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	119
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xdt512mag

1.2.2 Signal Properties Summary

Table 1-7 summarizes the pin functionality of the MC9S12XDP512. For available modules on other parts of the S12XD, S12XB and S12XA family please refer to Appendix E Derivative Differences.

Table 1-7. Signal Properties Summary (Sheet 1 of 4)

Pin Name Function 1	Pin Name Function 2	Pin Name Function 3	Pin Name Function 4	Pin Name Function 5	Power Supply	Internal Pull Resistor		Description
						CTRL	Reset State	
EXTAL	—	—	—	—	V_{DDPLL}	NA	NA	Oscillator pins
XTAL	—	—	—	—	V_{DDPLL}	NA	NA	
$\overline{\text{RESET}}$	—	—	—	—	V_{DDR}	PULLUP		External reset
TEST	—	—	—	—	N.A.	$\overline{\text{RESET}}$ pin	DOWN	Test input
VREGEN	—	—	—	—	V_{DDX}	PUCR	Up	Voltage regulator enable Input
XFC	—	—	—	—	V_{DDPLL}	NA	NA	PLL loop filter
BKGD	MODC	—	—	—	V_{DDX}	Always on	Up	Background debug
PAD[23:08]	AN[23:8]	—	—	—	V_{DDA}	PER0/ PER1	Disabled	Port AD I/O, Port AD inputs of ATD1 and analog inputs of ATD1
PAD[07:00]	AN[7:0]	—	—	—	V_{DDA}	PER1	Disabled	Port AD I/O, Port AD inputs of ATD0 and analog inputs of ATD0
PA[7:0]	—	—	—	—	V_{DDR}	PUCR	Disabled	Port A I/O
PB[7:0]	—	—	—	—	V_{DDR}	PUCR	Disabled	Port BI/O
PA[7:0]	ADDR[15:8]	IVD[15:8]	—	—	V_{DDR}	PUCR	Disabled	Port A I/O, address bus, internal visibility data
PB[7:1]	ADDR[7:1]	IVD[7:0]	—	—	V_{DDR}	PUCR	Disabled	Port B I/O, address bus, internal visibility data
PB0	ADDR0	$\overline{\text{UDS}}$			V_{DDR}	PUCR	Disabled	Port B I/O, address bus, upper data strobe
PC[7:0]	DATA[15:8]	—	—	—	V_{DDR}	PUCR	Disabled	Port C I/O, data bus
PD[7:0]	DATA[7:0]	—	—	—	V_{DDR}	PUCR	Disabled	Port D I/O, data bus
PE7	ECLKX2	$\overline{\text{XCLKS}}$	—	—	V_{DDR}	PUCR	Up	Port E I/O, system clock output, clock select
PE6	$\overline{\text{TAGHI}}$	MODB	—	—	V_{DDR}	While $\overline{\text{RESET}}$ pin is low: down		Port E I/O, tag high, mode input
PE5	$\overline{\text{RE}}$	MODA	$\overline{\text{TAGLO}}$	—	V_{DDR}	While $\overline{\text{RESET}}$ pin is low: down		Port E I/O, read enable, mode input, tag low input
PE4	ECLK	—	—	—	V_{DDR}	PUCR	Up	Port E I/O, bus clock output
PE3	$\overline{\text{LSTRB}}$	$\overline{\text{LDS}}$	EROMCTL	—	V_{DDR}	PUCR	Up	Port E I/O, low byte data strobe, EROMON control
PE2	R/$\overline{\text{W}}$	$\overline{\text{WE}}$	—	—	V_{DDR}	PUCR	Up	Port E I/O, read/write
PE1	$\overline{\text{IRQ}}$	—	—	—	V_{DDR}	PUCR	Up	Port E Input, maskable interrupt

Table 4-3. Multi-Channel Wrap Around Coding

WRAP3	WRAP2	WRAP1	WRAP0	Multiple Channel Conversions (MULT = 1) Wrap Around to AN0 after Converting
0	0	0	0	Reserved
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	AN10
1	0	1	1	AN11
1	1	0	0	AN12
1	1	0	1	AN13
1	1	1	0	AN14
1	1	1	1	AN15

4.3.2.2 ATD Control Register 1 (ATDCTL1)

Writes to this register will abort current conversion sequence but will not start a new sequence.

	7	6	5	4	3	2	1	0
R	ETRIGSEL	0	0	0	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0
W								
Reset	0	0	0	0	1	1	1	1

= Unimplemented or Reserved

Figure 4-4. ATD Control Register 1 (ATDCTL1)

Read: Anytime

Write: Anytime

Table 4-4. ATDCTL1 Field Descriptions

Field	Description
7 ETRIGSEL	**External Trigger Source Select** — This bit selects the external trigger source to be either one of the AD channels or one of the ETRIG[3:0] inputs. See device specification for availability and connectivity of ETRIG[3:0] inputs. If ETRIG[3:0] input option is not available, writing a 1 to ETRISEL only sets the bit but has no effect, that means one of the AD channels (selected by ETRIGCH[3:0]) remains the source for external trigger. The coding is summarized in Table 4-5.
3:0 ETRIGCH[3:0]	**External Trigger Channel Select** — These bits select one of the AD channels or one of the ETRIG[3:0] inputs as source for the external trigger. The coding is summarized in Table 4-5.

4.4.1.2 Analog Input Multiplexer

The analog input multiplexer connects one of the 16 external analog input channels to the sample and hold machine.

4.4.1.3 Sample Buffer Amplifier

The sample amplifier is used to buffer the input analog signal so that the storage node can be quickly charged to the sample potential.

4.4.1.4 Analog-to-Digital (A/D) Machine

The A/D machine performs analog to digital conversions. The resolution is program selectable at either 8 or 10 bits. The A/D machine uses a successive approximation architecture. It functions by comparing the stored analog sample potential with a series of digitally generated analog potentials. By following a binary search algorithm, the A/D machine locates the approximating potential that is nearest to the sampled potential.

When not converting the A/D machine disables its own clocks. The analog electronics continue drawing quiescent current. The power down (ADPU) bit must be set to disable both the digital clocks and the analog power consumption.

Only analog input signals within the potential range of V_{RL} to V_{RH} (A/D reference potentials) will result in a non-railed digital output codes.

4.4.2 Digital Sub-Block

This subsection explains some of the digital features in more detail. See register descriptions for all details.

4.4.2.1 External Trigger Input

The external trigger feature allows the user to synchronize ATD conversions to the external environment events rather than relying on software to signal the ATD module when ATD conversions are to take place. The external trigger signal (out of reset ATD channel 15, configurable in ATDCTL1) is programmable to be edge or level sensitive with polarity control. Table 4-27 gives a brief description of the different combinations of control bits and their effect on the external trigger function.

Table 4-27. External Trigger Control Bits

ETRIGLE	ETRIGP	ETRIGE	SCAN	Description
X	X	0	0	Ignores external trigger. Performs one conversion sequence and stops.
X	X	0	1	Ignores external trigger. Performs continuous conversion sequences.
0	0	1	X	Falling edge triggered. Performs one conversion sequence per trigger.
0	1	1	X	Rising edge triggered. Performs one conversion sequence per trigger.
1	0	1	X	Trigger active low. Performs continuous conversions while trigger is active.
1	1	1	X	Trigger active high. Performs continuous conversions while trigger is active.

During a conversion, if additional active edges are detected the overrun error flag ETORF is set.

7.2 External Signal Description

The ECT module has a total of eight external pins.

7.2.1 IOC7 — Input Capture and Output Compare Channel 7

This pin serves as input capture or output compare for channel 7.

7.2.2 IOC6 — Input Capture and Output Compare Channel 6

This pin serves as input capture or output compare for channel 6.

7.2.3 IOC5 — Input Capture and Output Compare Channel 5

This pin serves as input capture or output compare for channel 5.

7.2.4 IOC4 — Input Capture and Output Compare Channel 4

This pin serves as input capture or output compare for channel 4.

7.2.5 IOC3 — Input Capture and Output Compare Channel 3

This pin serves as input capture or output compare for channel 3.

7.2.6 IOC2 — Input Capture and Output Compare Channel 2

This pin serves as input capture or output compare for channel 2.

7.2.7 IOC1 — Input Capture and Output Compare Channel 1

This pin serves as input capture or output compare for channel 1.

7.2.8 IOC0 — Input Capture and Output Compare Channel 0

This pin serves as input capture or output compare for channel 0.

NOTE

For the description of interrupts see Section 7.4.3, “Interrupts”.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PWMCNT0	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT1	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT2	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT3	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT4	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT5	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT6	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT7	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMPER0	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER1	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER2	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER3	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER4	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER5	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER6	R W	Bit 7	6	5	4	3	2	1	Bit 0

= Unimplemented or Reserved

Figure 8-2. PWM Register Summary (Sheet 2 of 3)

15.2 External Signal Description

A single-wire interface pin called the background debug interface (BKGD) pin is used to communicate with the BDM system. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the background debug mode.

15.3 Memory Map and Register Definition

15.3.1 Module Memory Map

Table 15-1 shows the BDM memory map when BDM is active.

Table 15-1. BDM Memory Map

Global Address	Module	Size (Bytes)
0x7FFF00–0x7FFF0B	BDM registers	12
0x7FFF0C–0x7FFF0E	BDM firmware ROM	3
0x7FFF0F	Family ID (part of BDM firmware ROM)	1
0x7FFF10–0x7FFFFF	BDM firmware ROM	240

15.4 Functional Description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands: hardware and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see Section 15.4.3, “BDM Hardware Commands”. Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see Section 15.4.4, “Standard BDM Firmware Commands”. The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted (see Section 15.4.3, “BDM Hardware Commands”) and in secure mode (see Section 15.4.1, “Security”). Firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

15.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip EEPROM and Flash EEPROM are erased. This being the case, the UNSEC and ENBDM bit will get set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the EEPROM or Flash do not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the EEPROM and Flash.

BDM operation is not possible in any other mode than special single chip mode when the device is secured. The device can only be unsecured via BDM serial interface in special single chip mode. For more information regarding security, please see the S12X_9SEC Block Guide.

15.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

compared to the serial communication rate. 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.

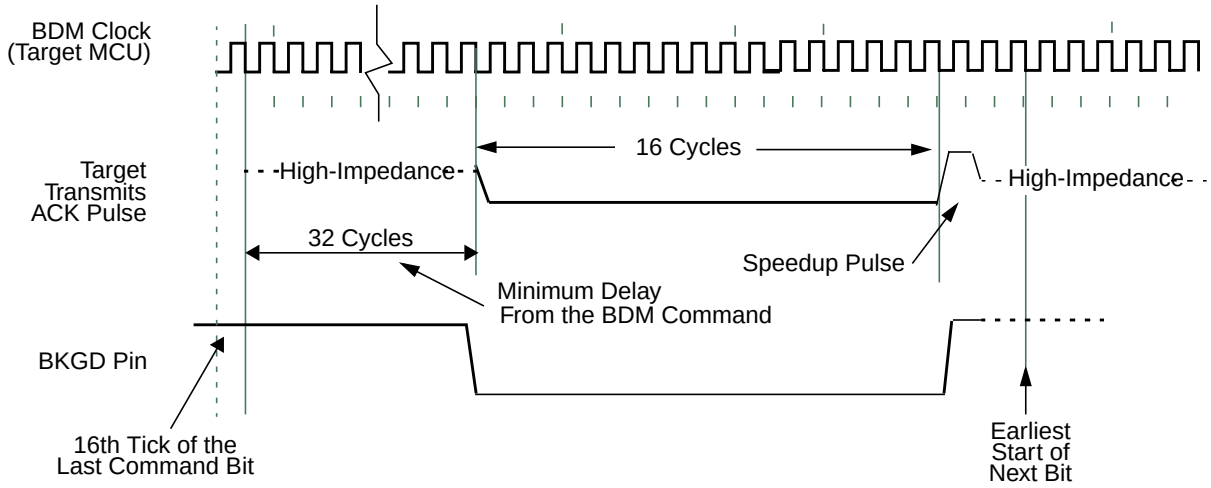

Figure 15-11. Target Acknowledge Pulse (ACK)

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 will not be issued meaning that the BDM command was not executed. After entering wait or stop mode, the BDM command is no longer pending.

Figure 15-12 shows the ACK handshake protocol in a command level timing diagram. The READ_BYTE instruction is used as an example. First, the 8-bit instruction opcode is sent by the host, followed by the address of the memory location to be read. The target BDM decodes the instruction. A bus cycle is grabbed (free or stolen) by the BDM and it executes the READ_BYTE operation. Having retrieved the data, the BDM issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the byte retrieval process. Note that data is sent in the form of a word and the host needs to determine which is the appropriate byte based on whether the address was odd or even.

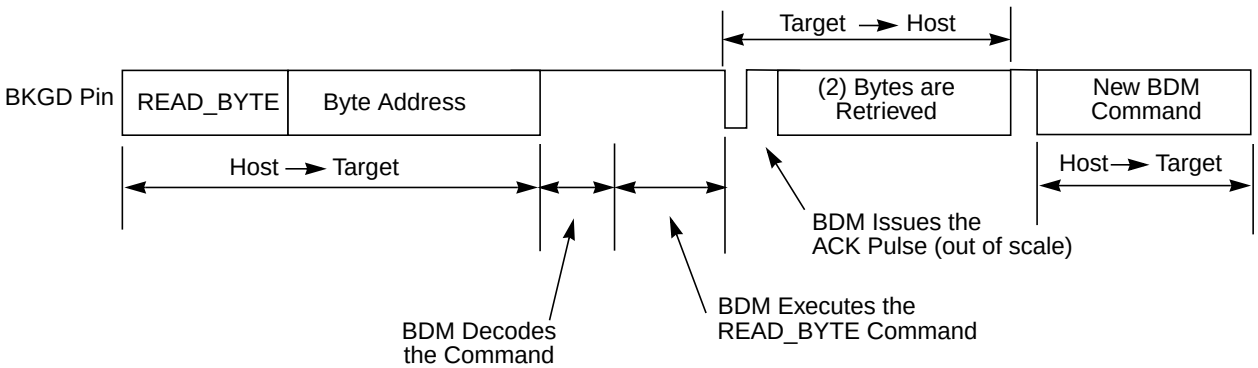

Figure 15-12. Handshake Protocol at Command Level

19.3.1.7 Debug State Control Registers

Each of the state sequencer states 1 to 3 features a dedicated control register to determine if transitions from that state are allowed depending upon comparator matches or tag hits and to define the next state for the state sequencer following a match. The 3 debug state control registers are located at the same address in the register address map (0x0027). Each register can be accessed using the COMRV bits in DBGC1 to blend in the required register (see Table 19-19).

Table 19-19. State Control Register Access Encoding

COMRV	Visible State Control Register
00	DBGSCR1
01	DBGSCR2
10	DBGSCR3
11	DBGSCR3

19.3.1.8 Debug State Control Register 1 (DBGSCR1)

0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W								
Reset	0	0	0	0	0	0	0	0

Unimplemented or Reserved

Figure 19-10. Debug State Control Register 1 (DBGSCR1)

Read: Anytime

Write: Anytime when DBG not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 00. The state control register 1 selects the targeted next state while in State1. The matches refer to the match channels of the comparator match control logic as depicted in Figure 19-1 and described in Section 19.3.1.11.1, “Debug Comparator Control Register (DBGXCTL)”. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 19-20. DBGSCR1 Field Descriptions

Field	Description
3–0 SC[3:0}	**State Control Bits** — These bits select the targeted next state while in State1, based upon the match event. See Table 19-21. The trigger priorities described in Table 19-38 dictate that in the case of simultaneous matches, the match on the lower channel number ([0,1,2,3) has priority. The SC[3:0] encoding ensures that a match leading to final state has priority over all other matches.

20.3.2.8.8 Debug Comparator Data Low Mask Register (DBGXDLM)

Address: 0x002F

	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 20-21. Debug Comparator Data Low Mask Register (DBGXDLM)

Read: Anytime

Write: Anytime when S12XDBG not armed.

Table 20-35. DBGXDLM Field Descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Data Low Mask Bits** — The Comparator data low mask bits control whether the selected comparator compares the data bus bits [7:0] to the corresponding comparator data compare bits. This register is available only for comparators A and C. 0 Do not compare corresponding data bit 1 Compare corresponding data bit

20.4 Functional Description

This section provides a complete functional description of the S12XDBG module. If the part is in secure mode, the S12XDBG module can generate breakpoints but tracing is not possible.

20.4.1 S12XDBG Operation

Arming the S12XDBG module by setting ARM in DBGC1 allows triggering, and storing of data in the trace buffer and can be used to cause breakpoints to the S12XCPU or the XGATE module. The DBG module is made up of four main blocks, the comparators, control logic, the state sequencer, and the trace buffer.

The comparators monitor the bus activity of the S12XCPU and XGATE modules. Comparators can be configured to monitor address and databus. Comparators can also be configured to mask out individual data bus bits during a compare and to use R/W and word/byte access qualification in the comparison. When a match with a comparator register value occurs the associated control logic can trigger the state sequencer to another state (see Figure 20-23). Either forced or tagged triggers are possible. Using a forced trigger, the trigger is generated immediately on a comparator match. Using a tagged trigger, at a comparator match, the instruction opcode is tagged and only if the instruction reaches the execution stage of the instruction queue is a trigger generated. In the case of a transition to Final State, bus tracing is triggered and/or a breakpoint can be generated. Tracing of both S12XCPU and/or XGATE bus activity is possible.

Independent of the state sequencer, a breakpoint can be triggered by the external $\overline{\text{TAGHI}}$ / $\overline{\text{TAGLO}}$ signals, by an XGATE S/W breakpoint request or by writing to the TRIG bit in the DBGC1 control register.

22.3.2.57 Port J Reduced Drive Register (RDRJ)

	7	6	5	4	3	2	1	0
R	RDRJ7	RDRJ6	RDRJ5	RDRJ4	0	RDRJ2	RDRJ1	RDRJ0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 22-59. Port J Reduced Drive Register (RDRJ)

Read: Anytime.

Write: Anytime.

This register configures the drive strength of each port J output pin as either full or reduced. If the port is used as input this bit is ignored.

Table 22-53. RDRJ Field Descriptions

Field	Description
7–0 RDRJ[7:4] RDRJ[2:0]	**Reduced Drive Port J** 0 Full drive strength at output. 1 Associated pin drives at about 1/6 of the full drive strength.

22.3.2.58 Port J Pull Device Enable Register (PERJ)

	7	6	5	4	3	2	1	0
R	PERJ7	PERJ6	PERJ5	PERJ4	0	PERJ2	PERJ1	PERJ0
W								
Reset	1	1	1	1	0	1	1	1

= Unimplemented or Reserved

Figure 22-60. Port J Pull Device Enable Register (PERJ)

Read: Anytime.

Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated, if the port is used as input or as wired-OR output. This bit has no effect if the port is used as push-pull output. Out of reset a pull-up device is enabled.

Table 22-54. PERJ Field Descriptions

Field	Description
7–0 PERJ[7:4] PERJ[2:0]	**Pull Device Enable Port J** 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

Table 23-4. PORTA Field Descriptions

Field	Description
7–0 PA[7:0]	Port A — Port A pins 7–0 are associated with address outputs ADDR15 through ADDR8 respectively inexpanded modesWhen this port is not used for external addresses, these pins can be used as general purpose I/O. If the data direction bits of the associated I/O pins are set to logic level “1”, a read returns the value of the port register, otherwise the buffered pin input state is read.

23.0.5.2 Port B Data Register (PORTB)

	7	6	5	4	3	2	1	0
R W	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Reset	0	0	0	0	0	0	0	0

Figure 23-4. Port B Data Register (PORTB)

Read: Anytime.

Write: Anytime.

Table 23-5. PORTB Field Descriptions

Field	Description
7–0 PB[7:0]	**Port B** — Port B pins 7–0 are associated with address outputs ADDR7 through ADDR1 respectively in expanded modes. Pin 0 is associated with output ADDR0 in emulation modes and special test mode and with Upper Data Select ($\overline{\text{UDS}}$) in normal expanded mode. When this port is not used for external addresses, these pins can be used as general purpose I/O. If the data direction bits of the associated I/O pins are set to logic level “1”, a read returns the value of the port register, otherwise the buffered pin input state is read.

23.0.5.3 Port A Data Direction Register (DDRA)

	7	6	5	4	3	2	1	0
R W	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset	0	0	0	0	0	0	0	0

Figure 23-5. Port A Data Direction Register (DDRA)

Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.

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

Table 24-61. Module Implementations on Derivatives

Number of Modules	MSCAN Modules					SPI Modules		
	CAN0	CAN1	CAN2	CAN3	CAN4	SPI0	SPI1	SPI2
5	yes	yes	yes	yes	yes	—	—	—
4	yes	yes	yes	—	yes	—	—	—
3	yes	yes	—	—	yes	yes	yes	yes
2	yes	—	—	—	yes	yes	yes	—
1	yes	—	—	—	—	yes	—	—

24.0.7 Ports

24.0.7.1 BKGD Pin

The BKGD pin is associated with the S12X_BDM and S12X_EBI modules. During reset, the BKGD pin is used as MODC input.

24.0.7.2 Port A and B

Port A pins PA[7:0] and Port B pins PB[7:0] can be used for either general-purpose I/O.

24.0.7.3 Port E

Port E pins PE[7:2] can be used for either general-purpose I/O or with the alternative functions.

Port E pin PE[7] an be used for either general-purpose I/O or as the free-running clock ECLKX2 output running at the core clock rate. The clock output is always enabled in emulation modes.

Port E pin PE[4] an be used for either general-purpose I/O or as the free-running clock ECLK output running at the bus clock rate or at the programmed divided clock rate. The clock output is always enabled in emulation modes.

Port E pin PE[1] can be used for either general-purpose input or as the level- or falling edge-sensitive $\overline{\text{IRQ}}$ interrupt input. $\overline{\text{IRQ}}$ will be enabled by setting the IRQEN configuration bit (Section 24.0.5.10, “IRQ Control Register (IRQCR)”) and clearing the I-bit in the CPU’s condition code register. It is inhibited at reset so this pin is initially configured as a simple input with a pull-up.

Port E pin PE[0] can be used for either general-purpose input or as the level-sensitive $\overline{\text{XIRQ}}$ interrupt input. $\overline{\text{XIRQ}}$ can be enabled by clearing the X-bit in the CPU’s condition code register. It is inhibited at reset so this pin is initially configured as a high-impedance input with a pull-up.

24.0.7.4 Port K

Port K pins PK[7:0] can be used for either general-purpose I/O.

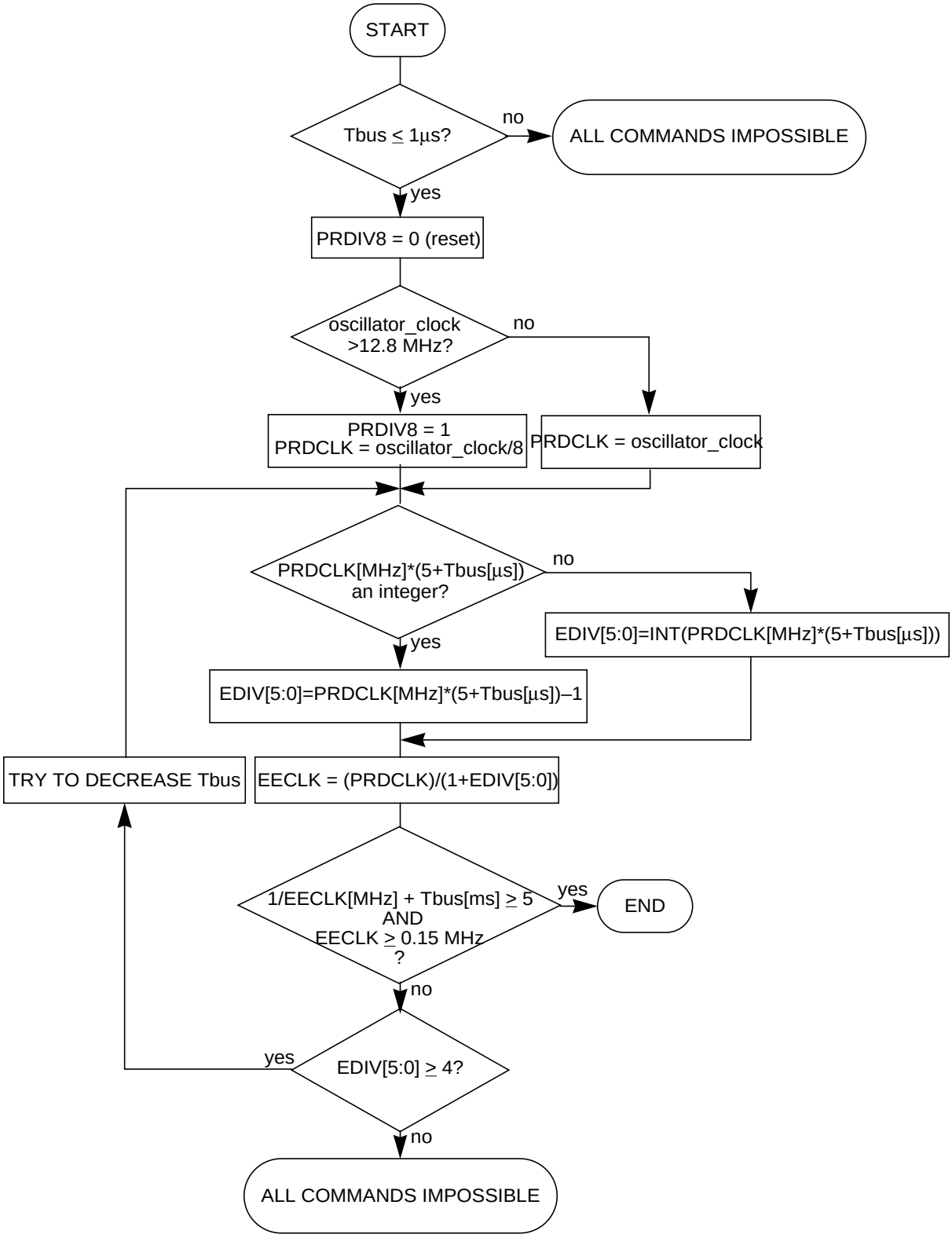

Figure 25-17. Determination Procedure for PRDIV8 and EDIV Bits

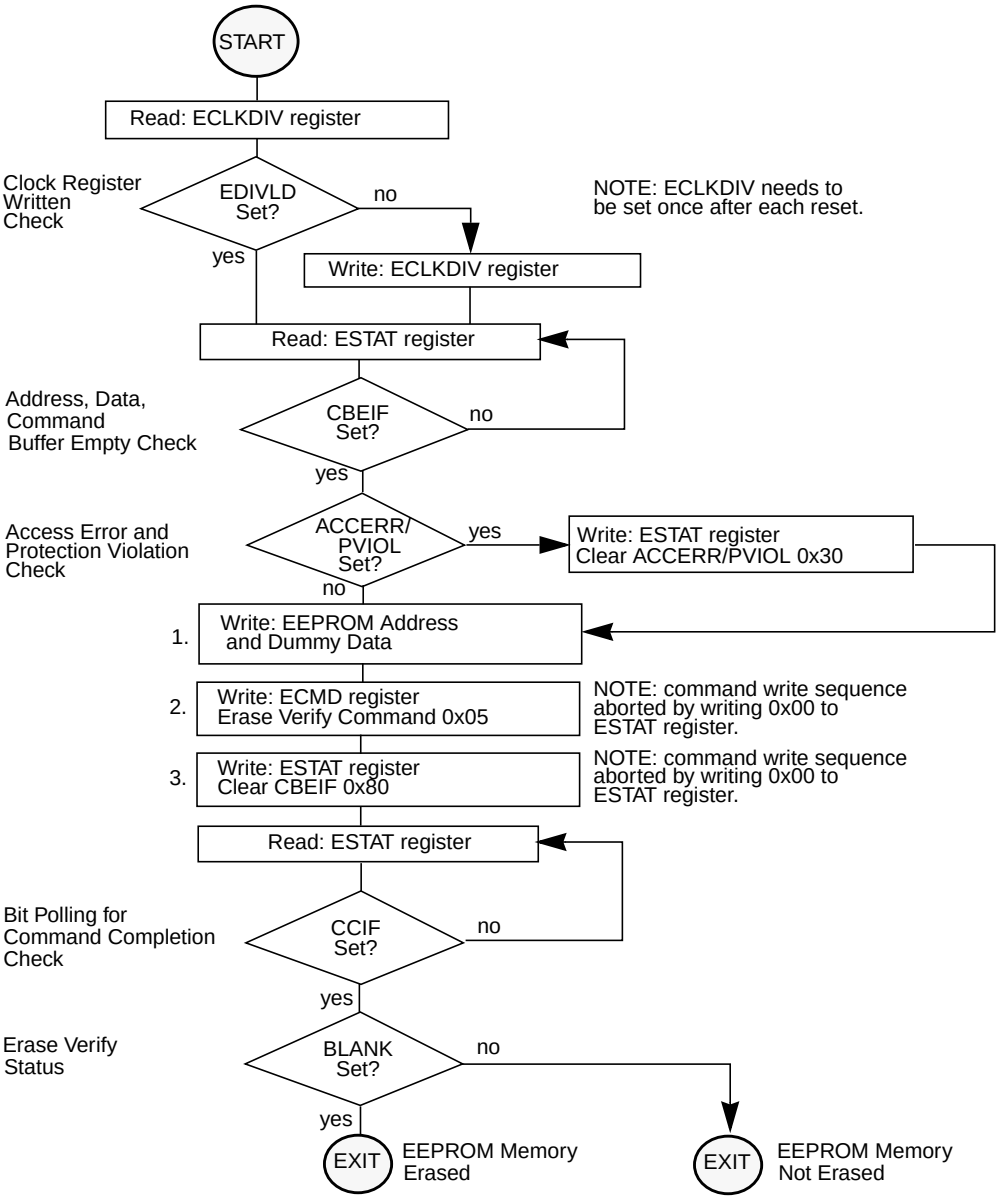

Figure 26-18. Example Erase Verify Command Flow

27.6.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0x7F_FF00–0x7F_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 27.3.2.2, “Flash Security Register (FSEC)”) and the KEYACC bit is set, a write to a backdoor key address in the Flash memory triggers a comparison between the written data and the backdoor key data stored in the Flash memory. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor keys stored in the Flash memory, the MCU will be unsecured. The data must be written to the backdoor keys sequentially starting with 0x7F_FF00–1 and ending with 0x7F_FF06–7. 0x0000 and 0xFFFF are not permitted as backdoor keys. While the KEYACC bit is set, reads of the Flash memory will return invalid data.

The user code stored in the Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 27.3.2.2, “Flash Security Register (FSEC)”), the MCU can be unsecured by the backdoor key access sequence described below:

1. Set the KEYACC bit in the Flash Configuration Register (FCNFG).
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0x7F_FF00.
3. Clear the KEYACC bit. Depending on the user code used to write the backdoor keys, a wait cycle (NOP) may be required before clearing the KEYACC bit.
4. If all four 16-bit words match the backdoor keys stored in Flash addresses 0x7F_FF00–0x7F_FF07, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 1:0.

The backdoor key access sequence is monitored by an internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following operations during the backdoor key access sequence will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor keys programmed in the Flash array.
2. If the four 16-bit words are written in the wrong sequence.
3. If more than four 16-bit words are written.
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF.
5. If the KEYACC bit does not remain set while the four 16-bit words are written.
6. If any two of the four 16-bit words are written on successive MCU clock cycles.

After the backdoor keys have been correctly matched, the MCU will be unsecured. Once the MCU is unsecured, the Flash security byte can be programmed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x7F_FF00–0x7F_FF07 in the Flash Configuration Field.

The security as defined in the Flash security byte (0x7F_FF0F) is not changed by using the backdoor key access sequence to unsecure. The backdoor keys stored in addresses 0x7F_FF00–0x7F_FF07 are

29.3.2.5.1 Flash Protection Restrictions

The general guideline is that Flash protection can only be added and not removed. Table 29-13 specifies all valid transitions between Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored and the FPROT register will remain unchanged. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS descriptions for additional restrictions.

Table 29-13. Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	
7	X	X	X	X	X	X	X	X

1 Allowed transitions marked with X.

29.3.2.6 Flash Status Register (FSTAT)

The FSTAT register defines the operational status of the module.

	7	6	5	4	3	2	1	0
R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	0	0
W								
Reset	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 29-10. Flash Status Register (FSTAT — Normal Mode)

	7	6	5	4	3	2	1	0
R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	0
W								
Reset	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 29-11. Flash Status Register (FSTAT — Special Mode)

29.4.2.6 Sector Erase Abort Command

The sector erase abort operation will terminate the active sector erase operation so that other sectors in a Flash block are available for read and program operations without waiting for the sector erase operation to complete.

An example flow to execute the sector erase abort operation is shown in Figure 29-29. The sector erase abort command write sequence is as follows:

1. Write to any Flash block address to start the command write sequence for the sector erase abort command. The address and data written are ignored.
2. Write the sector erase abort command, 0x47, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase abort command.

If the sector erase abort command is launched resulting in the early termination of an active sector erase operation, the ACCERR flag will set once the operation completes as indicated by the CCIF flag being set. The ACCERR flag sets to inform the user that the Flash sector may not be fully erased and a new sector erase command must be launched before programming any location in that specific sector. If the sector erase abort command is launched but the active sector erase operation completes normally, the ACCERR flag will not set upon completion of the operation as indicated by the CCIF flag being set. Therefore, if the ACCERR flag is not set after the sector erase abort command has completed, a Flash sector being erased when the abort command was launched will be fully erased. The maximum number of cycles required to abort a sector erase operation is equal to four FCLK periods (see Section 29.4.1.1, “Writing the FCLKDIV Register”) plus five bus cycles as measured from the time the CBEIF flag is cleared until the CCIF flag is set.

NOTE

Since the ACCERR bit in the FSTAT register may be set at the completion of the sector erase abort operation, a command write sequence is not allowed to be buffered behind a sector erase abort command write sequence. The CBEIF flag will not set after launching the sector erase abort command to indicate that a command should not be buffered behind it. If an attempt is made to start a new command write sequence with a sector erase abort operation active, the ACCERR flag in the FSTAT register will be set. A new command write sequence may be started after clearing the ACCERR flag, if set.

NOTE

The sector erase abort command should be used sparingly since a sector erase operation that is aborted counts as a complete program/erase cycle.

Table A-9. shows the configuration of the peripherals for run current measurement.

Table A-9. Peripheral Configurations for Run Supply Current Measurements

Peripheral	Configuration
MSCAN	configured to loop-back mode using a bit rate of 1Mbit/s
SPI	configured to master mode, continously transmit data (0x55 or 0xAA) at 1Mbit/s
SCI	configured into loop mode, continously transmit data (0x55) at speed of 57600 baud
IIC	operate in master mode and continously transmit data (0x55 or 0xAA) at the bit rate of 100Kbit/s
PWM	configured to toggle its pins at the rate of 40kHz
ECT	the peripheral shall be configured to output compare mode, Pulse accumulator and modulus counter enabled.
ATD	the peripheral is configured to operate at its maximum specified frequency and to continuously convert voltages on all input channels in sequence.
XGATE	XGATE fetches code from RAM, XGATE runs in an infinite loop , it reads the Status and Flag registers of CAN's, SPI's, SCI's in sequence and does some bit manipulation on the data
COP	COP Warchdog Rate 2^{24}
RTI	enabled, RTI Control Register (RTICTL) set to $FF
API	the module is configured to run from the RC oscillator clock source.
PIT	PIT is enabled, Micro-timer register 0 and 1 loaded with $0F and timer registers 0 to 3 are loaded with $03/07/0F/1F.
DBG	the module is enabled and the comparators are configured to trigger in outside range. The range covers all the code executed by the core.

0x00E0–0x00E7 Inter IC Bus (IIC0) Map (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00E5	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00E6	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00E7	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x00E8–0x00EF Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00E8	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00E9	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00EA	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00EB	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00EC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00ED	Reserved	R	0	0	0	0	0	0	0	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								

0x00F0–0x00F7 Serial Peripheral Interface (SPI1) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00F0	SPI1CR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
		W								
0x00F1	SPI1CR2	R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
		W								
0x00F2	SPI1BR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
		W								
0x00F3	SPI1SR	R	SPIF	0	SPTEF	MODF	0	0	0	0
		W								
0x00F4	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00F5	SPI1DR	R	Bit7	6	5	4	3	2	1	Bit0
		W								
0x00F6	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00F7	Reserved	R	0	0	0	0	0	0	0	0
		W								