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	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	16K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°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/mc9s12xa256vaa

1.1.5 Part ID Assignments & Maskset Numbers

The part ID is located in two 8-bit registers PARTIDH and PARTIDL (addresses 0x001A and 0x001B). The read-only value is a unique part ID for each revision of the chip. Table 1-6 shows the assigned part ID number and Mask Set number.

Table 1-6. Part Names, Masksets and Assigned Part ID Numbers

Part Names	Mask Set Number	Part ID[1]
MC9S12XDP512	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
MC9S12XDT512	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
MC9S12XA512	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
MC9S12XDT384	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
MC9S12XDQ256	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
	0M84E/1M84E	0xC000/0xC001
MC9S12XDT256	0L15Y/1L15Y/0M23S	0xC410/0xC4110xC412
	0M84E/1M84E	0xC000/0xC001
MC9S12XD256	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
	0M84E/1M84E	0xC000/0xC001
MC9S12XB256	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
	0M84E/1M84E	0xC000/0xC001
MC9S12XA256	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
	0M84E/1M84E	0xC000/0xC001
MC9S12XDG128	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
	0M42E/1M42E/2M42E	0xC100/0xC101/0xC102
MC9S12XD128	0L15Y/1L15Y/0M23S	0xC410/0xC4110xC412
	0M42E/1M42E/2M42E	0xC100/0xC101/0xC102
MC9S12XA128	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
	0M42E/1M42E/2M42E	0xC100/0xC101/0xC102
MC9S12XB128	0L15Y/1L15Y/0M23S	0xC410/0xC411/0xC412
	0M42E/1M42E/2M42E	0xC100/0xC101/0xC102

1 The coding is as follows:
Bit 15-12: Major family identifier
Bit 11-8: Minor family identifier
Bit 7-4: Major mask set revision number including FAB transfers
Bit 3-0: Minor — non full — mask set revision

1.2 Signal Description

This section describes signals that connect off-chip. It includes a pinout diagram, a table of signal properties, and detailed discussion of signals.

1.2.1 Device Pinout

The MC9S12XD family of devices offers pin-compatible packaged devices to assist with system development and accommodate expansion of the application.

Chapter 2
Clocks and Reset Generator (S12CRGV6)

2.1 Introduction

This specification describes the function of the clocks and reset generator (CRG).

2.1.1 Features

The main features of this block are:

- Phase locked loop (PLL) frequency multiplier
  - Reference divider
  - Automatic bandwidth control mode for low-jitter operation
  - Automatic frequency lock detector
  - Interrupt request on entry or exit from locked condition
  - Self clock mode in absence of reference clock
- System clock generator
  - Clock quality check
  - User selectable fast wake-up from Stop in self-clock mode for power saving and immediate program execution
  - Clock switch for either oscillator or PLL based system clocks
- Computer operating properly (COP) watchdog timer with time-out clear window
- System reset generation from the following possible sources:
  - Power on reset
  - Low voltage reset
  - Illegal address reset
  - COP reset
  - Loss of clock reset
  - External pin reset
- Real-time interrupt (RTI)

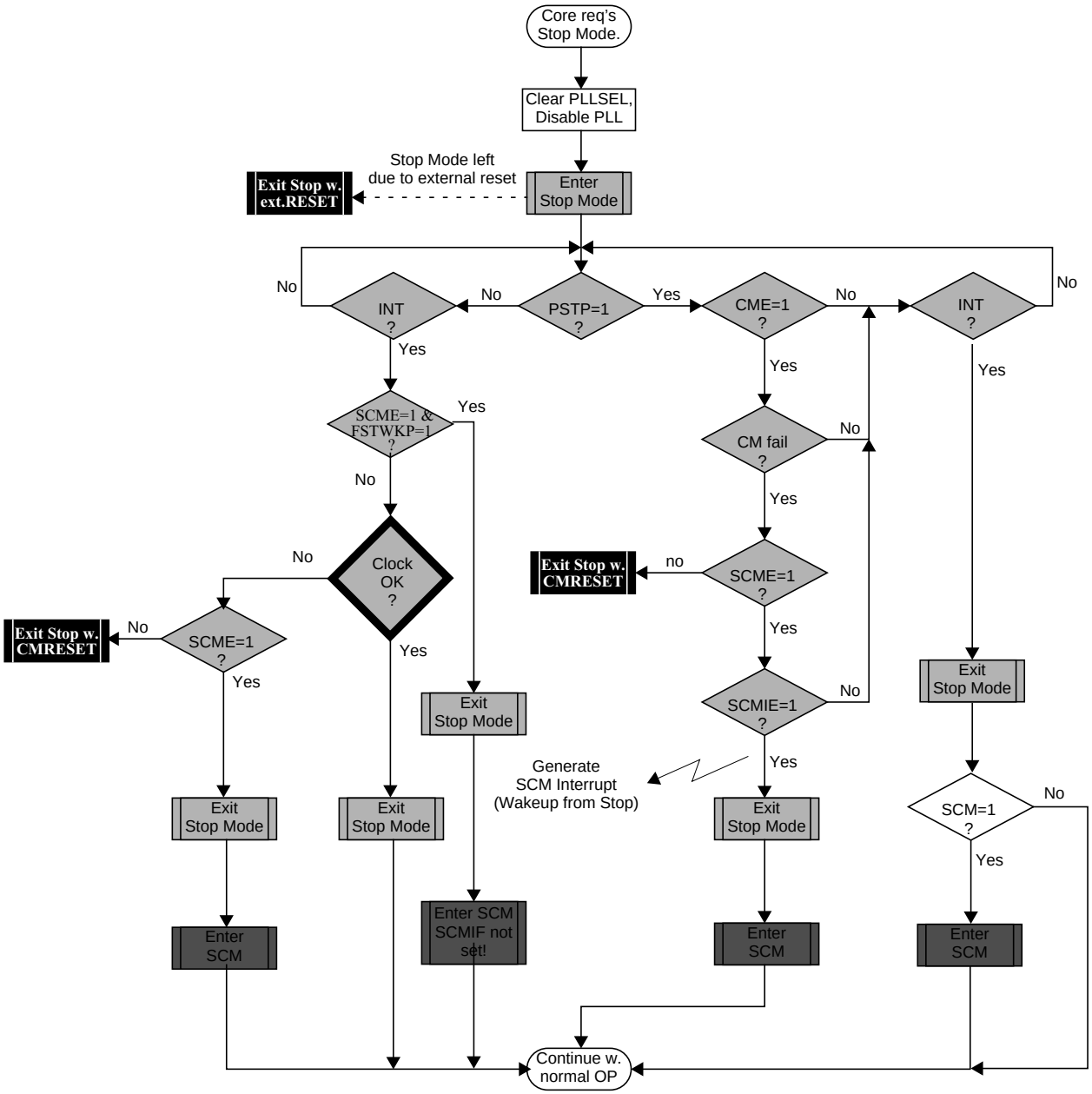

Figure 2-22. Stop Mode Entry/Exit Sequence

XNORL — Logical Exclusive NOR Immediate 8 bit Constant (Low Byte) — XNORL

Operation

~(RD.L ^ IMM8) ⇒ RD.L

Performs a bit wise logical exclusive NOR between the low byte of register RD and an immediate 8 bit constant and stores the result in the destination register RD.L. The high byte of RD is not affected.

CCR Effects

N	Z	V	C
Δ	Δ	0	—

N: Set if bit 7 of the result is set; cleared otherwise.
Z: Set if the 8 bit result is $00; cleared otherwise.
V: 0; cleared.
C: Not affected.

Code and CPU Cycles

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

When PACN1 overflows from 0x00FF to 0x0000, the interrupt flag PBOVF in PBFLG is set.

Full count register access will take place in one clock cycle.

NOTE

A separate read/write for high byte and low byte will give a different result than accessing them as a word.

When clocking pulse and write to the registers occurs simultaneously, write takes priority and the register is not incremented.

Table 9-5. IIC Divider and Hold Values (Sheet 3 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
4F	136	26	60	70
50	96	18	36	50
51	112	18	44	58
52	128	26	52	66
53	144	26	60	74
54	160	34	68	82
55	176	34	76	90
56	208	42	92	106
57	256	42	116	130
58	160	18	76	82
59	192	18	92	98
5A	224	34	108	114
5B	256	34	124	130
5C	288	50	140	146
5D	320	50	156	162
5E	384	66	188	194
5F	480	66	236	242
60	320	34	156	162
61	384	34	188	194
62	448	66	220	226
63	512	66	252	258
64	576	98	284	290
65	640	98	316	322
66	768	130	380	386
67	960	130	476	482
68	640	66	316	322
69	768	66	380	386
6A	896	130	444	450
6B	1024	130	508	514
6C	1152	194	572	578
6D	1280	194	636	642
6E	1536	258	764	770
6F	1920	258	956	962
70	1280	130	636	642
71	1536	130	764	770
72	1792	258	892	898
73	2048	258	1020	1026
74	2304	386	1148	1154
75	2560	386	1276	1282
76	3072	514	1532	1538
77	3840	514	1916	1922
78	2560	258	1276	1282
79	3072	258	1532	1538
7A	3584	514	1788	1794
7B	4096	514	2044	2050

stop mode has been reached. Hence after a system stop mode the handshake feature must be enabled again by sending the ACK_ENABLE command.

15.4.11 Serial Communication Time Out

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target will keep waiting for a rising edge on BKGD in order to answer the SYNC request pulse. If the rising edge is not detected, the target will keep waiting forever without any time-out limit.

Consider now the case where the host returns BKGD to logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target will keep waiting for another falling edge marking the start of a new bit. If, however, a new falling edge is not detected by the target within 512 clock cycles since the last falling edge, a time-out occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

If a read command is issued but the data is not retrieved within 512 serial clock cycles, a soft-reset will occur causing the command to be disregarded. The data is not available for retrieval after the time-out has occurred. This is the expected behavior if the handshake protocol is not enabled. However, consider the behavior where the BDM is running in a frequency much greater than the CPU frequency. In this case, the command could time out before the data is ready to be retrieved. In order to allow the data to be retrieved even with a large clock frequency mismatch (between BDM and CPU) when the hardware handshake protocol is enabled, the time out between a read command and the data retrieval is disabled. Therefore, the host could wait for more then 512 serial clock cycles and still be able to retrieve the data from an issued read command. However, once the handshake pulse (ACK pulse) is issued, the time-out feature is re-activated, meaning that the target will time out after 512 clock cycles. Therefore, the host needs to retrieve the data within a 512 serial clock cycles time frame after the ACK pulse had been issued. After that period, the read command is discarded and the data is no longer available for retrieval. Any negative edge in the BKGD pin after the time-out period is considered to be a new command or a SYNC request.

Note that whenever a partially issued command, or partially retrieved data, has occurred the time out in the serial communication is active. This means that if a time frame higher than 512 serial clock cycles is observed between two consecutive negative edges and the command being issued or data being retrieved is not complete, a soft-reset will occur causing the partially received command or data retrieved to be disregarded. The next negative edge in the BKGD pin, after a soft-reset has occurred, is considered by the target as the start of a new BDM command, or the start of a SYNC request pulse.

Table 17-13. PPAGE Field Descriptions

Field	Description
7–0 PIX[7:0]	**Program Page Index Bits 7–0** — These page index bits are used to select which of the 256 FLASH or ROM array pages is to be accessed in the Program Page Window.

The fixed 16K page from $4000–$7FFF (when ROMHM = 0) is the page number $FD.

The reset value of $FE ensures that there is linear Flash space available between addresses $4000 and $FFFF out of reset.

The fixed 16K page from $C000-$FFFF is the page number $FF.

17.3.2.9 RAM Write Protection Control Register (RAMWPC)

Address: 0x011C

	7	6	5	4	3	2	1	0
R	RWPE	0	0	0	0	0	AVIE	AVIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 17-17. RAM Write Protection Control Register (RAMWPC)

Read: Anytime

Write: Anytime

Table 17-14. RAMWPC Field Descriptions

Field	Description
7 RWPE	**RAM Write Protection Enable** — This bit enables the RAM write protection mechanism. When the RWPE bit is cleared, there is no write protection and any memory location is writable by the CPU module and the XGATE module. When the RWPE bit is set the write protection mechanism is enabled and write access of the CPU or to the XGATE RAM region. Write access performed by the XGATE module to outside of the XGATE RAM region or the shared region is suppressed as well in this case. 0 RAM write protection check is disabled, region boundary registers can be written. 1 RAM write protection check is enabled, region boundary registers cannot be written.
1 AVIE	**CPU Access Violation Interrupt Enable** — This bit enables the Access Violation Interrupt. If AVIE is set and AVIF is set, an interrupt is generated. 0 CPU Access Violation Interrupt Disabled. 1 CPU Access Violation Interrupt Enabled.
0 AVIF	**CPU Access Violation Interrupt Flag** — When set, this bit indicates that the CPU has tried to write a memory location inside the XGATE RAM region. This flag can be reset by writing'1' to the AVIF bit location. 0 No access violation by the CPU was detected. 1 Access violation by the CPU was detected.

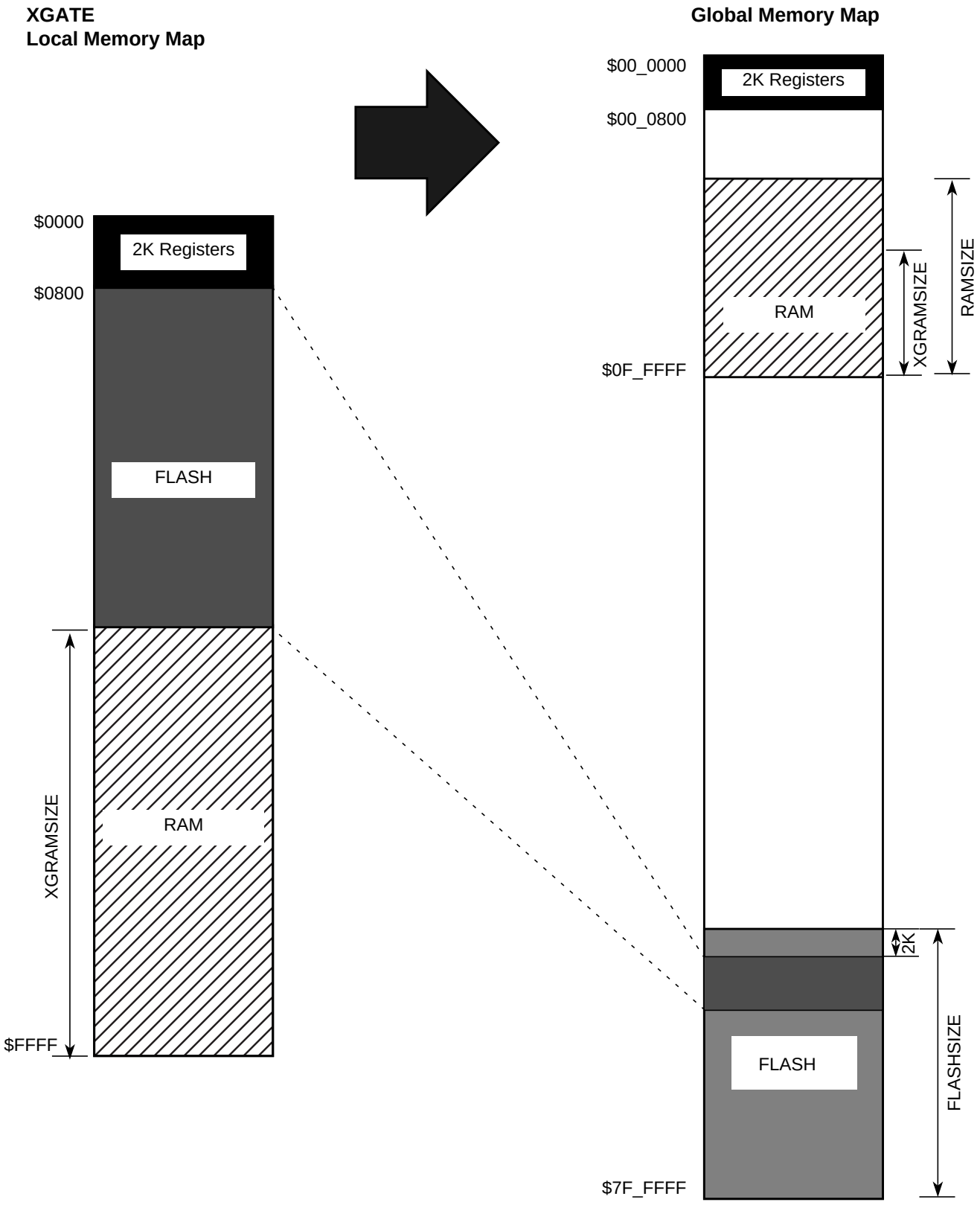

Figure 17-24. Local to Global Address Mapping (XGATE)

19.3.1.11.4 Debug Comparator Address Low Register (DBGXAL)

0x002B

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

Figure 19-17. Debug Comparator Address Low Register (DBGXAL)

Read: Anytime

Write: Anytime when DBG not armed.

Table 19-31. DBGXAL Field Descriptions

Field	Description
7–0 Bits [7:0]	**Comparator Address Low Compare Bits** — The comparator address low compare bits control whether the selected comparator will compare the address bus bits [7:0] to a logic 1 or logic 0. 0 Compare corresponding address bit to a logic 0 1 Compare corresponding address bit to a logic 1

19.3.1.11.5 Debug Comparator Data High Register (DBGXDH)

0x002C

	7	6	5	4	3	2	1	0
R W	Bit 15	14	13	12	11	10	9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 19-18. Debug Comparator Data High Register (DBGXDH)

Read: Anytime

Write: Anytime when DBG not armed.

Table 19-32. DBGXDH Field Descriptions

Field	Description
7–0 Bits [15:8]	**Comparator Data High Compare Bits** — The comparator data high compare bits control whether the selected comparator compares the data bus bits [15:8] to a logic 1 or logic 0. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparators A and C. 0 Compare corresponding data bit to a logic 0 1 Compare corresponding data bit to a logic 1

23.0.5.8 Port D Data Direction Register (DDRD)

	7	6	5	4	3	2	1	0
R W	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset	0	0	0	0	0	0	0	0

Figure 23-10. Port D Data Direction Register (DDRD)

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.

Table 23-11. DDRD Field Descriptions

Field	Description
7–0 DDRD[7:0]	Data Direction Port D — This register controls the data direction for port D. DDRD determines whether each pin is an input or output. A logic level “1” causes the associated port pin to be an output and a logic level “0” causes the associated pin to be a high-impedance input. 0 Associated pin is configured as input. 1 Associated pin is configured as output. **Note:** Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PORTD after changing the DDRD register.

23.0.5.9 Port E Data Register (PORTE)

	7	6	5	4	3	2	1	0
R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
W								
Alt. Func.	$\overline{\text{XCLKS}}$ or ECLKX2	MODB or $\overline{\text{TAGHI}}$	MODA or $\overline{\text{RE}}$ or $\overline{\text{TAGLO}}$	ECLK	EROMCTL or $\overline{\text{LSTRB}}$ or $\overline{\text{LDS}}$	R/$\overline{\text{W}}$ or $\overline{\text{WE}}$	$\overline{\text{IRQ}}$	$\overline{\text{XIRQ}}$
Reset	0	0	0	0	0	0	—[1]	—[1]

= Unimplemented or Reserved

Figure 23-11. Port E Data Register (PORTE)

1. These registers are reset to zero. Two bus clock cycles after reset release the register values are updated with the associated pin values.

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

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

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.

25.8.1 Description of EEPROM Interrupt Operation

The logic used for generating interrupts is shown in Figure 25-24.

The EEPROM module uses the CBEIF and CCIF flags in combination with the CBIE and CCIE enable bits to generate the EEPROM command interrupt request.

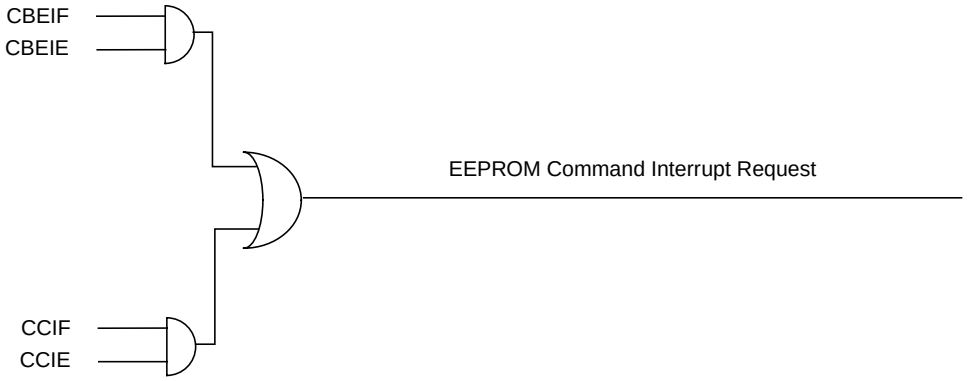

Figure 25-24. EEPROM Interrupt Implementation

For a detailed description of the register bits, refer to Section 25.3.2.4, “EEPROM Configuration Register (ECNFG)” and Section 25.3.2.6, “EEPROM Status Register (ESTAT)” .

CBEIF, PVIOL, and ACCERR are readable and writable, CCIF and BLANK are readable and not writable, remaining bits read 0 and are not writable in normal mode. FAIL is readable and writable in special mode. FAIL must be clear in special mode when starting a command write sequence.

Table 28-14. FSTAT Field Descriptions

Field	Description
7 CBEIF	**Command Buffer Empty Interrupt Flag** — The CBEIF flag indicates that the address, data and command buffers are empty so that a new command write sequence can be started. Writing a 0 to the CBEIF flag has no effect on CBEIF. Writing a 0 to CBEIF after writing an aligned word to the Flash address space, but before CBEIF is cleared, will abort a command write sequence and cause the ACCERR flag to be set. Writing a 0 to CBEIF outside of a command write sequence will not set the ACCERR flag. The CBEIF flag is cleared by writing a 1 to CBEIF. The CBEIF flag is used together with the CBEIE bit in the FCNFG register to generate an interrupt request (see Figure 28-32). 0 Command buffers are full. 1 Command buffers are ready to accept a new command.
6 CCIF	**Command Complete Interrupt Flag** — The CCIF flag indicates that there are no more commands pending. The CCIF flag is cleared when CBEIF is cleared and sets automatically upon completion of all active and pending commands. The CCIF flag does not set when an active commands completes and a pending command is fetched from the command buffer. Writing to the CCIF flag has no effect on CCIF. The CCIF flag is used together with the CCIE bit in the FCNFG register to generate an interrupt request (see Figure 28-32). 0 Command in progress. 1 All commands are completed.
5 PVIOL	**Protection Violation Flag** —The PVIOL flag indicates an attempt was made to program or erase an address in a protected area of the Flash memory during a command write sequence. Writing a 0 to the PVIOL flag has no effect on PVIOL. The PVIOL flag is cleared by writing a 1 to PVIOL. While PVIOL is set, it is not possible to launch a command or start a command write sequence. 0 No protection violation detected. 1 Protection violation has occurred.
4 ACCERR	**Access Error Flag** — The ACCERR flag indicates an illegal access has occurred to the Flash memory caused by either a violation of the command write sequence (see Section 28.4.1.2, “Command Write Sequence”), issuing an illegal Flash command (see Table 28-16), launching the sector erase abort command terminating a sector erase operation early (see Section 28.4.2.6, “Sector Erase Abort Command”) or the execution of a CPU STOP instruction while a command is executing (CCIF = 0). Writing a 0 to the ACCERR flag has no effect on ACCERR. The ACCERR flag is cleared by writing a 1 to ACCERR.While ACCERR is set, it is not possible to launch a command or start a command write sequence. If ACCERR is set by an erase verify operation or a data compress operation, any buffered command will not launch. 0 No access error detected. 1 Access error has occurred.
2 BLANK	**Flag Indicating the Erase Verify Operation Status** — When the CCIF flag is set after completion of an erase verify command, the BLANK flag indicates the result of the erase verify operation. The BLANK flag is cleared by the Flash module when CBEIF is cleared as part of a new valid command write sequence. Writing to the BLANK flag has no effect on BLANK. 0 Flash block verified as not erased. 1 Flash block verified as erased.
1 FAIL	**Flag Indicating a Failed Flash Operation** — The FAIL flag will set if the erase verify operation fails (selected Flash block verified as not erased). Writing a 0 to the FAIL flag has no effect on FAIL. The FAIL flag is cleared by writing a 1 to FAIL. 0 Flash operation completed without error. 1 Flash operation failed.

29.4.2.5 Mass Erase Command

The mass erase operation will erase all addresses in a Flash block using an embedded algorithm.

An example flow to execute the mass erase operation is shown in Figure 29-28. The mass erase command write sequence is as follows:

1. Write to a Flash block address to start the command write sequence for the mass erase command. The address and data written will be ignored.
2. Write the mass erase command, 0x41, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the mass erase command.

If a Flash block to be erased contains any protected area, the PVIOL flag in the FSTAT register will set and the mass erase command will not launch. Once the mass erase command has successfully launched, the CCIF flag in the FSTAT register will set after the mass erase operation has completed unless a new command write sequence has been buffered.

Table A-29. Example 1b: Normal Expanded Mode Timing V_{DD35} = 5.0 V ($\overline{EWAITE}$ = 1)

No.	C	Characteristic	Symbol	2 Stretch Cycles		3 Stretch Cycles		Unit
				Min	Max	Min	Max	
—	—	Frequency of internal bus	f_i	D.C.	40.0	D.C.	40.0	MHz
—	—	Internal cycle time	t_{cyc}	25	∞	25	∞	ns
—	—	Frequency of external bus	f_o	D.C.	13.3	D.C.	10.0	MHz
—	—	External cycle time (selected by EXSTR)	t_{cyce}	75	∞	100	∞	ns
1	—	External cycle time (EXSTR+1EWAIT)	t_{cycew}	100	∞	125	∞	ns
2	D	Address[1] valid to $\overline{RE}$ fall	t_{ADRE}	5	—	5	—	ns
3	D	Pulse width, $\overline{RE}$ [2]	PW_{RE}	85	—	110	—	ns
4	D	Address[1] valid to $\overline{WE}$ fall	t_{ADWE}	5	—	5	—	ns
5	D	Pulse width, $\overline{WE}$[2]	PW_{WE}	73	—	98	—	ns
6	D	Read data setup time (if ITHRS = 0)	t_{DSR}	24	—	24	—	ns
	D	Read data setup time (if ITHRS = 1)	t_{DSR}	28	—	28	—	ns
7	D	Read data hold time	t_{DHR}	0	—	0	—	ns
8	D	Read enable access time	t_{ACCR}	71	—	86	—	ns
9	D	Write data valid to $\overline{WE}$ fall	t_{WDWE}	7	—	7	—	ns
10	D	Write data setup time	t_{DSW}	81	—	106	—	ns
11	D	Write data hold time	t_{DHW}	8	—	8	—	ns
12	D	Address to $\overline{EWAIT}$ fall	t_{ADWF}	0	20	0	45	ns
13	D	Address to $\overline{EWAIT}$ rise	t_{ADWR}	37	47	62	72	ns

1 Includes the following signals: ADDRx, $\overline{UDS}$, $\overline{LDS}$, and $\overline{CSx}$.

2 Affected by $\overline{EWAIT}$.

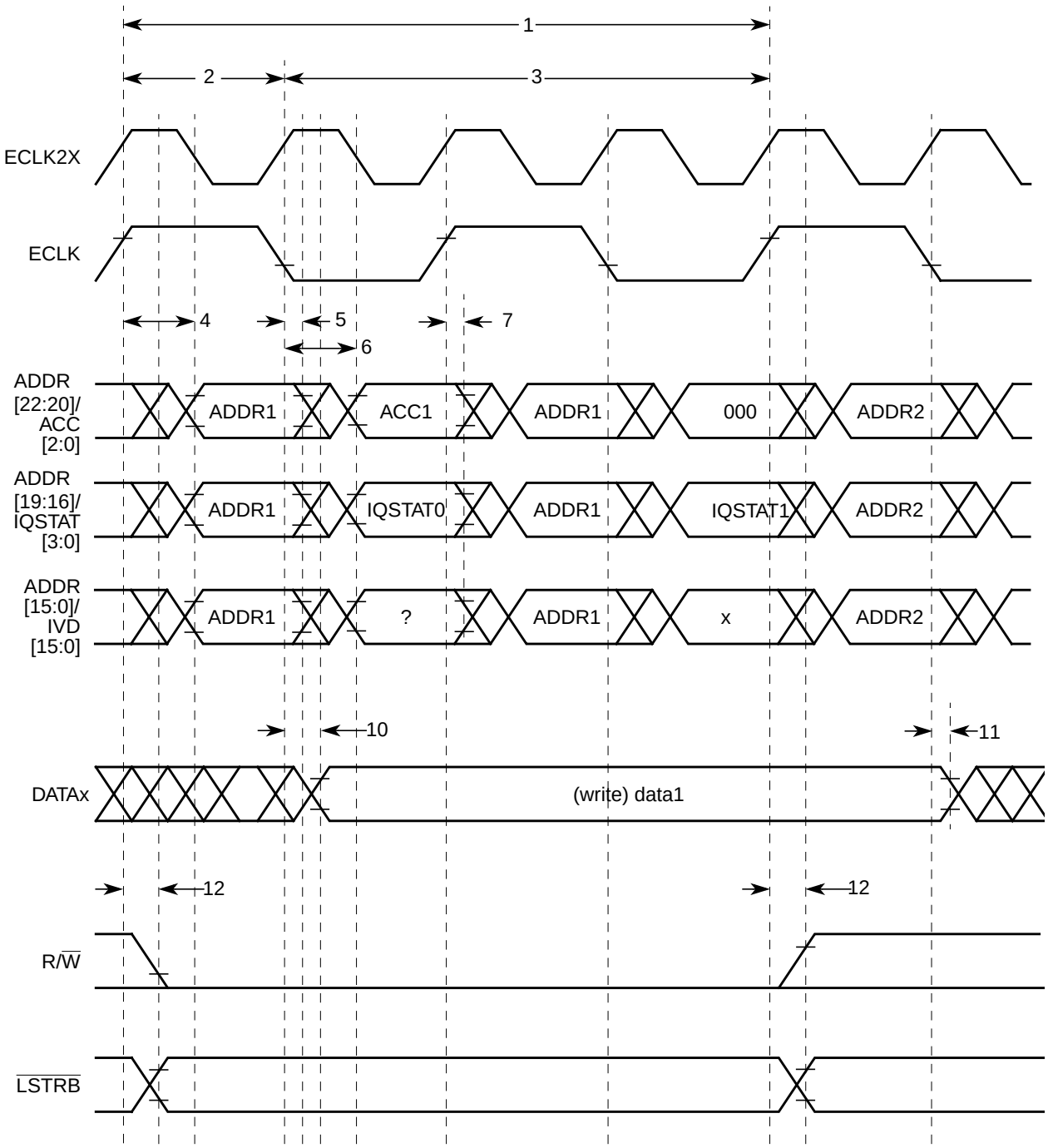

Figure A-16. Example 2b: Emulation Expanded Mode Ò Write with 1 Stretch Cycle

Table A-31. Example 2b: Emulation Expanded Mode Timing V_{DD35} = 5.0 V ($\overline{EWAITE}$ = 0)

No.	C	Characteristic[1]	Symbol	1 Stretch Cycle		2 Stretch Cycles		3 Stretch Cycles		Unit
				Min	Max	Min	Max	Min	Max	
—	—	Internal cycle time	t_{cyc}	25	25	25	25	25	25	ns
1	—	Cycle time	t_{cyce}	50	∞	75	∞	100	∞	ns
2	D	Pulse width, E high	PW_{EH}	11.5	14	11.5	14	11.5	14	ns
3	D	E falling to sampling E rising	t_{EFSR}	35	39.5	60	64.5	85	89.5	ns
4	D	Address delay time	t_{AD}	—	5	—	5	—	5	ns
5	D	Address hold time	t_{AH}	0	—	0	—	0	—	ns
6	D	IVD delay time[2]	t_{IVDD}	—	4.5	—	4.5	—	4.5	ns
7	D	IVD hold time[2]	t_{IVDH}	0	—	0	—	0	—	ns
8	D	Read data setup time Maskset L15Y	t_{DSR}	12	—	12	—	12	—	ns
	D	Read data setup time Maskset M84E	t_{DSR}	15	—	15	—	15	—	ns
9	D	Read data hold time	t_{DHR}	0	—	0	—	0	—	ns
10	D	Write data delay time	t_{DDW}	—	5	—	5	—	5	ns
11	D	Write data hold time	t_{DHW}	0	—	0	—	0	—	ns
12	D	Read/write data delay time[3]	t_{RWD}	–1	5	–1	5	–1	5	ns

1 Typical supply and silicon, room temperature only

2 Includes also ACCx, IQSTATx

3 Includes $\overline{LSTRB}$