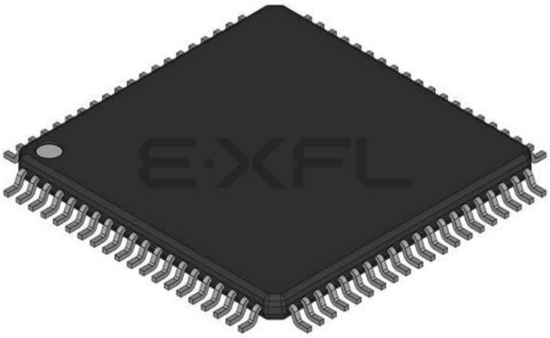

Welcome to E-XFL.COM

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	59
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	14K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12xd256maa

1.2.3.3 TEST — Test Pin

This input only pin is reserved for test. This pin has a pulldown device.

NOTE

The TEST pin must be tied to V_{SS} in all applications.

1.2.3.4 VREGEN — Voltage Regulator Enable Pin

This input only pin enables or disables the on-chip voltage regulator. The input has a pullup device.

1.2.3.5 XFC — PLL Loop Filter Pin

Please ask your Freescale representative for the interactive application note to compute PLL loop filter elements. Any current leakage on this pin must be avoided.

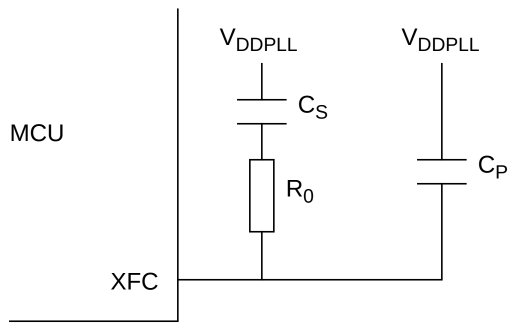

Figure 1-10. PLL Loop Filter Connections

1.2.3.6 BKGD / MODC — Background Debug and Mode Pin

The BKGD/MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The BKGD pin has a pullup device.

1.2.3.7 PAD[23:8] / AN[23:8] — Port AD Input Pins of ATD1

PAD[23:8] are general-purpose input or output pins and analog inputs AN[23:8] of the analog-to-digital converter ATD1.

1.2.3.8 PAD[7:0] / AN[7:0] — Port AD Input Pins of ATD0

PAD[7:0] are general-purpose input or output pins and analog inputs AN[7:0] of the analog-to-digital converter ATD0.

1.2.3.9 PAD[15:0] / AN[15:0] — Port AD Input Pins of ATD1

PAD[15:0] are general-purpose input or output pins and analog inputs AN[15:0] of the analog-to-digital converter ATD1.

Table 2-12. Outcome of Clock Loss in Wait Mode

CME	SCME	SCMIE	CRG Actions
0	X	X	Clock failure --> No action, clock loss not detected.
1	0	X	Clock failure --> CRG performs Clock Monitor Reset immediately
1	1	0	Clock failure --> Scenario 1: OSCCLK recovers prior to exiting wait mode. – MCU remains in wait mode, – VREG enabled, – PLL enabled, – SCM activated, – Start clock quality check, – Set SCMIF interrupt flag. *Some time later OSCCLK recovers.* – CM no longer indicates a failure, – 4096 OSCCLK cycles later clock quality check indicates clock o.k., – SCM deactivated, – PLL disabled depending on PLLWAI, – VREG remains enabled *(never gets disabled in wait mode)*. – MCU remains in wait mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* – Exit wait mode using OSCCLK as system clock (SYSCLK), – Continue normal operation. *or an External Reset is applied.* – Exit wait mode using OSCCLK as system clock, – Start reset sequence. Scenario 2: OSCCLK does not recover prior to exiting wait mode. – MCU remains in wait mode, – VREG enabled, – PLL enabled, – SCM activated, – Start clock quality check, – Set SCMIF interrupt flag, – Keep performing clock quality checks (could continue infinitely) while in wait mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* – Exit wait mode in SCM using PLL clock (f_{SCM}) as system clock, – Continue to perform additional clock quality checks until OSCCLK is o.k. again. or an External RESET is applied. – Exit wait mode in SCM using PLL clock (f_{SCM}) as system clock, – Start reset sequence, – Continue to perform additional clock quality checks until OSCCLKis o.k.again.
1	1	1	Clock failure --> – VREG enabled, – PLL enabled, – SCM activated, – Start clock quality check, – SCMIF set. SCMIF generates self clock mode wakeup interrupt. – Exit wait mode in SCM using PLL clock (f_{SCM}) as system clock, – Continue to perform a additional clock quality checks until OSCCLK is o.k. again.

Table 5-7. ATDCTL3 Field Descriptions (continued)

Field	Description
2 FIFO	**Result Register FIFO Mode** — If this bit is zero (non-FIFO mode), the A/D conversion results map into the result registers based on the conversion sequence; the result of the first conversion appears in the first result register, the second result in the second result register, and so on. If this bit is one (FIFO mode) the conversion counter is not reset at the beginning or ending of a conversion sequence; sequential conversion results are placed in consecutive result registers. In a continuously scanning conversion sequence, the result register counter will wrap around when it reaches the end of the result register file. The conversion counter value (CC2-0 in ATDSTAT0) can be used to determine where in the result register file, the current conversion result will be placed. Aborting a conversion or starting a new conversion by write to an ATDCTL register (ATDCTL5-0) clears the conversion counter even if FIFO=1. So the first result of a new conversion sequence, started by writing to ATDCTL5, will always be place in the first result register (ATDDDR0). Intended usage of FIFO mode is continuos conversion (SCAN=1) or triggered conversion (ETRIG=1). Finally, which result registers hold valid data can be tracked using the conversion complete flags. Fast flag clear mode may or may not be useful in a particular application to track valid data. 0 Conversion results are placed in the corresponding result register up to the selected sequence length. 1 Conversion results are placed in consecutive result registers (wrap around at end).
1–0 FRZ[1:0]	**Background Debug Freeze Enable** — When debugging an application, it is useful in many cases to have the ATD pause when a breakpoint (Freeze Mode) is encountered. These 2 bits determine how the ATD will respond to a breakpoint as shown in Table 5-9. Leakage onto the storage node and comparator reference capacitors may compromise the accuracy of an immediately frozen conversion depending on the length of the freeze period.

Table 5-8. Conversion Sequence Length Coding

S8C	S4C	S2C	S1C	Number of Conversions per Sequence
0	0	0	0	8
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	X	X	X	8

Table 5-9. ATD Behavior in Freeze Mode (Breakpoint)

FRZ1	FRZ0	Behavior in Freeze Mode
0	0	Continue conversion
0	1	Reserved
1	0	Finish current conversion, then freeze
1	1	Freeze Immediately

NEG Two's Complement NEG

Operation

–RS ⇒ RD (translates to SUB RD, R0, RS)
–RD ⇒ RD (translates to SUB RD, R0, RD)

Performs a two's complement on a general purpose register.

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 15 of the result is set; cleared otherwise.
Z: Set if the result is $0000; cleared otherwise.
V: Set if a two´s complement overflow resulted from the operation; cleared otherwise.
RS[15] & $RD[15]_{new}$
C: Set if there is a carry from the bit 15 of the result; cleared otherwise
RS[15] | $RD[15]_{new}$

Code and CPU Cycles

Source Form	Address Mode	Machine Code											Cycles
NEG RD, RS	TRI	0	0	0	1	1	RD	0	0	0	RS	0	0	P
NEG RD	TRI	0	0	0	1	1	RD	0	0	0	RD	0	0	P

SUBH Subtract Immediate 8 bit Constant (High Byte) SUBH

Operation

RD – IMM8:$00 ⇒ RD

Subtracts a signed immediate 8 bit constant from the content of high byte of register RD and using binary subtraction and stores the result in the high byte of destination register RD. This instruction can be used after an SUBL for a 16 bit immediate subtraction.

Example:

```
SUBL     R2,#LOWBYTE
SUBH     R2,#HIGHBYTE       ; R2 = R2 - 16 bit immediate
```

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 15 of the result is set; cleared otherwise.

Z: Set if the result is $0000; cleared otherwise.

V: Set if a two´s complement overflow resulted from the operation; cleared otherwise.
$RD[15]_{old}$ & $\overline{IMM8[7]}$ & $\overline{RD[15]_{new}}$ | $\overline{RD[15]_{old}}$ & $IMM8[7]$ & $RD[15]_{new}$

C: Set if there is a carry from the bit 15 of the result; cleared otherwise.
$\overline{RD[15]_{old}}$ & $IMM8[7]$ | $\overline{RD[15]_{old}}$ & $RD[15]_{new}$ | $IMM8[7]$ & $RD[15]_{new}$

Code and CPU Cycles

Source Form	Address Mode	Machine Code							Cycles
SUBH RD, #IMM8	IMM8	1	1	0	0	1	RD	IMM8	P

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
ICSYS	R W	SH37	SH26	SH15	SH04	TFMOD	PACMX	BUFEN	LATQ
Reserved	R W	Reserved							
TIMTST	R W	Timer Test Register							
PTPSR	R W	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
PTMCPSR	R W	PTMPS7	PTMPS6	PTMPS5	PTMPS4	PTMPS3	PTMPS2	PTMPS1	PTMPS0
PBCTL	R	0	PBEN	0	0	0	0	PBOVI	0
	W								
PBFLG	R	0	0	0	0	0	0	PBOVF	0
	W								
PA3H	R	PA3H7	PA3H6	PA3H5	PA3H4	PA3H3	PA3H2	PA3H1	PA3H0
	W								
PA2H	R	PA2H7	PA2H6	PA2H5	PA2H4	PA2H3	PA2H2	PA2H1	PA2H0
	W								
PA1H	R	PA1H7	PA1H6	PA1H5	PA1H4	PA1H3	PA1H2	PA1H1	PA1H0
	W								
PA0H	R	PA0H7	PA0H6	PA0H5	PA0H4	PA0H3	PA0H2	PA0H1	PA0H0
	W								
MCCNT (High)	R W	MCCNT15	MCCNT14	MCCNT13	MCCNT12	MCCNT11	MCCNT10	MCCNT9	MCCNT8
MCCNT (Low)	R W	MCCNT7	MCCNT6	MCCNT5	MCCNT4	MCCNT3	MCCNT2	MCCNT1	MCCNT9
TC0H (High)	R	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
	W								
TC0H (Low)	R	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0

(shaded) = Unimplemented or Reserved

Figure 7-2. ECT Register Summary (Sheet 4 of 5)

	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 7-63. Timer Input Capture Holding Register 3 High (TC3H)

	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 7-64. Timer Input Capture Holding Register 3 Low (TC3H)

Read: Anytime

Write: Has no effect.

All bits reset to zero.

These registers are used to latch the value of the input capture registers TC0–TC3. The corresponding IOSx bits in TIOS should be cleared (see Section 7.4.1.1, “IC Channels”).

7.4 Functional Description

This section provides a complete functional description of the ECT block, detailing the operation of the design from the end user perspective in a number of subsections.

9.5 Resets

The reset state of each individual bit is listed in Section 9.3, “Memory Map and Register Definition,” which details the registers and their bit-fields.

9.6 Interrupts

IICV2 uses only one interrupt vector.

Table 9-8. Interrupt Summary

Interrupt	Offset	Vector	Priority	Source	Description
IIC Interrupt	—	—	—	IBAL, TCF, IAAS bits in IBSR register	When either of IBAL, TCF or IAAS bits is set may cause an interrupt based on arbitration lost, transfer complete or address detect conditions

Internally there are three types of interrupts in IIC. The interrupt service routine can determine the interrupt type by reading the status register.

IIC Interrupt can be generated on

1. Arbitration lost condition (IBAL bit set)
2. Byte transfer condition (TCF bit set)
3. Address detect condition (IAAS bit set)

The IIC interrupt is enabled by the IBIE bit in the IIC control register. It must be cleared by writing 0 to the IBF bit in the interrupt service routine.

9.7 Initialization/Application Information

9.7.1 IIC Programming Examples

9.7.1.1 Initialization Sequence

Reset will put the IIC bus control register to its default status. Before the interface can be used to transfer serial data, an initialization procedure must be carried out, as follows:

1. Update the frequency divider register (IBFD) and select the required division ratio to obtain SCL frequency from system clock.
2. Update the IIC bus address register (IBAD) to define its slave address.
3. Set the IBEN bit of the IIC bus control register (IBCR) to enable the IIC interface system.
4. Modify the bits of the IIC bus control register (IBCR) to select master/slave mode, transmit/receive mode and interrupt enable or not.

12.3.2.4 SPI Status Register (SPISR)

	7	6	5	4	3	2	1	0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W								
Reset	0	0	1	0	0	0	0	0

= Unimplemented or Reserved

Figure 12-6. SPI Status Register (SPISR)

Read: Anytime

Write: Has no effect

Table 12-7. SPISR Field Descriptions

Field	Description
7 SPIF	**SPIF Interrupt Flag** — This bit is set after a received data byte has been transferred into the SPI data register. This bit is cleared by reading the SPISR register (with SPIF set) followed by a read access to the SPI data register. 0 Transfer not yet complete. 1 New data copied to SPIDR.
5 SPTEF	**SPI Transmit Empty Interrupt Flag** — If set, this bit indicates that the transmit data register is empty. To clear this bit and place data into the transmit data register, SPISR must be read with SPTEF = 1, followed by a write to SPIDR. Any write to the SPI data register without reading SPTEF = 1, is effectively ignored. 0 SPI data register not empty. 1 SPI data register empty.
4 MODF	**Mode Fault Flag** — This bit is set if the $\overline{SS}$ input becomes low while the SPI is configured as a master and mode fault detection is enabled, MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in Section 12.3.2.2, “SPI Control Register 2 (SPICR2)”. The flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to the SPI control register 1. 0 Mode fault has not occurred. 1 Mode fault has occurred.

17.4.3 Chip Access Restrictions

17.4.3.1 Illegal XGATE Accesses

A possible access error is flagged by the MMC and signalled to XGATE under the following conditions:

- XGATE performs misaligned word (in case of load-store or opcode or vector fetch accesses).
- XGATE accesses the register space (in case of opcode or vector fetch).
- XGATE performs a write to Flash in any modes (in case of load-store access).
- XGATE performs an access to a secured Flash in expanded modes (in case of load-store or opcode or vector fetch accesses).
- XGATE performs a write to non-XGATE region in RAM (RAM protection mechanism) (in case of load-store access).

For further details refer to the XGATE Block Guide.

17.4.3.2 Illegal CPU Accesses

After programming the protection mechanism registers (see Figure 1-17, Figure 1-18, Figure 1-19, and Figure 1-20) and setting the RWPE bit (see Figure 1-17) there are 3 regions recognized by the MMC module:

1. XGATE RAM region
2. CPU RAM region
3. Shared Region (XGATE AND CPU)

If the RWPE bit is set the CPU write accesses into the XGATE RAM region are blocked. If the CPU tries to write the XGATE RAM region the AVIF bit is set and an interrupt is generated if enabled. Furthermore if the XGATE tries to write to outside of the XGATE RAM or shared regions and the RWPE bit is set, the write access is suppressed and the access error will be flagged to the XGATE module (see Section 1.4.3.1, "Illegal XGATE Accesses" and the XGATE Block Guide).

The bottom address of the XGATE RAM region always starts at the lowest implemented RAM address.

The values stored in the boundary registers define the boundary addresses in 256 byte steps. The 256 byte block selected by any of the registers is always included in the respective region. For example setting the shared region lower boundary register (RAMSHL) to $C1 and the shared region upper boundary register (RAMSHU) to $E0 defines the shared region from address $0F_C100 to address $0F_E0FF in the global memory space (see Figure 1-25).

The interrupt requests generated by the MMC are listed in Table 1-23. Refer to the Device User Guide for the related interrupt vector address and interrupt priority.

Table 19-14. CDCM Encoding

CDCM	Description
00	Match2 mapped to comparator C match....... Match3 mapped to comparator D match.
01	Match2 mapped to comparator C/D inside range....... Match3 disabled.
10	Match2 mapped to comparator C/D outside range....... Match3 disabled.
11	Reserved

Table 19-15. ABCM Encoding

ABCM	Description
00	Match0 mapped to comparator A match....... Match1 mapped to comparator B match.
01	Match 0 mapped to comparator A/B inside range....... Match1 disabled.
10	Match 0 mapped to comparator A/B outside range....... Match1 disabled.
11	Reserved

19.3.1.5 Debug Trace Buffer Register (DBGTBH:DBGTBL)

0x0024

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

Figure 19-7. Debug Trace Buffer Register (DBGTBH)

0x0025

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

Figure 19-8. Debug Trace Buffer Register (DBGTBL)

Read: Anytime when unlocked and not secured and not armed.

Write: Aligned word writes when disarmed unlock the trace buffer for reading but do not affect trace buffer contents

Table 19-16. DBGTB Field Descriptions

Field	Description
15–0 Bit[15:0]	**Trace Buffer Data Bits** — The trace buffer register is a window through which the 64-bit wide data lines of the trace buffer may be read 16 bits at a time. Each valid read of DBGTB increments an internal trace buffer pointer which points to the next address to be read. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by writing to DBGTB with an aligned word write when the module is disarmed. The DBGTB register can be read only as an aligned word, any byte reads or misaligned access of these registers will return 0 and will not cause the trace buffer pointer to increment to the next trace buffer address. The same is true for word reads while the debugger is armed. System resets do not affect the trace buffer contents. The POR state is undefined.

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.

21.3.2.2 External Bus Interface Control Register 1 (EBICTL1)

	7	6	5	4	3	2	1	0
R	EWAITE	0	0	0	0	EXSTR2	EXSTR1	EXSTR0
W								
Reset	0	0	0	0	0	1	1	1

= Unimplemented or Reserved

Figure 21-4. External Bus Interface Control Register 1 (EBICTL1)

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

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

This register is used to configure the external access stretch (wait) function.

Table 21-5. EBICTL1 Field Descriptions

Field	Description
7 EWAITE	**External Wait Enable** — This bit enables the external access stretch function using the external $\overline{\text{EWAIT}}$ input pin. Enabling this feature may have effect on the minimum number of additional stretch cycles (refer to Table 21-6). External wait feature is only active if enabled in normal expanded mode and emulation expanded mode; function not available in all other operating modes. 0 External wait is disabled 1 External wait is enabled
2–0 EXSTR[2:0]	**External Access Stretch Bits 2, 1, 0** — This three bit field determines the amount of additional clock stretch cycles on every access to the external address space as shown in Table 21-6. The minimum number of stretch cycles depends on the EWAITE setting. Stretch cycles are added as programmed in normal expanded mode and emulation expanded mode; function not available in all other operating modes.

Table 21-6. External Access Stretch Bit Definition

EXSTR[2:0]	Number of Stretch Cycles	
	EWAITE = 0	EWAITE = 1
000	1 cycle	>= 2 cycles
001	2 cycles	>= 2 cycles
010	3 cycles	>= 3 cycles
011	4 cycles	>= 4 cycles
100	5 cycles	>= 5 cycles
101	6 cycles	>= 6 cycles
110	7 cycles	>= 7 cycles
111	8 cycles	>= 8 cycles

22.2 External Signal Description

This section lists and describes the signals that do connect off-chip.

22.2.1 Signal Properties

Table 22-1 shows all the pins and their functions that are controlled by the PIM. *Refer to Section 22.4, "Functional Description" for the availability of the individual pins in the different package options.*

NOTE

If there is more than one function associated with a pin, the priority is indicated by the position in the table from top (highest priority) to bottom (lowest priority).

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

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
—	BKGD	MODC[1]	I	MODC input during $\overline{RESET}$	BKGD
		BKGD	I/O	S12X_BDM communication pin	
A	PA[7:0]	ADDR[15:8] mux IVD[15:8][2]	O	High-order external bus address output (multiplexed with IVIS data)	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
B	PB[7:1]	ADDR[7:1] mux IVD[7:1][2]	O	Low-order external bus address output (multiplexed with IVIS data)	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
	PB[0]	ADDR[0] mux IVD0[2]	O	Low-order external bus address output (multiplexed with IVIS data)	
		$\overline{UDS}$	O	Upper data strobe	
		GPIO	I/O	General-purpose I/O	
C	PC[7:0]	DATA[15:8]	I/O	High-order bidirectional data input/output Configurable for reduced input threshold	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
D	PD[7:0]	DATA[7:0]	I/O	Low-order bidirectional data input/output Configurable for reduced input threshold	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	

22.3.2.11 S12X_EBI Ports, BKGD, VREGEN Pin Pull-up Control Register (PUCR)

	7	6	5	4	3	2	1	0
R	PUPKE	BKPUE	0	PUPEE	PUPDE	PUPCE	PUPBE	PUPAE
W								
Reset	1	1	0	1	0	0	0	0

= Unimplemented or Reserved

Figure 22-13. S12X_EBI Ports, BKGD, VREGEN Pin Pull-up Control Register (PUCR)

Read: Anytime in single-chip modes.

Write: Anytime, except BKPUE which is writable in special test mode only.

This register is used to enable pull-up devices for the associated ports A, B, C, D, E, and K. Pull-up devices are assigned on a per-port basis and apply to any pin in the corresponding port that is currently configured as an input.

Table 22-14. PUCR Field Descriptions

Field	Description
7 PUPKE	**Pull-up Port K Enable** 0 Port K pull-up devices are disabled. 1 Enable pull-up devices for Port K input pins.
6 BKPUE	**BKGD and VREGEN Pin Pull-up Enable** 0 BKGD and V_{REGEN} pull-up devices are disabled. 1 Enable pull-up devices on BKGD and V_{REGEN} pins.
4 PUPEE	**Pull-up Port E Enable** 0 Port E pull-up devices on bit 7, 4–0 are disabled. 1 Enable pull-up devices for Port E input pins bits 7, 4–0. **Note:** Bits 5 and 6 of Port E have pull-down devices which are only enabled during reset. This bit has no effect on these pins.
3 PUPDE	**Pull-up Port D Enable** 0 Port D pull-up devices are disabled. 1 Enable pull-up devices for all Port D input pins.
2 PUPCE	**Pull-up Port C Enable** 0 Port C pull-up devices are disabled. 1 Enable pull-up devices for all Port C input pins.
1 PUPBE	**Pull-up Port B Enable** 0 Port B pull-up devices are disabled. 1 Enable pull-up devices for all Port B input pins.
0 PUPAE	**Pull-up Port A Enable** 0 Port A pull-up devices are disabled. 1 Enable pull-up devices for all Port A input pins.

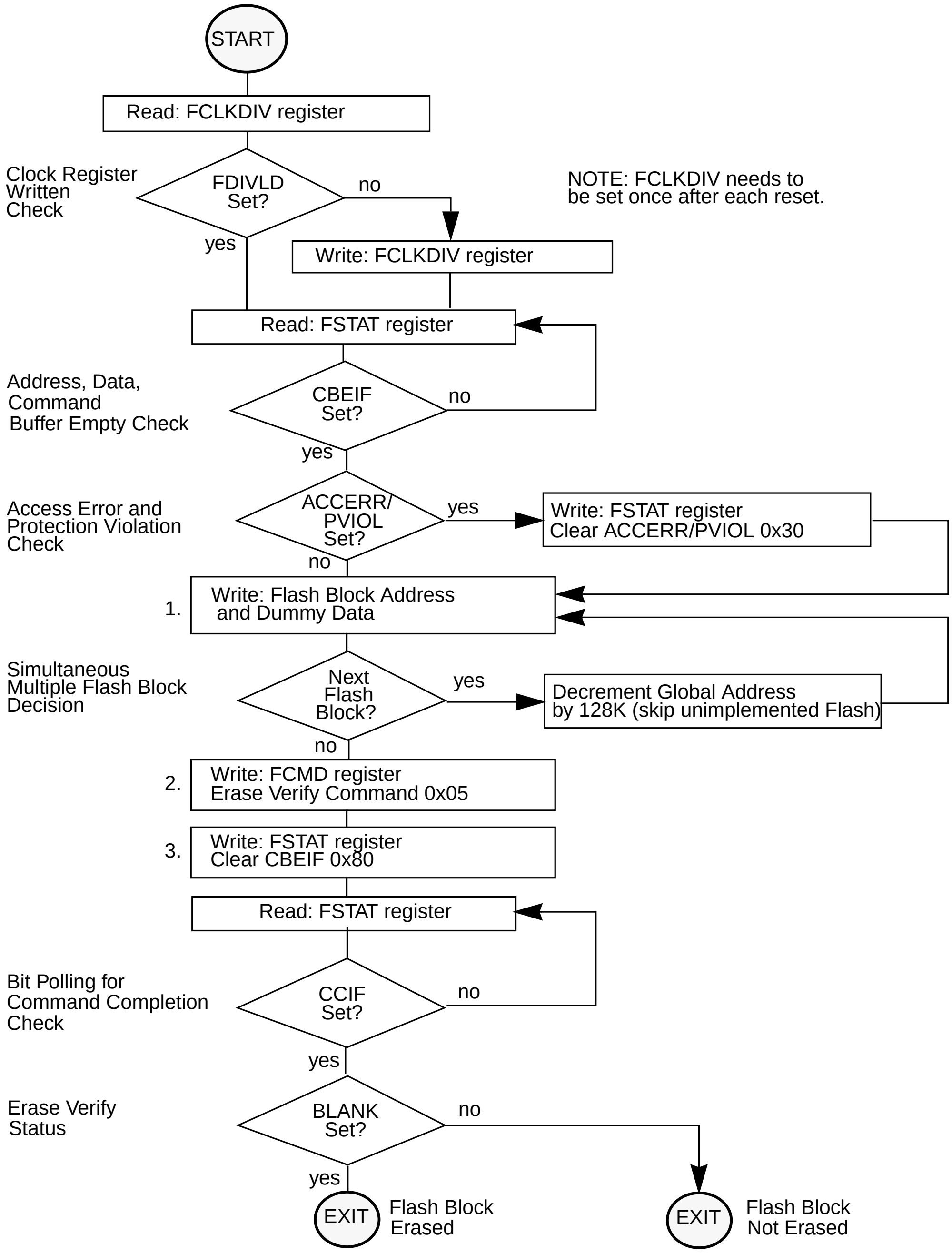

Figure 28-25. Example Erase Verify Command Flow