



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Not For New Designs
Core Processor	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I <sup>2</sup> C, LINbus, SCI, SPI
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	117
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 16x8/10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xhz512cag

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



	8.4.4	Stop Mode Operation
		Chapter 9
		Analog-to-Digital Converter (ATD10B16CV4)
		Block Description
91	Introdu	ction 377
<i>,</i> ,,,	9.1.1	Features
	9.1.2	Modes of Operation
	9.1.3	Block Diagram
9.2	Externa	l Signal Description
	9.2.1	ANx (x = 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0) — Analog Input Channel <i>x</i> Pins
		379
	9.2.2	ETRIG3, ETRIG2, ETRIG1, ETRIG0 — External Trigger Pins
	9.2.3	V <sub>RH</sub> , V <sub>RL</sub> — High Reference Voltage Pin, Low Reference Voltage Pin
	9.2.4	V <sub>DDA</sub> , V <sub>SSA</sub> — Analog Circuitry Power Supply Pins
9.3	Memor	y Map and Register Definition
	9.3.1	Module Memory Map
	9.3.2	Register Descriptions
9.4	Functio	nal Description
	9.4.1	Analog Sub-block
	9.4.2	Digital Sub-Block
~ ~	9.4.3	Operation in Low Power Modes
9.5	Initializ	ation/Application Information
	9.5.1	Setting up and starting an A/D conversion
0.6	9.5.2	Aborting an A/D conversion
9.6	Kesets	
9./	Interrup	ns

# Chapter 10

# Liquid Crystal Display (LCD32F4BV1) Block Description

10.1	Introduc	tion	1
	10.1.1	Features	1
	10.1.2	Modes of Operation	1
	10.1.3	Block Diagram	2
10.2	External	Signal Description	3
	10.2.1	BP[3:0] — Analog Backplane Pins	3
	10.2.2	FP[31:0] — Analog Frontplane Pins	3
	10.2.3	VLCD — LCD Supply Voltage Pin	3
10.3	Memory	Map and Register Definition	3
	10.3.1	Module Memory Map	3
	10.3.2	Register Descriptions	5



Chapter 4 4 Kbyte EEPROM Module (S12XEETX4KV2)

# 4.1.4 Block Diagram

A block diagram of the EEPROM module is shown in Figure 4-1.



Figure 4-1. EETX4K Block Diagram

# 4.2 External Signal Description

The EEPROM module contains no signals that connect off-chip.

# 4.3 Memory Map and Register Definition

This section describes the memory map and registers for the EEPROM module.

## 4.3.1 Module Memory Map

The EEPROM memory map is shown in Figure 4-2. The HCS12X architecture places the EEPROM memory addresses between global addresses 0x13\_F000 and 0x13\_FFFF. The EPROT register, described in Section 4.3.2.5, "EEPROM Protection Register (EPROT)", can be set to protect the upper region in the EEPROM memory from accidental program or erase. The EEPROM addresses covered by this protectable



Chapter 4 4 Kbyte EEPROM Module (S12XEETX4KV2)

## 4.4.1.2 Command Write Sequence

The EEPROM command controller is used to supervise the command write sequence to execute program, erase, erase verify, sector erase abort, and sector modify algorithms.

Before starting a command write sequence, the ACCERR and PVIOL flags in the ESTAT register must be clear (see Section 4.3.2.6, "EEPROM Status Register (ESTAT)") and the CBEIF flag should be tested to determine the state of the address, data and command buffers. If the CBEIF flag is set, indicating the buffers are empty, a new command write sequence can be started. If the CBEIF flag is clear, indicating the buffers are not available, a new command write sequence will overwrite the contents of the address, data and command buffers.

A command write sequence consists of three steps which must be strictly adhered to with writes to the EEPROM module not permitted between the steps. However, EEPROM register and array reads are allowed during a command write sequence. The basic command write sequence is as follows:

- 1. Write to one address in the EEPROM memory.
- 2. Write a valid command to the ECMD register.
- 3. Clear the CBEIF flag in the ESTAT register by writing a 1 to CBEIF to launch the command.

The address written in step 1 will be stored in the EADDR registers and the data will be stored in the EDATA registers. If the CBEIF flag in the ESTAT register is clear when the first EEPROM array write occurs, the contents of the address and data buffers will be overwritten and the CBEIF flag will be set. When the CBEIF flag is cleared, the CCIF flag is cleared on the same bus cycle by the EEPROM command controller indicating that the command was successfully launched. For all command write sequences except sector erase abort, the CBEIF flag will set four bus cycles after the CCIF flag is cleared indicating that the address, data, and command buffers are ready for a new command write sequence to begin. For sector erase abort operations, the CBEIF flag will remain clear until the operation completes. Except for the sector erase abort command, a buffered command will wait for the active operation to be completed before being launched. The sector erase abort command is launched when the CBEIF flag is cleared as part of a sector erase abort command write sequence. Once a command is launched, the completion of the command operation is indicated by the setting of the CCIF flag in the ESTAT register. The CCIF flag will set upon completion of all active and buffered commands.

# 4.4.2 EEPROM Commands

Table 4-9 summarizes the valid EEPROM commands along with the effects of the commands on the EEPROM block.

ECMDB	Command	Function on EEPROM Memory
0x05	Erase Verify	Verify all memory bytes in the EEPROM block are erased. If the EEPROM block is erased, the BLANK flag in the ESTAT register will set upon command completion.
0x20	Program	Program a word (two bytes) in the EEPROM block.
0x40	Sector Erase	Erase all four memory bytes in a sector of the EEPROM block.

Table 4-9. EEPROM Command Description

Chapter 5 XGATE (S12XGATEV2)

Field	Description
4 XGSS	<ul> <li>XGATE Single Step — This bit forces the execution of a single instruction if the XGATE is in DEBUG Mode and no software error has occurred (XGSWEIF cleared).</li> <li>Read:</li> <li>0 No single step in progress</li> <li>1 Single step in progress</li> <li>Write</li> <li>0 No effect</li> <li>1 Execute a single RISC instruction</li> <li>Note: Invoking a Single Step will cause the XGATE to temporarily leave Debug Mode until the instruction has been executed.</li> </ul>
3 XGFACT	<ul> <li>Fake XGATE Activity — This bit forces the XGATE to flag activity to the MCU even when it is idle. When it is set the MCU will never enter system stop mode which assures that peripheral modules will be clocked during XGATE idle periods</li> <li>Read:</li> <li>0 XGATE will only flag activity if it is not idle or in debug mode.</li> <li>1 XGATE will always signal activity to the MCU.</li> <li>Write:</li> <li>0 Only flag activity if not idle or in debug mode.</li> <li>1 Always signal XGATE activity.</li> </ul>
1 XGSWEIF	<ul> <li>XGATE Software Error Interrupt Flag — This bit signals a pending Software Error Interrupt. It is set if the RISC core detects an error condition (see Section 5.4.5, "Software Error Detection"). The RISC core is stopped while this bit is set. Clearing this bit will terminate the current thread and cause the XGATE to become idle. Read:</li> <li>0 Software Error Interrupt is not pending</li> <li>1 Software Error Interrupt is pending if XGIE is set</li> <li>Write:</li> <li>0 No effect</li> <li>1 Clears the XGSWEIF bit</li> </ul>
0 XGIE	XGATE Interrupt Enable — This bit acts as a global interrupt enable for the XGATE module         Read:         0 All XGATE interrupts disabled         1 All XGATE interrupts enabled         Write:         0 Disable all XGATE interrupts         1 Enable all XGATE interrupts

## Table 5-1. XGMCTL Field Descriptions (Sheet 3 of 3)





## Logical AND Immediate 8 bit Constant (Low Byte)



## Operation

RD.L & IMM8  $\Rightarrow$  RD.L

Performs a bit wise logical AND 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**

Ν	Ζ	V	С

 $|\Delta|\Delta|0|-$ 

- 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	
ANDL RD, #IMM8	IMM8	1	0	0	0	0	RD	IMM8	Р



LDW

## Load Word from Memory

# LDW

## Operation

$$\begin{split} M[RB, \#OFFS5] &\Rightarrow RD \\ M[RB, RI] &\Rightarrow RD \\ M[RB, RI] &\Rightarrow RD; RI+2 \Rightarrow RI^1 \\ RI-2 &\Rightarrow RI; M[RS, RI] \Rightarrow RD \\ IMM16 &\Rightarrow RD (translates to LDL RD, \#IMM16[7:0]; LDH RD, \#IMM16[15:8]) \end{split}$$

Loads a 16 bit value into the register RD.

## **CCR Effects**

	Ν	Ζ	V	С
Γ				

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

## **Code and CPU Cycles**

Source Form	Address Mode		Machine Code							Cycles		
LDW RD, (RB, #OFFS5)	IDO5	0	1	0	0	1	RD	RB	OFFS5		PR	
LDW RD, (RB, RI)	IDR	0	1	1	0	1	RD	RB	RI	0	0	PR
LDW RD, (RB, RI+)	IDR+	0	1	1	0	1	RD	RB	RI	0	1	PR
LDW RD, (RB, -RI)	-IDR	0	1	1	0	1	RD	RB	RI	1	0	PR
LDW RD, #IMM16	IMM8	1	1	1	1	0	RD	IMM16[7:0]		Р		
	IMM8	1	1	1	1	1	RD	IMM16[15:8]			Р	

1. If the same general purpose register is used as index (RI) and destination register (RD), the content of the register will not be incremented after the data move: M[RB, RI] ⇒ RD



EXTAL input frequency. In full stop mode (PSTP = 0), the EXTAL pin is pulled down by an internal resistor of typical 200 k $\Omega$ .

## NOTE

Freescale recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier.

Loop controlled circuit is not suited for overtone resonators and crystals.



Figure 8-2. Loop Controlled Pierce Oscillator Connections (XCLKS = 0)

NOTE

Full swing Pierce circuit is not suited for overtone resonators and crystals without a careful component selection.



\* R<sub>s</sub> can be zero (shorted) when use with higher frequency crystals. Refer to manufacturer's data.







MC9S12XHZ512 Data Sheet, Rev. 1.06



## 11.4.1.3.5 Dither Bit (DITH)

The purpose of the dither mode is to increase the minimum length of output pulses without decreasing the PWM resolution, in order to limit the pulse distortion introduced by the slew rate control of the outputs. If dither mode is selected the output pattern will repeat after two timer counter overflows. For the same output frequency, the shortest output pulse will have twice the length while dither feature is selected. To achieve the same output frame frequency, the prescaler of the MC10B12C module has to be set to twice the division rate if dither mode is selected; e.g., with the same prescaler division rate the repeat rate of the output pattern is the same as well as the shortest output pulse with or without dither mode selected.

The DITH bit in control register 0 enables or disables the dither function.

DITH = 0: dither function is disabled.

When DITH is cleared and assuming left aligned operation and RECIRC = 0, the PWM output will start at a logic low level at the beginning of the PWM period (motor controller timer counter = 0x000). The PWM output remains low until the motor controller timer counter matches the 11-bit PWM duty cycle value, DUTY, contained in D[10:0] in MCDCx. When a match (output compare between motor controller timer counter and DUTY) occurs, the PWM output will toggle to a logic high level and will remain at a logic high level until the motor controller timer counter overflows (reaches the contents of MCPER – 1). After the motor controller timer counter resets to 0x000, the PWM output will return to a logic low level. This completes one PWM period. The PWM period repeats every P counts (as defined by the bits P[10:0] in the motor controller period register) of the motor controller timer counter. If DUTY >= P, the output will be static low. If DUTY = 0x0000, the output will be continuously at a logic high level. The relationship between the motor controller timer clock, motor controller timer counter value, and PWM output while DITH = 0 is shown in Figure 11-17.



DITH = 1: dither function is enabled

Please note if DITH = 1, the bit P0 in the motor controller period register will be internally forced to 0 and read always as 0.

When DITH is set and assuming left aligned operation and RECIRC = 0, the PWM output will start at a logic low level at the beginning of the PWM period (when the motor controller timer counter = 0x000). The PWM output remains low until the motor controller timer counter matches the 10-bit PWM duty cycle



## NOTE

A separate read/write for high byte and low byte gives a different result than accessing the register as a word.

If the RDMCL bit in the MDCCTL register is cleared, reads of the MDCCNT register will return the present value of the count register. If the RDMCL bit is set, reads of the MDCCNT register will return the contents of the load register.

With a 0x0000 write to the MDCCNT register, the modulus counter stays at zero and does not set the MCZIF flag in the SSDFLG register.

If modulus mode is not enabled (MODMC = 0), a write to the MDCCNT register immediately updates the load register and the counter register with the value written to it. The modulus counter will count down from this value and will stop at 0x0000.

If modulus mode is enabled (MODMC = 1), a write to the MDCCNT register updates the load register with the value written to it. The count register will not be updated with the new value until the next counter underflow. The FLMC bit in the MDCCTL register can be used to immediately update the count register with the new value if an immediate load is desired.

The modulus down counter clock frequency is the bus frequency divided by 64 or 512.

# 12.3.2.6 Integration Accumulator Register (ITGACC)



Write: Never.

## NOTE

A separate read for high byte and low byte gives a different result than accessing the register as a word.

MC9S12XHZ512 Data Sheet, Rev. 1.06



#### Chapter 13 Inter-Integrated Circuit (IICV3) Block Description

- Acknowledge bit generation/detection
- Bus busy detection
- General Call Address detection
- Compliant to ten-bit address



IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
85	112	36	44	64
86	128	40	52	72
87	152	40	64	84
88	112	28	40	60
89	128	28	48	68
8A	144	36	56	76
8B	160	36	64	84
8C	176	44	72	92
8D	192	44	80	100
8E	224	52	96	116
8F	272	52	120	140
90	192	36	72	100
91	224	36	88	116
92	256	52	104	132
93	288	52	120	148
94	320	68	136	164
95	352	68	152	180
96	416	84	184	212
97	512	84	232	260
98	320	36	152	164
99	384	36	184	196
9A	448	68	216	228
9B	512	68	248	260
9C	576	100	280	292
9D	640	100	312	324
9E	768	132	376	388
9F	960	132	472	484
A0	640	68	312	324
A1	768	68	376	388
A2	896	132	440	452
A3	1024	132	504	516
A4	1152	196	568	580
A5	1280	196	632	644
A6	1536	260	760	772
A7	1920	260	952	964
A8	1280	132	632	644
A9	1536	132	760	772
AA	1792	260	888	900
AB	2048	260	1016	1028
AC	2304	388	1144	1156
AD	2560	388	1272	1284
AE	3072	516	1528	1540
AF	3840	516	1912	1924
BU	2560	260	12/2	1284
В1	3072	260	1528	1540

## Table 13-7. IIC Divider and Hold Values (Sheet 5 of 6)



#### Chapter 14 Freescale's Scalable Controller Area Network (S12MSCANV3)

1. Read: Anytime

Write: Anytime when not in initialization mode

## NOTE

The CANTIER register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

#### Table 14-14. CANTIER Register Field Descriptions

Field	Description
2-0 TXEIE[2:0	<ul> <li>Transmitter Empty Interrupt Enable</li> <li>No interrupt request is generated from this event.</li> <li>A transmitter empty (transmit buffer available for transmission) event causes a transmitter empty interrupt request.</li> </ul>

## 14.3.2.9 MSCAN Transmitter Message Abort Request Register (CANTARQ)

The CANTARQ register allows abort request of queued messages as described below.

Module Base + 0x0008

Access: User read/write<sup>(1)</sup>

	7	6	5	4	3	2	1	0		
R	0	0	0	0	0					
W						ADIRQZ	ADIRQI	ADIRQU		
Reset:	0	0	0	0	0	0	0	0		
		= Unimplemented								

#### Figure 14-12. MSCAN Transmitter Message Abort Request Register (CANTARQ)

1. Read: Anytime

Write: Anytime when not in initialization mode

#### NOTE

The CANTARQ register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

#### Table 14-15. CANTARQ Register Field Descriptions

Field	Description
2-0 ABTRQ[2:0]	<ul> <li>Abort Request — The CPU sets the ABTRQx bit to request that a scheduled message buffer (TXEx = 0) be aborted. The MSCAN grants the request if the message has not already started transmission, or if the transmission is not successful (lost arbitration or error). When a message is aborted, the associated TXE (see Section 14.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and abort acknowledge flags (ABTAK, see Section 14.3.2.10, "MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)") are set and a transmit interrupt occurs if enabled. The CPU cannot reset ABTRQx. ABTRQx is reset whenever the associated TXE flag is set.</li> <li>0 No abort request</li> <li>1 Abort request pending</li> </ul>





Figure 14-28. Identifier Register 2 (IDR2) — Extended Identifier Mapping

Field	Description
7-0 ID[14:7]	<b>Extended Format Identifier</b> — The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

## Table 14-29. IDR2 Register Field Descriptions — Extended

#### Module Base + 0x00X3

	7	6	5	4	3	2	1	0
R W	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
Reset:	x	х	х	x	x	x	х	x

Figure 14-29. Identifier Register 3 (IDR3) — Extended Identifier Mapping

#### Table 14-30. IDR3 Register Field Descriptions — Extended

Field	Description
7-1 ID[6:0]	<b>Extended Format Identifier</b> — The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
0 RTR	<ul> <li>Remote Transmission Request — This flag reflects the status of the remote transmission request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent.</li> <li>0 Data frame</li> <li>1 Remote frame</li> </ul>



Chapter 19 Enhanced Capture Timer (ECT16B8CV3)



Figure 19-68. Detailed Timer Block Diagram in Latch Mode when PRNT = 1

MC9S12XHZ512 Data Sheet, Rev. 1.06

Chapter 22 S12X Debug (S12XDBGV3) Module

# 22.3.2.2 Debug Status Register (DBGSR)

Address: 0x0021



#### Figure 22-4. Debug Status Register (DBGSR)

## Read: Anytime

Write: Never

## Table 22-8. DBGSR Field Descriptions

Field	Description
7 TBF	<b>Trace Buffer Full</b> — The TBF bit indicates that the trace buffer has stored 64 or more lines of data since it was last armed. If this bit is set, then all 64 lines will be valid data, regardless of the value of DBGCNT bits CNT[6:0]. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit
6 EXTF	External Tag Hit Flag — The EXTF bit indicates if a tag hit condition from an external TAGHI/TAGLO tag was met since arming. This bit is cleared when ARM in DBGC1 is written to a one. 0 External tag hit has not occurred 1 External tag hit has occurred
2–0 SSF[2:0]	<b>State Sequencer Flag Bits</b> — The SSF bits indicate in which state the State Sequencer is currently in. During a debug session on each transition to a new state these bits are updated. If the debug session is ended by software clearing the ARM bit, then these bits retain their value to reflect the last state of the state sequencer before disarming. If a debug session is ended by an internal trigger, then the state sequencer returns to state0 and these bits are cleared to indicate that state0 was entered during the session. On arming the module the state sequencer enters state1 and these bits are forced to SSF[2:0] = 001. See Table 22-9.

#### Table 22-9. SSF[2:0] — State Sequence Flag Bit Encoding

SSF[2:0]	Current State
000	State0 (disarmed)
001	State1
010	State2
011	State3
100	Final State
101,110,111	Reserved

Field	Description
6 CSZ	<ul> <li>Access Type Indicator — This bit indicates if the access was a byte or word size access. This bit only contains valid information when tracing CPU12X activity in Detail Mode.</li> <li>0 Word Access</li> <li>1 Byte Access</li> </ul>
5 CRW	<ul> <li>Read Write Indicator — This bit indicates if the corresponding stored address corresponds to a read or write access. This bit only contains valid information when tracing CPU12X activity in Detail Mode.</li> <li>Write Access</li> <li>Read Access</li> </ul>
4 COCF	<ul> <li>CPU12X Opcode Fetch Indicator — This bit indicates if the stored address corresponds to an opcode fetch cycle. This bit only contains valid information when tracing the XGATE accesses in Detail Mode.</li> <li>0 Stored information does not correspond to opcode fetch cycle</li> <li>1 Stored information corresponds to opcode fetch cycle</li> </ul>
3 XACK	<ul> <li>XGATE Access Indicator — This bit indicates if the stored XGATE address corresponds to a free cycle. This bit only contains valid information when tracing the CPU12X accesses in Detail Mode.</li> <li>0 Stored information corresponds to free cycle</li> <li>1 Stored information does not correspond to free cycle</li> </ul>
2 XSZ	<ul> <li>Access Type Indicator — This bit indicates if the access was a byte or word size access. This bit only contains valid information when tracing XGATE activity in Detail Mode.</li> <li>0 Word Access</li> <li>1 Byte Access</li> </ul>
1 XRW	Read Write Indicator — This bit indicates if the corresponding stored address corresponds to a read or write access. This bit only contains valid information when tracing XGATE activity in Detail Mode. 0 Write Access 1 Read Access
0 XOCF	<ul> <li>XGATE Opcode Fetch Indicator — This bit indicates if the stored address corresponds to an opcode fetch cycle. This bit only contains valid information when tracing the CPU12X accesses in Detail Mode.</li> <li>0 Stored information does not correspond to opcode fetch cycle</li> <li>1 Stored information corresponds to opcode fetch cycle</li> </ul>

#### Table 22-46. CXINF Field Descriptions (continued)

## 22.4.5.4 Reading Data from Trace Buffer

The data stored in the Trace Buffer can be read using either the background debug module (BDM) module, the XGATE or the CPU12X provided the S12XDBG module is not armed, is configured for tracing and the system not secured. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by an aligned word write to DBGTB when the module is disarmed.

The Trace Buffer can only be read through the DBGTB register using aligned word reads, any byte or misaligned reads return 0 and do not cause the trace buffer pointer to increment to the next trace buffer address. The Trace Buffer data is read out first-in first-out. By reading CNT in DBGCNT the number of valid 64-bit lines can be determined. DBGCNT will not decrement as data is read.

Whilst reading an internal pointer is used to determine the next line to be read. After a tracing session, the pointer points to the oldest data entry, thus if no overflow has occurred, the pointer points to line0, otherwise it points to the line with the oldest entry. The pointer is initialized by each aligned write to DBGTBH to point to the oldest data again. This enables an interrupted trace buffer read sequence to be easily restarted from the oldest data entry.



# 25.4.3 Chip Access Restrictions

## 25.4.3.1 Illegal XGATE Accesses

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

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

For further details refer to the XGATE Block Guide.

## 25.4.3.2 Illegal CPU Accesses

After programming the protection mechanism registers (see Figure 25-17, Figure 25-18, Figure 25-19, and Figure 25-20) and setting the RWPE bit (see Figure 25-17) there are 3 regions recognized by the MMC module:

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

If the RWPE bit is set the CPU write accesses into the XGATE RAM region are blocked. If the CPU tries to write the XGATE RAM region the AVIF bit is set and an interrupt is generated if enabled. Furthermore if the XGATE tries to write to outside of the XGATE RAM or shared regions and the RWPE bit is set, the write access is suppressed and the access error will be flagged to the XGATE module (see Section 25.4.3.1, "Illegal XGATE Accesses and the XGATE Block Guide).

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

The values stored in the boundary registers define the boundary addresses in 256 byte steps. The 256 byte block selected by any of the registers is always included in the respective region. For example setting the shared region lower boundary register (RAMSHL) to 0xC1 and the shared region upper boundary register (RAMSHU) to 0xE0 defines the shared region from address 0x0F\_C100 to address 0x0F\_E0FF in the global memory space (see Figure 25-20).

The interrupt requests generated by the MMC are listed in Table 25-23. Refer to the Device User Guide for the related interrupt vector address and interrupt priority.

The following conditions must be satisfied to ensure correct operation of the RAM protection mechanism:

- Value stored in RAMXGU must be lower than the value stored in RAMSHL.
- Value stored RAMSHL must be lower or equal than the value stored in RAMSHU.



Chapter 25 Memory Mapping Control (S12XMMCV3)



#### **Appendix A Electrical Characteristics**



Figure A-1. ATD Accuracy Definitions

## NOTE

Figure A-1 shows only definitions, for specification values refer to Table A-13.

#### MC9S12XHZ512 Data Sheet, Rev. 1.06



**Appendix A Electrical Characteristics** 



# A.9.2 Normal Expanded Mode (External Wait Feature Enabled)

Figure A-11. Example 1b: Normal Expanded Mode — Stretched Read Access