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	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	91
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xdt256mal

Chapter 5Analog-to-Digital Converter (S12ATD10B8CV3)

Chapter 6
XGATE (S12XGATEV2)

1.2.4.2 V_{DDR1}, V_{DDR2}, V_{SSR1}, V_{SSR2} — Power and Ground Pins for I/O Drivers and for Internal Voltage Regulator

External power and ground for I/O drivers and input to the internal voltage regulator. Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high-frequency characteristics and place them as close to the MCU as possible. Bypass requirements depend on how heavily the MCU pins are loaded.

1.2.4.3 V_{DD1}, V_{DD2}, V_{SS1}, V_{SS2} — Core Power Pins

Power is supplied to the MCU through V_{DD} and V_{SS}. Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high-frequency characteristics and place them as close to the MCU as possible. This 2.5-V supply is derived from the internal voltage regulator. There is no static load on those pins allowed. The internal voltage regulator is turned off, if V_{REGEN} is tied to ground.

NOTE

No load allowed except for bypass capacitors.

1.2.4.4 V_{DDA}, V_{SSA} — Power Supply Pins for ATD and V_{REG}

V_{DDA}, V_{SSA} are the power supply and ground input pins for the voltage regulator and the analog-to-digital converters.

1.2.4.5 V_{RH}, V_{RL} — ATD Reference Voltage Input Pins

V_{RH} and V_{RL} are the reference voltage input pins for the analog-to-digital converter.

1.2.4.6 V_{DDPLL}, V_{SSPLL} — Power Supply Pins for PLL

Provides operating voltage and ground for the oscillator and the phased-locked loop. This allows the supply voltage to the oscillator and PLL to be bypassed independently. This 2.5-V voltage is generated by the internal voltage regulator.

NOTE

No load allowed except for bypass capacitors.

EXTAL input frequency. In full stop mode (PSTP = 0), the EXTAL pin is pulled down by an internal resistor of typical 200 kΩ.

NOTE

Freescale recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier.

Loop controlled circuit is not suited for overtone resonators and crystals.

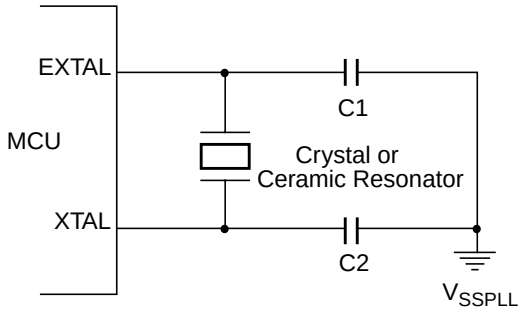
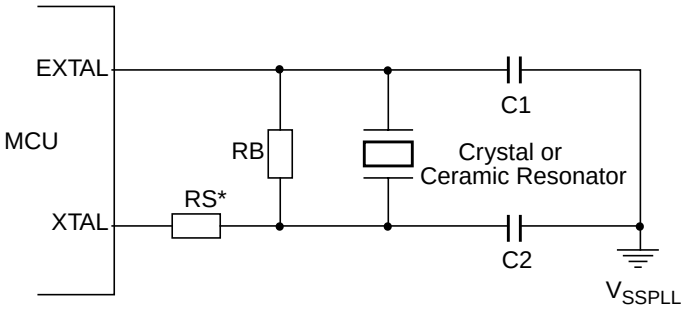


Figure 3-2. Loop Controlled Pierce Oscillator Connections ($\overline{XCLKS}$ = 1)

NOTE

Full swing Pierce circuit is not suited for overtone resonators and crystals without a careful component selection.



* R_S can be zero (shorted) when use with higher frequency crystals. Refer to manufacturer's data.

Figure 3-3. Full Swing Pierce Oscillator Connections ($\overline{XCLKS}$ = 0)

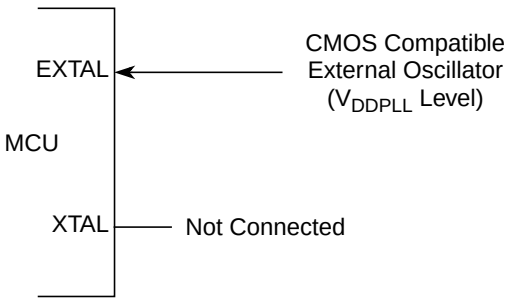


Figure 3-4. External Clock Connections ($\overline{XCLKS}$ = 0)

5.3.2.6 ATD Control Register 5 (ATDCTL5)

This register selects the type of conversion sequence and the analog input channels sampled. Writes to this register will abort current conversion sequence and start a new conversion sequence.

	7	6	5	4	3	2	1	0
R	DJM	DSGN	SCAN	MULT	0	CC	CB	CA
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-8. ATD Control Register 5 (ATDCTL5)

Read: Anytime

Write: Anytime

Table 5-13. ATDCTL5 Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — This bit controls justification of conversion data in the result registers. See Section 5.3.2.13, “ATD Conversion Result Registers (ATDDRx),” for details. 0 Left justified data in the result registers 1 Right justified data in the result registers
6 DSGN	**Result Register Data Signed or Unsigned Representation** — This bit selects between signed and unsigned conversion data representation in the result registers. Signed data is represented as 2’s complement. Signed data is not available in right justification. See Section 5.3.2.13, “ATD Conversion Result Registers (ATDDRx),” for details. 0 Unsigned data representation in the result registers 1 Signed data representation in the result registers Table 5-14 summarizes the result data formats available and how they are set up using the control bits. Table 5-15 illustrates the difference between the signed and unsigned, left justified output codes for an input signal range between 0 and 5.12 Volts.
5 SCAN	**Continuous Conversion Sequence Mode** — This bit selects whether conversion sequences are performed continuously or only once. 0 Single conversion sequence 1 Continuous conversion sequences (scan mode)
4 MULT	**Multi-Channel Sample Mode** — When MULT is 0, the ATD sequence controller samples only from the specified analog input channel for an entire conversion sequence. The analog channel is selected by channel selection code (control bits CC/CB/CA located in ATDCTL5). When MULT is 1, the ATD sequence controller samples across channels. The number of channels sampled is determined by the sequence length value (S8C, S4C, S2C, S1C). The first analog channel examined is determined by channel selection code (CC, CB, CA control bits); subsequent channels sampled in the sequence are determined by incrementing the channel selection code. 0 Sample only one channel 1 Sample across several channels
2–0 CC, CB, CA	**Analog Input Channel Select Code** — These bits select the analog input channel(s) whose signals are sampled and converted to digital codes. Table 5-16 lists the coding used to select the various analog input channels. In the case of single channel scans (MULT = 0), this selection code specified the channel examined. In the case of multi-channel scans (MULT = 1), this selection code represents the first channel to be examined in the conversion sequence. Subsequent channels are determined by incrementing channel selection code; selection codes that reach the maximum value wrap around to the minimum value.

RTS — Return to Scheduler — RTS

Operation

Terminates the current thread of program execution and remains idle until a new thread is started by the hardware scheduler.

CCR Effects

N	Z	V	C
—	—	—	—

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code															Cycles
RTS	INH	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	PA

If the corresponding NOVWx bit of the ICOVW register is set, the capture register or its holding register cannot be written by an event unless they are empty (see Section 7.4.1.1, “IC Channels”). This will prevent the captured value from being overwritten until it is read or latched in the holding register.

2. IC Queue Mode (LATQ = 0)

   The main timer value is memorized in the IC register by a valid input pin transition (see Figure 7-67 and Figure 7-68).

   If the corresponding NOVWx bit of the ICOVW register is cleared, with a new occurrence of a capture, the value of the IC register will be transferred to its holding register and the IC register memorizes the new timer value.

   If the corresponding NOVWx bit of the ICOVW register is set, the capture register or its holding register cannot be written by an event unless they are empty (see Section 7.4.1.1, “IC Channels”).

   In queue mode, reads of the holding register will latch the corresponding pulse accumulator value to its holding register.

7.4.1.1.3 Delayed IC Channels

There are four delay counters in this module associated with IC channels 0–3. The use of this feature is explained in the diagram and notes below.

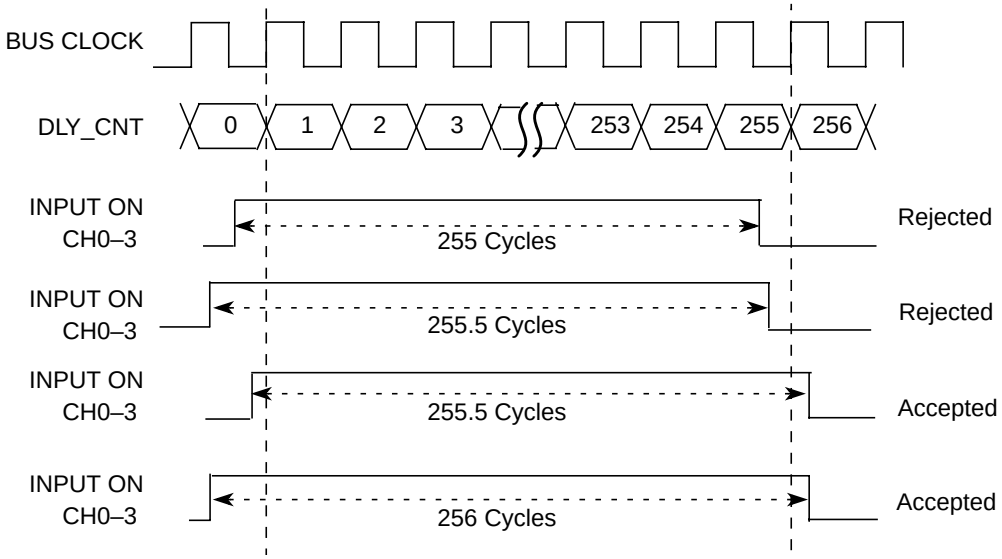


Figure 7-72. Channel Input Validity with Delay Counter Feature

In Figure 7-72 a delay counter value of 256 bus cycles is considered.

1. Input pulses with a duration of (DLY_CNT – 1) cycles or shorter are rejected.
2. Input pulses with a duration between (DLY_CNT – 1) and DLY_CNT cycles may be rejected or accepted, depending on their relative alignment with the sample points.
3. Input pulses with a duration between (DLY_CNT – 1) and DLY_CNT cycles may be rejected or accepted, depending on their relative alignment with the sample points.
4. Input pulses with a duration of DLY_CNT or longer are accepted.

9.1.2 Modes of Operation

The IIC functions the same in normal, special, and emulation modes. It has two low power modes: wait and stop modes.

9.1.3 Block Diagram

The block diagram of the IIC module is shown in Figure 9-1.

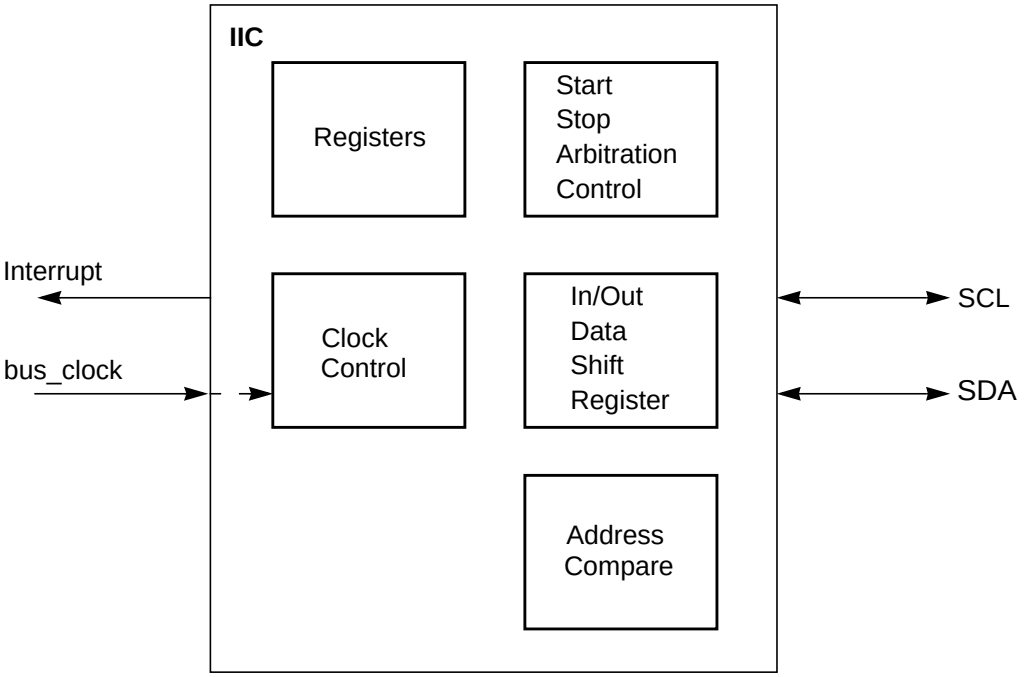


Figure 9-1. IIC Block Diagram

Differently from the normal bit transfer (where the host initiates the transmission), the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge in the BKGD pin. The hardware handshake protocol in Figure 15-11 specifies the timing when the BKGD pin is being driven, so the host should follow this timing constraint in order to avoid the risk of an electrical conflict in the BKGD pin.

NOTE

The only place the BKGD pin can have an electrical conflict is when one side is driving low and the other side is issuing a speedup pulse (high). Other "highs" are pulled rather than driven. However, at low rates the time of the speedup pulse can become lengthy and so the potential conflict time becomes longer as well.

The ACK handshake protocol does not support nested ACK pulses. If a BDM command is not acknowledge by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDM command. When the CPU enters wait or stop while the host issues a hardware command (e.g., WRITE_BYTE), the target discards the incoming command due to the wait or stop being detected. Therefore, the command is not acknowledged by the target, which means that the ACK pulse will not be issued in this case. After a certain time the host (not aware of stop or wait) should decide to abort any possible pending ACK pulse in order to be sure a new command can be issued. Therefore, the protocol provides a mechanism in which a command, and its corresponding ACK, can be aborted.

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 15.4.8, "Hardware Handshake Abort Procedure".

15.4.8 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 15.4.9, "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 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 if the serial interface is running on a different clock rate than the bus. 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

Address: 0x012C

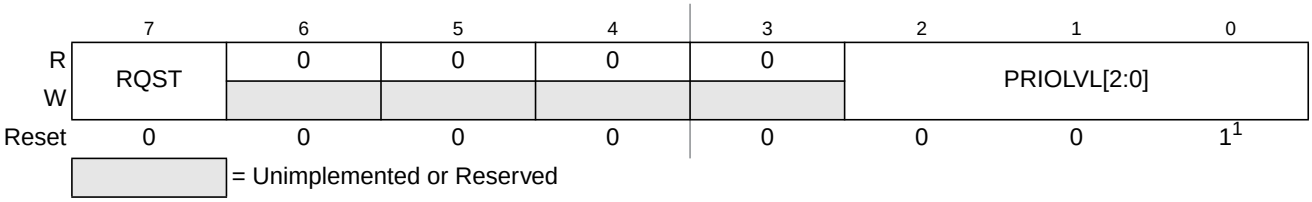


Figure 16-10. Interrupt Request Configuration Data Register 4 (INT_CFDATA4)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012D

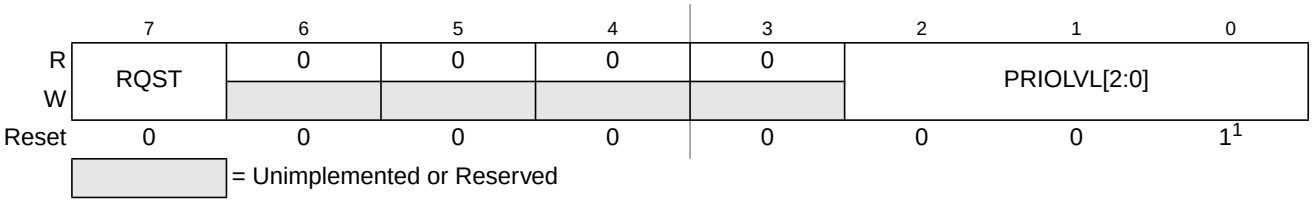


Figure 16-11. Interrupt Request Configuration Data Register 5 (INT_CFDATA5)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012E

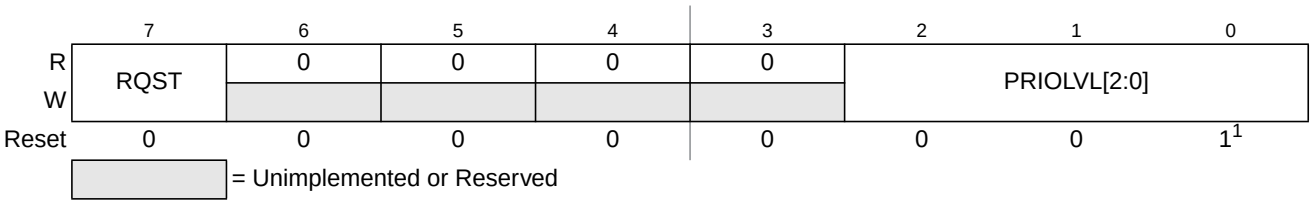


Figure 16-12. Interrupt Request Configuration Data Register 6 (INT_CFDATA6)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012F

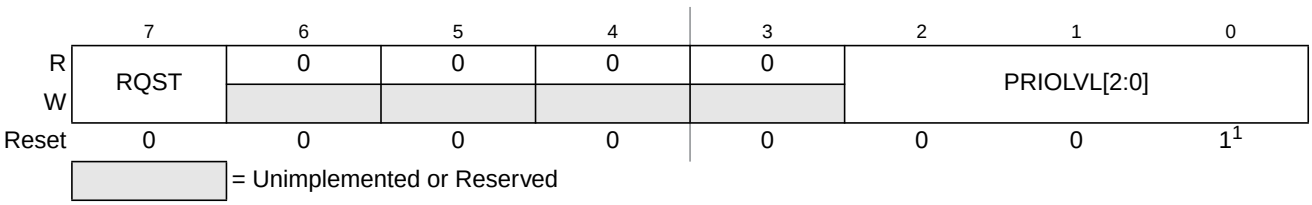


Figure 16-13. Interrupt Request Configuration Data Register 7 (INT_CFDATA7)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Read: Anytime

Write: Anytime

17.3.2.12 RAM Shared Region Upper Boundary Register (RAMSHU)

Address: 0x011F

	7	6	5	4	3	2	1	0
R	1	SHU6	SHU5	SHU4	SHU3	SHU2	SHU1	SHU0
W								
Reset	1	1	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 17-20. RAM Shared Region Upper Boundary Register (RAMSHU)

Read: Anytime

Write: Anytime when RWPE = 0

Table 17-17. RAMSHU Field Descriptions

Field	Description
6–0 SHU[6:0]	**RAM Shared Region Upper Boundary Bits 6–0** — These bits define the upper boundary of the shared memory in multiples of 256 bytes. The block selected by this register is included in the region. See Figure 1-25 for details.

17.4 Functional Description

The MMC block performs several basic functions of the S12X sub-system operation: MCU operation modes, priority control, address mapping, select signal generation and access limitations for the system. Each aspect is described in the following subsections.

17.4.1 MCU Operating Mode

- Normal single-chip mode

  There is no external bus in this mode. The MCU program is executed from the internal memory and no external accesses are allowed.
- Special single-chip mode

  This mode is generally used for debugging single-chip operation, boot-strapping or security related operations. The active background debug mode is in control of the CPU code execution and the BDM firmware is waiting for serial commands sent through the BKGD pin. There is no external bus in this mode.
- Emulation single-chip mode

  Tool vendors use this mode for emulation systems in which the user's target application is normal single-chip mode. Code is executed from external or internal memory depending on the set-up of the EROMON bit (see Section 1.3.2.5, "MMC Control Register (MMCCTL1)"). The external bus is active in both cases to allow observation of internal operations (internal visibility).

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x011D	RAMXGU	R	1	XGU6	XGU5	XGU4	XGU3	XGU2	XGU1	XGU0
		W								
0x011E	RAMSHL	R	1	SHL6	SHL5	SHL4	SHL3	SHL2	SHL1	SHL0
		W								
0x011F	RAMSHU	R	1	SHU6	SHU5	SHU4	SHU3	SHU2	SHU1	SHU0
		W								

= Unimplemented or Reserved

Figure 18-2. MMC Register Summary

18.3.2 Register Descriptions

18.3.2.1 MMC Control Register (MMCCTL0)

Address: 0x000A PRR

	7	6	5	4	3	2	1	0
R	0	0	0	0	CS3E	CS2E	CS1E	CS0E
W								
Reset	0	0	0	0	0	0	0	$\overline{\text{ROMON}}$[1]

1. ROMON is bit[0] of the register MMCTL1 (see Figure 18-10)

= Unimplemented or Reserved

Figure 18-3. MMC Control Register (MMCCTL0)

Read: Anytime. In emulation modes read operations will return the data from the external bus. In all other modes the data is read from this register.

Write: Anytime. In emulation modes write operations will also be directed to the external bus.

Table 18-4. Chip Selects Function Activity

Register Bit	Chip Modes					
	NS	SS	NX	ES	EX	ST
CS3E, CS2E, CS1E, CS0E	Disabled[1]	Disabled	Enabled[2]	Disabled	Enabled	Enabled

1 Disabled: feature always inactive.

2 Enabled: activity is controlled by the appropriate register bit value.

The MMCCTL0 register is used to control external bus functions, i.e., availability of chip selects.

CAUTION

XGATE write access to this register during an CPU access which makes use of this register could lead to unexpected results.

18.4.4.1 Master Bus Prioritization regarding Access Conflicts on Target Buses

The arbitration scheme allows only one master to be connected to a target at any given time. The following rules apply when prioritizing accesses from different masters to the same target bus:

- CPU always has priority over BDM and XGATE.
- XGATE access to PRU registers constitutes a special case. It is always granted and stalls the CPU for its duration.
- XGATE has priority over BDM.
- BDM has priority over CPU and XGATE when its access is stalled for more than 128 cycles. In the later case the suspect master will be stalled after finishing the current operation and the BDM will gain access to the bus.
- In emulation modes all internal accesses are visible on the external bus as well and the external bus is used during access to the PRU registers.

18.4.5 Interrupts

18.4.5.1 Outgoing Interrupt Requests

The following interrupt requests can be triggered by the MMC module:

CPU access violation: The CPU access violation signals to the CPU detection of an error condition in the CPU application code which is resulted in write access to the protected XGATE RAM area (see Section 18.4.3.2, “Illegal CPU Accesses).

18.5 Initialization/Application Information

18.5.1 CALL and RTC Instructions

CALL and RTC instructions are uninterruptable CPU instructions that automate page switching in the program page window. The CALL instruction is similar to the JSR instruction, but the subroutine that is called can be located anywhere in the local address space or in any Flash or ROM page visible through the program page window. The CALL instruction calculates and stacks a return address, stacks the current PPAGE value and writes a new instruction-supplied value to the PPAGE register. The PPAGE value controls which of the 256 possible pages is visible through the 16 Kbyte program page window in the 64 Kbyte local CPU memory map. Execution then begins at the address of the called subroutine.

During the execution of the CALL instruction, the CPU performs the following steps:

1. Writes the current PPAGE value into an internal temporary register and writes the new instruction-supplied PPAGE value into the PPAGE register
2. Calculates the address of the next instruction after the CALL instruction (the return address) and pushes this 16-bit value onto the stack
3. Pushes the temporarily stored PPAGE value onto the stack
4. Calculates the effective address of the subroutine, refills the queue and begins execution at the new address

Table 22-1. Pin Functions and Priorities (Sheet 5 of 7)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
P	PP7	PWM7	I/O	Pulse Width Modulator input/output channel 7	GPIO
		SCK2	I/O	Serial Peripheral Interface 2 serial clock pin	
		GPIO/KWP7	I/O	General-purpose I/O with interrupt	
	PP6	PWM6	O	Pulse Width Modulator output channel 6	
		$\overline{SS2}$	I/O	Serial Peripheral Interface 2 slave select output in master mode, input for slave mode or master mode.	
		GPIO/KWP6	I/O	General-purpose I/O with interrupt	
	PP5	PWM5	O	Pulse Width Modulator output channel 5	
		MOSI2	I/O	Serial Peripheral Interface 2 master out/slave in pin	
		GPIO/KWP5	I/O	General-purpose I/O with interrupt	
	PP4	PWM4	O	Pulse Width Modulator output channel 4	
		MISO2	I/O	Serial Peripheral Interface 2 master in/slave out pin	
		GPIO/KWP4	I/O	General-purpose I/O with interrupt	
	PP3	PWM3	O	Pulse Width Modulator output channel 3	
		$\overline{SS1}$	I/O	Serial Peripheral Interface 1 slave select output in master mode, input for slave mode or master mode.	
		GPIO/KWP3	I/O	General-purpose I/O with interrupt	
	PP2	PWM2	O	Pulse Width Modulator output channel 2	
		SCK1	I/O	Serial Peripheral Interface 1 serial clock pin	
		GPIO/KWP2	I/O	General-purpose I/O with interrupt	
	PP1	PWM1	O	Pulse Width Modulator output channel 1	
		MOSI1	I/O	Serial Peripheral Interface 1 master out/slave in pin	
		GPIO/KWP1	I/O	General-purpose I/O with interrupt	
	PP0	PWM0	O	Pulse Width Modulator output channel 0	
		MISO1	I/O	Serial Peripheral Interface 1 master in/slave out pin	
		GPIO/KWP0	I/O	General-purpose I/O with interrupt	

Table 22-1. Pin Functions and Priorities (Sheet 6 of 7)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
H	PH7	$\overline{SS2}$	I/O	Serial Peripheral Interface 2 slave select output in master mode, input for slave mode or master mode	GPIO
		TXD5	O	Serial Communication Interface 5 transmit pin	
		GPIO/KWH7	I/O	General-purpose I/O with interrupt	
	PH6	SCK2	I/O	Serial Peripheral Interface 2 serial clock pin	
		RXD5	I	Serial Communication Interface 5 receive pin	
		GPIO/KWH6	I/O	General-purpose I/O with interrupt	
	PH5	MOSI2	I/O	Serial Peripheral Interface 2 master out/slave in pin	
		TXD4	O	Serial Communication Interface 4 transmit pin	
		GPIO/KWH5	I/O	General-purpose I/O with interrupt	
	PH4	MISO2	I/O	Serial Peripheral Interface 2 master in/slave out pin	
		RXD4	I	Serial Communication Interface 4 receive pin	
		GPIO/KWH4	I/O	General-purpose I/O with interrupt	
	PH3	$\overline{SS1}$	I/O	Serial Peripheral Interface 1 slave select output in master mode, input for slave mode or master mode.	
		GPIO/KWH3	I/O	General-purpose I/O with interrupt	
	PH2	SCK1	I/O	Serial Peripheral Interface 1 serial clock pin	
		GPIO/KWH2	I/O	General-purpose I/O with interrupt	
	PH1	MOSI1	I/O	Serial Peripheral Interface 1 master out/slave in pin	
		GPIO/KWH1	I/O	General-purpose I/O with interrupt	
	PH0	MISO1	I/O	Serial Peripheral Interface 1 master in/slave out pin	
		GPIO/KWH0	I/O	General-purpose I/O with interrupt	

23.0.5.69 Port AD1 Data Direction Register 1 (DDR1AD1)

	7	6	5	4	3	2	1	0
R W	DDR1AD115	DDR1AD114	DDR1AD113	DDR1AD112	DDR1AD111	DDR1AD110	DDR1AD19	DDR1AD18
Reset	0	0	0	0	0	0	0	0

Figure 23-71. Port AD1 Data Direction Register 1 (DDR1AD1)

Read: Anytime.

Write: Anytime.

This register configures pins PAD as either input or output.

Table 23-62. DDR1AD1 Field Descriptions

Field	Description
7–0 DDR1AD1[15:8]	**Data Direction Port AD1 Register 1** 0 Associated pin is configured as input. 1 Associated pin is configured as output. **Note:** Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTAD11 register, when changing the DDR1AD1 register. **Note:** To use the digital input function on port AD1 the ATD1 digital input enable register (ATD1DIEN1) has to be set to logic level “1”.

23.0.5.70 Port AD1 Reduced Drive Register 0 (RDR0AD1)

	7	6	5	4	3	2	1	0
R W	RDR0AD123	RDR0AD122	RDR0AD121	RDR0AD120	RDR0AD119	RDR0AD118	RDR0AD117	RDR0AD116
Reset	0	0	0	0	0	0	0	0

Figure 23-72. Port AD1 Reduced Drive Register 0 (RDR0AD1)

Read: Anytime.

Write: Anytime.

This register configures the drive strength of each PAD[23:16] output pin as either full or reduced. If the port is used as input this bit is ignored.

Table 23-63. RDR0AD1 Field Descriptions

Field	Description
7–0 RDR0AD1[23:16]	**Reduced Drive Port AD1 Register 0** 0 Full drive strength at output. 1 Associated pin drives at about 1/6 of the full drive strength.

28.3.2.7 Flash Command Register (FCMD)

The FCMD register is the Flash command register.

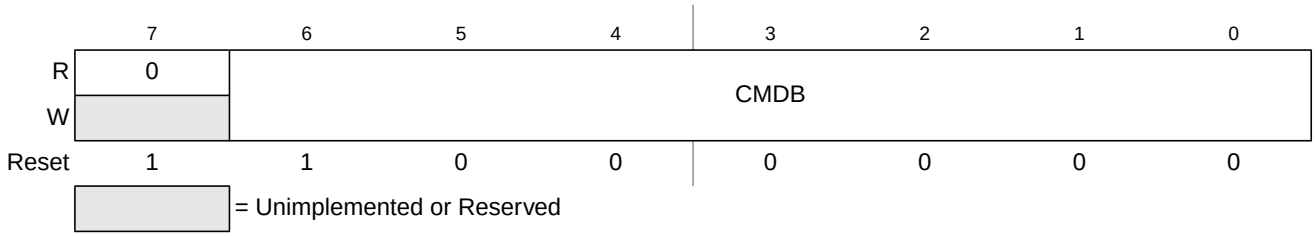


Figure 28-14. Flash Command Register (FCMD)

All CMDB bits are readable and writable during a command write sequence while bit 7 reads 0 and is not writable.

Table 28-15. FCMD Field Descriptions

Field	Description
6:0 CMDB[6:0]	**Flash Command** — Valid Flash commands are shown in Table 28-16. Writing any command other than those listed in Table 28-16 sets the ACCERR flag in the FSTAT register.

Table 28-16. Valid Flash Command List

CMDB[6:0]	NVM Command
0x05	Erase Verify
0x06	Data Compress
0x20	Word Program
0x40	Sector Erase
0x41	Mass Erase
0x47	Sector Erase Abort

28.3.2.8 Flash Control Register (FCTL)

The FCTL register is the Flash control register.

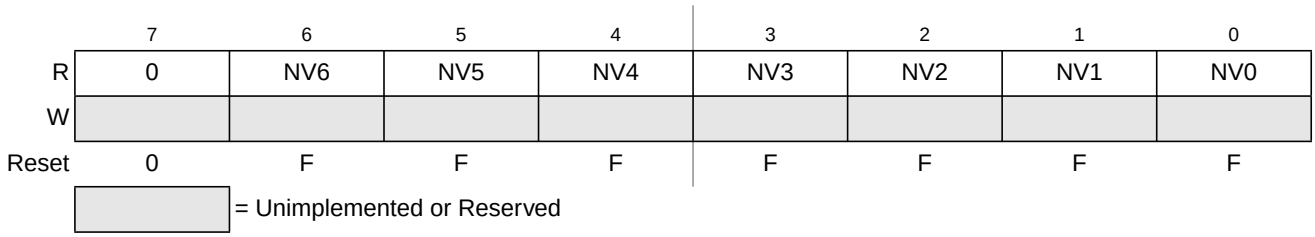


Figure 28-15. Flash Control Register (FCTL)

All bits in the FCTL register are readable but are not writable.

The FCTL NV bits are loaded from the Flash nonvolatile byte located at global address 0x7F_FF0E during the reset sequence, indicated by F in Figure 28-15.

28.4.2.2.1 Data Compress Operation

The Flash module contains a 16-bit multiple-input signature register (MISR) for each Flash block to generate a 16-bit signature based on selected Flash array data. If multiple Flash blocks are selected for simultaneous compression, then the signature from each Flash block is further compressed to generate a single 16-bit signature. The final 16-bit signature, found in the FDATA registers after the data compress operation has completed, is based on the following logic equation which is executed on every data compression cycle during the operation:

MISR[15:0] = {MISR[14:0], ^MISR[15,4,2,1]} ^ DATA[15:0] ***Eqn. 28-1***

where MISR is the content of the internal signature register associated with each Flash block and DATA is the data to be compressed as shown in Figure 28-27.

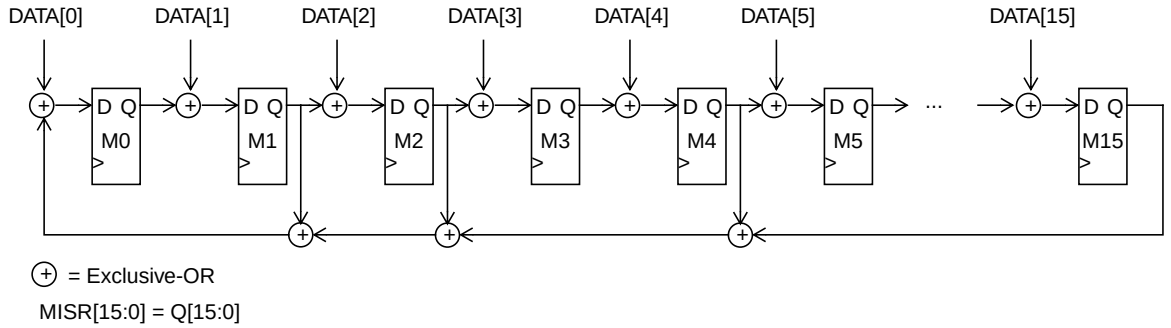


Figure 28-27. 16-Bit MISR Diagram

During the data compress operation, the following steps are executed:

1. MISR for each Flash block is reset to 0xFFFF.
2. Initialized DATA equal to 0xFFFF is compressed into the MISR for each selected Flash block which results in the MISR containing 0x0001.
3. DATA equal to the selected Flash array data range is read and compressed into the MISR for each selected Flash block with addresses incrementing.
4. DATA equal to the selected Flash array data range is read and compressed into the MISR for each selected Flash block with addresses decrementing.
5. If Flash block 0 is selected for compression, DATA equal to the contents of the MISR for Flash block 0 is compressed into the MISR for Flash block 0. If data in Flash block 0 was not selected for compression, the MISR for Flash block 0 contains 0xFFFF.
6. If Flash block 1 is selected for compression, DATA equal to the contents of the MISR for Flash block 1 is compressed into the MISR for Flash block 0.
7. If Flash block 2 is selected for compression, DATA equal to the contents of the MISR for Flash block 2 is compressed into the MISR for Flash block 0.
8. If Flash block 3 is selected for compression, DATA equal to the contents of the MISR for Flash block 3 is compressed into the MISR for Flash block 0.
9. If Flash block 4 is selected for compression, DATA equal to the contents of the MISR for Flash block 4 is compressed into the MISR for Flash block 0.

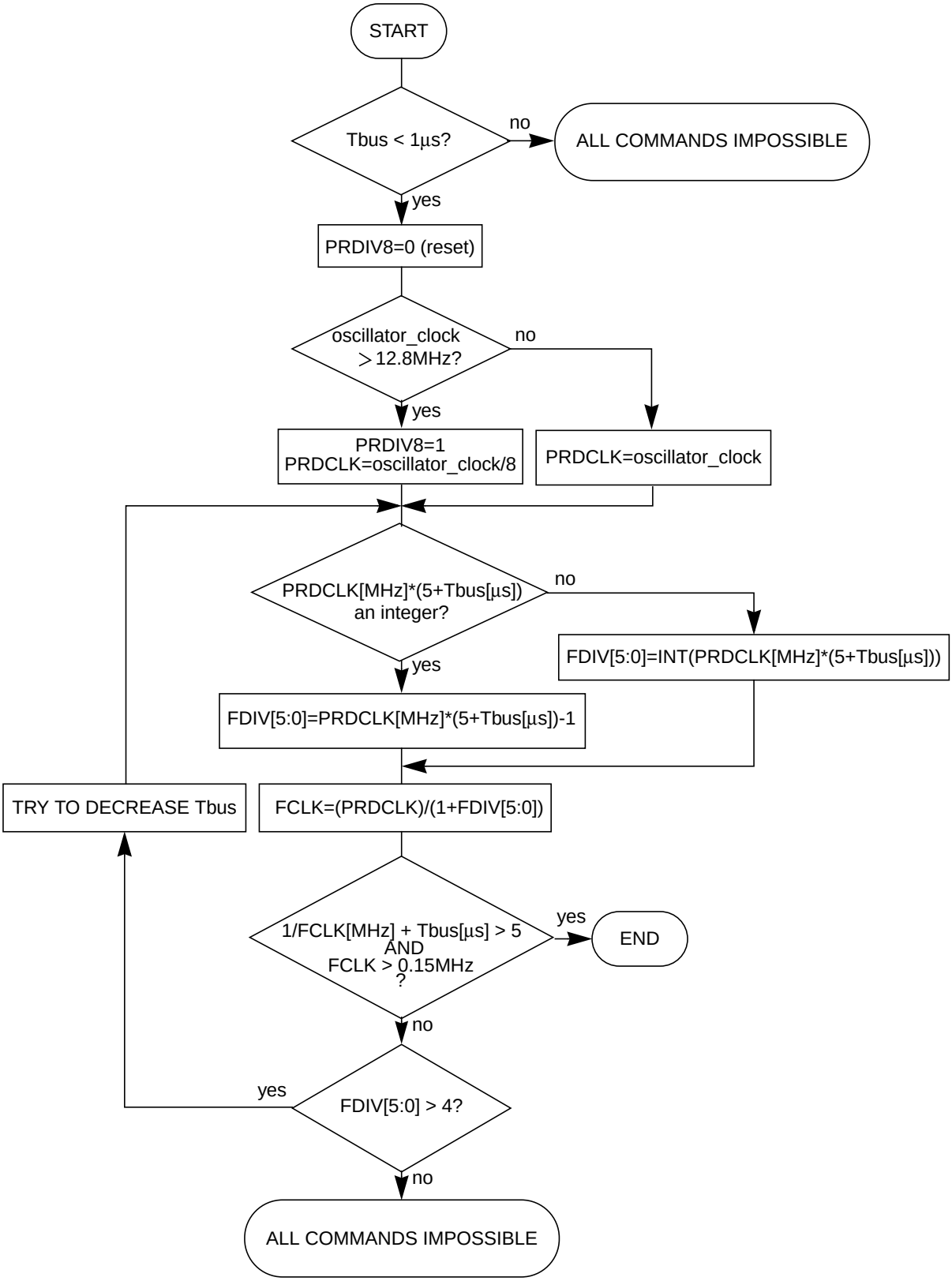


Figure 29-22. Determination Procedure for PRDIV8 and FDIV Bits

0x000E–0x000F External Bus Interface (S12XEBI) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000E	EBICTL0	R	ITHRS	0	HDBE	ASIZ4	ASIZ3	ASIZ2	ASIZ1	ASIZ0
		W								
0x000F	EBICTL1	R	EWAITE	0	0	0	0	EXSTR2	EXSTR1	EXSTR0
		W								

0x0010–0x0017 Module Mapping Control (S12XMMC) Map 2 of 4

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0010	GPAGE	R	0	GP6	GP5	GP4	GP3	GP2	GP1	GP0
		W								
0x0011	DIRECT	R	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
		W								
0x0012	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0013	MMCCTL1	R	0	0	0	0	0	EROMON	ROMHM	ROMON
		W								
0x0014	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0015	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0016	RPAGE	R	RP7	RP6	RP5	RP4	RP3	RP2	RP1	RP0
		W								
0x0017	EPAGE	R	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
		W								

0x0018–0x001B Miscellaneous Peripheral

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0018	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0019	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x001A	PARTIDH	R	1	1	0	0	0	1	0	0
		W								
0x001B	PARTIDL	R	0	0	0	0	0	0	0	0
		W								

0x001C–0x001F Port Integration Module (PIM) Map 3 of 5

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x001C	ECLKCTL	R	NECLK	NCLKX2	0	0	0	0	EDIV1	EDIV0
		W								