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	32K 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/mc9s12xdp512mag

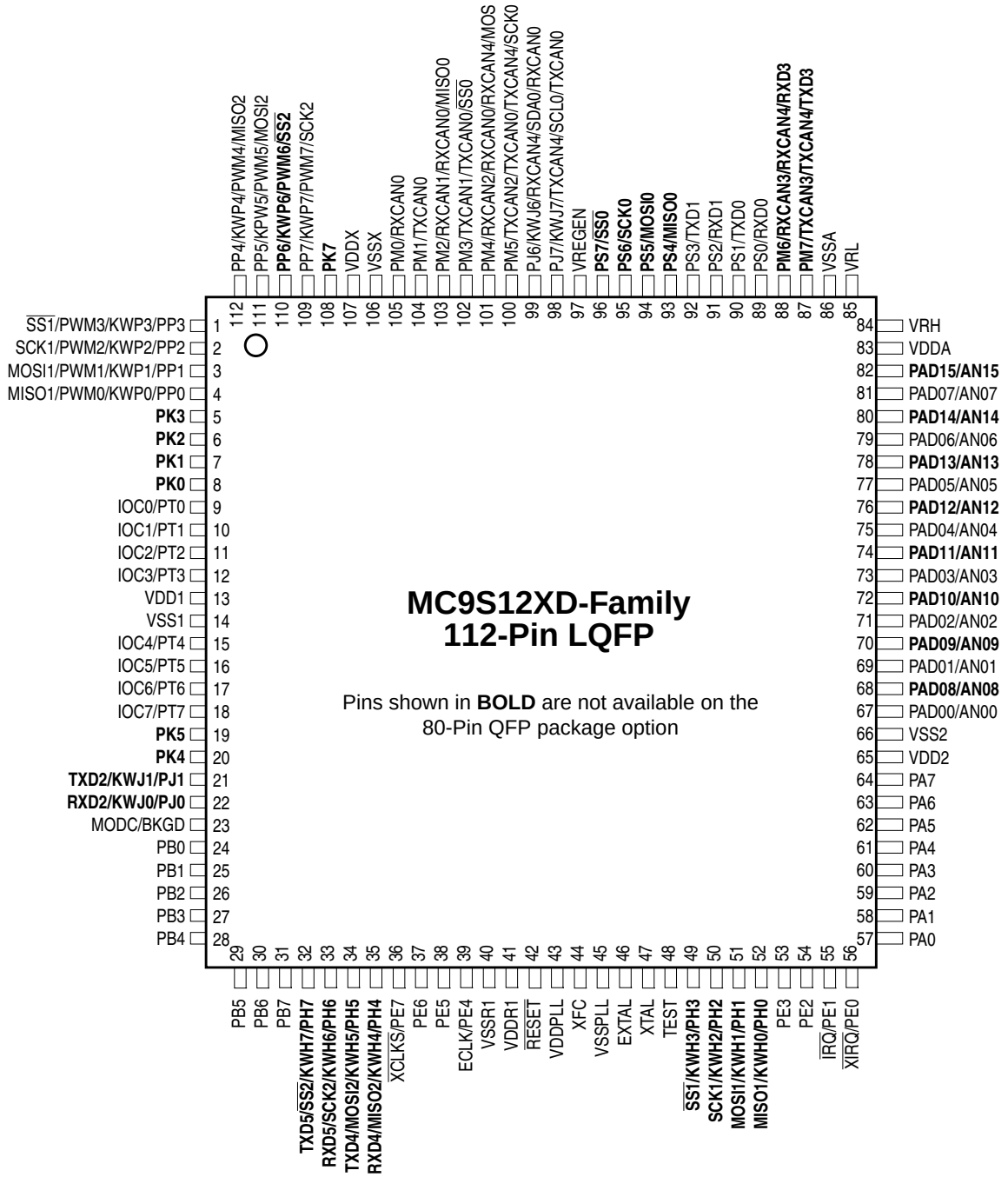

Figure 1-8. MC9S12XD Family Pin Assignments 112-Pin LQFP Package

1.8 ATD0 External Trigger Input Connection

The ATD_10B8C module includes four external trigger inputs ETRIG0, ETRIG1, ETRIG, and ETRIG3. The external trigger allows the user to synchronize ATD conversion to external trigger events. Table 1-15 shows the connection of the external trigger inputs on MC9S12XDP512RMV2.

Table 1-15. ATD0 External Trigger Sources

External Trigger Input	Connected to . .
ETRIG0	Pulse width modulator channel 1
ETRIG1	Pulse width modulator channel 3
ETRIG2	Periodic interrupt timer hardware trigger0 PITTRIG[0].
ETRIG3	Periodic interrupt timer hardware trigger1 PITTRIG[1].

See Section Chapter 5, “Analog-to-Digital Converter (S12ATD10B8CV2) for information about the analog-to-digital converter module. When this section refers to freeze mode this is equivalent to active BDM mode.

1.9 ATD1 External Trigger Input Connection

The ATD_10B16C module includes four external trigger inputs ETRIG0, ETRIG1, ETRIG, and ETRIG3. The external trigger feature allows the user to synchronize ATD conversion to external trigger events. Table 1-16 shows the connection of the external trigger inputs on MC9S12XDP512RMV2.

Table 1-16. ATD1 External Trigger Sources

External Trigger Input	Connected to . .
ETRIG0	Pulse width modulator channel 1
ETRIG1	Pulse width modulator channel 3
ETRIG2	Periodic interrupt timer hardware trigger0 PITTRIG[0].
ETRIG3	Periodic interrupt timer hardware trigger1 PITTRIG[1].

See Section Chapter 4, “Analog-to-Digital Converter (ATD10B16CV4) Block Description for information about the analog-to-digital converter module. When this section refers to freeze mode this is equivalent to active BDM mode.

2.3.2.12 CRG COP Timer Arm/Reset Register (ARMCOP)

This register is used to restart the COP time-out period.

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

Figure 2-15. ARMCOP Register Diagram

Read: Always reads 0x_00

Write: Anytime

When the COP is disabled (CR[2:0] = “000”) writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

Writing any value other than 0x_55 or 0x_AA causes a COP reset. To restart the COP time-out period you must write 0x_55 followed by a write of 0x_AA. Other instructions may be executed between these writes but the sequence (0x_55, 0x_AA) must be completed prior to COP end of time-out period to avoid a COP reset. Sequences of 0x_55 writes or sequences of 0x_AA writes are allowed. When the WCOP bit is set, 0x_55 and 0x_AA writes must be done in the last 25% of the selected time-out period; writing any value in the first 75% of the selected period will cause a COP reset.

6.8.1.8 Dyadic Addressing (DYA)

In this mode the result of an operation between two registers is stored in one of the registers used as operands.

RD = RD ∗ RS is the general register to register format, with register RD being the first operand and RS the second. RD and RS can be any of the 8 general purpose registers R0 … R7. If R0 is used as the destination register, only the condition code flags are updated. This addressing mode is used only for shift operations with a variable shift value

Examples:

```
LSL     R4,R5    ; R4 = R4 << R5
LSR     R4,R5    ; R4 = R4 >> R5
```

6.8.1.9 Triadic Addressing (TRI)

In this mode the result of an operation between two or three registers is stored into a third one.
RD = RS1 ∗ RS2 is the general format used in the order RD, RS1, RS1. RD, RS1, RS2 can be any of the 8 general purpose registers R0 … R7. If R0 is used as the destination register RD, only the condition code flags are updated. This addressing mode is used for all arithmetic and logical operations.

Examples:

```
ADC     R5,R6,R7          ; R5 = R6 + R7 + Carry
SUB     R5,R6,R7          ; R5 = R6 - R7
```

6.8.1.10 Relative Addressing 9-Bit Wide (REL9)

A 9-bit signed word address offset is included in the instruction word. This addressing mode is used for conditional branch instructions.

Examples:

```
BCC     REL9              ; PC = PC + 2 + (REL9 << 1)
BEQ     REL9              ; PC = PC + 2 + (REL9 << 1)
```

6.8.1.11 Relative Addressing 10-Bit Wide (REL10)

An 11-bit signed word address offset is included in the instruction word. This addressing mode is used for the unconditional branch instruction.

Examples:

```
BRA     REL10             ; PC = PC + 2 + (REL10 << 1)
```

6.8.1.12 Index Register plus Immediate Offset (IDO5)

(RS, #offset5) provides an unsigned offset from the base register.

Examples:

```
LDB     R4,(R1,#offset)   ; loads a byte from R1+offset into R4
STW     R4,(R1,#offset)   ; stores R4 as a word to R1+offset
```

7.3.2.11 Timer System Control Register 2 (TSCR2)

	7	6	5	4	3	2	1	0
R	TOI	0	0	0	TCRE	PR2	PR1	PR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-16. Timer System Control Register 2 (TSCR2)

Read or write: Anytime

All bits reset to zero.

Table 7-14. TSCR2 Field Descriptions

Field	Description
7 TOI	**Timer Overflow Interrupt Enable** 0 Timer overflow interrupt disabled. 1 Hardware interrupt requested when TOF flag set.
3 TCRE	**Timer Counter Reset Enable** — This bit allows the timer counter to be reset by a successful channel 7 output compare. This mode of operation is similar to an up-counting modulus counter. 0 Counter reset disabled and counter free runs. 1 Counter reset by a successful output compare on channel 7. **Note:** If register TC7 = 0x0000 and TCRE = 1, then the TCNT register will stay at 0x0000 continuously. If register TC7 = 0xFFFF and TCRE = 1, the TOF flag will never be set when TCNT is reset from 0xFFFF to 0x0000.
2:0 PR[2:0]	**Timer Prescaler Select** — These three bits specify the division rate of the main Timer prescaler when the PRNT bit of register TSCR1 is set to 0. The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero. See Table 7-15.

Table 7-15. Prescaler Selection

PR2	PR1	PR0	Prescale Factor
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

NOTE

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

See Section 8.4.2.3, “PWM Period and Duty” for more information.

NOTE

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

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

- Polarity = 0 (PPOL x =0)

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

  Duty Cycle = [PWMDTYx / PWMPERx] * 100%

For boundary case programming values, please refer to Section 8.4.2.8, “PWM Boundary Cases”.

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

Figure 8-16. PWM Channel Duty Registers (PWMDTYx)

Read: Anytime

Write: Anytime

8.3.2.15 PWM Shutdown Register (PWMSDN)

The PWMSDN register provides for the shutdown functionality of the PWM module in the emergency cases. For proper operation, channel 7 must be driven to the active level for a minimum of two bus clocks.

	7	6	5	4	3	2	1	0
R	PWMIF	PWMIE	0	PWMLVL	0	PWM7IN	PWM7INL	PWM7ENA
W			PWMRSTRT					
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-17. PWM Shutdown Register (PWMSDN)

Read: Anytime

Write: Anytime

14.3.2.3 Autonomous Periodical Interrupt Control Register (VREGAPICL)

The VREGAPICL register allows the configuration of the VREG_3V3 autonomous periodical interrupt features.

	7	6	5	4	3	2	1	0
R	APICLK	0	0	0	0	APIFE	APIE	APIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-4. Autonomous Periodical Interrupt Control Register (VREGAPICL)

Table 14-4. VREGAPICL Field Descriptions

Field	Description
7 APICLK	**Autonomous Periodical Interrupt Clock Select Bit** — Selects the clock source for the API. Writable only if APIFE = 0; APICLK cannot be changed if APIFE is set by the same write operation. 0 Autonomous periodical interrupt clock used as source. 1 Bus clock used as source.
2 APIFE	**Autonomous Periodical Interrupt Feature Enable Bit** — Enables the API feature and starts the API timer when set. 0 Autonomous periodical interrupt is disabled. 1 Autonomous periodical interrupt is enabled and timer starts running.
1 APIE	**Autonomous Periodical Interrupt Enable Bit** 0 API interrupt request is disabled. 1 API interrupt will be requested whenever APIF is set.
0 APIF	**Autonomous Periodical Interrupt Flag** — APIF is set to 1 when the in the API configured time has elapsed. This flag can only be cleared by writing a 1 to it. Clearing of the flag has precedence over setting. Writing a 0 has no effect. If enabled (APIE = 1), APIF causes an interrupt request. 0 API timeout has not yet occurred. 1 API timeout has occurred.

- Software control of BDM operation during wait mode
- Software selectable clocks
- Global page access functionality
- Enabled but not active out of reset in emulation modes
- CLKSW bit set out of reset in emulation mode.
- When secured, hardware commands are allowed to access the register space in special single chip mode, if the Flash and EEPROM erase tests fail.
- Family ID readable from firmware ROM at global address 0x7FFF0F (value for HCS12X devices is 0xC1)
- BDM hardware commands are operational until system stop mode is entered (all bus masters are in stop mode)

15.1.2 Modes of Operation

BDM is available in all operating modes but must be enabled before firmware commands are executed. Some systems may have a control bit that allows suspending the function during background debug mode.

15.1.2.1 Regular Run Modes

All of these operations refer to the part in run mode and not being secured. The BDM does not provide controls to conserve power during run mode.

- Normal modes

  General operation of the BDM is available and operates the same in all normal modes.
- Special single chip mode

  In special single chip mode, background operation is enabled and active out of reset. This allows programming a system with blank memory.
- Emulation modes

  In emulation mode, background operation is enabled but not active out of reset. This allows debugging and programming a system in this mode more easily.

15.1.2.2 Secure Mode Operation

If the device is in secure mode, the operation of the BDM is reduced to a small subset of its regular run mode operation. Secure operation prevents access to Flash or EEPROM other than allowing erasure. For more information please see Section 15.4.1, “Security”.

16.1.4 Block Diagram

Figure 16-1 shows a block diagram of the XINT module.

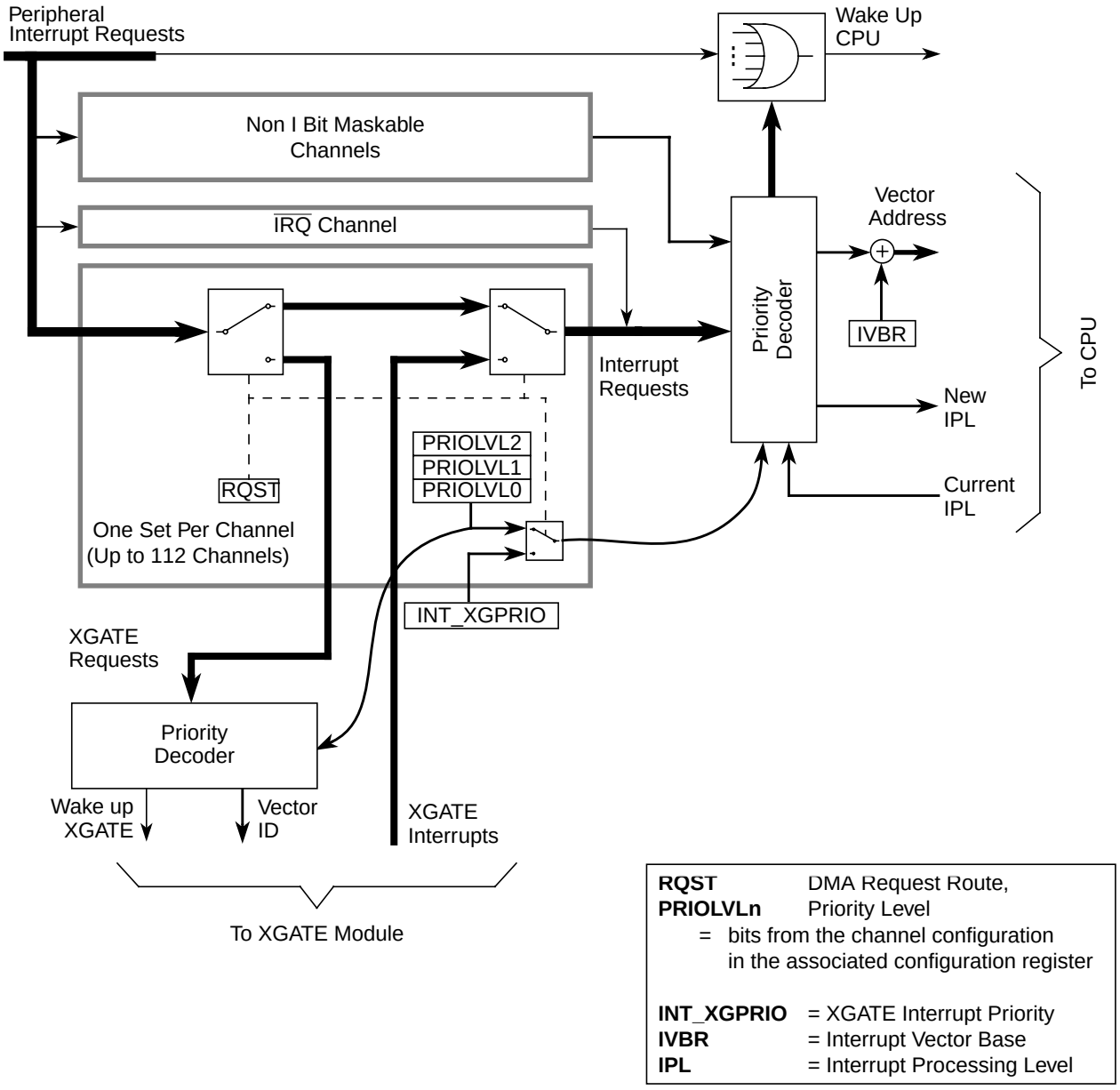

Figure 16-1. XINT Block Diagram

18.3.2.2 Mode Register (MODE)

Address: 0x000B PRR

	7	6	5	4	3	2	1	0
R	MODC	MODB	MODA	0	0	0	0	0
W								
Reset	MODC[1]	MODB[1]	MODA[1]	0	0	0	0	0

1. External signal (see Table 18-2).

= Unimplemented or Reserved

Figure 18-4. Mode Register (MODE)

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

Write: Only if a transition is allowed (see Figure 18-5). In emulation modes write operations will be also directed to the external bus.

The MODE bits of the MODE register are used to establish the MCU operating mode.

CAUTION

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

Table 18-7. MODE Field Descriptions

Field	Description
7–5 MODC, MODB, MODA	**Mode Select Bits** — These bits control the current operating mode during $\overline{\text{RESET}}$ high (inactive). The external mode pins MODC, MODB, and MODA determine the operating mode during $\overline{\text{RESET}}$ low (active). The state of the pins is latched into the respective register bits after the $\overline{\text{RESET}}$ signal goes inactive (see Figure 18-5). Write restrictions exist to disallow transitions between certain modes. Figure 18-5 illustrates all allowed mode changes. Attempting non authorized transitions will not change the MODE bits, but it will block further writes to these register bits except in special modes. Both transitions from normal single-chip mode to normal expanded mode and from emulation single-chip to emulation expanded mode are only executed by writing a value of 3'b101 (write once). Writing any other value will not change the MODE bits, but will block further writes to these register bits. Changes of operating modes are not allowed when the device is secured, but it will block further writes to these register bits except in special modes. In emulation modes reading this address returns data from the external bus which has to be driven by the emulator. It is therefore responsibility of the emulator hardware to provide the expected value (i.e. a value corresponding to normal single chip mode while the device is in emulation single-chip mode or a value corresponding to normal expanded mode while the device is in emulation expanded mode).

Table 23-2. PIM Memory Map (Sheet 3 of 3)

Address	Use	Access
0x026B	Port J Reduced Drive Register (RDRJ)	Read / Write[1]
0x026C	Port J Pull Device Enable Register (PERJ)	Read / Write[1]
0x026D	Port J Polarity Select Register (PPSJ)	Read / Write[1]
0x026E	Port J Interrupt Enable Register (PIEJ)	Read / Write[1]
0x026F	Port J Interrupt Flag Register (PIFJ)	Read / Write[1]
0x0270	PIM Reserved	—
0x0271	Port AD0 Data Register 1 (PT1AD0)	Read / Write
0x0272	PIM Reserved	—
0x0273	Port AD0 Data Direction Register 1 (DDR1AD0)	Read / Write
0x0274	PIM Reserved	—
0x0275	Port AD0 Reduced Drive Register 1 (RDR1AD0)	Read / Write
0x0276	PIM Reserved	—
0x0277	Port AD0 Pull Up Enable Register 1 (PER1AD0)	Read / Write
0x0278	Port AD1 Data Register 0 (PT0AD1)	Read / Write
0x0279	Port AD1 Data Register 1 (PT1AD1)	Read / Write
0x027A	Port AD1 Data Direction Register 0 (DDR0AD1)	Read / Write
0x027B	Port AD1 Data Direction Register 1 (DDR1AD1)	Read / Write
0x027C	Port AD1 Reduced Drive Register 0 (RDR0AD1)	Read / Write
0x027D	Port AD1 Reduced Drive Register 1 (RDR1AD1)	Read / Write
0x027E	Port AD1 Pull Up Enable Register 0 (PER0AD1)	Read / Write
0x027F	Port AD1 Pull Up Enable Register 1 (PER1AD1)	Read / Write

1. Write access not applicable for one or more register bits. Refer to Section 23.0.5, “Register Descriptions”.

23.0.5 Register Descriptions

Table 23-3 summarizes the effect on the various configuration bits, data direction (DDR), output level (IO), reduced drive (RDR), pull enable (PE), pull select (PS), and interrupt enable (IE) for the ports.

The configuration bit PS is used for two purposes:

1. Configure the sensitive interrupt edge (rising or falling), if interrupt is enabled.
2. Select either a pull-up or pull-down device if PE is active.

Table 23-16. ECLKCTL Field Descriptions (continued)

Field	Description
6 NCLKX2	**No ECLKX2** — This bit controls the availability of a free-running clock on the ECLKX2 pin. This clock has a fixed rate of twice the internal bus clock. Clock output is always active in emulation modes and if enabled in all other operating modes. 0 ECLKX2 is enabled 1 ECLKX2 is disabled
1–0 EDIV[1:0]	**Free-Running ECLK Divider** — These bits determine the rate of the free-running clock on the ECLK pin. The usage of the bits is shown in Table 23-17. Divider is always disabled in emulation modes and active as programmed in all other operating modes.

Table 23-17. Free-Running ECLK Clock Rate

EDIV[1:0]	Rate of Free-Running ECLK
00	ECLK = Bus clock rate
01	ECLK = Bus clock rate divided by 2
10	ECLK = Bus clock rate divided by 3
11	ECLK = Bus clock rate divided by 4

23.0.5.14 IRQ Control Register (IRQCR)

	7	6	5	4	3	2	1	0
R	IRQE	IRQEN	0	0	0	0	0	0
W								
Reset	0	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 23-16. IRQ Control Register (IRQCR)

Read: See individual bit descriptions below.

Write: See individual bit descriptions below.

Table 23-18. IRQCR Field Descriptions

Field	Description
7 IRQE	**IRQ Select Edge Sensitive Only** Special modes: Read or write anytime. Normal and emulation modes: Read anytime, write once. 0 $\overline{\text{IRQ}}$ configured for low level recognition. 1 $\overline{\text{IRQ}}$ configured to respond only to falling edges. Falling edges on the $\overline{\text{IRQ}}$ pin will be detected anytime IRQE = 1 and will be cleared only upon a reset or the servicing of the $\overline{\text{IRQ}}$ interrupt.
6 IRQEN	**External IRQ Enable** Read or write anytime. 0 External $\overline{\text{IRQ}}$ pin is disconnected from interrupt logic. 1 External $\overline{\text{IRQ}}$ pin is connected to interrupt logic.

Table 24-1. Pin Functions and Priorities (Sheet 3 of 5)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
M	PM7	TXCAN4	O	MSCAN4 transmit pin	GPIO
		GPIO	I/O	General-purpose I/O	
	PM6	RXCAN4	I	MSCAN4 receive pin	
		GPIO	I/O	General-purpose I/O	
	PM5	TXCAN0	O	MSCAN0 transmit pin	
		TXCAN4	O	MSCAN4 transmit pin	
		SCK0	I/O	Serial Peripheral Interface 0 serial clock pin *If CAN0 is routed to PM[3:2] the SPI0 can still be used in bidirectional master mode.*	
		GPIO	I/O	General-purpose I/O	
	PM4	RXCAN0	I	MSCAN0 receive pin	
		RXCAN4	I	MSCAN4 receive pin	
		MOSI0	I/O	Serial Peripheral Interface 0 master out/slave in pin *If CAN0 is routed to PM[3:2] the SPI0 can still be used in bidirectional master mode.*	
		GPIO	I/O	General-purpose I/O	
	PM3	TXCAN0	O	MSCAN0 transmit pin	
		$\overline{SS0}$	I/O	Serial Peripheral Interface 0 slave select output in master mode, input for slave mode or master mode.	
		GPIO	I/O	General-purpose I/O	
	PM2	RXCAN0	I	MSCAN0 receive pin	
		MISO0	I/O	Serial Peripheral Interface 0 master in/slave out pin	
		GPIO	I/O	General-purpose I/O	
	PM1	TXCAN0	O	MSCAN0 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PM0	RXCAN0	I	MSCAN0 receive pin	
		GPIO	I/O	General-purpose I/O	

24.0.5.9 ECLK Control Register (ECLKCTL)

	7	6	5	4	3	2	1	0	
R	NECLK	NCLKX2	0	0	0	0	EDIV1	EDIV0	
W									
Reset[1]	Mode Dependent	1	0	0	0	0	0	0	Mode
SS	0	1	0	0	0	0	0	0	Special Single-Chip
ES	1	1	0	0	0	0	0	0	Emulation Single-Chip
ST	0	1	0	0	0	0	0	0	Special Test
EX	0	1	0	0	0	0	0	0	Emulation Expanded
NS	1	1	0	0	0	0	0	0	Normal Single-Chip
NX	0	1	0	0	0	0	0	0	Normal Expanded

= Unimplemented or Reserved

Figure 24-11. ECLK Control Register (ECLKCTL)

1. Reset values in emulation modes are identical to those of the target mode.

Read: Anytime.

Write: Anytime.

The ECLKCTL register is used to control the availability of the free-running clocks and the free-running clock divider.

Table 24-12. ECLKCTL Field Descriptions

Field	Description
7 NECLK	**No ECLK** — This bit controls the availability of a free-running clock on the ECLK pin. Clock output is always active in emulation modes and if enabled in all other operating modes. 0 ECLK enabled 1 ECLK disabled
6 NCLKX2	**No ECLKX2** — This bit controls the availability of a free-running clock on the ECLKX2 pin. This clock has a fixed rate of twice the internal bus clock. Clock output is always active in emulation modes and if enabled in all other operating modes. 0 ECLKX2 is enabled 1 ECLKX2 is disabled
1–0 EDIV[1:0]	**Free-Running ECLK Divider** — These bits determine the rate of the free-running clock on the ECLK pin. The usage of the bits is shown in Table 24-13. Divider is always disabled in emulation modes and active as programmed in all other operating modes.

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

Table 24-21. PERT Field Descriptions

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

24.0.5.18 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 24-20. 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 24-22. 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.

24.0.5.19 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 24-21. Port S Data Register (PTS)

Read: Anytime.

Write: Anytime.

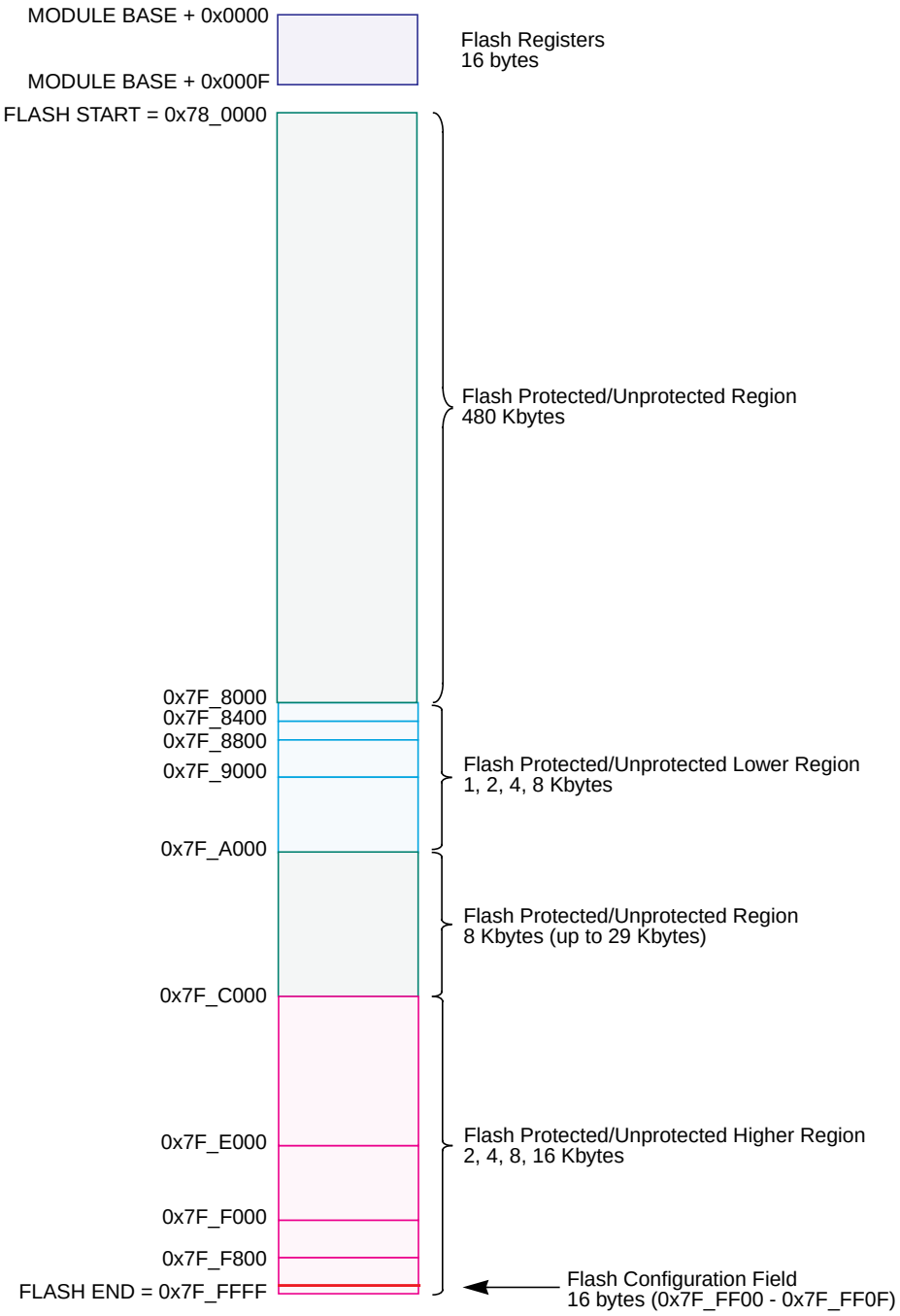

Figure 27-2. Flash Memory Map

28.4.2.1 Erase Verify Command

The erase verify operation will verify that a Flash block is erased.

An example flow to execute the erase verify operation is shown in Figure 28-25. The erase verify command write sequence is as follows:

1. Write to a Flash block address to start the command write sequence for the erase verify command. The address and data written will be ignored. Multiple Flash blocks can be simultaneously erase verified by writing to the same relative address in each Flash block.
2. Write the erase verify command, 0x05, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the erase verify command.

After launching the erase verify command, the CCIF flag in the FSTAT register will set after the operation has completed unless a new command write sequence has been buffered. The number of bus cycles required to execute the erase verify operation is equal to the number of addresses in a Flash block plus 14 bus cycles as measured from the time the CBEIF flag is cleared until the CCIF flag is set. Upon completion of the erase verify operation, the BLANK flag in the FSTAT register will be set if all addresses in the selected Flash blocks are verified to be erased. If any address in a selected Flash block is not erased, the erase verify operation will terminate and the BLANK flag in the FSTAT register will remain clear. The MRDS bits in the FTSTMOD register will determine the sense-amp margin setting during the erase verify operation.

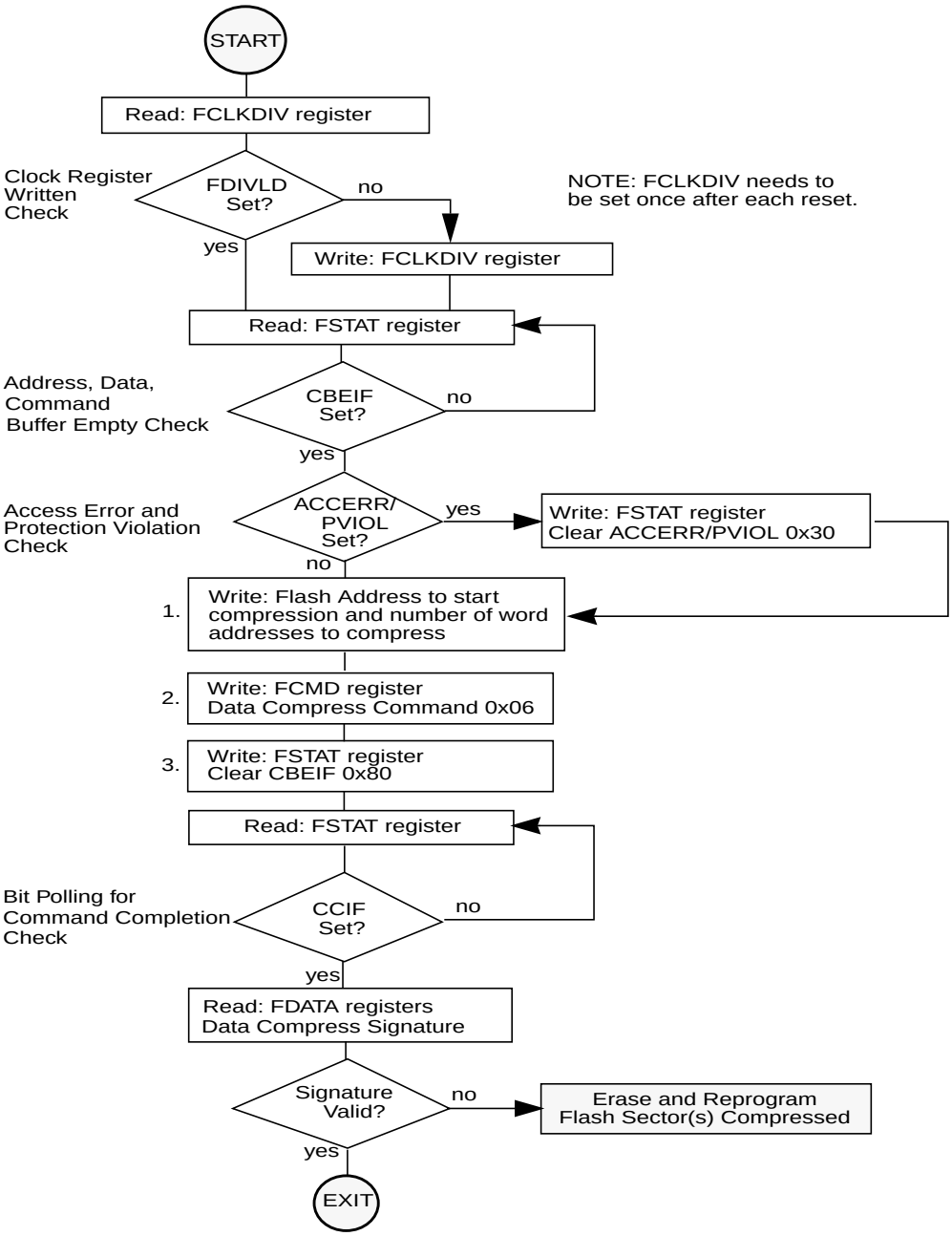

Figure 29-24. Example Data Compress Command Flow

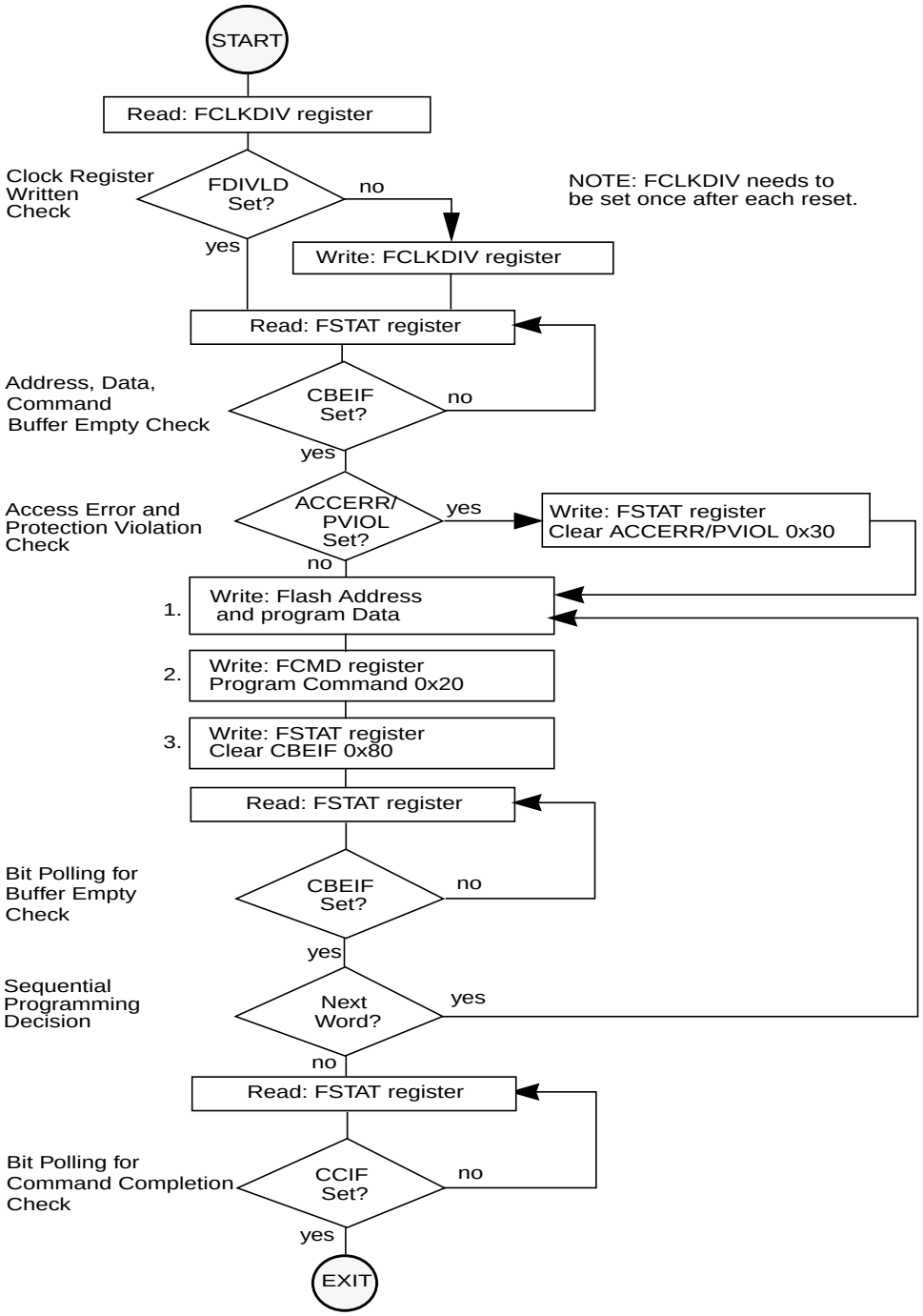

Figure 29-26. Example Program Command Flow