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	91
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 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12xa512cal

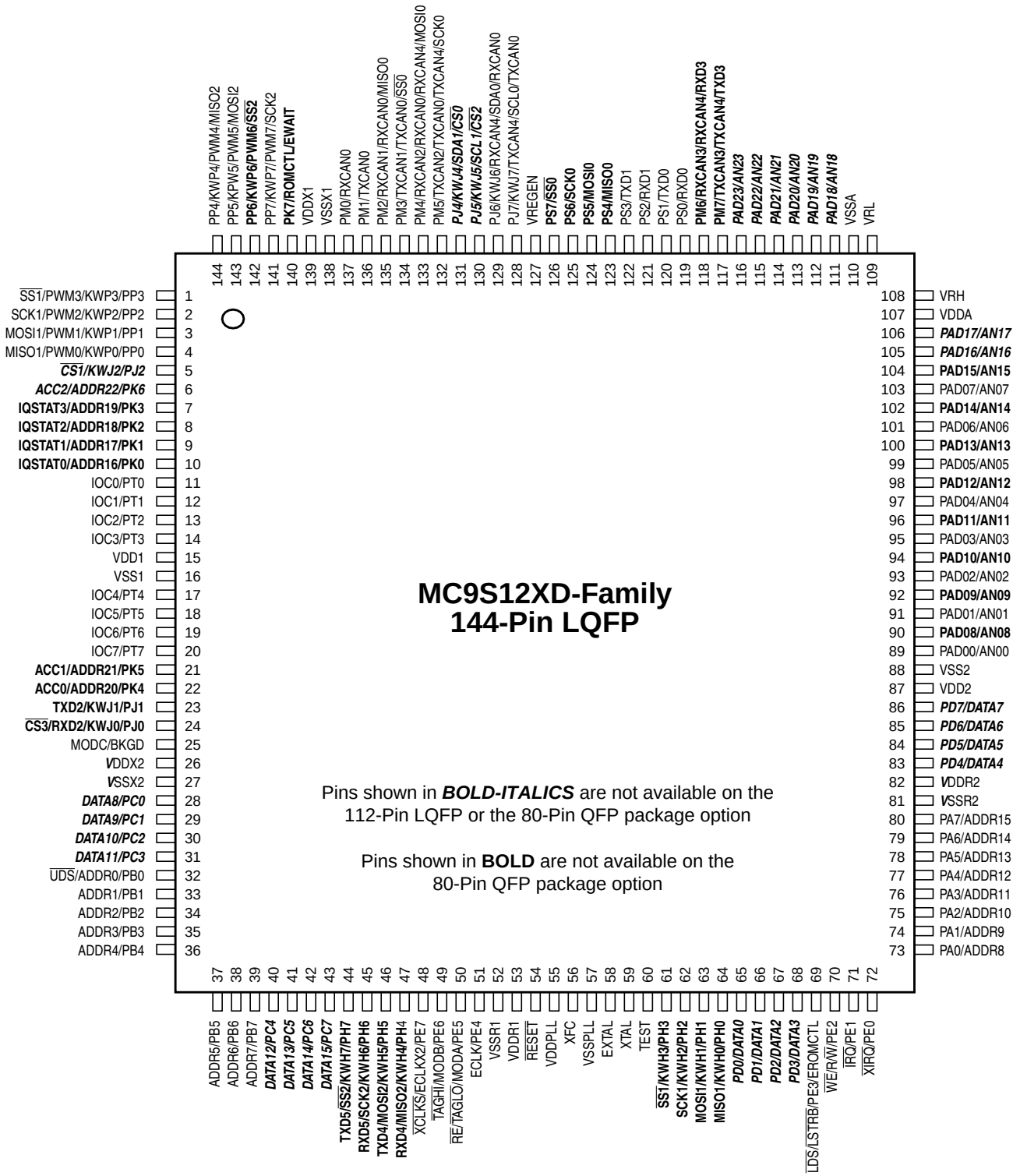

Figure 1-7. MC9S12XD Family Pin Assignment 144-Pin LQFP Package

1.3 System Clock Description

The clock and reset generator module (CRG) provides the internal clock signals for the core and all peripheral modules. Figure 1-12 shows the clock connections from the CRG to all modules.

See 79Chapterf or details on clock generation.

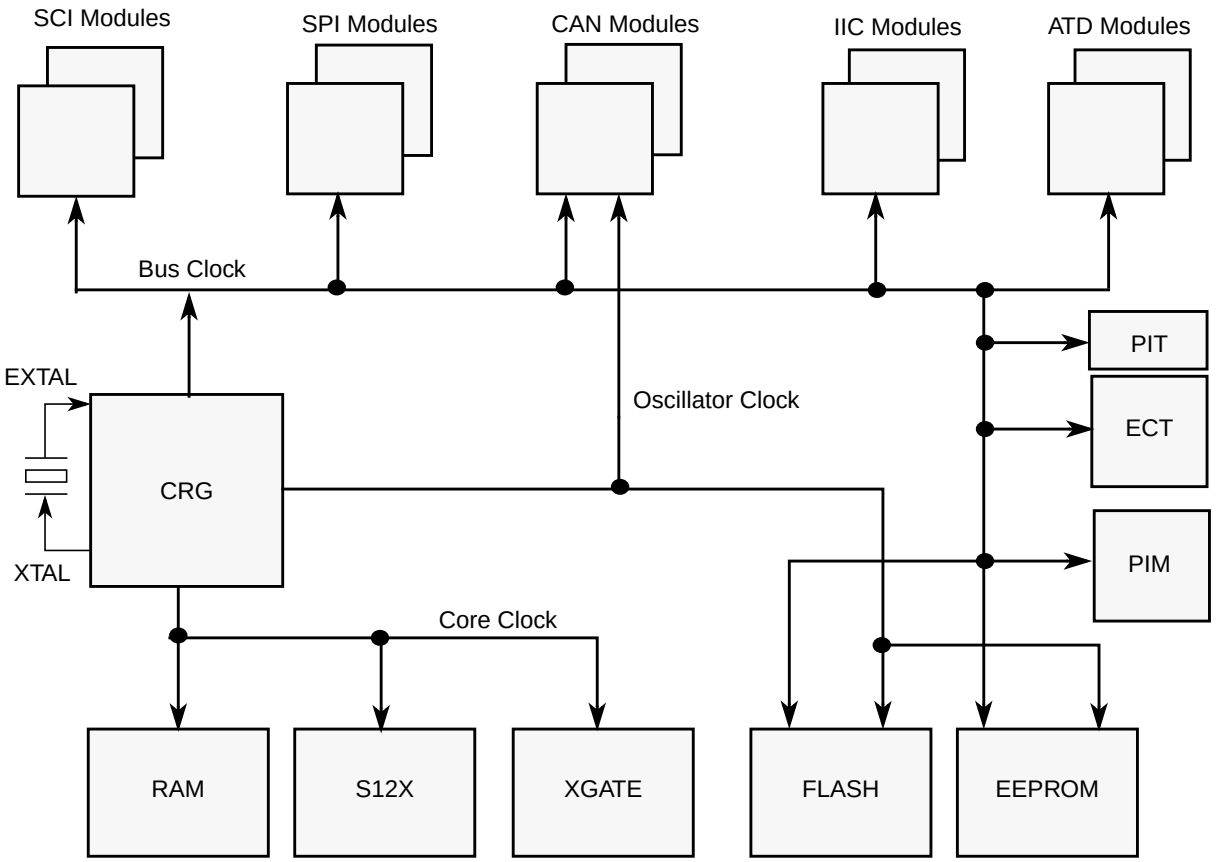

Figure 1-14. MC9S12XD Family Clock Connections

The MCU's system clock can be supplied in several ways enabling a range of system operating frequencies to be supported:

- The on-chip phase locked loop (PLL)
- the PLL self clocking
- the oscillator

The clock generated by the PLL or oscillator provides the main system clock frequencies core clock and bus clock. As shown in Figure 1-12, this system clocks are used throughout the MCU to drive the core, the memories, and the peripherals.

The program Flash memory and the EEPROM are supplied by the bus clock and the oscillator clock.The oscillator clock is used as a time base to derive the program and erase times for the NVM's. See the Flash and EEPROM section for more details on the operation of the NVM's.

Table 2-7. RTI Frequency Divide Rates for RTDEC = 0

RTR[3:0]	RTR[6:4] =							
	000 (OFF)	**001 (2^{10})**	**010 (2^{11})**	**011 (2^{12})**	**100 (2^{13})**	**101 (2^{14})**	**110 (2^{15})**	**111 (2^{16})**
0000 (÷1)	OFF*	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
0001 (÷2)	OFF	$2x2^{10}$	$2x2^{11}$	$2x2^{12}$	$2x2^{13}$	$2x2^{14}$	$2x2^{15}$	$2x2^{16}$
0010 (÷3)	OFF	$3x2^{10}$	$3x2^{11}$	$3x2^{12}$	$3x2^{13}$	$3x2^{14}$	$3x2^{15}$	$3x2^{16}$
0011 (÷4)	OFF	$4x2^{10}$	$4x2^{11}$	$4x2^{12}$	$4x2^{13}$	$4x2^{14}$	$4x2^{15}$	$4x2^{16}$
0100 (÷5)	OFF	$5x2^{10}$	$5x2^{11}$	$5x2^{12}$	$5x2^{13}$	$5x2^{14}$	$5x2^{15}$	$5x2^{16}$
0101 (÷6)	OFF	$6x2^{10}$	$6x2^{11}$	$6x2^{12}$	$6x2^{13}$	$6x2^{14}$	$6x2^{15}$	$6x2^{16}$
0110 (÷7)	OFF	$7x2^{10}$	$7x2^{11}$	$7x2^{12}$	$7x2^{13}$	$7x2^{14}$	$7x2^{15}$	$7x2^{16}$
0111 (÷8)	OFF	$8x2^{10}$	$8x2^{11}$	$8x2^{12}$	$8x2^{13}$	$8x2^{14}$	$8x2^{15}$	$8x2^{16}$
1000 (÷9)	OFF	$9x2^{10}$	$9x2^{11}$	$9x2^{12}$	$9x2^{13}$	$9x2^{14}$	$9x2^{15}$	$9x2^{16}$
1001 (÷10)	OFF	$10x2^{10}$	$10x2^{11}$	$10x2^{12}$	$10x2^{13}$	$10x2^{14}$	$10x2^{15}$	$10x2^{16}$
1010 (÷11)	OFF	$11x2^{10}$	$11x2^{11}$	$11x2^{12}$	$11x2^{13}$	$11x2^{14}$	$11x2^{15}$	$11x2^{16}$
1011 (÷12)	OFF	$12x2^{10}$	$12x2^{11}$	$12x2^{12}$	$12x2^{13}$	$12x2^{14}$	$12x2^{15}$	$12x2^{16}$
1100 (÷13)	OFF	$13x2^{10}$	$13x2^{11}$	$13x2^{12}$	$13x2^{13}$	$13x2^{14}$	$13x2^{15}$	$13x2^{16}$
1101 (÷14)	OFF	$14x2^{10}$	$14x2^{11}$	$14x2^{12}$	$14x2^{13}$	$14x2^{14}$	$14x2^{15}$	$14x2^{16}$
1110 (÷15)	OFF	$15x2^{10}$	$15x2^{11}$	$15x2^{12}$	$15x2^{13}$	$15x2^{14}$	$15x2^{15}$	$15x2^{16}$
1111 (÷16)	OFF	$16x2^{10}$	$16x2^{11}$	$16x2^{12}$	$16x2^{13}$	$16x2^{14}$	$16x2^{15}$	$16x2^{16}$

* Denotes the default value out of reset.This value should be used to disable the RTI to ensure future backwards compatibility.

6.8.6 Instruction Coding

Table 6-17 summarizes all XGATE instructions in the order of their machine coding.

Table 6-17. Instruction Set Summary (Sheet 1 of 3)

Functionality	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Return to Scheduler and Others																
BRK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NOP	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
RTS	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
SIF	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Semaphore Instructions																
CSEM IMM3	0	0	0	0	0	IMM3			1	1	1	1	0	0	0	0
CSEM RS	0	0	0	0	0	RS			1	1	1	1	0	0	0	1
SSEM IMM3	0	0	0	0	0	IMM3			1	1	1	1	0	0	1	0
SSEM RS	0	0	0	0	0	RS			1	1	1	1	0	0	1	1
Single Register Instructions																
SEX RD	0	0	0	0	0	RD			1	1	1	1	0	1	0	0
PAR RD	0	0	0	0	0	RD			1	1	1	1	0	1	0	1
JAL RD	0	0	0	0	0	RD			1	1	1	1	0	1	1	0
SIF RS	0	0	0	0	0	RS			1	1	1	1	0	1	1	1
Special Move instructions																
TFR RD,CCR	0	0	0	0	0	RD			1	1	1	1	1	0	0	0
TFR CCR,RS	0	0	0	0	0	RS			1	1	1	1	1	0	0	1
TFR RD,PC	0	0	0	0	0	RD			1	1	1	1	1	0	1	0
Shift instructions Dyadic																
BFFO RD, RS	0	0	0	0	1	RD			RS			1	0	0	0	0
ASR RD, RS	0	0	0	0	1	RD			RS			1	0	0	0	1
CSL RD, RS	0	0	0	0	1	RD			RS			1	0	0	1	0
CSR RD, RS	0	0	0	0	1	RD			RS			1	0	0	1	1
LSL RD, RS	0	0	0	0	1	RD			RS			1	0	1	0	0
LSR RD, RS	0	0	0	0	1	RD			RS			1	0	1	0	1
ROL RD, RS	0	0	0	0	1	RD			RS			1	0	1	1	0
ROR RD, RS	0	0	0	0	1	RD			RS			1	0	1	1	1
Shift instructions immediate																
ASR RD, #IMM4	0	0	0	0	1	RD			IMM4				1	0	0	1
CSL RD, #IMM4	0	0	0	0	1	RD			IMM4				1	0	1	0
CSR RD, #IMM4	0	0	0	0	1	RD			IMM4				1	0	1	1
LSL RD, #IMM4	0	0	0	0	1	RD			IMM4				1	1	0	0
LSR RD, #IMM4	0	0	0	0	1	RD			IMM4				1	1	0	1
ROL RD, #IMM4	0	0	0	0	1	RD			IMM4				1	1	1	0
ROR RD, #IMM4	0	0	0	0	1	RD			IMM4				1	1	1	1

13.3.0.9 PIT Count Register 0 to 3 (PITCNT0–3)

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

Figure 13-15. PIT Count Register 0 (PITCNT0)

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

Figure 13-16. PIT Count Register 1 (PITCNT1)

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

Figure 13-17. PIT Count Register 2 (PITCNT2)

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

Figure 13-18. PIT Count Register 3 (PITCNT3)

Read: Anytime

Write: Has no meaning or effect

Table 13-9. PITCNT0–3 Field Descriptions

Field	Description
15:0 PCNT[15:0]	**PIT Count Bits 15-0** — These bits represent the current 16-bit modulus down-counter value. The read access for the count register must take place in one clock cycle as a 16-bit access.

13.4 Functional Description

Figure 13-19 shows a detailed block diagram of the PIT module. The main parts of the PIT are status, control and data registers, two 8-bit down-counters, four 16-bit down-counters and an interrupt/trigger interface.

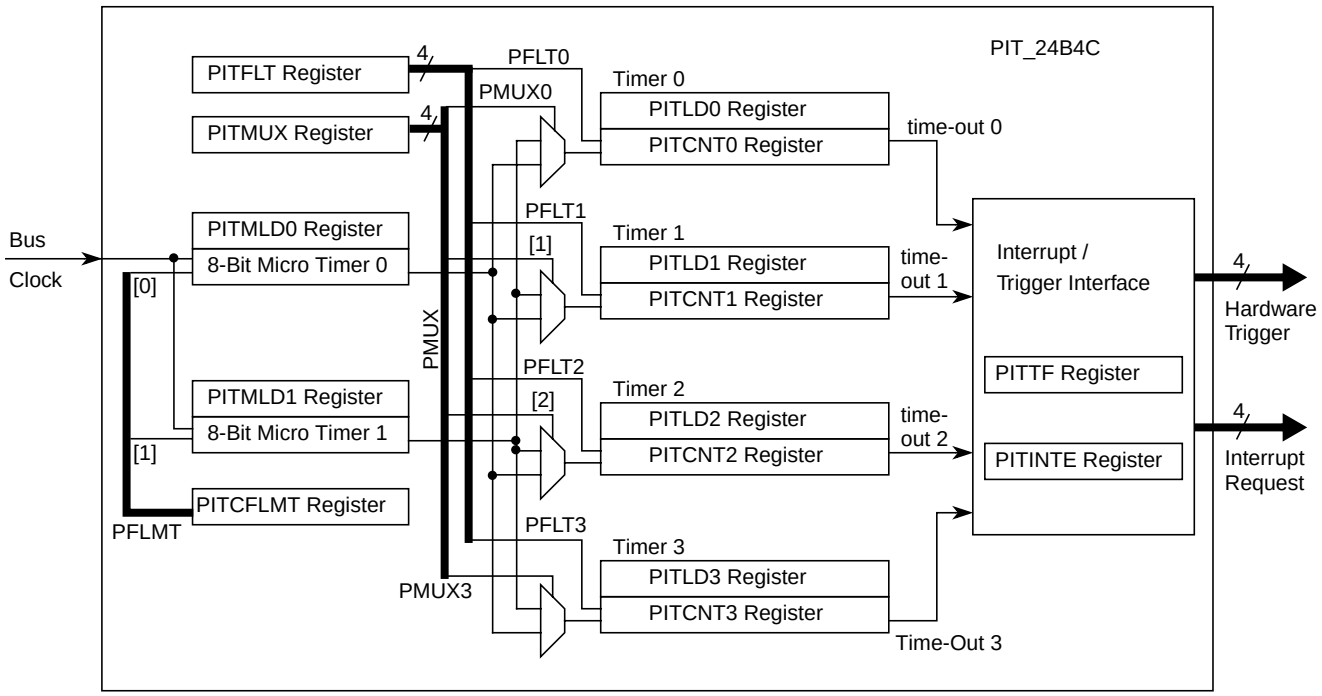

Figure 13-19. PIT Detailed Block Diagram

13.4.1 Timer

As shown in Figure 13-1and Figure 13-19, the 24-bit timers are built in a two-stage architecture with four 16-bit modulus down-counters and two 8-bit modulus down-counters. The 16-bit timers are clocked with two selectable micro time bases which are generated with 8-bit modulus down-counters. Each 16-bit timer is connected to micro time base 0 or 1 via the PMUX[3:0] bit setting in the PIT Multiplex (PITMUX) register.

A timer channel is enabled if the module enable bit PITE in the PIT control and force load micro timer (PITCFLMT) register is set and if the corresponding PCE bit in the PIT channel enable (PITCE) register is set. Two 8-bit modulus down-counters are used to generate two micro time bases. As soon as a micro time base is selected for an enabled timer channel, the corresponding micro timer modulus down-counter will load its start value as specified in the PITMTLD0 or PITMTLD1 register and will start down-counting. Whenever the micro timer down-counter has counted to zero the PITMTLD register is reloaded and the connected 16-bit modulus down-counters count one cycle.

14.2.4 VDDPLL, VSSPLL — Regulator Output2 (PLL) Pins

Signals V_{DDPLL}/V_{SSPLL} are the secondary outputs of VREG_3V3 that provide the power supply for the PLL and oscillator. These signals are connected to device pins to allow external decoupling capacitors (220 nF, X7R ceramic).

In Shutdown Mode, an external supply driving V_{DDPLL}/V_{SSPLL} can replace the voltage regulator.

14.2.5 V_{REGEN} — Optional Regulator Enable Pin

This optional signal is used to shutdown VREG_3V3. In that case, V_{DD}/V_{SS} and V_{DDPLL}/V_{SSPLL} must be provided externally. Shutdown mode is entered with VREGEN being low. If VREGEN is high, the VREG_3V3 is either in Full Performance Mode or in Reduced Power Mode.

For the connectivity of VREGEN, see device specification.

NOTE

Switching from FPM or RPM to shutdown of VREG_3V3 and vice versa is not supported while MCU is powered.

14.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in VREG_3V3.

If enabled in the system, the VREG_3V3 will abort all read and write accesses to reserved registers within it’s memory slice.

14.3.1 Module Memory Map

Table 14-2 provides an overview of all used registers.

Table 14-2. Memory Map

Address Offset	Use	Access
0x0000	HT Control Register (VREGHTCL)	—
0x0001	Control Register (VREGCTRL)	R/W
0x0002	Autonomous Periodical Interrupt Control Register (VREGAPICL)	R/W
0x0003	Autonomous Periodical Interrupt Trimming Register (VREGAPITR)	R/W
0x0004	Autonomous Periodical Interrupt Period High (VREGAPIRH)	R/W
0x0005	Autonomous Periodical Interrupt Period Low (VREGAPIRL)	R/W
0x0006	Reserved 06	—
0x0007	Reserved 07	—

16.3.1 Register Descriptions

This section describes in address order all the XINT registers and their individual bits.

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0121	IVBR	R	IVB_ADDR[7:0]							
		W								
0x0126	INT_XGPRIO	R	0	0	0	0	0	XILVL[2:0]		
		W								
0x0127	INT_CFADDR	R	INT_CFADDR[7:4]				0	0	0	0
		W								
0x0128	INT_CFDATA0	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x0129	INT_CFDATA1	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012A	INT_CFDATA2	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012B	INT_CFDATA3	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012C	INT_CFDATA4	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012D	INT_CFDATA5	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012E	INT_CFDATA6	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012F	INT_CFDATA7	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								

= Unimplemented or Reserved

Figure 16-2. XINT Register Summary

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.

18.3.2.12 RAM Shared Region Upper Boundary Register (RAMSHU)

Address: 0x011F

	7	6	5	4	3	2	1	0
R	1	SHU6	SHU5	SHU4	SHU3	SHU2	SHU1	SHU0
W								
Reset	1	1	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 18-20. RAM Shared Region Upper Boundary Register (RAMSHU)

Read: Anytime

Write: Anytime when RWPE = 0

Table 18-18. RAMSHU Field Descriptions

Field	Description
6–0 SHU[6:0]	**RAM Shared Region Upper Boundary Bits 6–0** — These bits define the upper boundary of the shared memory in multiples of 256 bytes. The block selected by this register is included in the region. See Figure 18-25 for details.

25.4.2.5 Sector Erase Abort Command

The sector erase abort operation will terminate the active sector erase or sector modify operation so that other sectors in an EEPROM block are available for read and program operations without waiting for the sector erase or sector modify operation to complete.

An example flow to execute the sector erase abort operation is shown in Figure 25-22. The sector erase abort command write sequence is as follows:

1. Write to any EEPROM memory address to start the command write sequence for the sector erase abort command. The address and data written are ignored.
2. Write the sector erase abort command, 0x47, to the ECMD register.
3. Clear the CBEIF flag in the ESTAT register by writing a 1 to CBEIF to launch the sector erase abort command.

If the sector erase abort command is launched resulting in the early termination of an active sector erase or sector modify operation, the ACCERR flag will set once the operation completes as indicated by the CCIF flag being set. The ACCERR flag sets to inform the user that the EEPROM sector may not be fully erased and a new sector erase or sector modify command must be launched before programming any location in that specific sector. If the sector erase abort command is launched but the active sector erase or sector modify operation completes normally, the ACCERR flag will not set upon completion of the operation as indicated by the CCIF flag being set. If the sector erase abort command is launched after the sector modify operation has completed the sector erase step, the program step will be allowed to complete. The maximum number of cycles required to abort a sector erase or sector modify operation is equal to four EECLK periods (see Section 25.4.1.1, “Writing the ECLKDIV Register”) plus five bus cycles as measured from the time the CBEIF flag is cleared until the CCIF flag is set.

NOTE

Since the ACCERR bit in the ESTAT register may be set at the completion of the sector erase abort operation, a command write sequence is not allowed to be buffered behind a sector erase abort command write sequence. The CBEIF flag will not set after launching the sector erase abort command to indicate that a command should not be buffered behind it. If an attempt is made to start a new command write sequence with a sector erase abort operation active, the ACCERR flag in the ESTAT register will be set. A new command write sequence may be started after clearing the ACCERR flag, if set.

NOTE

The sector erase abort command should be used sparingly since a sector erase operation that is aborted counts as a complete program/erase cycle.

26.4.2.6 Sector Modify Command

The sector modify operation will erase both words in a sector of EEPROM memory followed by a reprogram of the addressed word using an embedded algorithm.

An example flow to execute the sector modify operation is shown in Figure 26-23. The sector modify command write sequence is as follows:

1. Write to an EEPROM memory address to start the command write sequence for the sector modify command. The EEPROM address written determines the sector to be erased and word to be reprogrammed while byte address bit 0 is ignored.
2. Write the sector modify command, 0x60, to the ECMD register.
3. Clear the CBEIF flag in the ESTAT register by writing a 1 to CBEIF to launch the sector erase command.

If an EEPROM sector to be modified is in a protected area of the EEPROM memory, the PVIOL flag in the ESTAT register will set and the sector modify command will not launch. Once the sector modify command has successfully launched, the CCIF flag in the ESTAT register will set after the sector modify operation has completed unless a new command write sequence has been buffered.

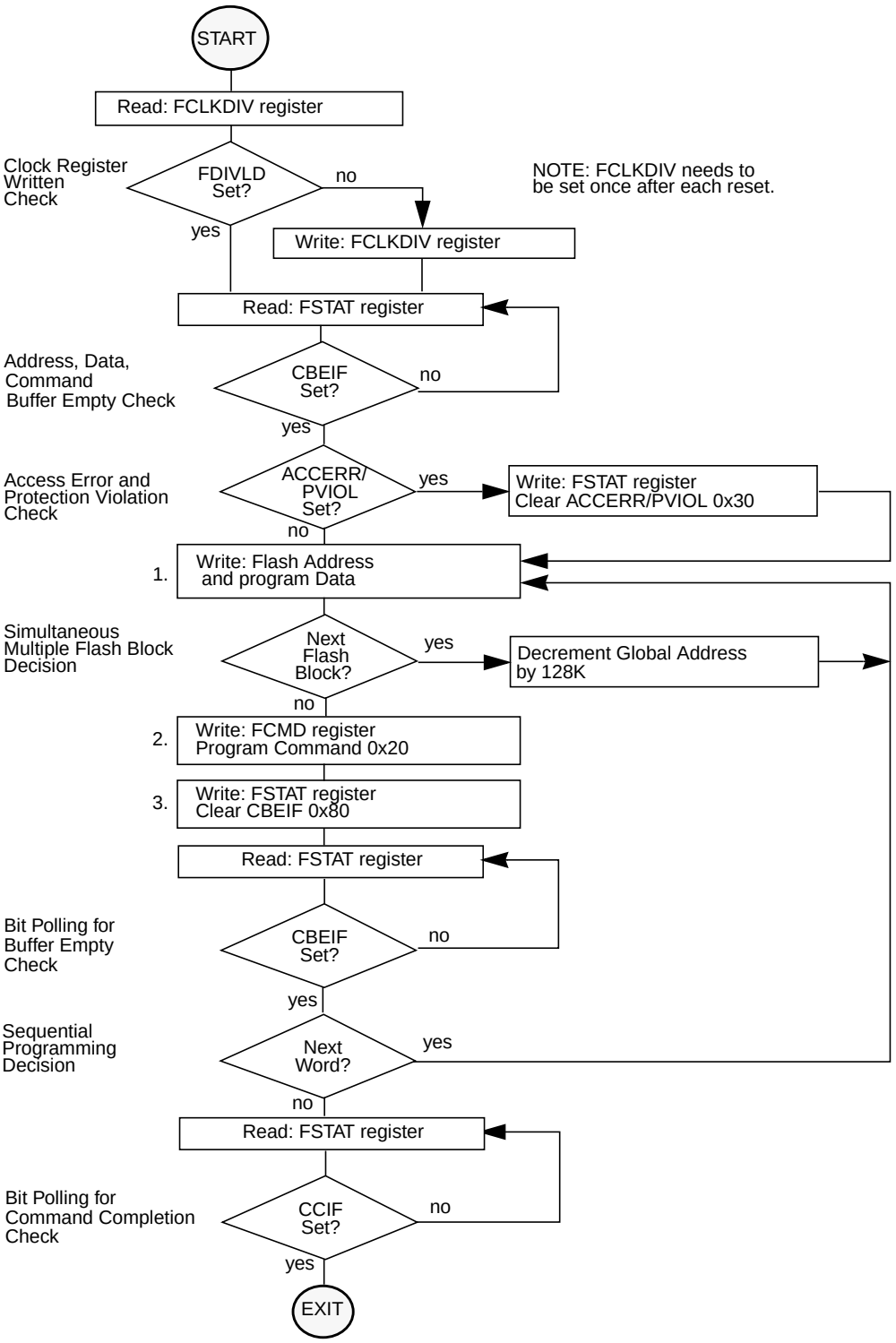

Figure 27-28. Example Program Command Flow

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.

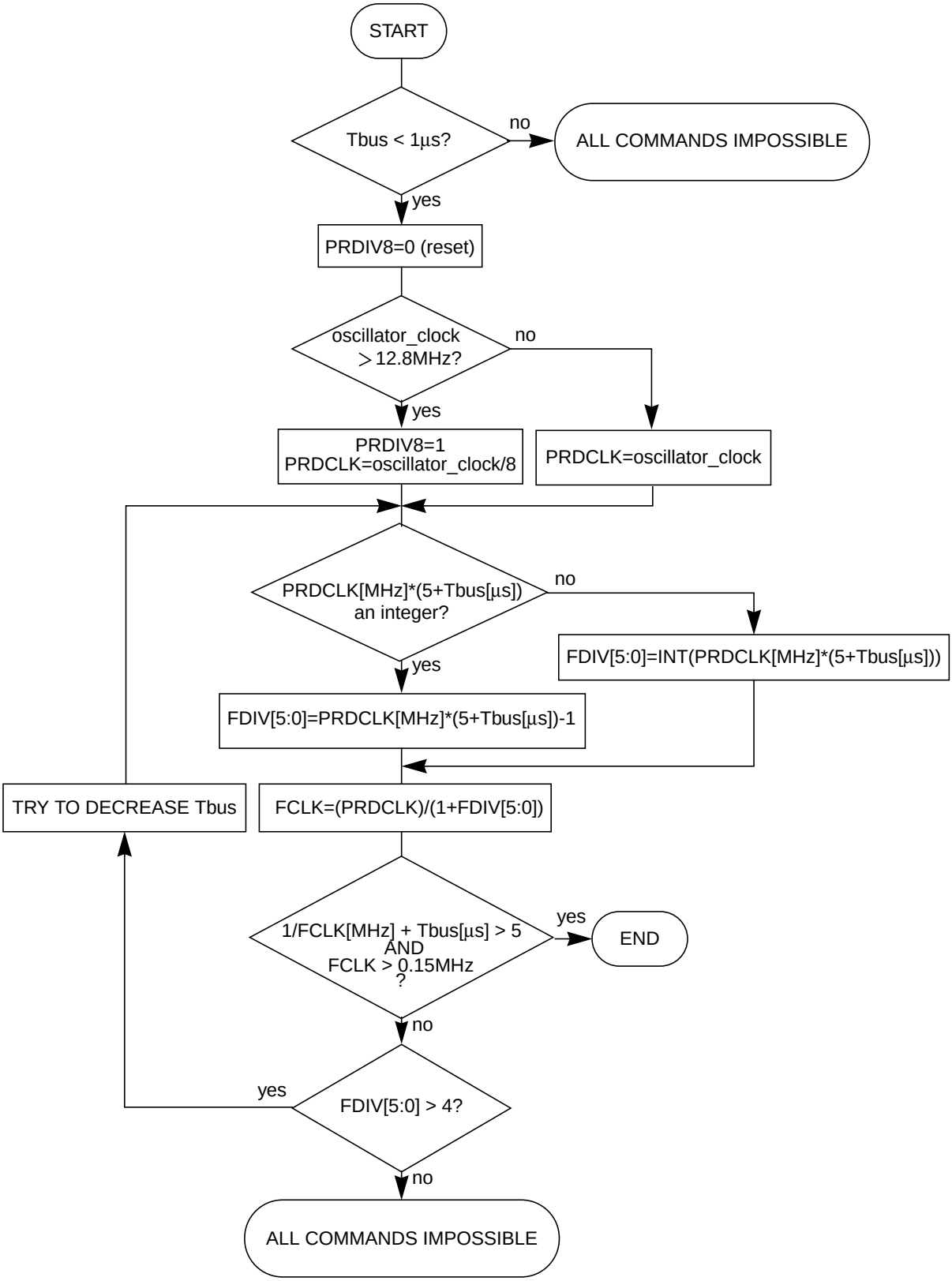

Figure 29-22. Determination Procedure for PRDIV8 and FDIV Bits

A.3 NVM, Flash, and EEPROM

NOTE

Unless otherwise noted the abbreviation NVM (nonvolatile memory) is used for both Flash and EEPROM.

A.3.1 NVM Timing

The time base for all NVM program or erase operations is derived from the oscillator. A minimum oscillator frequency f_{NVMOSC} is required for performing program or erase operations. The NVM modules do not have any means to monitor the frequency and will not prevent program or erase operation at frequencies above or below the specified minimum. Attempting to program or erase the NVM modules at a lower frequency a full program or erase transition is not assured.

The Flash and EEPROM program and erase operations are timed using a clock derived from the oscillator using the FCLKDIV and ECLKDIV registers respectively. The frequency of this clock must be set within the limits specified as f_{NVMOP}.

The minimum program and erase times shown in Table A-17 are calculated for maximum f_{NVMOP} and maximum f_{bus}. The maximum times are calculated for minimum f_{NVMOP} and a f_{bus} of 2 MHz.

A.3.1.1 Single Word Programming

The programming time for single word programming is dependant on the bus frequency as a well as on the frequency f_{NVMOP} and can be calculated according to the following formula.

$$t_{swpgm} = 9 \cdot \frac{1}{f_{NVMOP}} + 25 \cdot \frac{1}{f_{bus}}$$

A.3.1.2 Burst Programming

This applies only to the Flash where up to 64 words in a row can be programmed consecutively using burst programming by keeping the command pipeline filled. The time to program a consecutive word can be calculated as:

$$t_{bwpgm} = 4 \cdot \frac{1}{f_{NVMOP}} + 9 \cdot \frac{1}{f_{bus}}$$

The time to program a whole row is:

$$t_{brpgm} = t_{swpgm} + 63 \cdot t_{bwpgm}$$

Burst programming is more than 2 times faster than single word programming.

A.8 External Bus Timing

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

- Crystal input within 45% to 55% duty
- Equal loads of pins
- Pad full drive (reduced drive must be off)

A.8.1 Normal Expanded Mode (External Wait Feature Disabled)

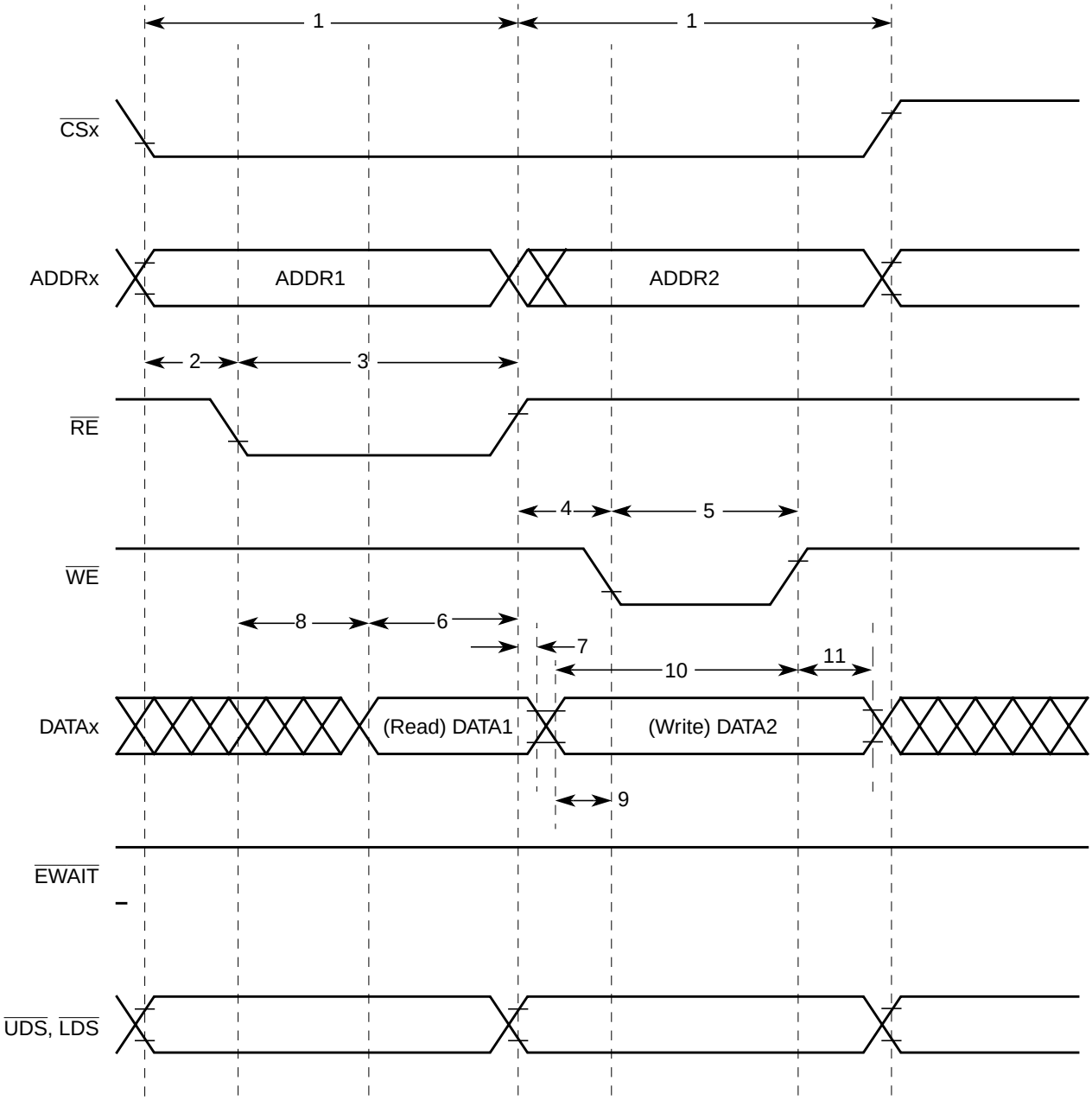

Figure A-11. Example 1a: Normal Expanded Mode — Read Followed by Write

E.5 Peripheral Sets S12XD - Family

Table E-5. S12XD Peripherals

Device	Package	XGATE	CAN	SCI	SPI	IIC	ECT	PIT	ATD0	ATD1	I/O
9S12XDP512	144LQFP	yes	5	6	3	2	8	4	8ch[1]	16ch[2]	119
	112LQFP		5	6	3	1	8	4	8ch[1]	8ch[3]	91
9S12XDT512	144LQFP		3	6	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		3	6	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		3	2	3	1	8	4	8ch[1]		59
9S12XDT384	144LQFP		3	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		3	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		3	2	3	1	8	4	8ch[1]		59
9S12XDQ256	144LQFP		4	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		4	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		4	2	3	1	8	4	8ch[1]		59
9S12XDT256	144LQFP		3	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		3	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		3	2	3	1	8	4	8ch[1]		59
9S12XD256	144LQFP		1	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		1	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		1	2	3	1	8	4	8ch[1]		59
3S12XDT256	144LQFP		3	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		3	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		3	2	3	1	8	4	8ch[1]		59
9S12XDG128	112LQFP	yes but XGATE has no Flash Access	2	2	2	1	8	4		16ch[4]	91
	80QFP		2	2	2	1	8	4		8ch[5]	59
3S12XDG128	112LQFP		2	2	2	1	8	4		16ch[4]	91
	80QFP		2	2	2	1	8	4		8ch[5]	59
9S12XD128	112LQFP		1	2	2	1	8	4		16ch[4]	91
	80QFP		1	2	2	1	8	4		8ch[5]	59
9S12XD64	80QFP		1	2	2	1	8	2		8ch[5]	59

1 ATD0 routed to PAD[7:0]