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	Obsolete
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	119
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 24x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12xd256vag

Chapter 19
S12X Debug (S12XDBGV2) Module

Chapter 20
S12X Debug (S12XDBGV3) Module

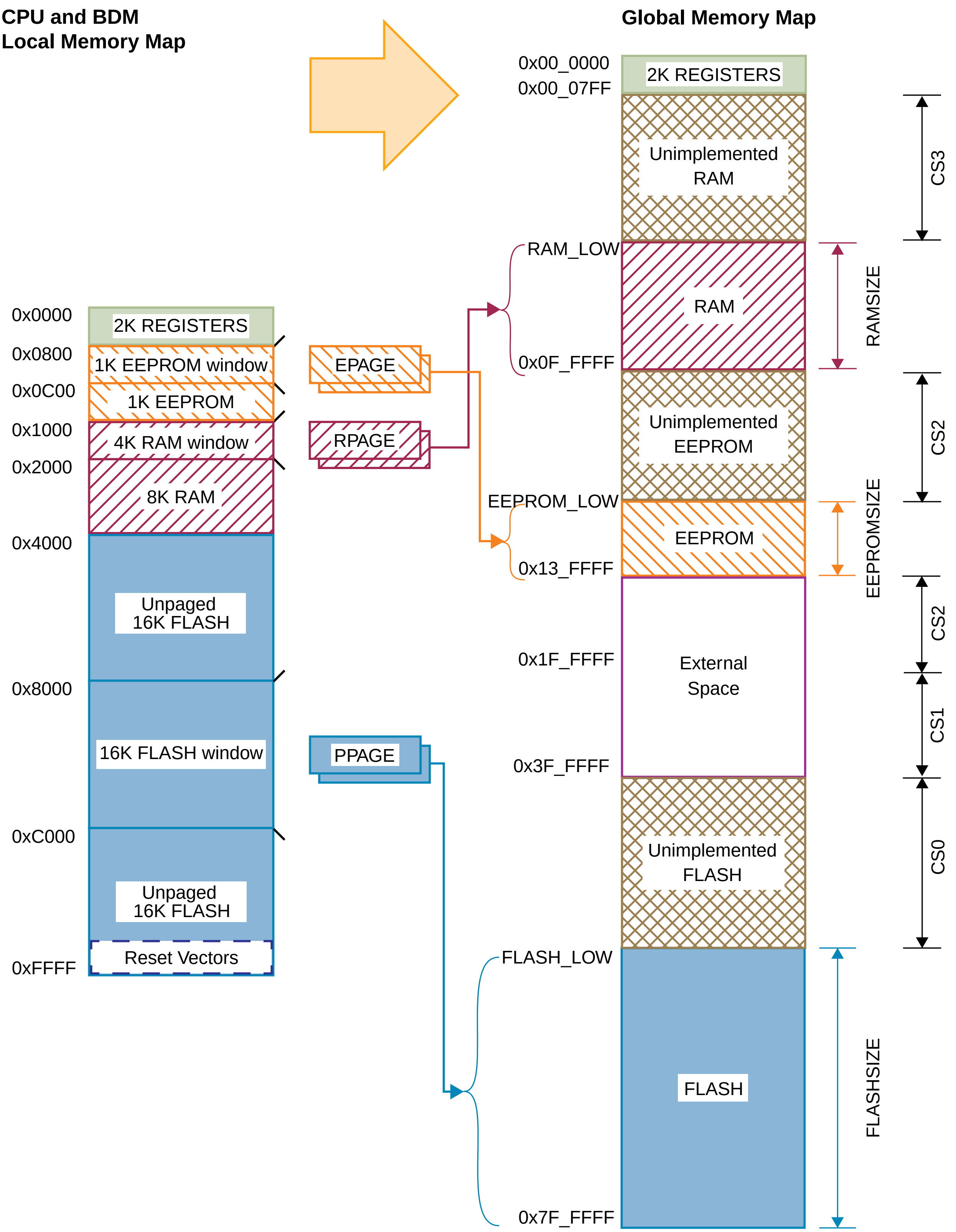

Figure 1-3. S12X CPU & BDM Global Address Mapping

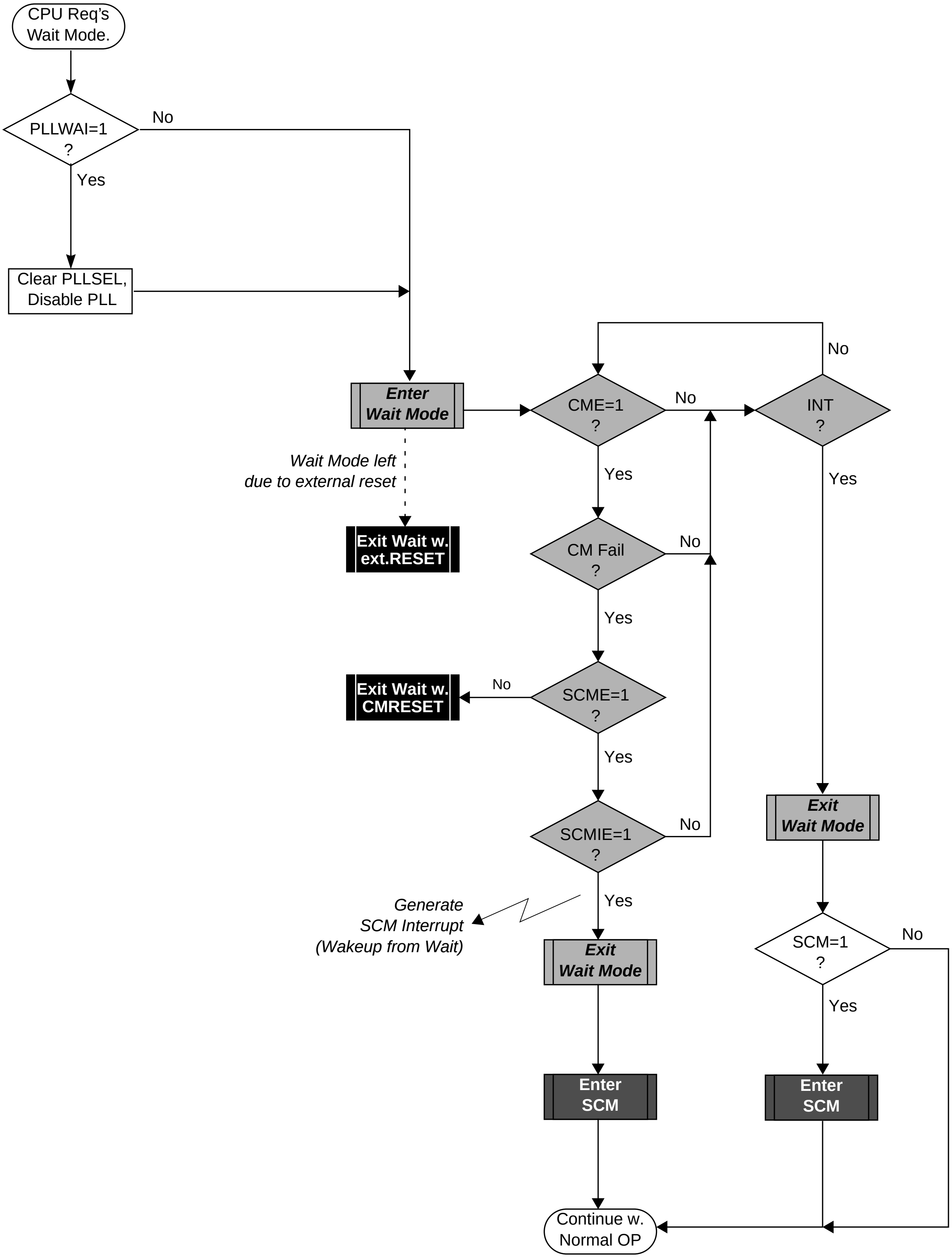

Figure 2-21. Wait Mode Entry/Exit Sequence

In freeze mode all clocks of the XGATE module may be stopped, depending on the module configuration (see Section 6.3.1.1, “XGATE Control Register (XGMCTL)”).

6.1.4 Block Diagram

Figure Figure 6-1 shows a block diagram of the XGATE.

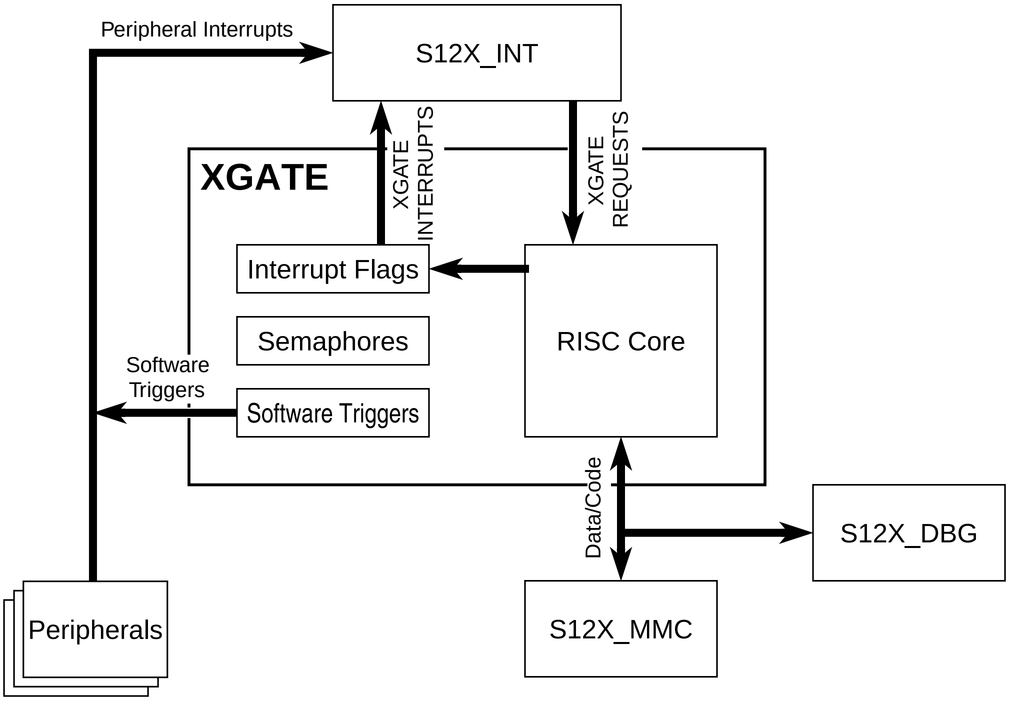

Figure 6-1. XGATE Block Diagram

6.2 External Signal Description

The XGATE module has no external pins.

MOV Move Register Content MOV

Operation

RS ⇒ RD (translates to OR RD, R0, RS)

Copies the content of RS to RD.

CCR Effects

N	Z	V	C
Δ	Δ	0	—

N: Set if bit 15 of the result is set; cleared otherwise.
Z: Set if the result is $0000; cleared otherwise.
V: 0; cleared.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
MOV RD, RS	TRI	0	0	0	1	0	RD	0 0 0	RS	1	0	P

6.9 Initialization and Application Information

6.9.1 Initialization

The recommended initialization of the XGATE is as follows:

1. Clear the XGE bit to suppress any incoming service requests.
2. Make sure that no thread is running on the XGATE. This can be done in several ways:
   a) Poll the XGCHID register until it reads $00. Also poll XGDBG and XGSWEIF to make sure that the XGATE has not been stopped.

   b) Enter Debug Mode by setting the XGDBG bit. Clear the XGCHID register. Clear the XGDBG bit.

   The recommended method is a).
3. Set the XGVBR register to the lowest address of the XGATE vector space.
4. Clear all Channel ID flags.
5. Copy XGATE vectors and code into the RAM.
6. Initialize the S12X_INT module.
7. Enable the XGATE by setting the XGE bit.

The following code example implements the XGATE initialization sequence.

6.9.2 Code Example (Transmit "Hello World!" on SCI)

```
                        CPU   S12X
                        ;############################################
                        ;#                   SYMBOLS                #
                        ;############################################
SCI_REGS                EQU  $00C8           ;SCI register space
SCIBDH                  EQU  SCI_REGS+$00    ;SCI Baud Rate Register
SCIBDL                  EQU  SCI_REGS+$00    ;SCI Baud Rate Register
SCICR2                  EQU  SCI_REGS+$03    ;SCI Control Register 2
SCISR1                  EQU  SCI_REGS+$04    ;SCI Status  Register 1
SCIDRL                  EQU  SCI_REGS+$07    ;SCI Control Register 2
TIE                     EQU  $80             ;TIE bit mask
TE                      EQU  $08             ;TE bit mask
RE                      EQU  $04             ;RE bit mask
SCI_VEC                 EQU  $D6             ;SCI vector number

INT_REGS                     EQU             $0120           ;S12X_INT register space
INT_CFADDR              EQU  INT_REGS+$07    ;Interrupt Configuration Address Register
INT_CFDATA              EQU  INT_REGS+$08    ;Interrupt Configuration Data Registers
RQST                    EQU  $80             ;RQST bit mask

XGATE_REGS              EQU  $0380           ;XGATE register space
XGMCTL                  EQU  XGATE_REGS+$00  ;XGATE Module Control Register
XGMCTL_CLEAR            EQU  $FA02           ;Clear all XGMCTL bits
XGMCTL_ENABLE           EQU  $8282           ;Enable XGATE
XGCHID                  EQU  XGATE_REGS+$02  ;XGATE Channel ID Register
XGVBR                   EQU  XGATE_REGS+$06  ;XGATE ISP Select Register
XGIF                    EQU  XGATE_REGS+$08  ;XGATE Interrupt Flag Vector
```

11.3.2.7 SCI Status Register 1 (SCISR1)

The SCISR1 and SCISR2 registers provides inputs to the MCU for generation of SCI interrupts. Also, these registers can be polled by the MCU to check the status of these bits. The flag-clearing procedures require that the status register be read followed by a read or write to the SCI data register.It is permissible to execute other instructions between the two steps as long as it does not compromise the handling of I/O, but the order of operations is important for flag clearing.

	7	6	5	4	3	2	1	0
R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	1	1	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 11-10. SCI Status Register 1 (SCISR1)

Read: Anytime

Write: Has no meaning or effect

Table 11-10. SCISR1 Field Descriptions

Field	Description
7 TDRE	**Transmit Data Register Empty Flag** — TDRE is set when the transmit shift register receives a byte from the SCI data register. When TDRE is 1, the transmit data register (SCIDRH/L) is empty and can receive a new value to transmit.Clear TDRE by reading SCI status register 1 (SCISR1), with TDRE set and then writing to SCI data register low (SCIDRL). 0 No byte transferred to transmit shift register 1 Byte transferred to transmit shift register; transmit data register empty
6 TC	**Transmit Complete Flag** — TC is set low when there is a transmission in progress or when a preamble or break character is loaded. TC is set high when the TDRE flag is set and no data, preamble, or break character is being transmitted.When TC is set, the TXD pin becomes idle (logic 1). Clear TC by reading SCI status register 1 (SCISR1) with TC set and then writing to SCI data register low (SCIDRL). TC is cleared automatically when data, preamble, or break is queued and ready to be sent. TC is cleared in the event of a simultaneous set and clear of the TC flag (transmission not complete). 0 Transmission in progress 1 No transmission in progress
5 RDRF	**Receive Data Register Full Flag** — RDRF is set when the data in the receive shift register transfers to the SCI data register. Clear RDRF by reading SCI status register 1 (SCISR1) with RDRF set and then reading SCI data register low (SCIDRL). 0 Data not available in SCI data register 1 Received data available in SCI data register
4 IDLE	**Idle Line Flag** — IDLE is set when 10 consecutive logic 1s (if M = 0) or 11 consecutive logic 1s (if M =1) appear on the receiver input. Once the IDLE flag is cleared, a valid frame must again set the RDRF flag before an idle condition can set the IDLE flag.Clear IDLE by reading SCI status register 1 (SCISR1) with IDLE set and then reading SCI data register low (SCIDRL). 0 Receiver input is either active now or has never become active since the IDLE flag was last cleared 1 Receiver input has become idle **Note:** When the receiver wakeup bit (RWU) is set, an idle line condition does not set the IDLE flag.

18.3 Memory Map and Registers

18.3.1 Module Memory Map

A summary of the registers associated with the MMC block is shown in Figure 18-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000A	MMCCTL0	R	0	0	0	0	CS3E	CS2E	CS1E	CS0E
		W	▨	▨	▨	▨				
0x000B	MODE	R	MODC	MODB	MODA	0	0	0	0	0
		W				▨	▨	▨	▨	▨
0x0010	GPAGE	R	0	GP6	GP5	GP4	GP3	GP2	GP1	GP0
		W	▨							
0x0011	DIRECT	R	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
		W								
0x0012	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x0013	MMCCTL1	R	0	0	0	0	0	EROMON	ROMHM	ROMON
		W	▨	▨	▨	▨	▨			
0x0014	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x0015	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x0016	RPAGE	R	RP7	RP6	RP5	RP4	RP3	RP2	RP1	RP0
		W								
0x0017	EPAGE	R	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
		W								
0x0030	PPAGE	R	PIX7	PIX6	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
		W								
0x0031	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented or Reserved

Figure 18-2. MMC Register Summary

Table 19-21. State1 Sequencer Next Sate Selection

SC[3:0]	Description
0000	Any match triggers to state2
0001	Any match triggers to state3
0010	Any match triggers to final state
0011	Match2 triggers to State2....... Other matches have no effect
0100	Match2 triggers to State3....... Other matches have no effect
0101	Match2 triggers to final state....... Other matches have no effect
0110	Match0 triggers to State2....... Match1 triggers to State3....... Other matches have no effect
0111	Match1 triggers to State3....... Match0 triggers final state....... Other matches have no effect
1000	Match0 triggers to State2....... Match2 triggers to State3....... Other matches have no effect
1001	Match2 triggers to State3....... Match0 triggers final state....... Other matches have no effect
1010	Match1 triggers to State2....... Match3 triggers to State3....... Other matches have no effect
1011	Match3 triggers to State3....... Match1 triggers to final state....... Other matches have no effect
1100	Reserved
1101	Reserved
1110	Reserved
1111	Reserved

19.3.1.9 Debug State Control Register 2 (DBGSCR2)

0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W	▨	▨	▨	▨				
Reset	0	0	0	0	0	0	0	0

▨ Unimplemented or Reserved

Figure 19-11. Debug State Control Register 2 (DBGSCR2)

Read: Anytime

Write: Anytime when DBG not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 01. The state control register 2 selects the targeted next state while in State2. The matches refer to the match channels of the comparator match control logic as depicted in Figure 19-1 and described in Section 19.3.1.11.1, “Debug Comparator Control Register (DBGXCTL)”. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

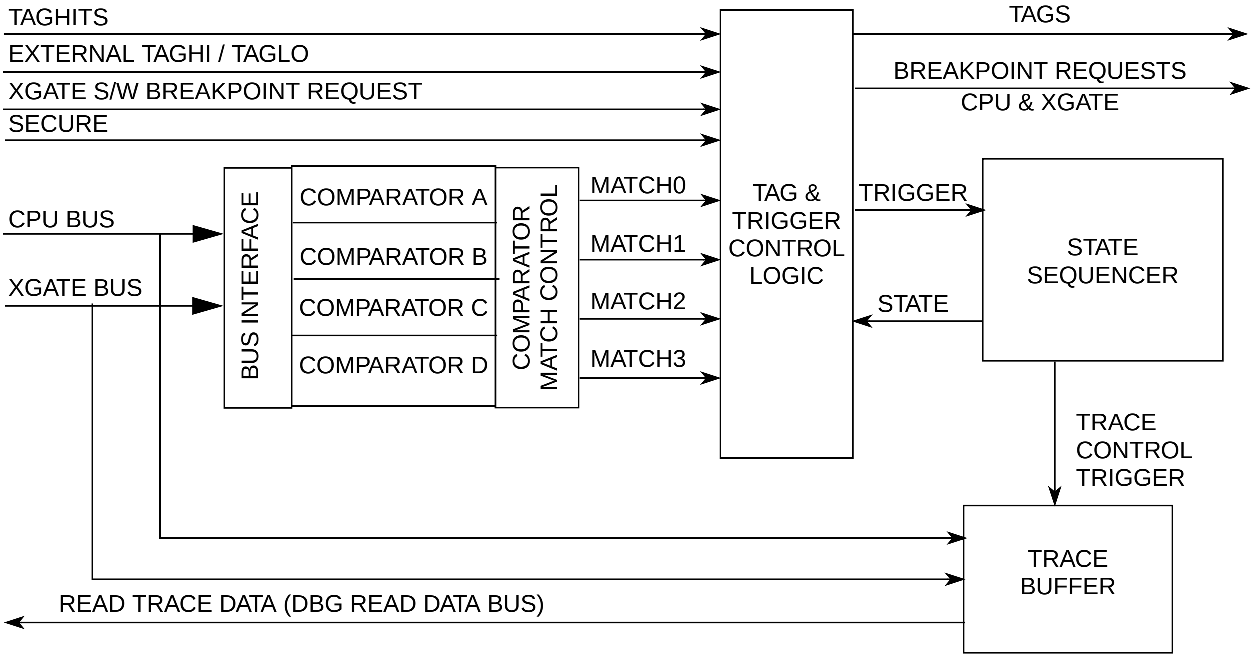

Figure 19-22. DBG Overview

19.4.2 Comparator Modes

The DBG contains 4 comparators, A, B, C, and D. Each comparator can be configured to monitor either CPU or XGATE busses using the SRC bit in the corresponding comparator control register. Each comparator compares the selected address bus with the address stored in DBGXAH, DBGXAM and DBGXAL. Furthermore comparators A and C also compare the data buses to the data stored in DBGXDH, DBGXDL and allow masking of individual data bus bits.

All comparators are disabled in BDM and during BDM accesses.

The comparator match control logic (see Figure 19-22) configures comparators to monitor the busses for an exact address or an address range, whereby either an access inside or outside the specified range generates a match condition. The comparator configuration is controlled by the control register contents and the range control by the DBGC2 contents.

On a match a trigger can initiate a transition to another state sequencer state (see Section 19.4.3, “Trigger Modes”). The comparator control register also allows the type of access to be included in the comparison through the use of the RWE,RW,SZE and SZ bits. The RWE bit controls whether read or write comparison is enabled for the associated comparator and the RW bit selects either a read or write access for a valid match. Similarly the SZE and SZ bits allows the size of access (word or byte) to be considered in the compare. Only comparators B and D feature SZE and SZ.

The TAG bit in each comparator control register is used to determine the triggering condition. By setting TAG, the comparator will qualify a match with the output of opcode tracking logic and a trigger occurs before the tagged instruction executes (tagged-type trigger). Whilst tagging the RW, RWE, SZE and SZ bits are ignored and the comparator register must be loaded with the exact opcode address.

19.4.7.5 DBG Breakpoint Priorities

XGATE software breakpoints have the highest priority. Active tracing sessions are terminated immediately.

If a TRIG triggers occur after begin or mid aligned tracing has already been triggered by a comparator instigated transition to final state, then TRIG no longer has an effect. When the associated tracing session is complete, the breakpoint occurs. Similarly if a TRIG is followed by a subsequent trigger from a comparator channel whose BRK=0, it has no effect, since tracing has already started.

If a comparator tag hit occurs simultaneously with an external $\overline{\text{TAGHI}}$/$\overline{\text{TAGLO}}$ hit, the state sequencer enters State0. $\overline{\text{TAGHI}}$/$\overline{\text{TAGLO}}$ triggers are always end aligned, to end tracing immediately, independent of the tracing trigger alignment bits TALIGN[1:0].

19.4.7.5.1 DBG Breakpoint priorities, mapping and BDM interfacing

Breakpoint operation is dependent on the state of the BDM module. If the BDM module is active, the CPU is executing out of BDM firmware and S12X breakpoints are disabled. In addition, while executing a BDM TRACE command, tagging into BDM is disabled.

Table 19-45. Breakpoint Mapping Summary

DBGBRK[1] (DBGC1[3])[1]	BDM bit (DBGC1[4])	BDM enabled	BDM active	Type of Debug Session
0	X	X	X	No Breakpoint
1	0	0	0	Breakpoint to SWI
1	0	1	X	Illegal Configuration. Do Not Use.[2]
1	1	0	0	Illegal Configuration. Do Not Use. [3]
1	1	1	0	Breakpoint to BDM
1	1	1	1	No Breakpoint

[1] All sources except XGATE software BKP, which are independent of this bit.

[2] The DBGC1[4] bit (BDM) must be set if using the BDM interface together with the DBG module. Failure to set this bit could result in XGATE generated breakpoints to SWI during BDM firmware execution corrupting the S12X PC return address, should the user have entered BDM via the BACKGROUND command or BGND instruction.

[3] End aligned tagged Breakpoint to SWI. Begin, Mid aligned and Forced Breakpoints disabled

If BDM is not active, the breakpoint will give priority to BDM requests over SWI requests if the breakpoint happens to coincide with a SWI instruction in the user's code. On returning from BDM, the SWI from user code gets executed.

BDM cannot be entered from a breakpoint unless the ENABLE bit is set in the BDM. If entry to BDM via a BGND instruction is attempted and the ENABLE bit in the BDM is cleared, the CPU actually executes the BDM firmware code. It checks the ENABLE and returns if ENABLE is not set. If not serviced by the monitor then the breakpoint is re-asserted when the BDM returns to normal CPU flow.

If the comparator register contents coincide with the SWI/BDM vector address then an SWI in user code and DBG breakpoint could occur simultaneously. The CPU ensures that BDM requests have a higher priority than SWI requests. Returning from the BDM/SWI service routine care must be taken to avoid re-triggering a breakpoint.

22.3.2.72 Port AD1 Pull Up Enable Register 0 (PER0AD1)

	7	6	5	4	3	2	1	0
R W	PER0AD123	PER0AD122	PER0AD121	PER0AD120	PER0AD119	PER0AD118	PER0AD117	PER0AD116
Reset	0	0	0	0	0	0	0	0

Figure 22-74. Port AD1 Pull Up Enable Register 0 (PER0AD1)

Read: Anytime.

Write: Anytime.

This register activates a pull-up device on the respective PAD[23:16] pin if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull-up device is enabled.

Table 22-65. PER0AD1 Field Descriptions

Field	Description
7–0 PER0AD1[23:16]	**Pull Device Enable Port AD1 Register 0** 0 Pull-up device is disabled. 1 Pull-up device is enabled.

22.3.2.73 Port AD1 Pull Up Enable Register 1 (PER1AD1)

	7	6	5	4	3	2	1	0
R W	PER1AD115	PER1AD114	PER1AD113	PER1AD112	PER1AD111	PER1AD110	PER1AD19	PER1AD18
Reset	0	0	0	0	0	0	0	0

Figure 22-75. Port AD1 Pull Up Enable Register 1 (PER1AD1)

Read: Anytime.

Write: Anytime.

This register activates a pull-up device on the respective PAD[15:08] pin if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull-up device is enabled.

Table 22-66. PER1AD1 Field Descriptions

Field	Description
7–0 PER1AD1[15:8]	**Pull Device Enable Port AD1 Register 1** 0 Pull-up device is disabled. 1 Pull-up device is enabled.

Table 23-1. Pin Functions and Priorities (Sheet 2 of 7)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
E	PE[7]	$\overline{\mathrm{XCLKS}}$[1]	I	External clock selection input during $\overline{\mathrm{RESET}}$	Mode dependent[3]
		ECLKX2	I	Free-running clock output at Core Clock rate (ECLK x 2)	
		GPIO	I/O	General-purpose I/O	
	PE[6]	MODB[1]	I	MODB input during $\overline{\mathrm{RESET}}$	
		$\overline{\mathrm{TAGHI}}$	I	Instruction tagging low pin Configurable for reduced input threshold	
		GPIO	I/O	General-purpose I/O	
	PE[5]	MODA[1]	I	MODA input during $\overline{\mathrm{RESET}}$	
		$\overline{\mathrm{RE}}$	O	Read enable signal	
		$\overline{\mathrm{TAGLO}}$	I	Instruction tagging low pin Configurable for reduced input threshold	
		GPIO	I/O	General-purpose I/O	
	PE[4]	ECLK	O	Free-running clock output at the Bus Clock rate or programmable divided in normal modes	
		GPIO	I/O	General-purpose I/O	
	PE[3]	EROMCTL[1]	I	EROMON bit control input during $\overline{\mathrm{RESET}}$	
		$\overline{\mathrm{LSTRB}}$	O	Low strobe bar output	
		$\overline{\mathrm{LDS}}$	O	Lower data strobe	
		GPIO	I/O	General-purpose I/O	
	PE[2]	R/$\overline{\mathrm{W}}$	O	Read/write output for external bus	
		$\overline{\mathrm{WE}}$	O	Write enable signal	
		GPIO	I/O	General-purpose I/O	
	PE[1]	$\overline{\mathrm{IRQ}}$	I	Maskable level- or falling edge-sensitive interrupt input	
		GPIO	I/O	General-purpose I/O	
	PE[0]	$\overline{\mathrm{XIRQ}}$	I	Non-maskable level-sensitive interrupt input	
		GPIO	I/O	General-purpose I/O	

Table 23-3. Pin Configuration Summary

DDR	IO	RDR	PE	PS[1]	IE[2]	Function	Pull Device	Interrupt
0	x	x	0	x	0	Input	Disabled	Disabled
0	x	x	1	0	0	Input	Pull Up	Disabled
0	x	x	1	1	0	Input	Pull Down	Disabled
0	x	x	0	0	1	Input	Disabled	Falling edge
0	x	x	0	1	1	Input	Disabled	Rising edge
0	x	x	1	0	1	Input	Pull Up	Falling edge
0	x	x	1	1	1	Input	Pull Down	Rising edge
1	0	0	x	x	0	Output, full drive to 0	Disabled	Disabled
1	1	0	x	x	0	Output, full drive to 1	Disabled	Disabled
1	0	1	x	x	0	Output, reduced drive to 0	Disabled	Disabled
1	1	1	x	x	0	Output, reduced drive to 1	Disabled	Disabled
1	0	0	x	0	1	Output, full drive to 0	Disabled	Falling edge
1	1	0	x	1	1	Output, full drive to 1	Disabled	Rising edge
1	0	1	x	0	1	Output, reduced drive to 0	Disabled	Falling edge
1	1	1	x	1	1	Output, reduced drive to 1	Disabled	Rising edge

[1] Always “0” on Port A, B, C, D, E, K, AD0, and AD1.

[2] Applicable only on Port P, H, and J.

NOTE

All register bits in this module are completely synchronous to internal clocks during a register read.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PORTA	R W	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PORTB	R W	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
DDRA	R W	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
DDRB	R W	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0

▨ = Unimplemented or Reserved

Figure 23-2. PIM Register Summary (Sheet 1 of 7)

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
Non-PIM Address Range	R	Non-PIM Address Range							
	W								
PORTK	R	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
	W								
DDRK	R	DDRK7	DDRK6	DDRK5	DDRK4	DDRK3	DDRK2	DDRK1	DDRK0
	W								
Non-PIM Address Range	R	Non-PIM Address Range							
	W								
PTT	R	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0
	W								
PTIT	R	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
	W	▨	▨	▨	▨	▨	▨	▨	▨
DDRT	R	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
	W								
RDRT	R	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0
	W								
PERT	R	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0
	W								
PPST	R	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
Reserved	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
PTS	R	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
	W								
PTIS	R	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
	W	▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented or Reserved

Figure 23-2. PIM Register Summary (Sheet 3 of 7)

Table 24-17. PTT Field Descriptions

Field	Description
7–0 PTT[7:0]	**Port T** — Port T bits 7–0 are associated with ECT channels IOC7–IOC0 (*refer to ECT section*). When not used with the ECT, these pins can be used as general purpose I/O. If the data direction bits of the associated I/O pins are set to logic level “1”, a read returns the value of the port register, otherwise the buffered pin input state is read.

24.0.5.14 Port T Input Register (PTIT)

	7	6	5	4	3	2	1	0
R	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset[1]	—	—	—	—	—	—	—	—

▨ = Unimplemented or Reserved

Figure 24-16. Port T Input Register (PTIT)

[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.

Table 24-18. PTIT Field Descriptions

Field	Description
7–0 PTIT[7:0]	**Port T Input** — This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

24.0.5.15 Port T Data Direction Register (DDRT)

	7	6	5	4	3	2	1	0
R W	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
Reset	0	0	0	0	0	0	0	0

Figure 24-17. Port T Data Direction Register (DDRT)

Read: Anytime.

Write: Anytime.

This register configures each port T pin as either input or output.

The ECT forces the I/O state to be an output for each timer port associated with an enabled output compare. In this case the data direction bits will not change.

If the associated SCI transmit or receive channel is enabled this register has no effect on the pins. The pin is forced to be an output if a SCI transmit channel is enabled, it is forced to be an input if the SCI receive channel is enabled.

The DDRS bits revert to controlling the I/O direction of a pin when the associated channel is disabled.

Table 24-23. DDRS Field Descriptions

Field	Description
7–0 DDRS[7:0]	**Data Direction Port S** 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 PTS or PTIS registers, when changing the DDRS register.

24.0.5.22 Port S Reduced Drive Register (RDRS)

	7	6	5	4	3	2	1	0
R W	RDRS7	RDRS6	RDRS5	RDRS4	RDRS3	RDRS2	RDRS1	RDRS0
Reset	0	0	0	0	0	0	0	0

Figure 24-24. Port S Reduced Drive Register (RDRS)

Read: Anytime.

Write: Anytime.

This register configures the drive strength of each port S output pin as either full or reduced. If the port is used as input this bit is ignored.

Table 24-24. RDRS Field Descriptions

Field	Description
7–0 RDRS[7:0]	**Reduced Drive Port S** 0 Full drive strength at output. 1 Associated pin drives at about 1/6 of the full drive strength.

24.0.5.23 Port S Pull Device Enable Register (PERS)

	7	6	5	4	3	2	1	0
R W	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0
Reset	1	1	1	1	1	1	1	1

Figure 24-25. Port S Pull Device Enable Register (PERS)

Read: Anytime.

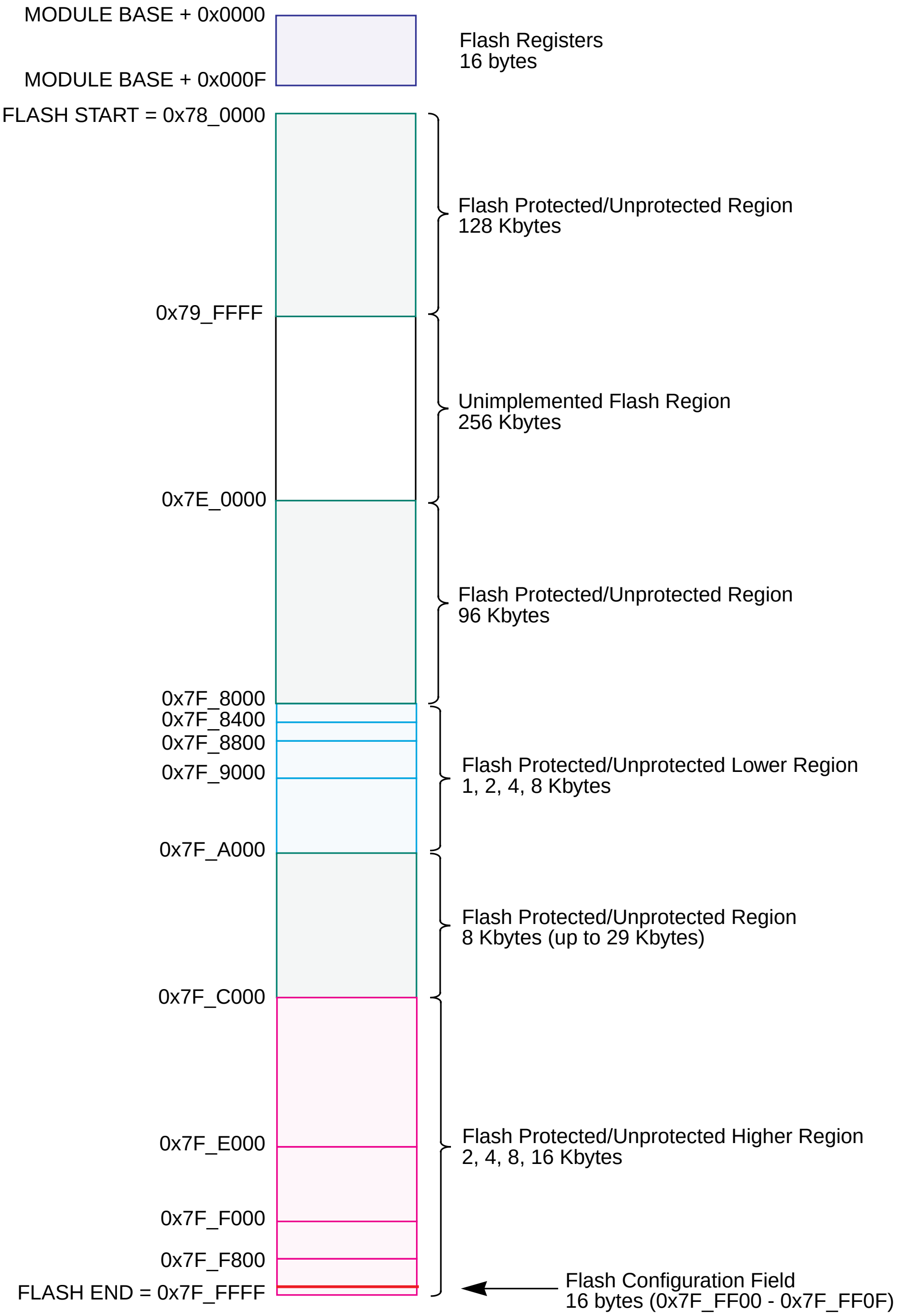
MODULE BASE + 0x0000
MODULE BASE + 0x000F
Flash Registers
16 bytes
FLASH START = 0x78_0000
Flash Protected/Unprotected Region
128 Kbytes
0x79_FFFF
Unimplemented Flash Region
256 Kbytes
0x7E_0000
Flash Protected/Unprotected Region
96 Kbytes
0x7F_8000
0x7F_8400
0x7F_8800
0x7F_9000
Flash Protected/Unprotected Lower Region
1, 2, 4, 8 Kbytes
0x7F_A000
Flash Protected/Unprotected Region
8 Kbytes (up to 29 Kbytes)
0x7F_C000
0x7F_E000
Flash Protected/Unprotected Higher Region
2, 4, 8, 16 Kbytes
0x7F_F000
0x7F_F800
FLASH END = 0x7F_FFFF
Flash Configuration Field
16 bytes (0x7F_FF00 - 0x7F_FF0F)

Figure 28-2. Flash Memory Map

The relative deviation of t_{nom} is at its maximum for one clock period, and decreases towards zero for larger number of clock periods (N).

Defining the jitter as:

$$J(N) = \max\left(\left|1 - \frac{t_{max}(N)}{N \cdot t_{nom}}\right|, \left|1 - \frac{t_{min}(N)}{N \cdot t_{nom}}\right|\right)$$

For N < 1000, the following equation is a good fit for the maximum jitter:

$$J(N) = \frac{j_1}{\sqrt{N}} + j_2$$

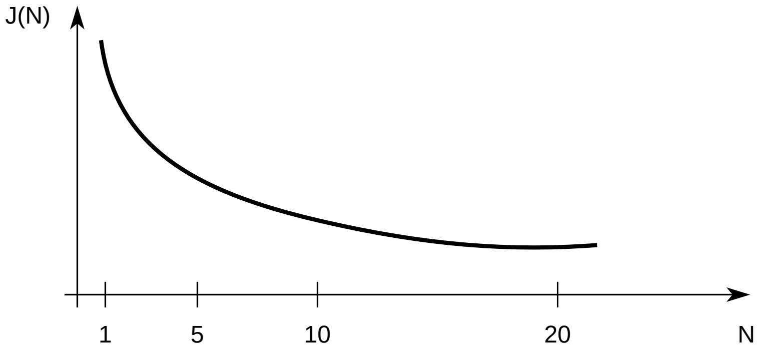

Figure A-5. Maximum Bus Clock Jitter Approximation

This is very important to notice with respect to timers, serial modules where a prescaler will eliminate the effect of the jitter to a large extent.