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	50MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, 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	16K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
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/product-detail/nxp-semiconductors/s912xet256j2maa

Table 1-9. Pin-Out Summary (Sheet 6 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP[1] 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
D13	108	84	60	VRH				
C13	109	85	61	VRL				
E13	110	86	62	VSSA1				
B14	111			PAD18	AN18			
A14	112			PAD19	AN19			
C12	113			PAD20	AN20			
B13	114			PAD21	AN21			
D12	115			PAD22	AN22			
B12	116			PAD23	AN23			
C11	117	87		PM7	TXCAN3	TXCAN4	TXD3	
A13	118	88		PM6	RXCAN3	RXCAN4	RXD3	
D11	119	89	63	PS0	RXD0			
B11	120	90	64	PS1	TXD0			
C10	121	91	65	PS2	RXD1			
A12	122	92	66	PS3	TXD1			
VSSX				VSSX6				
VDDX				VDDX6				
B10	123	93		PS4	MISO0			
A11	124	94		PS5	MOSI0			
A10	125	95		PS6	SCK0			
C9	126	96		PS7	$\overline{SS0}$			
B9	127	97	67	TEST				
D9	128	98	68	PJ7	KWJ7	TXCAN4	SCL0	TXCAN0
A9	129	99	69	PJ6	KWJ6	RXCAN4	SDA0	RXCAN0
C8	130			PJ5	KWJ5	SCL1	$\overline{CS2}$	
B8				PF0	$\overline{CS0}$			
D8	131			PJ4	KWJ4	SDA1	$\overline{CS0}$	
A8				PF1	$\overline{CS1}$			
D7	132	100	70	PM5	TXCAN2	TXCAN0	TXCAN4	SCK0

Table 2-51. DDRH Register Field Descriptions (continued)

Field	Description
1 DDRH	**Port H data direction**— This register controls the data direction of pin 1. The enabled SCI6 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI1 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
0 DDRH	**Port H data direction**— This register controls the data direction of pin 0. The enabled SCI6 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI1 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTH or PTIH registers, when changing the DDRH register.

2.3.56 Port H Reduced Drive Register (RDRH)

Address 0x0263 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	RDRH7	RDRH6	RDRH5	RDRH4	RDRH3	RDRH2	RDRH1	RDRH0
Reset	0	0	0	0	0	0	0	0

Figure 2-54. Port H Reduced Drive Register (RDRH)

1. Read: Anytime.
   Write: Anytime.

Table 2-52. RDRH Register Field Descriptions

Field	Description
7-0 RDRH	**Port H reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.4.2.9 Interrupt flag register (PIFx)

If the pin is used as an interrupt input this register holds the interrupt flag after a valid pin event.

2.4.2.10 Module routing register (MODRR, PTRRR, PTLRR, PTFRR)

This register supports the re-routing of the CAN0, CAN4, SPI2-0, SCI7-3, IIC0, TIM and CS[3:0] pins to alternative ports. This allows a software re-configuration of the pinouts of the different package options with respect to above peripherals.

2.4.3 Pins and Ports

NOTE

Please refer to the SOC Guide to determine the pin availability in the different package options.

2.4.3.1 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.

2.4.3.2 Port A, B

Port A pins PA[7:0] and Port B pins PB[7:0] can be used for either general-purpose I/O with the external bus interface. In this case Port A and Port B are associated with the external address bus outputs ADDR15-ADDR8 and ADDR7-ADDR0, respectively. PB0 is the ADDR0 or $\overline{\text{UDS}}$ output.

2.4.3.3 Port C, D

Port C pins PC[7:0] and Port D pins PD[7:0] can be used for either general-purpose I/O with the external bus interface. In this case Port C and Port D are associated with the external data bus inputs/outputs DATA15-DATA8 and DATA7-DATA0, respectively.

These pins are configured for reduced input threshold in certain operating modes (refer to S12X_EBI section).

2.4.3.4 Port E

Port E is associated with the external bus control outputs $\overline{\text{RW}}$, $\overline{\text{LSTRB}}$, $\overline{\text{LDS}}$ and $\overline{\text{RE}}$, the free-running clock outputs ECLK and ECLK2X, as well as with the $\overline{\text{TAGHI}}$, $\overline{\text{TAGLO}}$, MODA and MODB and interrupt inputs $\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$.

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] can 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.

```
;many cases assemblers are "direct page aware" and can
;automatically select direct mode.
```

3.3.2.5 MMC Control Register (MMCCTL1)

Address: 0x0013 PRR

	7	6	5	4	3	2	1	0
R	TGMRAMON	0	EEEIFRON	PGMIFRON	RAMHM	EROMON	ROMHM	ROMON
W								
Reset	0	0	0	0	0	EROMCTL	0	ROMCTL

= Unimplemented or Reserved

Figure 3-10. MMC Control Register (MMCCTL1)

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: Refer to each bit description. In emulation modes write operations will also be directed to the external bus.

CAUTION

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

Table 3-11. MMCCTL1 Field Descriptions

Field	Description
7 TGMRAMON	**EEE Tag RAM and FTM SCRATCH RAM visible in the memory map** Write: Anytime This bit is used to made the EEE Tag RAM nd FTM SCRATCH RAM visible in the global memory map. 0 Not visible in the memory map. 1 Visible in the memory map.
5 EEEIFRON	**EEE IFR visible in the memory map** Write: Anytime This bit is used to made the IFR sector of EEE DATA FLASH visible in the global memory map. 0 Not visible in the memory map. 1 Visible in the memory map.
4 PGMIFRON	**Program IFR visible in the memory map** Write: Anytime This bit is used to made the IFR sector of the Program Flash visible in the global memory map. 0 Not visible in the memory map. 1 Visible in the memory map.
3 RAMHM	**RAM only in higher Half of the memory map** Write: Once in normal and emulation modes and anytime in special modes 0 Accesses to $4000–$7FFF will be mapped to 14_4000-14_7FFF in the global memory space (external access). 1 Accesses to $4000–$7FFF will be mapped to $0F_C000-$0F_FFFF in the global memory space (RAM area).

Expansion of the BDM Local Address Map

PPAGE, RPAGE, and EPAGE registers are also used for the expansion of the BDM local address to the global address. These registers can be read and written by the BDM.

The BDM expansion scheme is the same as the CPU expansion scheme.

3.4.2.2 Global Addresses Based on the Global Page

CPU Global Addresses Based on the Global Page

The seven global page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEE as well as additional external memory.

The GPAGE Register is used only when the CPU is executing a global instruction (see Section 3.3.2.3, “Global Page Index Register (GPAGE)). The generated global address is the result of concatenation of the CPU local address [15:0] with the GPAGE register [22:16] (see Figure 3-7).

BDM Global Addresses Based on the Global Page

The seven BDMGPR Global Page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEE as well as additional external memory.

The BDM global page index register (BDMGPR) is used only in the case the CPU is executing a firmware command which uses a global instruction (like GLDD, GSTD) or by a BDM hardware command (like WRITE_W, WRITE_BYTE, READ_W, READ_BYTE). See the BDM Block Guide for further details.

The generated global address is a result of concatenation of the BDM local address with the BDMGPR register [22:16] in the case of a hardware command or concatenation of the CPU local address and the BDMGPR register [22:16] in the case of a firmware command (see Figure 3-18).

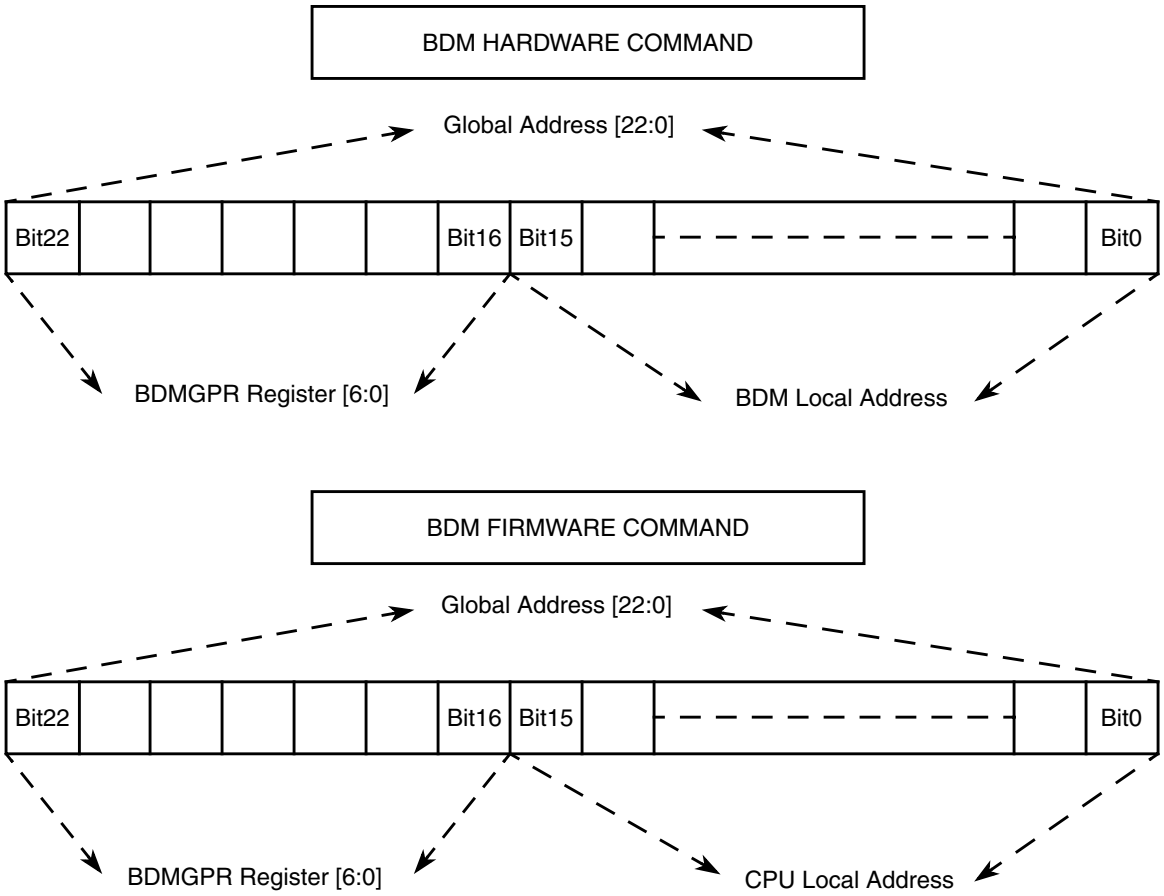

Figure 3-18. BDMGPR Address Mapping

3.4.2.3 Implemented Memory Map

The global memory spaces reserved for the internal resources (RAM, EEE, and FLASH) are not determined by the MMC module. Size of the individual internal resources are however fixed in the design of the device cannot be changed by the user. Please refer to the Device User Guide for further details. Figure 3-19 and Table 3-17 show the memory spaces occupied by the on-chip resources. Please note that the memory spaces have fixed top addresses.

Table 3-17. Global Implemented Memory Space

Internal Resource	$Address
RAM	RAM_LOW = 0x10_0000 minus RAMSIZE(1)
FLASH	FLASH_LOW = 0x80_0000 minus FLASHSIZE(2)

1. RAMSIZE is the hexadecimal value of RAM SIZE in bytes
2. FLASHSIZE is the hexadecimal value of FLASH SIZE in bytes

When the device is operating in expanded modes except emulation single-chip mode, accesses to global addresses which are not occupied by the on-chip resources (unimplemented areas or external memory space) result in accesses to the external bus (see Figure 3-19).

5.1.1 Glossary or Terms

bus clock	System Clock. Refer to CRG Block Guide.
expanded modes	Normal Expanded Mode Emulation Single-Chip Mode Emulation Expanded Mode Special Test Mode
single-chip modes	Normal Single-Chip Mode Special Single-Chip Mode
emulation modes	Emulation Single-Chip Mode Emulation Expanded Mode
normal modes	Normal Single-Chip Mode Normal Expanded Mode
special modes	Special Single-Chip Mode Special Test Mode
NS	Normal Single-Chip Mode
SS	Special Single-Chip Mode
NX	Normal Expanded Mode
ES	Emulation Single-Chip Mode
EX	Emulation Expanded Mode
ST	Special Test Mode
external resource	Addresses outside MCU
PRR	Port Replacement Registers
PRU	Port Replacement Unit
EMULMEM	External emulation memory
access source	CPU or BDM or XGATE

5.1.2 Features

The XEBI includes the following features:

- Output of up to 23-bit address bus and control signals to be used with a non-muxed external bus
- Bidirectional 16-bit external data bus with option to disable upper half
- Visibility of internal bus activity

5.1.3 Modes of Operation

- Single-chip modes

  The external bus interface is not available in these modes.
- Expanded modes

  Address, data, and control signals are activated on the external bus in normal expanded mode and special test mode.
- Emulation modes

  The external bus is activated to interface to an external tool for emulation of normal expanded mode or normal single-chip mode applications.

8.3.2 Register Descriptions

This section consists of the S12XDBG control and trace buffer register descriptions in address order. Each comparator has a bank of registers that are visible through an 8-byte window between 0x0028 and 0x002F in the S12XDBG module register address map. When ARM is set in DBGC1, the only bits in the S12XDBG module registers that can be written are ARM, TRIG, and COMRV[1:0].

8.3.2.1 Debug Control Register 1 (DBGC1)

Address: 0x0020

	7	6	5	4	3	2	1	0
R	ARM	0	XGSBPE	BDM	DBGBRK		COMRV	
W		TRIG						
Reset	0	0	0	0	0	0	0	0

Figure 8-3. Debug Control Register (DBGC1)

Read: Anytime

Write: Bits 7, 1, 0 anytime
Bit 6 can be written anytime but always reads back as 0.
Bits 5:2 anytime S12XDBG is not armed.

NOTE

If a write access to DBGC1 with the ARM bit position set occurs simultaneously to a hardware disarm from an internal trigger event, then the ARM bit is cleared due to the hardware disarm.

NOTE

When disarming the S12XDBG by clearing ARM with software, the contents of bits[5:2] are not affected by the write, since up until the write operation, ARM = 1 preventing these bits from being written. These bits must be cleared using a second write if required.

Table 8-5. DBGC1 Field Descriptions

Field	Description
7 ARM	**Arm Bit** — The ARM bit controls whether the S12XDBG module is armed. This bit can be set and cleared by user software and is automatically cleared on completion of a tracing session, or if a breakpoint is generated with tracing not enabled. On setting this bit the state sequencer enters State1. 0 Debugger disarmed 1 Debugger armed
6 TRIG	**Immediate Trigger Request Bit** — This bit when written to 1 requests an immediate trigger independent of comparator or external tag signal status. When tracing is complete a forced breakpoint may be generated depending upon DBGBRK and BDM bit settings. This bit always reads back a 0. Writing a 0 to this bit has no effect. If TSOURCE are clear no tracing is carried out. If tracing has already commenced using BEGIN- or MID trigger alignment, it continues until the end of the tracing session as defined by the TALIGN bit settings, thus TRIG has no affect. In secure mode tracing is disabled and writing to this bit has no effect. 0 Do not trigger until the state sequencer enters the Final State. 1 Trigger immediately .

The programmer's model of the XGATE RISC core is shown in Figure 10-22. The processor offers a set of seven general purpose registers (R1 - R7), which serve as accumulators and index registers. An additional eighth register (R0) is tied to the value "$0000". Registers R1 and R7 have additional functionality. R1 is preloaded with the initial data pointer of the channel's service request vector (see Figure 10-23). R7 is either preloaded with the content of XGISP74 if the interrupt priority of the current channel is in the range 7 to 4, or it is with preloaded the content of XGISP31 if the interrupt priority of the current channel is in the range 3 to 1. The remaining general purpose registers will be reset to an unspecified value at the beginning of each thread.

The 16 bit program counter allows the addressing of a 64 kbyte address space.

The condition code register contains four bits: the sign bit (S), the zero flag (Z), the overflow flag (V), and the carry bit (C). The initial content of the condition code register is undefined.

10.4.3 Memory Map

The XGATE's RISC core is able to access an address space of 64K bytes. The allocation of memory blocks within this address space is determined on chip level. Refer to the **S12X_MMC Section** for a detailed information.

The XGATE vector block assigns a start address and a data pointer to each XGATE channel. Its position in the XGATE memory map can be adjusted through the XGVBR register (see Section 10.3.1.7, "XGATE Vector Base Address Register (XGVBR)"). Figure 10-23 shows the layout of the vector block. Each vector consists of two 16 bit words. The first contains the start address of the service routine. This value will be loaded into the program counter before a service routine is executed. The second word is a pointer to the service routine's data space. This value will be loaded into register R1 before a service routine is executed.

ASR — Arithmetic Shift Right — ASR

Operation

n = RS or IMM4

Shifts the bits in register RD *n* positions to the right. The higher *n* bits of the register RD become filled with the sign bit (RD[15]). The carry flag will be updated to the bit contained in RD[n-1] before the shift for *n* > 0.

n can range from 0 to 16.

In immediate address mode, *n* is determined by the operand IMM4. *n* is considered to be 16 if IMM4 is equal to 0.

In dyadic address mode, *n* is determined by the content of RS. *n* is considered to be 16 if the content of RS is greater than 15.

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.
$RD[15]_{old}$ ^ $RD[15]_{new}$

C: Set if *n* > 0 and RD[n-1] = 1; if n = 0 unaffected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code												Cycles
ASR RD, #IMM4	IMM4	0	0	0	0	1	RD	IMM4		1	0	0	1	P
ASR RD, RS	DYA	0	0	0	0	1	RD	RS	1	0	0	0	1	P

CMP — Compare — CMP

Operation

RS1 – RS2 ⇒ NONE (translates to SUB R0, RS1, RS2)
RD – IMM16 ⇒ NONE (translates to CMPL RD, #IMM16[7:0]; CPCH RD, #IMM16[15:8])

Subtracts two 16 bit values and discards the result.

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.
RS1[15] & $\overline{RS2[15]}$ & $\overline{result[15]}$ | $\overline{RS1[15]}$ & RS2[15] & result[15]
RD[15] & $\overline{IMM16[15]}$ & $\overline{result[15]}$ | $\overline{RD[15]}$ & IMM16[15] & result[15]

C: Set if there is a carry from the bit 15 of the result; cleared otherwise.
$\overline{RS1[15]}$ & RS2[15] | $\overline{RS1[15]}$ & result[15] | RS2[15] & result[15]
$\overline{RD[15]}$ & IMM16[15] | $\overline{RD[15]}$ & result[15] | IMM16[15] & result[15]

Code and CPU Cycles

Source Form	Address Mode	Machine Code											Cycles
CMP RS1, RS2	TRI	0	0	0	1	1	0	0	0	RS1	RS2	0	0	P
CMP RS, #IMM16	IMM8	1	1	0	1	0	RS			IMM16[7:0]				P
	IMM8	1	1	0	1	1	RS			IMM16[15:8]				P

13.1.3 Block Diagram

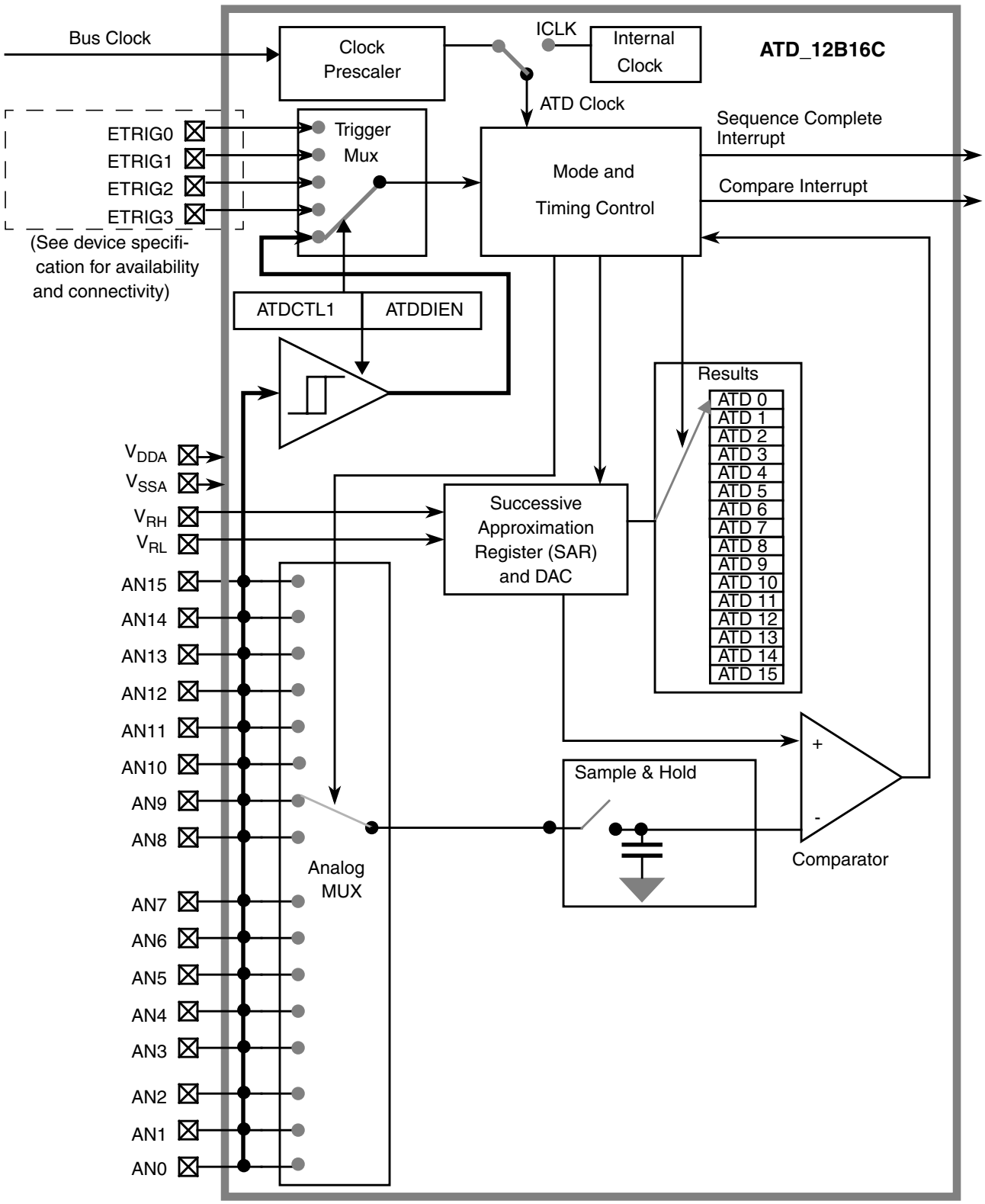

Figure 13-1. ADC12B16C Block Diagram

Offset Module Base + 0x0011

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

= Unimplemented or Reserved

Figure 25-23. Flash Reserved0 Register (FRSV0)

All bits in the FRSV0 register read 0 and are not writable.

25.3.2.16 Flash Reserved1 Register (FRSV1)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0012

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

= Unimplemented or Reserved

Figure 25-24. Flash Reserved1 Register (FRSV1)

All bits in the FRSV1 register read 0 and are not writable.

25.3.2.17 Flash Reserved2 Register (FRSV2)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0013

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

= Unimplemented or Reserved

Figure 25-25. Flash Reserved2 Register (FRSV2)

All bits in the FRSV2 register read 0 and are not writable.

Address & Name	7	6	5	4	3	2	1	0

= Unimplemented or Reserved

Figure 27-4. FTM512K3 Register Summary (continued)

27.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

Offset Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIV[6:0]						
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 27-5. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bits 6–0 are write once and bit 7 is not writable.

Table 27-8. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded** 0 FCLKDIV register has not been written 1 FCLKDIV register has been written since the last reset
6–0 FDIV[6:0]	**Clock Divider Bits** — FDIV[6:0] must be set to effectively divide OSCCLK down to generate an internal Flash clock, FCLK, with a target frequency of 1 MHz for use by the Flash module to control timed events during program and erase algorithms. Table 27-9 shows recommended values for FDIV[6:0] based on OSCCLK frequency. Please refer to Section 27.4.1, “Flash Command Operations,” for more information.

CAUTION

The FCLKDIV register should never be written while a Flash command is executing (CCIF=0). The FCLKDIV register is writable during the Flash reset sequence even though CCIF is clear.

The Full Partition D-Flash command (see Section 28.4.2.15) is used to program the EEE nonvolatile information register fields where address 0x12_0000 defines the D-Flash partition for user access and address 0x12_0004 defines the buffer RAM partition for EEE operations.

Table 28-7. EEE Nonvolatile Information Register Fields

Global Address (EEEIFRON)	Size (Bytes)	Description
0x12_0000 – 0x12_0001	2	D-Flash User Partition (DFPART) Refer to Section 28.4.2.15, “Full Partition D-Flash Command”
0x12_0002 – 0x12_0003	2	D-Flash User Partition (duplicate[1])
0x12_0004 – 0x12_0005	2	Buffer RAM EEE Partition (ERPART) Refer to Section 28.4.2.15, “Full Partition D-Flash Command”
0x12_0006 – 0x12_0007	2	Buffer RAM EEE Partition (duplicate[1])
0x12_0008 – 0x12_007F	120	Reserved

1. Duplicate value used if primary value generates a double bit fault when read during the reset sequence.

28.3.2 Register Descriptions

The Flash module contains a set of 20 control and status registers located between Flash module base + 0x0000 and 0x0013. A summary of the Flash module registers is given in Figure 28-4 with detailed descriptions in the following subsections.

CAUTION

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and Memory Controller behavior.

Address & Name		7	6	5	4	3	2	1	0
0x0000 FCLKDIV	R	FDIVLD	FDIV6	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
	W								
0x0002 FCCOBIX	R	0	0	0	0	0	CCOBIX2	CCOBIX1	CCOBIX0
	W								
0x0003 FECCRIX	R	0	0	0	0	0	ECCRIX2	ECCRIX1	ECCRIX0
	W								
0x0004 FCNFG	R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
	W								

Figure 28-4. FTM768K4 Register Summary

Table 28-14. FECCRIX Field Descriptions

Field	Description
2-0 ECCRIX[2:0]	**ECC Error Register Index**— The ECCRIX bits are used to select which word of the FECCR register array is being read. See Section 28.3.2.13, “Flash ECC Error Results Register (FECCR),” for more details.

28.3.2.5 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU or XGATE.

Offset Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 28-9. Flash Configuration Register (FCNFG)

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

Table 28-15. FCNFG Field Descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Section 28.3.2.7)
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 28.3.2.8). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated

Table 28-76. EEPROM Emulation Query Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 28-30)
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

28.4.2.22 Partition D-Flash Command

The Partition D-Flash command allows the user to allocate sectors within the D-Flash block for applications and a partition within the buffer RAM for EEPROM access. The D-Flash block consists of 128 sectors with 256 bytes per sector. The Erase All Blocks command must be run prior to launching the Partition D-Flash command.

Table 28-77. Partition D-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x20	Not required
001	Number of 256 byte sectors for the D-Flash user partition (DFPART)	
010	Number of 256 byte sectors for buffer RAM EEE partition (ERPART)	

Upon clearing CCIF to launch the Partition D-Flash command, the following actions are taken to define a partition within the D-Flash block for direct access (DFPART) and a partition within the buffer RAM for EEE use (ERPART):

- Validate the DFPART and ERPART values provided:
  - DFPART <= 128 (maximum number of 256 byte sectors in D-Flash block)
  - ERPART <= 16 (maximum number of 256 byte sectors in buffer RAM)
  - If ERPART > 0, 128 - DFPART >= 12 (minimum number of 256 byte sectors in the D-Flash block required to support EEE)
  - If ERPART > 0, ((128-DFPART)/ERPART) >= 8 (minimum ratio of D-Flash EEE space to buffer RAM EEE space to support EEE)
- Erase verify the D-Flash block and the EEE nonvolatile information register
- Program DFPART to the EEE nonvolatile information register at global address 0x12_0000 (see Table 28-7)

Command Write Sequence — An MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.

D-Flash Memory — The D-Flash memory constitutes the nonvolatile memory store required for EEE. Memory space in the D-Flash memory not required for EEE can be partitioned to provide nonvolatile memory space for applications.

D-Flash Sector — The D-Flash sector is the smallest portion of the D-Flash memory that can be erased. The D-Flash sector consists of four 64 byte rows for a total of 256 bytes.

EEE (Emulated EEPROM) — A method to emulate the small sector size features and endurance characteristics associated with an EEPROM.

EEE IFR — Nonvolatile information register located in the D-Flash block that contains data required to partition the D-Flash memory and buffer RAM for EEE. The EEE IFR is visible in the global memory map by setting the EEEIFRON bit in the MMCCTL1 register.

NVM Command Mode — An NVM mode using the CPU to setup the FCCOB register to pass parameters required for Flash command execution.

Phrase — An aligned group of four 16-bit words within the P-Flash memory. Each phrase includes eight ECC bits for single bit fault correction and double bit fault detection within the phrase.

P-Flash Memory — The P-Flash memory constitutes the main nonvolatile memory store for applications.

P-Flash Sector — The P-Flash sector is the smallest portion of the P-Flash memory that can be erased. Each P-Flash sector contains 1024 bytes.

Program IFR — Nonvolatile information register located in the P-Flash block that contains the Device ID, Version ID, and the Program Once field. The Program IFR is visible in the global memory map by setting the PGMIFRON bit in the MMCCTL1 register.

29.1.2 Features

29.1.2.1 P-Flash Features

- 1024 Kbytes of P-Flash memory composed of three 256 Kbyte Flash blocks and two 128 Kbyte Flash blocks. The 256 Kbyte Flash block consists of two 128 Kbyte sections each divided into 128 sectors of 1024 bytes. The 128 Kbyte Flash blocks are each divided into 128 sectors of 1024 bytes.
- Single bit fault correction and double bit fault detection within a 64-bit phrase during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation
- Ability to program up to one phrase in each P-Flash block simultaneously
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

Table 29-12. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED(1)
10	UNSECURED
11	SECURED

1. Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 29.5.

29.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

Offset Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CCOBIX[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 29-7. FCCOB Index Register (FCCOBIX)

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 29-13. FCCOBIX Field Descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to select which word of the FCCOB register array is being read or written to. See Section 29.3.2.11, “Flash Common Command Object Register (FCCOB),” for more details.

29.3.2.4 Flash ECCR Index Register (FECCRIX)

The FECCRIX register is used to index the FECCR register for ECC fault reporting.

Offset Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ECCRIX[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 29-8. FECCR Index Register (FECCRIX)

ECCRIX bits are readable and writable while remaining bits read 0 and are not writable.

A.7.3 External Bus Timing

The following conditions are assumed for all following external bus timing values:

- Crystal input within 45% to 55% duty
- Equal 25 pF load on all pins
- Pad full drive (reduced drive must be off)

A.7.3.1 Normal Expanded Mode (External Wait Feature Disabled)

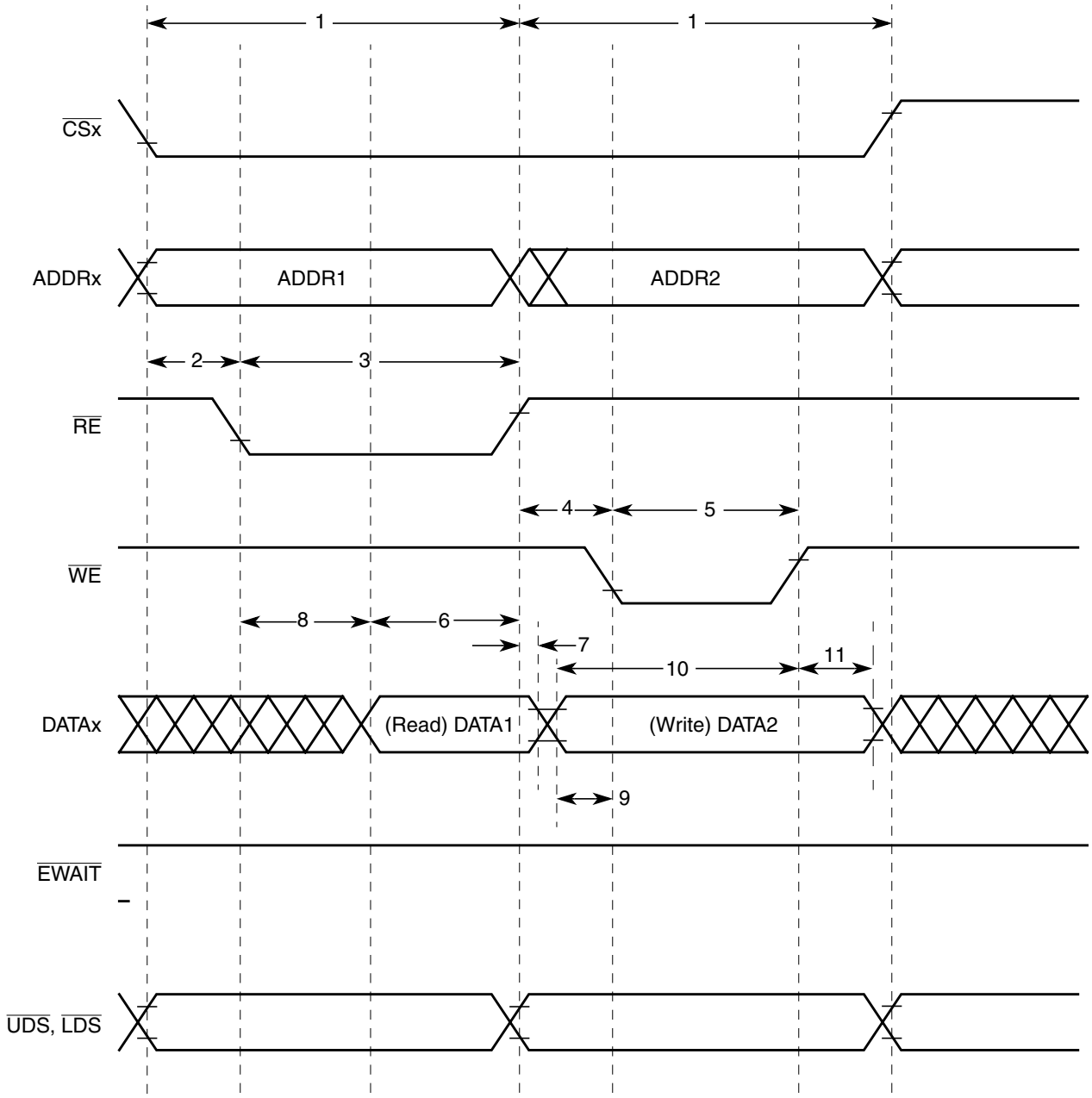

Figure A-12. Example 1a: Normal Expanded Mode — Read Followed by Write