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	91
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 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°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=mc9s12xd256mal

Section Number	Title	Page

5.3.2.10 ATD Status Register 1 (ATDSTAT1)

This read-only register contains the conversion complete flags.

	7	6	5	4	3	2	1	0
R	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-12. ATD Status Register 1 (ATDSTAT1)

Read: Anytime

Write: Anytime, no effect

Table 5-20. ATDSTAT1 Field Descriptions

Field	Description
7–0 CCF[7:0]	**Conversion Complete Flag x (x = 7, 6, 5, 4, 3, 2, 1, 0)** — A conversion complete flag is set at the end of each conversion in a conversion sequence. The flags are associated with the conversion position in a sequence (and also the result register number). Therefore, CCF0 is set when the first conversion in a sequence is complete and the result is available in result register ATDDR0; CCF1 is set when the second conversion in a sequence is complete and the result is available in ATDDR1, and so forth. A flag CCFx (x = 7, 6, 5, 4, 3, 2,1, 70) is cleared when one of the following occurs: A) Write to ATDCTL5 (a new conversion sequence is started) B) If AFFC=0 and read of ATDSTAT1 followed by read of result register ATDDRx C) If AFFC=1 and read of result register ATDDRx In case of a concurrent set and clear on CCFx: The clearing by method A) will overwrite the set. The clearing by methods B) or C) will be overwritten by the set. 0 Conversion number x not completed 1 Conversion number x has completed, result ready in ATDDRx

BLS Branch if Lower or Same BLS

Operation

If C | Z = 1, then PC + $0002 + (REL9 << 1) ⇒ PC

Branch instruction to compare unsigned numbers.

Branch if RS1 ≤ RS2:

```
SUB     R0,RS1,RS2
BLS     REL9
```

CCR Effects

N	Z	V	C
—	—	—	—

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code								Cycles
BLS REL9	REL9	0	0	1	1	0	0	1	REL9	PP/P

BNE Branch if Not Equal BNE

Operation

If Z = 0, then PC + $0002 + (REL9 << 1) ⇒ PC

Tests the Zero flag and branches if Z = 0.

CCR Effects

N	Z	V	C
—	—	—	—

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code								Cycles
BNE REL9	REL9	0	0	1	0	0	1	0	REL9	PP/P

To calculate the output frequency in center aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by twice the value in the period register for that channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / (2*PWMPERx)
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx = 0)

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

  Duty Cycle = [PWMDTYx / PWMPERx] * 100%

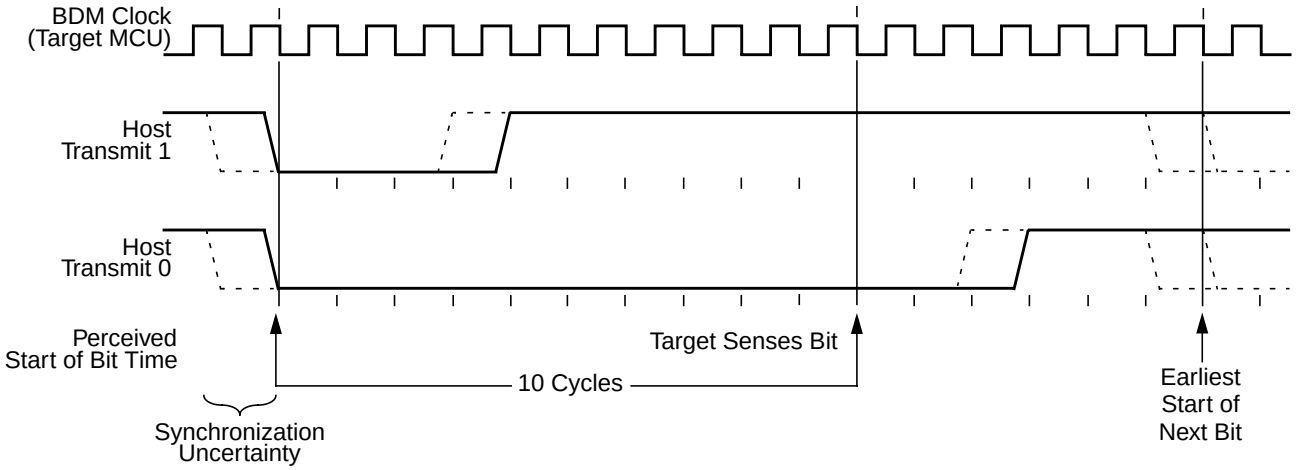

Figure 15-8. BDM Host-to-Target Serial Bit Timing

The receive cases are more complicated. Figure 15-9 shows the host receiving a logic 1 from the target system. Since the host is asynchronous to the target, there is up to one clock-cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target. The host holds the BKGD pin low long enough for the target to recognize it (at least two target clock cycles). The host must release the low drive before the target drives a brief high speedup pulse seven target clock cycles after the perceived start of the bit time. The host should sample the bit level about 10 target clock cycles after it started the bit time.

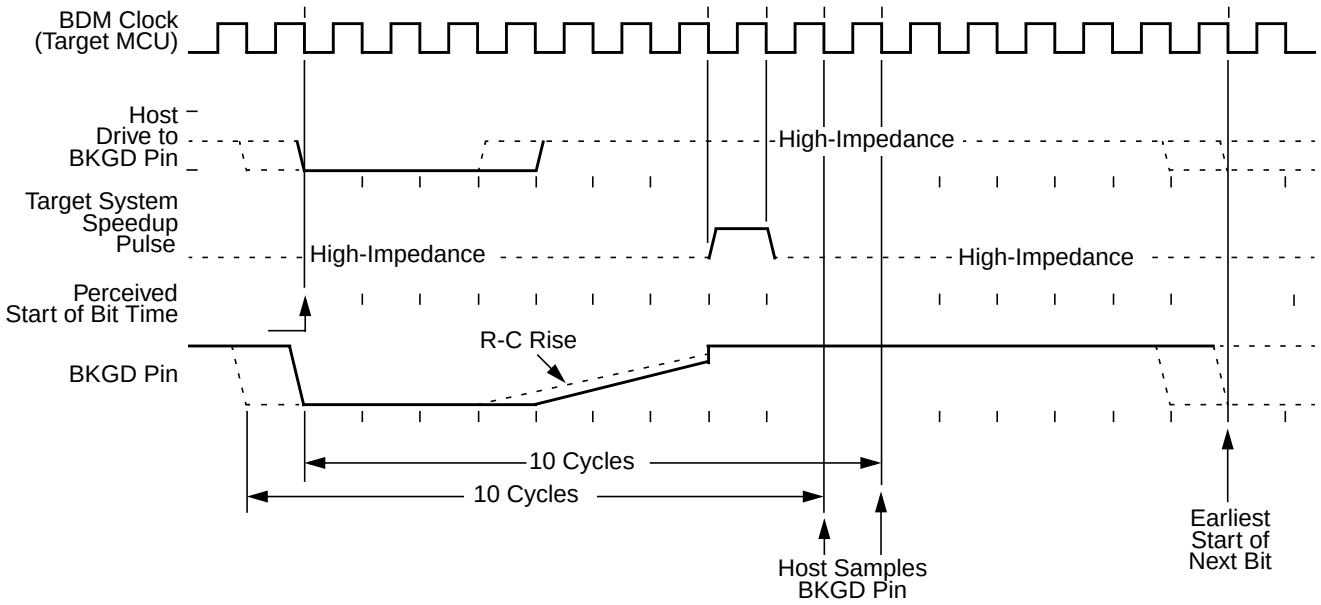

Figure 15-9. BDM Target-to-Host Serial Bit Timing (Logic 1)

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

17.4.4.2 Access Conflicts on Target Buses

The arbitration scheme allows only one master to be connected to a target at any given time. The following rules apply when prioritizing accesses from different masters to the same target bus:

- CPU always has priority over XGATE.
- BDM access has priority over XGATE.
- XGATE access to PRU registers constitutes a special case. It is always granted and stalls the CPU and BDM for its duration.
- In emulation modes all internal accesses are visible on the external bus as well.
- During access to the PRU registers, the external bus is reserved.

17.4.5 Interrupts

17.4.5.1 Outgoing Interrupt Requests

The following interrupt requests can be triggered by the MMC module:

CPU access violation: The CPU access violation signals to the CPU detection of an error condition in the CPU application code which is resulted in write access to the protected XGATE RAM area (see Section 1.4.3.2, “Illegal CPU Accesses”).

17.5 Initialization/Application Information

17.5.1 CALL and RTC Instructions

CALL and RTC instructions are uninterruptable CPU instructions that automate page switching in the program page window. The CALL instruction is similar to the JSR instruction, but the subroutine that is called can be located anywhere in the local address space or in any Flash or ROM page visible through the program page window. The CALL instruction calculates and stacks a return address, stacks the current PPAGE value and writes a new instruction-supplied value to the PPAGE register. The PPAGE value controls which of the 256 possible pages is visible through the 16 Kbyte program page window in the 64 Kbyte local CPU memory map. Execution then begins at the address of the called subroutine.

During the execution of the CALL instruction, the CPU performs the following steps:

1. Writes the current PPAGE value into an internal temporary register and writes the new instruction-supplied PPAGE value into the PPAGE register
2. Calculates the address of the next instruction after the CALL instruction (the return address) and pushes this 16-bit value onto the stack
3. Pushes the temporarily stored PPAGE value onto the stack
4. Calculates the effective address of the subroutine, refills the queue and begins execution at the new address

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x011D	RAMXGU	R	1	XGU6	XGU5	XGU4	XGU3	XGU2	XGU1	XGU0
		W								
0x011E	RAMSHL	R	1	SHL6	SHL5	SHL4	SHL3	SHL2	SHL1	SHL0
		W								
0x011F	RAMSHU	R	1	SHU6	SHU5	SHU4	SHU3	SHU2	SHU1	SHU0
		W								

= Unimplemented or Reserved

Figure 18-2. MMC Register Summary

18.3.2 Register Descriptions

18.3.2.1 MMC Control Register (MMCCTL0)

Address: 0x000A PRR

	7	6	5	4	3	2	1	0
R	0	0	0	0	CS3E	CS2E	CS1E	CS0E
W								
Reset	0	0	0	0	0	0	0	$\overline{\text{ROMON}}$[1]

1. ROMON is bit[0] of the register MMCTL1 (see Figure 18-10)

= Unimplemented or Reserved

Figure 18-3. MMC Control Register (MMCCTL0)

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

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

Table 18-4. Chip Selects Function Activity

Register Bit	Chip Modes					
	NS	SS	NX	ES	EX	ST
CS3E, CS2E, CS1E, CS0E	Disabled[1]	Disabled	Enabled[2]	Disabled	Enabled	Enabled

1 Disabled: feature always inactive.

2 Enabled: activity is controlled by the appropriate register bit value.

The MMCCTL0 register is used to control external bus functions, i.e., availability of chip selects.

CAUTION

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

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	W								
PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
	W								
DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
	W								
DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
	W								
PORTC	R	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
	W								
PORTD	R	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	W								
DDRC	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
	W								
DDRD	R	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
	W								
PORTE	R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
	W								
DDRE	R	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	0	0
	W								
Non-PIM Address Range	R	Non-PIM Address Range							
	W								
PUCR	R	PUPKE	BKPUE	0	PUPEE	PUPDE	PUPCE	PUPBE	PUPAE
	W								
RDRIV	R	RDPK	0	0	RDPE	RDPD	RDPC	RDPB	RDPA
	W								
Non-PIM Address Range	R	Non-PIM Address Range							
	W								

= Unimplemented or Reserved

Figure 22-2. PIM Register Summary (Sheet 1 of 6)

22.3.2.24 Port S Input Register (PTIS)

	7	6	5	4	3	2	1	0
R	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
W								
Reset[1]	—	—	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 22-26. Port S Input Register (PTIS)

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.

Write: Never, writes to this register have no effect.

This register always reads back the buffered state of the associated pins. This also can be used to detect overload or short circuit conditions on output pins.

Table 23-42. PPSP Field Descriptions

Field	Description
7–0 PPSP[7:0]	**Polarity Select Port P** 0 Falling edge on the associated port P pin sets the associated flag bit in the PIFP register.A pull-up device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input. 1 Rising edge on the associated port P pin sets the associated flag bit in the PIFP register.A pull-down device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input.

23.0.5.44 Port P Interrupt Enable Register (PIEP)

	7	6	5	4	3	2	1	0
R W	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
Reset	0	0	0	0	0	0	0	0

Figure 23-46. Port P Interrupt Enable Register (PIEP)

Read: Anytime.

Write: Anytime.

This register disables or enables on a per-pin basis the edge sensitive external interrupt associated with Port P.

Table 23-43. PIEP Field Descriptions

Field	Description
7–0 PIEP[7:0]	**Interrupt Enable Port P** 0 Interrupt is disabled (interrupt flag masked). 1 Interrupt is enabled.

23.0.5.45 Port P Interrupt Flag Register (PIFP)

	7	6	5	4	3	2	1	0
R W	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
Reset	0	0	0	0	0	0	0	0

Figure 23-47. Port P Interrupt Flag Register (PIFP)

Read: Anytime.

Write: Anytime.

25.3.2 Register Descriptions

The EEPROM module also contains a set of 12 control and status registers located between EEPROM module base + 0x0000 and 0x000B. A summary of the EEPROM module registers is given in Figure 25-3. Detailed descriptions of each register bit are provided.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
ECLKDIV	R	EDIVLD	PRDIV8	EDIV5	EDIV4	EDIV3	EDIV2	EDIV1	EDIV0
	W								
RESERVED1	R	0	0	0	0	0	0	0	0
	W								
RESERVED2	R	0	0	0	0	0	0	0	0
	W								
ECNFG	R	CBEIE	CCIE	0	0	0	0	0	0
	W								
EPROT	R	EPOPEN	RNV6	RNV5	RNV4	EPDIS	EPS2	EPS1	EPS0
	W								
ESTAT	R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	0	0
	W								
ECMD	R	0	CMDB						
	W								
RESERVED3	R	0	0	0	0	0	0	0	0
	W								
EADDRHI	R	0	0	0	0	0	0	EABHI	
	W								
EADDRLO	R	EABLO							
	W								
EDATAHI	R	EDHI							
	W								
EDATALO	R	EDLO							
	W								

= Unimplemented or Reserved

Figure 25-3. EETX2K Register Summary

The FCTL register is loaded from the Flash Configuration Field byte at global address 0x7F_FF0E during the reset sequence, indicated by F in Figure 27-15.

Table 27-19. FCTL Field Descriptions

Field	Description
7-0 NV[7:0]	**Non volatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the Device User Guide for proper use of the NV bits.

27.3.2.9 Flash Address Registers (FADDR)

The FADDRHI and FADDRLO registers are the Flash address registers.

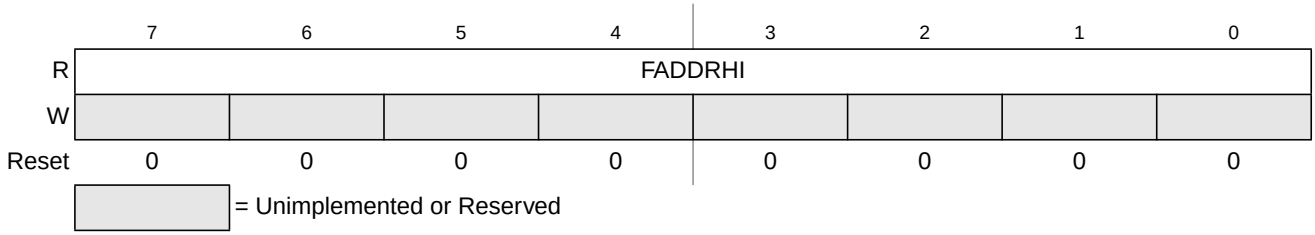

Figure 27-16. Flash Address High Register (FADDRHI)

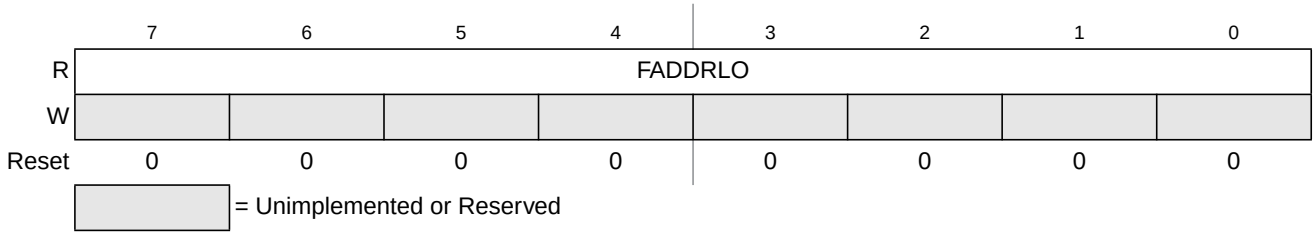

Figure 27-17. Flash Address Low Register (FADDRLO)

All FADDRHI and FADDRLO bits are readable but are not writable. After an array write as part of a command write sequence, the FADDR registers will contain the mapped MCU address written.

27.3.2.10 Flash Data Registers (FDATA)

The FDATAHI and FDATALO registers are the Flash data registers.

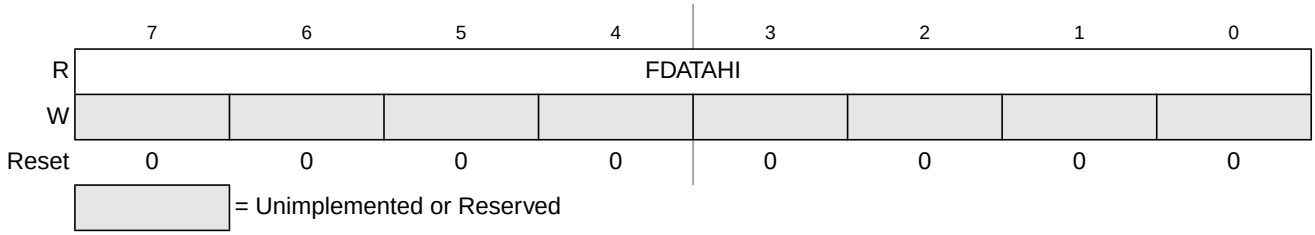

Figure 27-18. Flash Data High Register (FDATAHI)

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 29-14. 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 29-30). 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 29-30). 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 29.4.1.2, “Command Write Sequence”), issuing an illegal Flash command (see Table 29-16), launching the sector erase abort command terminating a sector erase operation early (see Section 29.4.2.6, “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 (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.

A.5 Reset, Oscillator, and PLL

This section summarizes the electrical characteristics of the various startup scenarios for oscillator and phase-locked loop (PLL).

A.5.1 Startup

Table A-21 summarizes several startup characteristics explained in this section. Detailed description of the startup behavior can be found in the Clock and Reset Generator (CRG) Block Guide.

Table A-21. Startup Characteristics

Conditions are shown in Table A-4unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	D	Reset input pulse width, minimum input time	PW_{RSTL}	2	—	—	t_{osc}
2	D	Startup from reset	n_{RST}	192	—	196	n_{osc}
3	D	Interrupt pulse width, $\overline{IRQ}$ edge-sensitive mode	PW_{IRQ}	25[1]	—	—	ns
4	D	Wait recovery startup time	t_{WRS}	—	—	14	t_{cyc}
5	D	Fast wakeup from STOP[2]	t_{fws}	—	50	—	µs

1 1 t_{cycle} at 40Mhz Bus Clock

2 V_{DD1}/V_{DD2} filter capacitors 220 nF, V_{DD35} = 5 V, T= 25°C

A.5.1.1 POR

The release level V_{PORR} and the assert level V_{PORA} are derived from the V_{DD} supply. They are also valid if the device is powered externally. After releasing the POR reset the oscillator and the clock quality check are started. If after a time t_{CQOUT} no valid oscillation is detected, the MCU will start using the internal self clock. The fastest startup time possible is given by n_{uposc}.

A.5.1.2 SRAM Data Retention

Provided an appropriate external reset signal is applied to the MCU, preventing the CPU from executing code when V_{DD35} is out of specification limits, the SRAM contents integrity is guaranteed if after the reset the PORF bit in the CRG flags register has not been set.

A.5.1.3 External Reset

When external reset is asserted for a time greater than PW_{RSTL} the CRG module generates an internal reset, and the CPU starts fetching the reset vector without doing a clock quality check, if there was an oscillation before reset.

A.5.1.4 Stop Recovery

Out of stop the controller can be woken up by an external interrupt. A clock quality check as after POR is performed before releasing the clocks to the system.

Appendix G
Detailed Register Map

The following tables show the detailed register map of the MC9S12XD-Family.

0x0000–0x0009 Port Integration Module (PIM) Map 1 of 5

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0000	PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA 0
		W								
0x0001	PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		W								
0x0002	DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
0x0003	DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
0x0004	PORTC	R	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		W								
0x0005	PORTD	R	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
		W								
0x0006	DDRC	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W								
0x0007	DDRD	R	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		W								
0x0008	PORTE	R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
		W								
0x0009	DDRE	R	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	0	0
		W								

0x000A–0x000B Module Mapping Control (S12XMMC) Map 1 of 4

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000A	MMCCTL0	R	0	0	0	0	0	CS2E	CS1E	CS0E
		W								
0x000B	MODE	R	MODC	MODB	MODA	0	0	0	0	0
		W								

0x000C–0x000D Port Integration Module (PIM) Map 2 of 5

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000C	PUCR	R	PUPKE	BKPUE	0	PUPEE	PUPDE	PUPCE	PUPBE	PUPAE
		W								
0x000D	RDRIV	R	RDPK	0	0	RDPE	RDPD	RDPC	RDPB	RDPA
		W								

0x0040–0x007F Enhanced Capture Timer 16-Bit 8-Channels (ECT) Map (Sheet 1 of 3)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0040	TIOS	R	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
		W								
0x0041	CFORC	R	0	0	0	0	0	0	0	0
		W	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
0x0042	OC7M	R	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
		W								
0x0043	OC7D	R	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
		W								
0x0044	TCNT (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x0045	TCNT (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0046	TSCR1	R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
		W								
0x0047	TTOV	R	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
		W								
0x0048	TCTL1	R	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
		W								
0x0049	TCTL2	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
		W								
0x004A	TCTL3	R	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
		W								
0x004B	TCTL4	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
		W								
0x004C	TIE	R	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I
		W								
0x004D	TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
		W								
0x004E	TFLG1	R	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F
		W								
0x004F	TFLG2	R	TOF	0	0	0	0	0	0	0
		W								
0x0050	TC0 (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x0051	TC0 (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0052	TC1 (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x0053	TC1 (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0054	TC2 (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x0055	TC2 (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								