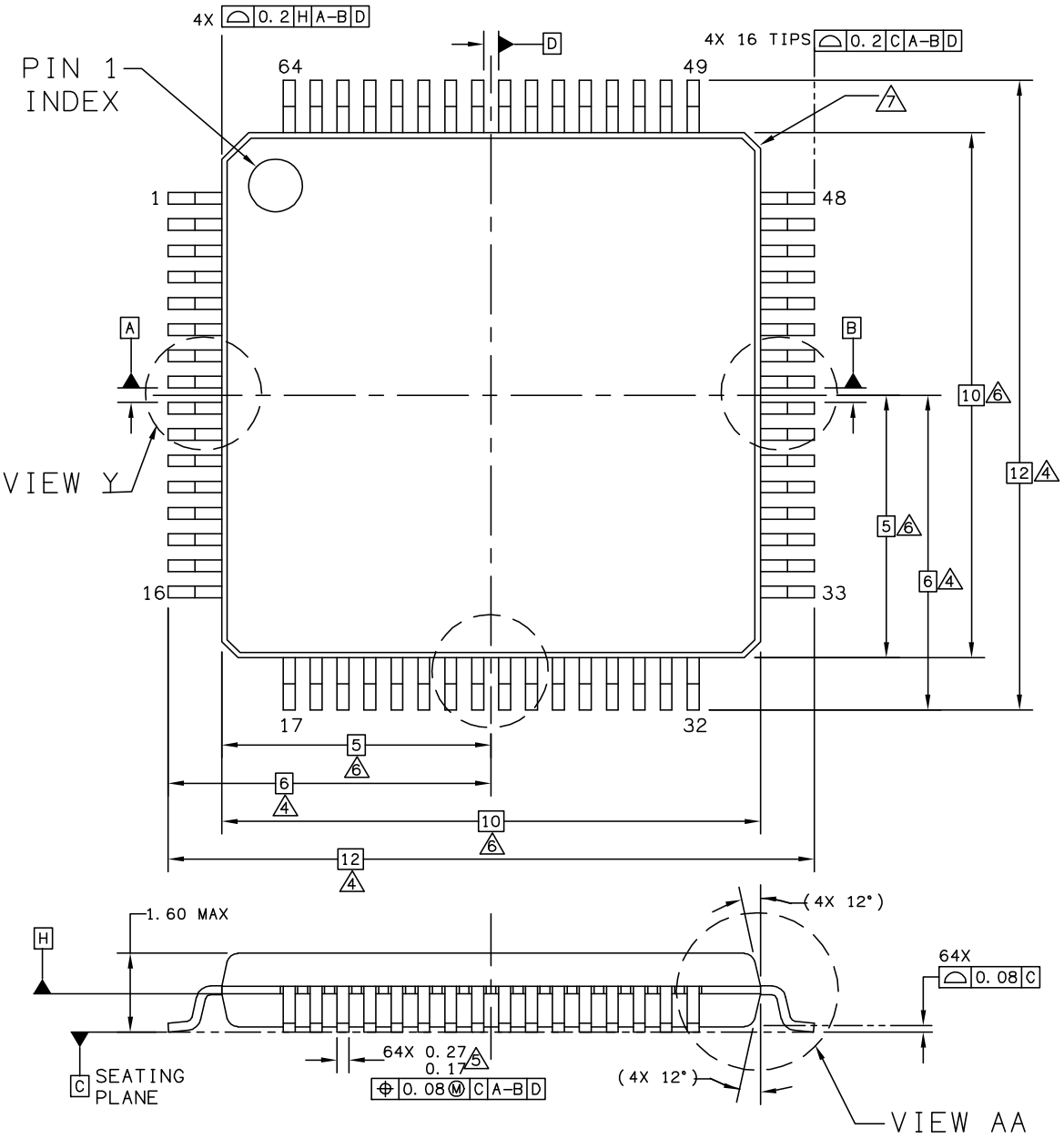
The timing parameters are captured exclusively during command execution (CCIF=0), excluding any time spent on the command write sequence to load and start the command. The formula above and the number of cycles in the following tables apply for the cases where the commands executed successfully in a new device, reflected in the minimum and typical timing parameters; however, due to aging, some of the commands will adjust their execution according to different margin settings and may eventually take longer to run than what the formula may return. The Max and Lfmax timing columns in the tables below already reflect this adjustment where applicable.

A summary of key timing parameters can be found from Table A-34 to Table A-38.

Table A-34. NVM Clock Timing Characteristics

Num	Rating	Symbol	Min	Typ	Max	Unit
1	Bus frequency	f_{NVMBUS}	1	25	25	MHz
2	Operating frequency	f_{NVMOP}	0.8	1.0	1.05	MHz

D.2 64 LQFP Mechanical Dimensions



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 64LD LQFP, 10 X 10 X 1.4 PKG, 0.5 PITCH, CASE OUTLINE	DOCUMENT NO: 98ASS23234W	REV: E
	CASE NUMBER: 840F-02	11 AUG 2006
	STANDARD: JEDEC MS-026 BCD	