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	Obsolete
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	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 24x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12xdt256cag

The S12XD, S12XA and S12XB family devices are offered in the following packages:

- 144-pin LQFP package with an external bus interface (address/data bus)
- 112-pin LQFP without external bus interface
- 80-pin QFP without external bus interface

See Appendix E Derivative Differences for package options.

CAUTION

Most the I/O Pins have different functionality depending on the module configuration. Not all functions are shown in the following pinouts. Please refer to Table 1-7 for a complete description. For avalability of the modules on different family members refer to Appendix E Derivative Differences. For pinout explanations of the different parts refer to E.7 Pinout explanations:

The following conditions apply when the PLL is in automatic bandwidth control mode (AUTO = 1):

- The TRACK bit is a read-only indicator of the mode of the filter.
- The TRACK bit is set when the VCO frequency is within a certain tolerance, Δ_{trk}, and is clear when the VCO frequency is out of a certain tolerance, Δ_{unt}.
- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance, Δ_{Lock}, and is cleared when the VCO frequency is out of a certain tolerance, Δ_{unl}.
- Interrupt requests can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

The PLL can also operate in manual mode (AUTO = 0). Manual mode is used by systems that do not require an indicator of the lock condition for proper operation. Such systems typically operate well below the maximum system frequency (f_{sys}) and require fast start-up. The following conditions apply when in manual mode:

- ACQ is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the ACQ bit should be asserted to configure the filter in acquisition mode.
- After turning on the PLL by setting the PLLON bit software must wait a given time (t_{acq}) before entering tracking mode (ACQ = 0).
- After entering tracking mode software must wait a given time (t_{al}) before selecting the PLLCLK as the source for system and core clocks (PLLSEL = 1).

2.4.1.2 System Clocks Generator

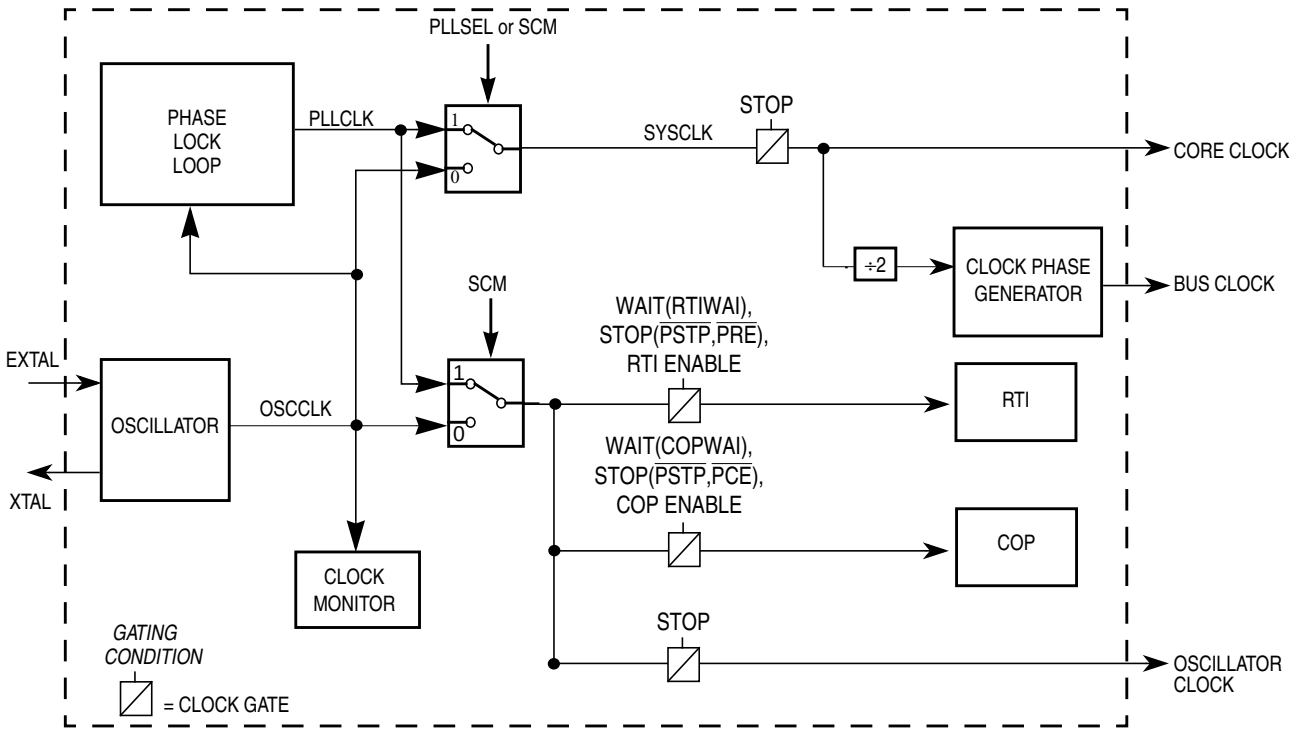

Figure 2-17. System Clocks Generator

Figure 6-22 gives an example of the typical usage of the XGATE hardware semaphores.

Two concurrent threads are running on the system. One is running on the S12X_CPU and the other is running on the RISC core. They both have a critical section of code that accesses the same system resource. To guarantee that the system resource is only accessed by one thread at a time, the critical code sequence must be embedded in a semaphore lock/release sequence as shown.

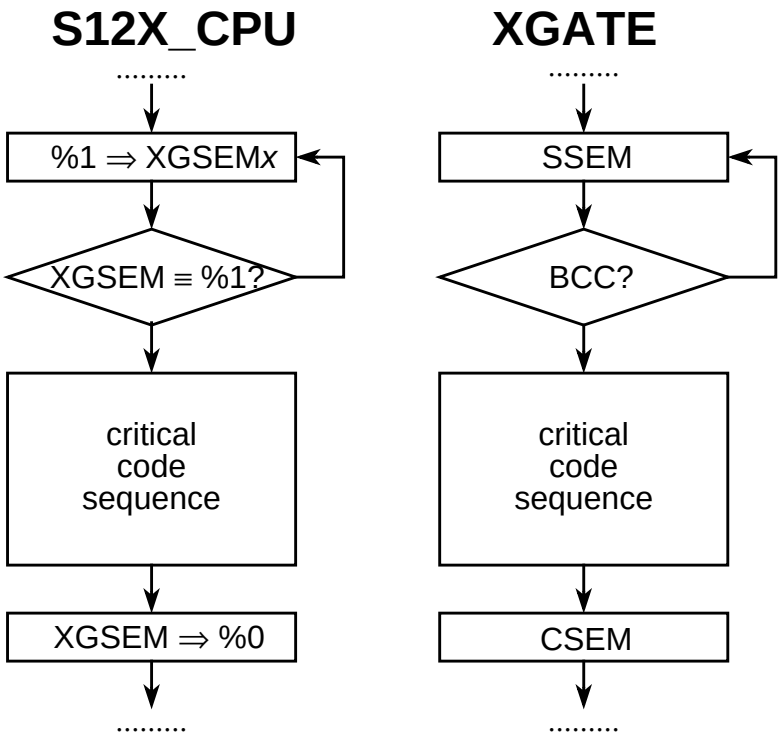

Figure 6-22. Algorithm for Locking and Releasing Semaphores

6.4.5 Software Error Detection

The XGATE module will immediately terminate program execution after detecting an error condition caused by erratic application code. There are three error conditions:

- Execution of an illegal opcode
- Illegal vector or opcode fetches
- Illegal load or store accesses

All opcodes which are not listed in section Section 6.8, “Instruction Set” are illegal opcodes. Illegal vector and opcode fetches as well as illegal load and store accesses are defined on chip level. Refer to the **S12X_MMC Section** for a detailed information.

Table 6-17. Instruction Set Summary (Sheet 3 of 3)

Functionality	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANDH RD, #IMM8	1	0	0	0	1	RD			IMM8							
BITL RD, #IMM8	1	0	0	1	0	RD			IMM8							
BITH RD, #IMM8	1	0	0	1	1	RD			IMM8							
ORL RD, #IMM8	1	0	1	0	0	RD			IMM8							
ORH RD, #IMM8	1	0	1	0	1	RD			IMM8							
XNORL RD, #IMM8	1	0	1	1	0	RD			IMM8							
XNORH RD, #IMM8	1	0	1	1	1	RD			IMM8							
Arithmetic Immediate Instructions																
SUBL RD, #IMM8	1	1	0	0	0	RD			IMM8							
SUBH RD, #IMM8	1	1	0	0	1	RD			IMM8							
CMPL RS, #IMM8	1	1	0	1	0	RS			IMM8							
CPCH RS, #IMM8	1	1	0	1	1	RS			IMM8							
ADDL RD, #IMM8	1	1	1	0	0	RD			IMM8							
ADDH RD, #IMM8	1	1	1	0	1	RD			IMM8							
LDL RD, #IMM8	1	1	1	1	0	RD			IMM8							
LDH RD, #IMM8	1	1	1	1	1	RD			IMM8							

7.3.2.31 Timer Input Capture Holding Registers 0–3 (TCxH)

	15	14	13	12	11	10	9	8
R	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-57. Timer Input Capture Holding Register 0 High (TC0H)

	7	6	5	4	3	2	1	0
R	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-58. Timer Input Capture Holding Register 0 Low (TC0H)

	15	14	13	12	11	10	9	8
R	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-59. Timer Input Capture Holding Register 1 High (TC1H)

	7	6	5	4	3	2	1	0
R	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-60. Timer Input Capture Holding Register 1 Low (TC1H)

	15	14	13	12	11	10	9	8
R	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-61. Timer Input Capture Holding Register 2 High (TC2H)

	7	6	5	4	3	2	1	0
R	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-62. Timer Input Capture Holding Register 2 Low (TC2H)

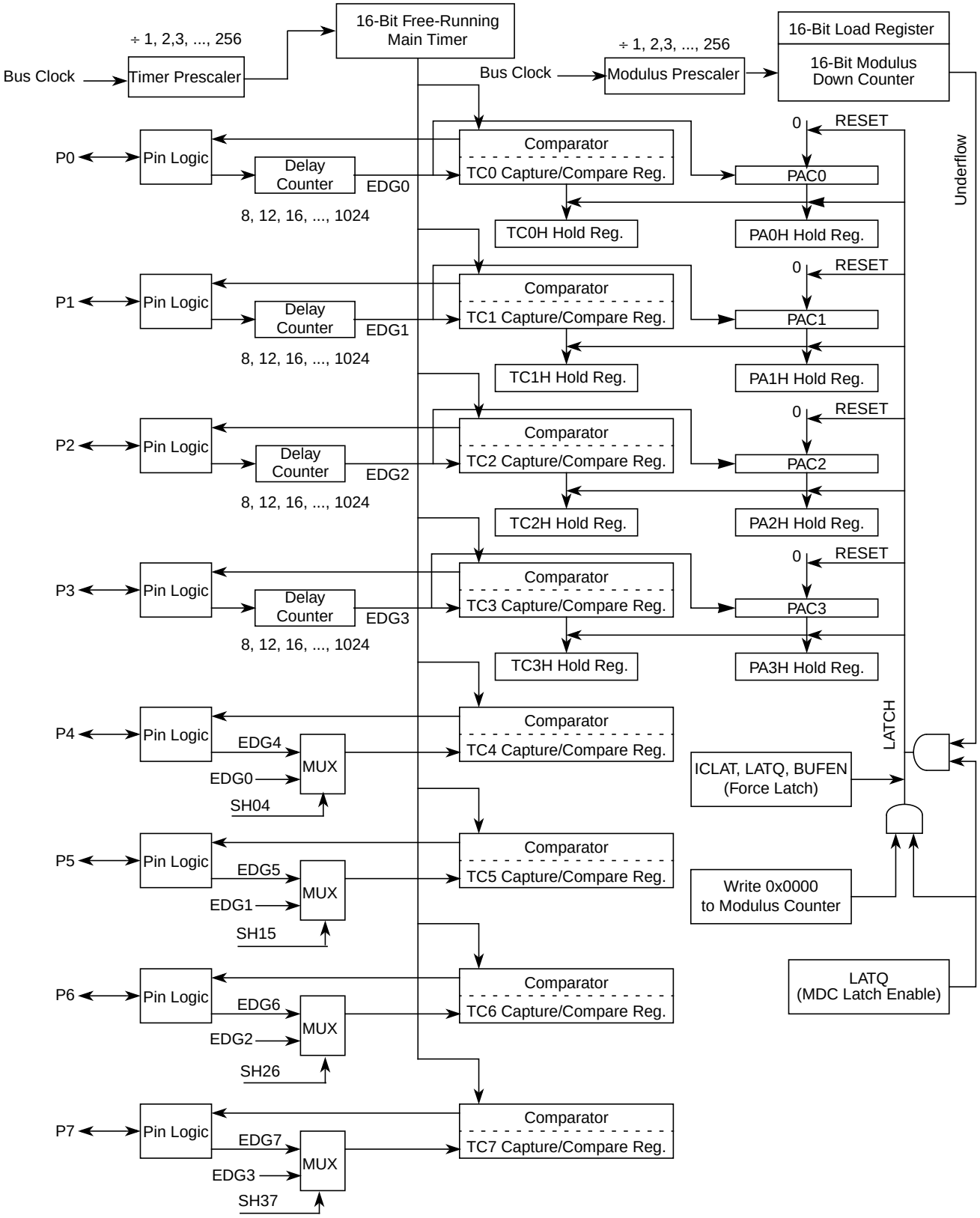

Figure 7-66. Detailed Timer Block Diagram in Latch Mode when PRNT = 1

Each channel counter can be read at anytime without affecting the count or the operation of the PWM channel.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. When the channel is disabled (PWMEx = 0), the counter stops. When a channel becomes enabled (PWMEx = 1), the associated PWM counter continues from the count in the PWMCNTx register. This allows the waveform to continue where it left off when the channel is re-enabled. When the channel is disabled, writing “0” to the period register will cause the counter to reset on the next selected clock.

NOTE

If the user wants to start a new “clean” PWM waveform without any “history” from the old waveform, the user must write to channel counter (PWMCNTx) prior to enabling the PWM channel (PWMEx = 1).

Generally, writes to the counter are done prior to enabling a channel in order to start from a known state. However, writing a counter can also be done while the PWM channel is enabled (counting). The effect is similar to writing the counter when the channel is disabled, except that the new period is started immediately with the output set according to the polarity bit.

NOTE

Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.

The counter is cleared at the end of the effective period (see Section 8.4.2.5, “Left Aligned Outputs” and Section 8.4.2.6, “Center Aligned Outputs” for more details).

Table 8-10. PWM Timer Counter Conditions

Counter Clears ($00)	Counter Counts	Counter Stops
When PWMCNTx register written to any value	When PWM channel is enabled (PWMEx = 1). Counts from last value in PWMCNTx.	When PWM channel is disabled (PWMEx = 0)
Effective period ends		

8.4.2.5 Left Aligned Outputs

The PWM timer provides the choice of two types of outputs, left aligned or center aligned. They are selected with the CAEx bits in the PWMCAE register. If the CAEx bit is cleared (CAEx = 0), the corresponding PWM output will be left aligned.

In left aligned output mode, the 8-bit counter is configured as an up counter only. It compares to two registers, a duty register and a period register as shown in the block diagram in Figure 8-19. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register resets the counter and the output flip-flop, as shown in Figure 8-19, as well as performing a load from the double buffer period and duty register to the associated registers, as described in Section 8.4.2.3, “PWM Period and Duty”. The counter counts from 0 to the value in the period register – 1.

11.2 External Signal Description

The SCI module has a total of two external pins.

11.2.1 TXD — Transmit Pin

The TXD pin transmits SCI (standard or infrared) data. It will idle high in either mode and is high impedance anytime the transmitter is disabled.

11.2.2 RXD — Receive Pin

The RXD pin receives SCI (standard or infrared) data. An idle line is detected as a line high. This input is ignored when the receiver is disabled and should be terminated to a known voltage.

11.3 Memory Map and Register Definition

This section provides a detailed description of all the SCI registers.

11.3.1 Module Memory Map and Register Definition

The memory map for the SCI module is given below in Figure 11-2. The address listed for each register is the address offset. The total address for each register is the sum of the base address for the SCI module and the address offset for each register.

within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

As soon as the SYNC request is detected by the target, any partially received command or bit retrieved is discarded. This is referred to as a soft-reset, equivalent to a time-out in the serial communication. After the SYNC response, the target will consider the next negative edge (issued by the host) as the start of a new BDM command or the start of new SYNC request.

Another use of the SYNC command pulse is to abort a pending ACK pulse. The behavior is exactly the same as in a regular SYNC command. Note that one of the possible causes for a command to not be acknowledged by the target is a host-target synchronization problem. In this case, the command may not have been understood by the target and so an ACK response pulse will not be issued.

15.4.10 Instruction Tracing

When a TRACE1 command is issued to the BDM in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. Once this has occurred, the CPU is forced to return to the standard BDM firmware and the BDM is active and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Once back in standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

Be aware when tracing through the user code that the execution of the user code is done step by step but all peripherals are free running. Hence possible timing relations between CPU code execution and occurrence of events of other peripherals no longer exist.

Do not trace the CPU instruction BGND used for soft breakpoints. Tracing the BGND instruction will result in a return address pointing to BDM firmware address space.

When tracing through user code which contains stop or wait instructions the following will happen when the stop or wait instruction is traced:

The CPU enters stop or wait mode and the TRACE1 command can not be finished before leaving the low power mode. This is the case because BDM active mode can not be entered after CPU executed the stop instruction. However all BDM hardware commands except the BACKGROUND command are operational after tracing a stop or wait instruction and still being in stop or wait mode. If system stop mode is entered (all bus masters are in stop mode) no BDM command is operational.

As soon as stop or wait mode is exited the CPU enters BDM active mode and the saved PC value points to the entry of the corresponding interrupt service routine.

In case the handshake feature is enabled the corresponding ACK pulse of the TRACE1 command will be discarded when tracing a stop or wait instruction. Hence there is no ACK pulse when BDM active mode is entered as part of the TRACE1 command after CPU exited from stop or wait mode. All valid commands sent during CPU being in stop or wait mode or after CPU exited from stop or wait mode will have an ACK pulse. The handshake feature becomes disabled only when system

17.5.3 On-Chip ROM Control

The MCU offers two modes to support emulation. In the first mode (called generator) the emulator provides the data instead of the internal FLASH and traces the CPU actions. In the other mode (called observer) the internal FLASH provides the data and all internal actions are made visible to the emulator.

17.5.3.1 ROM Control in Single-Chip Modes

In single-chip modes the MCU has no external bus. All memory accesses and program fetches are internal (see Figure 1-27).

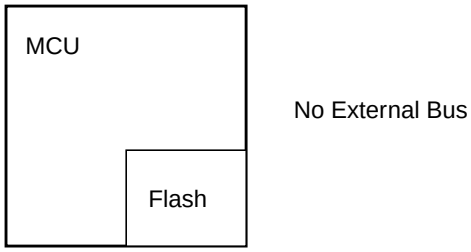

Figure 17-27. ROM in Single Chip Modes

17.5.3.2 ROM Control in Emulation Single-Chip Mode

In emulation single-chip mode the external bus is connected to the emulator. If the EROMON bit is set, the internal FLASH provides the data and the emulator can observe all internal CPU actions on the external bus. If the EROMON bit is cleared, the emulator provides the data (generator) and traces the all CPU actions (see Figure 1-28).

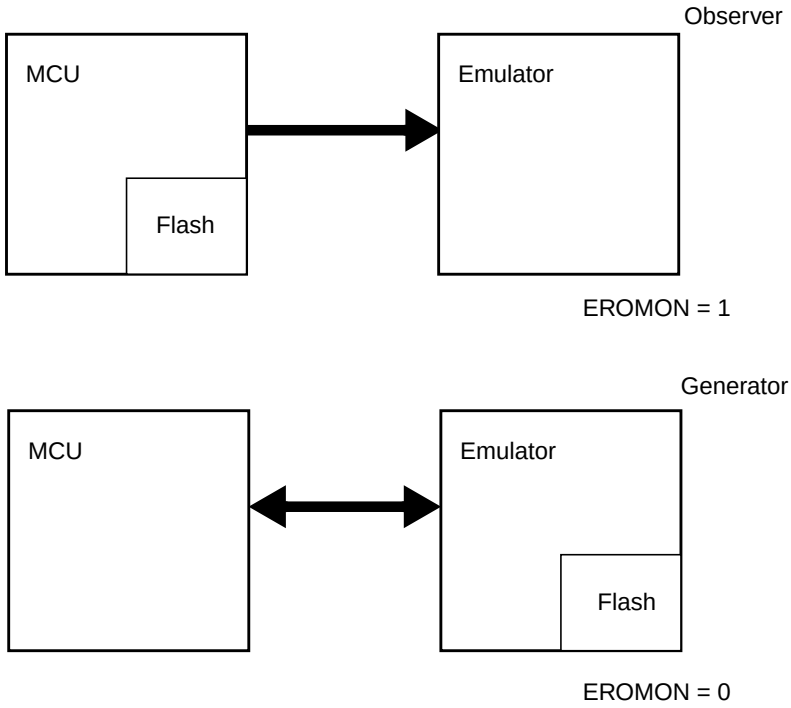

Figure 17-28. ROM in Emulation Single-Chip Mode

Chapter 18
Memory Mapping Control (S12XMMCV3)

18.1 Introduction

This section describes the functionality of the module mapping control (MMC) sub-block of the S12X platform. The block diagram of the MMC is shown in Figure 18-1.

The MMC module controls the multi-master priority accesses, the selection of internal resources and external space. Internal buses, including internal memories and peripherals, are controlled in this module. The local address space for each master is translated to a global memory space.

Table 19-27. DBGXCTL Field Descriptions (continued)

Field	Description
6 (COMP B/D) SZ	**Size Comparator Value Bit** — The SZ bit selects either word or byte access size in comparison for the associated comparator. This bit is ignored if the SZE bit is cleared or if the TAG bit in the same register is set. This bit position has NDB functionality for comparators A and C 0 Word access size will be compared 1 Byte access size will be compared
5 TAG	**Tag Select** — This bit controls whether the comparator match will cause a trigger or tag the opcode at the matched address. Tagged opcodes trigger only if they reach the execution stage of the instruction queue. 0 Trigger immediately on match 1 On match, tag the opcode. If the opcode is about to be executed a trigger is generated
4 BRK	**Break** — This bit controls whether a comparator match can cause an immediate breakpoint independent of state sequencer state. The module breakpoints must be enabled using the DBGC1 bits DBGBRK[1:0]. 0 Breakpoints may only be generated from this channel when the state machine reaches final state. 1 A match on this channel generates an immediate breakpoint, tracing, if active, is terminated and the module disarmed.
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is not used if RWE = 0. 0 Write cycle will be matched 1 Read cycle will be matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is not useful for tagged operations. 1 Read/Write is used in comparison 0 Read/Write is not used in comparison
1 SRC	**SRC** — Determines mapping of comparator to CPU or XGATE 0 The comparator is mapped to CPU busses 1 The comparator is mapped to XGATE address and data busses
0 COMPE	**Comparator Enable Bit**— Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled for state sequence triggers or tag generation

Table 19-28 shows the effect for RWE and RW on the comparison conditions. These bits are not useful for tagged operations since the trigger occurs based on the tagged opcode reaching the execution stage of the instruction queue. Thus, these bits are ignored if tagged triggering is selected.

Table 19-28. Read or Write Comparison Logic Table

RWE Bit	RW Bit	RW Signal	Comment
0	x	0	RW not used in comparison
0	x	1	RW not used in comparison
1	0	0	Write data bus
1	0	1	No match
1	1	0	No match
1	1	1	Read data bus

20.3.2.2 Debug Status Register (DBGSR)

Address: 0x0021

	7	6	5	4	3	2	1	0
R	TBF	EXTF	0	0	0	SSF2	SSF1	SSF0
W								
Reset	—	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-4. Debug Status Register (DBGSR)

Read: Anytime

Write: Never

Table 20-6. DBGSR Field Descriptions

Field	Description
7 TBF	**Trace Buffer Full** — The TBF bit indicates that the trace buffer has stored 64 or more lines of data since it was last armed. If this bit is set, then all 64 lines will be valid data, regardless of the value of DBGCNT bits CNT[6:0]. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit
6 EXTF	**External Tag Hit Flag** — The EXTF bit indicates if a tag hit condition from an external $\overline{\text{TAGHI}}$/$\overline{\text{TAGLO}}$ tag was met since arming. This bit is cleared when ARM in DBGC1 is written to a one. 0 External tag hit has not occurred 1 External tag hit has occurred
2–0 SSF[2:0]	**State Sequencer Flag Bits** — The SSF bits indicate in which state the State Sequencer is currently in. During a debug session on each transition to a new state these bits are updated. If the debug session is ended by software clearing the ARM bit, then these bits retain their value to reflect the last state of the state sequencer before disarming. If a debug session is ended by an internal trigger, then the state sequencer returns to state0 and these bits are cleared to indicate that state0 was entered during the session. On arming the module the state sequencer enters state1 and these bits are forced to SSF[2:0] = 001. See Table 20-7.

Table 20-7. SSF[2:0] — State Sequence Flag Bit Encoding

SSF[2:0]	Current State
000	State0 (disarmed)
001	State1
010	State2
011	State3
100	Final State
101,110,111	Reserved

Table 20-45. Breakpoint Mapping Summary

DBGBRK[1] (DBGC1[3])	BDM Bit (DBGC1[4])	BDM Enabled	BDM Active	S12X Breakpoint Mapping
0	X	X	X	No Breakpoint
1	0	X	0	Breakpoint to SWI
1	0	X	1	No Breakpoint
1	1	0	X	Breakpoint to SWI
1	1	1	0	Breakpoint to BDM
1	1	1	1	No Breakpoint

BDM cannot be entered from a breakpoint unless the ENABLE bit is set in the BDM. If entry to BDM via a BGND instruction is attempted and the ENABLE bit in the BDM is cleared, the S12XCPU actually executes the BDM firmware code. It checks the ENABLE and returns if ENABLE is not set. If not serviced by the monitor then the breakpoint is re-asserted when the BDM returns to normal S12XCPU flow.

If the comparator register contents coincide with the SWI/BDM vector address then an SWI in user code and DBG breakpoint could occur simultaneously. The S12XCPU ensures that BDM requests have a higher priority than SWI requests. Returning from the BDM/SWI service routine care must be taken to avoid re triggering a breakpoint.

NOTE

When program control returns from a tagged breakpoint using an RTI or BDM GO command without program counter modification it will return to the instruction whose tag generated the breakpoint. To avoid re triggering a breakpoint at the same location reconfigure the S12XDBG module in the SWI routine, if configured for an SWI breakpoint, or over the BDM interface by executing a TRACE command before the GO to increment the program flow past the tagged instruction.

An XGATE software breakpoint is forced immediately, the tracing session terminated and the XGATE module execution stops. The user can thus determine if an XGATE breakpoint has occurred by reading out the XGATE program counter over the BDM interface.

21.4 Functional Description

This section describes the functions of the external bus interface. The availability of external signals and functions in relation to the operating mode is initially summarized and described in more detail in separate sub-sections.

21.4.1 Operating Modes and External Bus Properties

A summary of the external bus interface functions for each operating mode is shown in Table 21-7.

Table 21-7. Summary of Functions

Properties (if Enabled)	Single-Chip Modes		Expanded Modes			
	Normal Single-Chip	Special Single-Chip	Normal Expanded	Emulation Single-Chip	Emulation Expanded	Special Test
Timing Properties						
PRR access[1]	2 cycles read internal write internal	2 cycles read internal write internal	2 cycles read internal write internal	2 cycles read external write int & ext	2 cycles read external write int & ext	2 cycles read internal write internal
Internal access visible externally	—	—	—	1 cycle	1 cycle	1 cycle
External address access and unimplemented area access[2]	—	—	Max. of 2 to 9 programmed cycles or n cycles of ext. wait[3]	1 cycle	Max. of 2 to 9 programmed cycles or n cycles of ext. wait[3]	1 cycle
Flash area address access[4]	—	—	—	1 cycle	1 cycle	1 cycle
Signal Properties						
Bus signals	—	—	ADDR[22:1] DATA[15:0]	ADDR[22:20]/ACC[2:0] ADDR[19:16]/IQSTAT[3:0] ADDR[15:0]/IVD[15:0] DATA[15:0]	ADDR[22:20]/ACC[2:0] ADDR[19:16]/IQSTAT[3:0] ADDR[15:0]/IVD[15:0] DATA[15:0]	ADDR[22:0] DATA[15:0]
Data select signals (if 16-bit data bus)	—	—	$\overline{UDS}$ $\overline{LDS}$	ADDR0 $\overline{LSTRB}$	ADDR0 $\overline{LSTRB}$	ADDR0 $\overline{LSTRB}$
Data direction signals	—	—	$\overline{RE}$ $\overline{WE}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$
External wait feature	—	—	$\overline{EWAIT}$	—	$\overline{EWAIT}$	—
Reduced input threshold enabled on	—	—	Refer to Table 21-3	DATA[15:0] $\overline{EWAIT}$	DATA[15:0] $\overline{EWAIT}$	Refer to Table 21-3

1 Incl. S12X_EBI registers

2 Refer to S12X_MMC section.

3 If EWAITE = 1, the minimum number of external bus cycles is 3.

4 Available only if configured appropriately by ROMON and EROMON (refer to S12X_MMC section).

22.3.2.22 Port T Polarity Select Register (PPST)

	7	6	5	4	3	2	1	0
R W	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
Reset	0	0	0	0	0	0	0	0

Figure 22-24. Port T Polarity Select Register (PPST)

Read: Anytime.

Write: Anytime.

This register selects whether a pull-down or a pull-up device is connected to the pin.

Table 22-26. PPST Field Descriptions

Field	Description
7–0 PPST[7:0]	**Pull Select Port T** 0 A pull-up device is connected to the associated port T pin, if enabled by the associated bit in register PERT and if the port is used as input. 1 A pull-down device is connected to the associated port T pin, if enabled by the associated bit in register PERT and if the port is used as input.

22.3.2.23 Port S Data Register (PTS)

	7	6	5	4	3	2	1	0
R W	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
SCI/SPI	$\overline{SS0}$	SCK0	MOSI0	MISO0	TXD1	RXD1	TXD0	RXD0
Reset	0	0	0	0	0	0	0	0

Figure 22-25. Port S Data Register (PTS)

Read: Anytime.

Write: Anytime.

Port S pins 7–4 are associated with the SPI0. The SPI0 pin configuration is determined by several status bits in the SPI0 module. *Refer to SPI section for details.* When not used with the SPI0, these pins can be used as general purpose I/O.

Port S bits 3–0 are associated with the SCI1 and SCI0. The SCI ports associated with transmit pins 3 and 1 are configured as outputs if the transmitter is enabled. The SCI ports associated with receive pins 2 and 0 are configured as inputs if the receiver is enabled. *Refer to SCI section for details.* When not used with the SCI, 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.

Table 24-1. Pin Functions and Priorities (Sheet 5 of 5)

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	
J	PJ7	TXCAN4	O	MSCAN4 transmit pin	GPIO
		SCL0	O	Inter Integrated Circuit 0 serial clock line	
		TXCAN0	O	MSCAN0 transmit pin	
		GPIO/KWJ7	I/O	General-purpose I/O with interrupt	
	PJ6	RXCAN4	I	MSCAN4 receive pin	
		SDA0	I/O	Inter Integrated Circuit 0 serial data line	
		RXCAN0	I	MSCAN0 receive pin	
		GPIO/KWJ6	I/O	General-purpose I/O with interrupt	
	PJ1	GPIO/KWJ1	I/O	General-purpose I/O with interrupt	
	PJ0	GPIO/KWJ0	I/O	General-purpose I/O with interrupt	
AD1	PAD[15:00]	GPIO	I/O	General-purpose I/O	GPIO
		AN[15:0]	I	ATD1 analog inputs	

1. Function active when $\overline{RESET}$ asserted.

25.1.4 Block Diagram

A block diagram of the EEPROM module is shown in Figure 25-1.

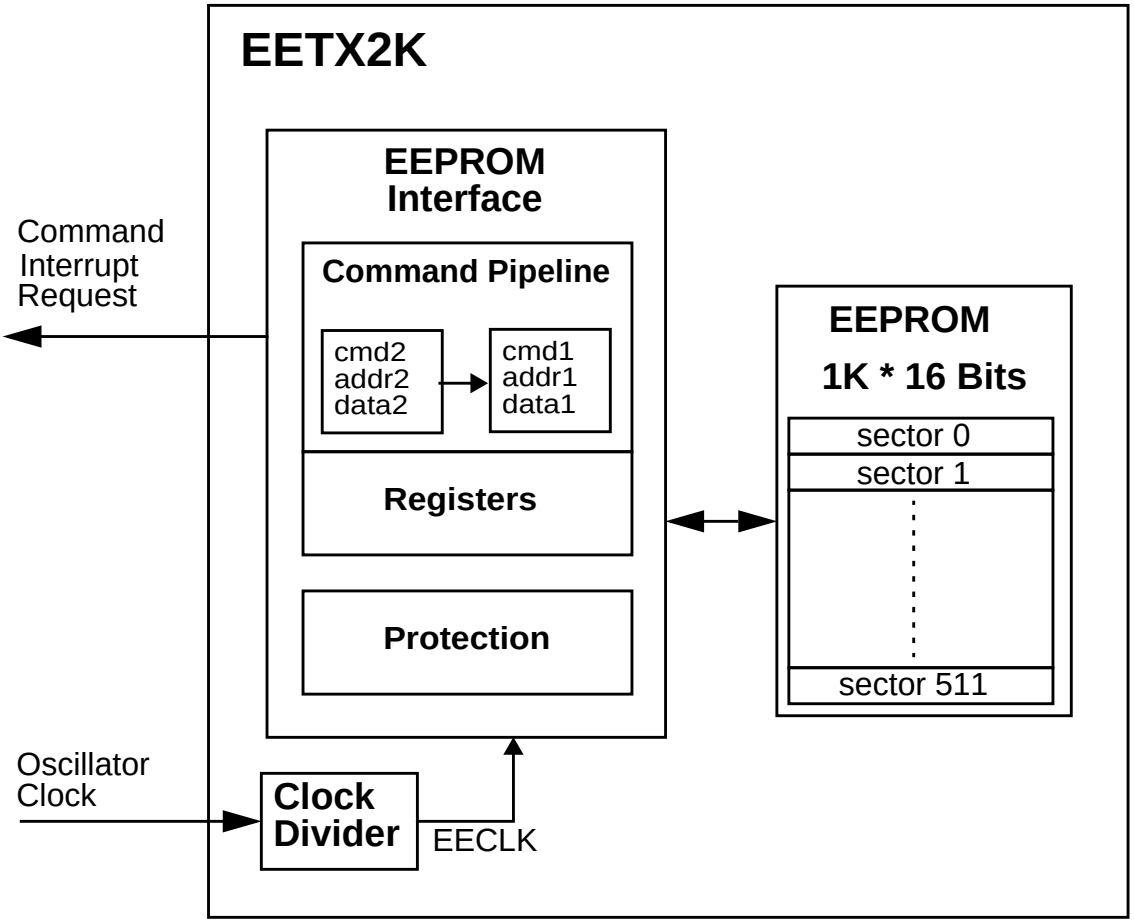

Figure 25-1. EETX2K Block Diagram

25.2 External Signal Description

The EEPROM module contains no signals that connect off-chip.

25.3 Memory Map and Register Definition

This section describes the memory map and registers for the EEPROM module.

25.3.1 Module Memory Map

The EEPROM memory map is shown in Figure 25-2. The HCS12X architecture places the EEPROM memory addresses between global addresses 0x13_F800 and 0x13_FFFF. The EPROT register, described in Section 25.3.2.5, “EEPROM Protection Register (EPROT)”, can be set to protect the upper region in the EEPROM memory from accidental program or erase. The EEPROM addresses covered by this protectable

26.5 Operating Modes

26.5.1 Wait Mode

If a command is active (CCIF = 0) when the MCU enters the wait mode, the active command and any buffered command will be completed.

The EEPROM module can recover the MCU from wait mode if the CBEIF and CCIF interrupts are enabled (see Section 26.8, “Interrupts”).

26.5.2 Stop Mode

If a command is active (CCIF = 0) when the MCU enters the stop mode, the operation will be aborted and, if the operation is program, sector erase, mass erase, or sector modify, the EEPROM array data being programmed or erased may be corrupted and the CCIF and ACCERR flags will be set. If active, the high voltage circuitry to the EEPROM memory will immediately be switched off when entering stop mode. Upon exit from stop mode, the CBEIF flag is set and any buffered command will not be launched. The ACCERR flag must be cleared before starting a command write sequence (see Section 26.4.1.2, “Command Write Sequence”).

NOTE

As active commands are immediately aborted when the MCU enters stop mode, it is strongly recommended that the user does not use the STOP instruction during program, sector erase, mass erase, or sector modify operations.

26.5.3 Background Debug Mode

In background debug mode (BDM), the EPROT register is writable. If the MCU is unsecured, then all EEPROM commands listed in Table 26-10 can be executed. If the MCU is secured and is in special single chip mode, the only command available to execute is mass erase.

26.6 EEPROM Module Security

The EEPROM module does not provide any security information to the MCU. After each reset, the security state of the MCU is a function of information provided by the Flash module (see the specific FTX Block Guide).

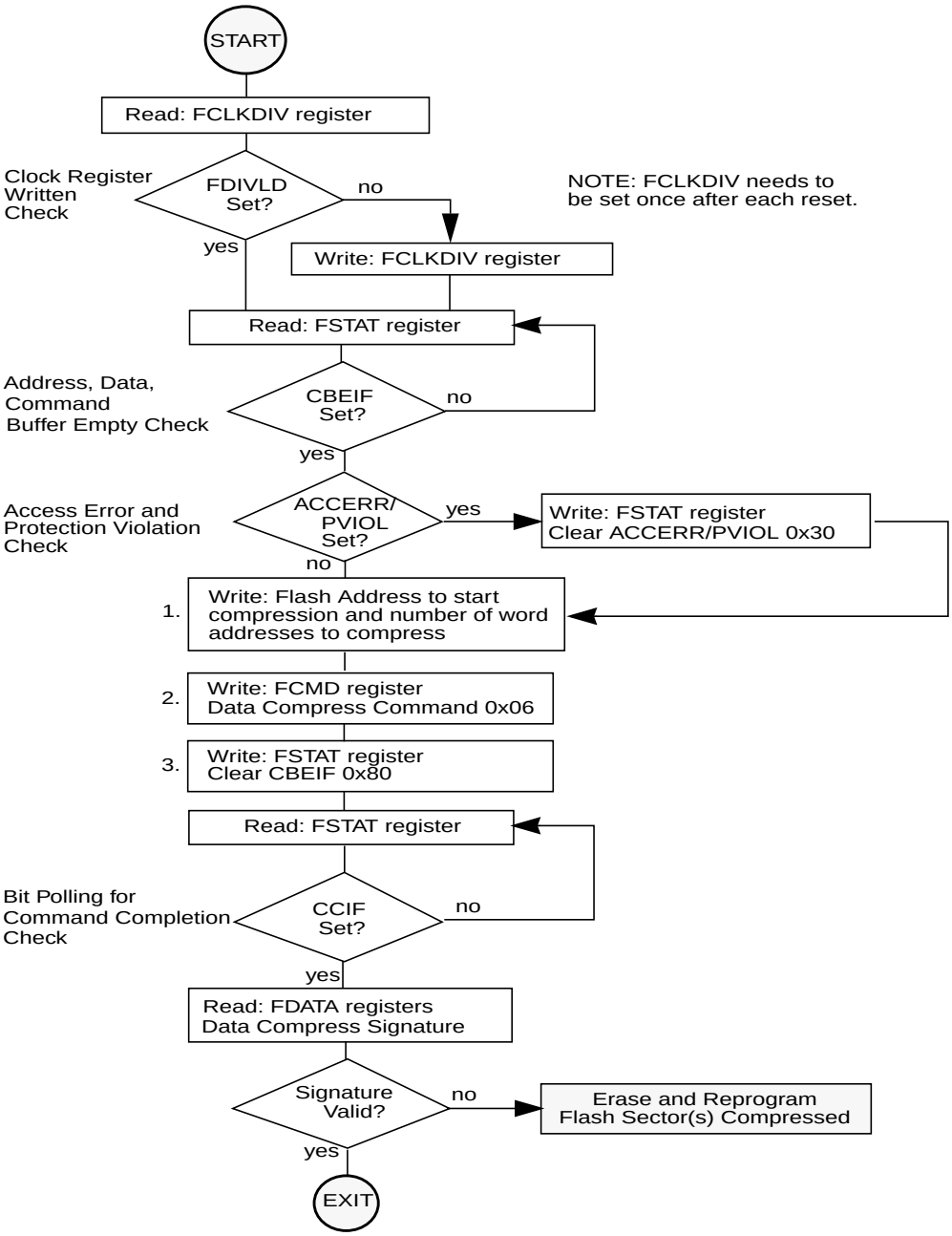

Figure 29-24. Example Data Compress Command Flow