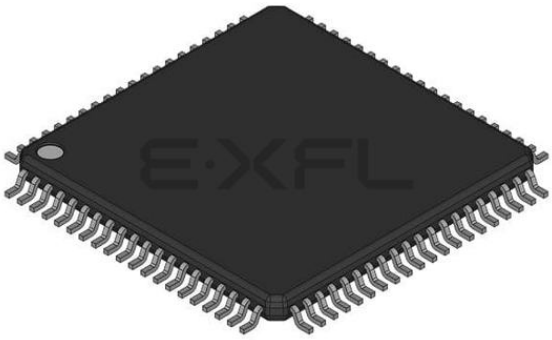

Welcome to E-XFL.COM

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	40MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s9s12xs256j0maar

Chapter 16
Timer Module (TIM16B8CV2)

Chapter 17
Voltage Regulator (S12VREGL3V3V1)

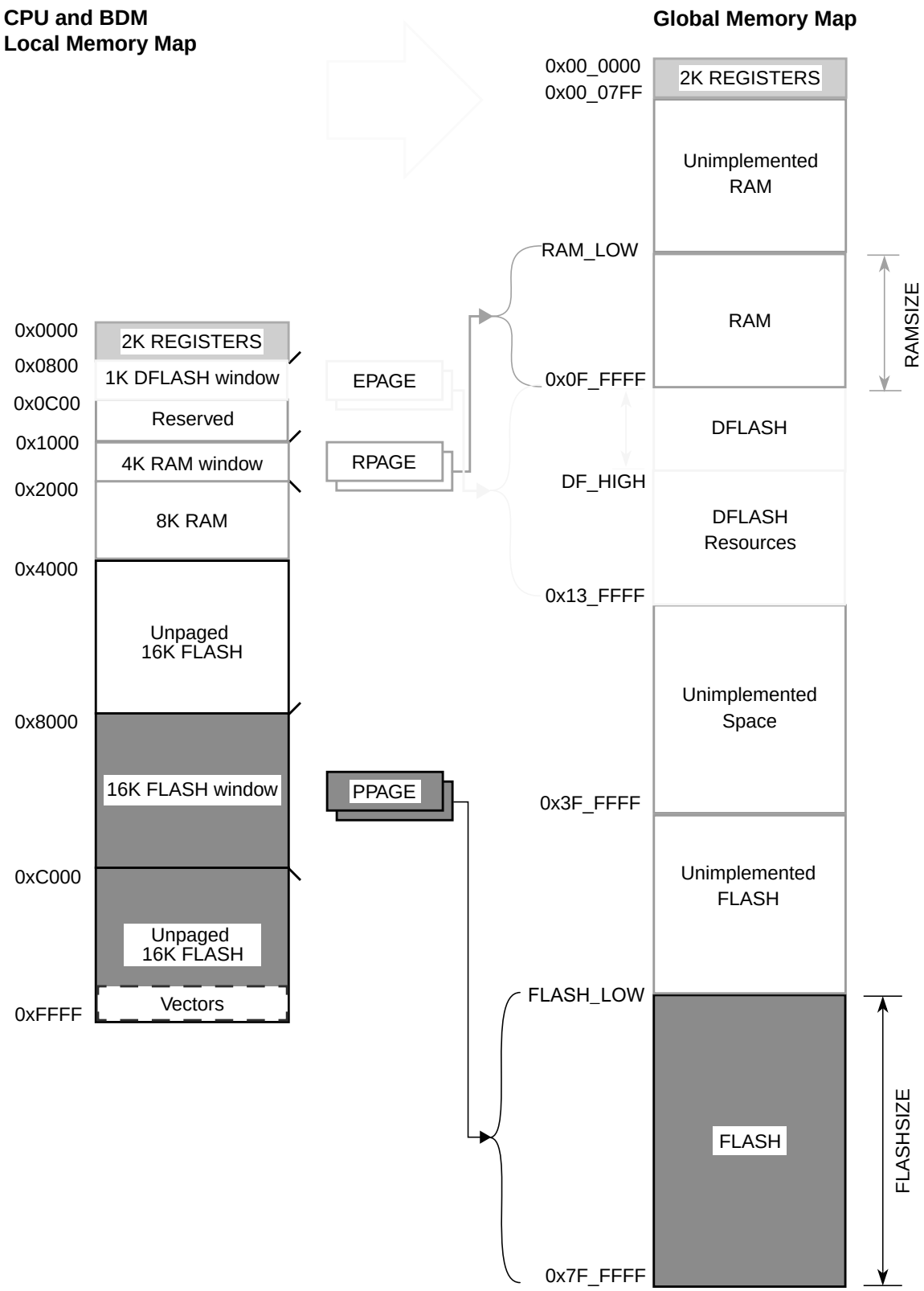

Figure 1-2. S12XS Family Global Memory Map

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0256 WOMM	R	WOMM7	WOMM6	WOMM5	WOMM4	WOMM3	WOMM2	WOMM1	WOMM0
	W								
0x0257 MODRR	R	MODRR7	MODRR6	0	MODRR4	0	0	0	0
	W								
0x0258 PTP	R	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
	W								
0x0259 PTIP	R	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
	W								
0x025A DDRP	R	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
	W								
0x025B RDRP	R	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
	W								
0x025C PERP	R	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
	W								
0x025D PPSP	R	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
	W								
0x025E PIEP	R	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
	W								
0x025F PIFP	R	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
	W								
0x0260 PTH	R	PTH7	PTH6	PTH5	PTH4	PTH3	PTH2	PTH1	PTH0
	W								
0x0261 PTIH	R	PTIH7	PTIH6	PTIH5	PTIH4	PTIH3	PTIH2	PTIH1	PTIH0
	W								
0x0262 DDRH	R	DDRH7	DDRH6	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0
	W								
0x0263 RDRH	R	RDRH7	RDRH6	RDRH5	RDRH4	RDRH3	RDRH2	RDRH1	RDRH0
	W								
0x0264 PERH	R	PERH7	PERH6	PERH5	PERH4	PERH3	PERH2	PERH1	PERH0
	W								

= Unimplemented or Reserved

1 Read: Always reads 0x00
Write: Unimplemented

2.3.16 Port K Data Register (PORTK)

Address 0x0032 (PRR) Access: User read/write[1]

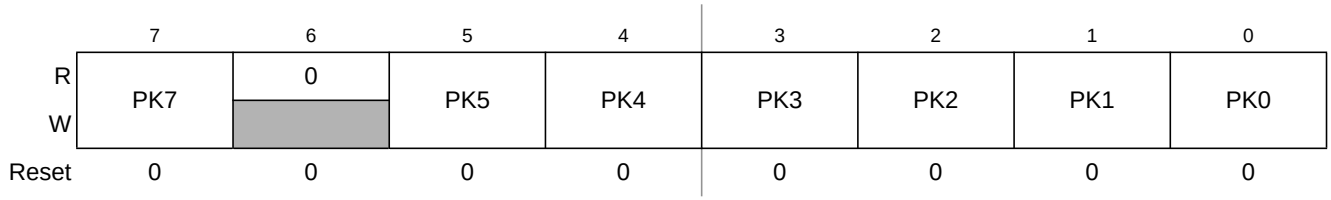

Figure 2-14. Port K Data Register (PORTK)

1 Read: Anytime, the data source depends on the data direction value
Write: Anytime

Table 2-14. PORTK Register Field Descriptions

Field	Description
7,5-0 PK	**Port K general purpose input/output data**—Data Register The associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read.

2.3.17 Port K Data Direction Register (DDRK)

Address 0x0033 (PRR) Access: User read/write[1]

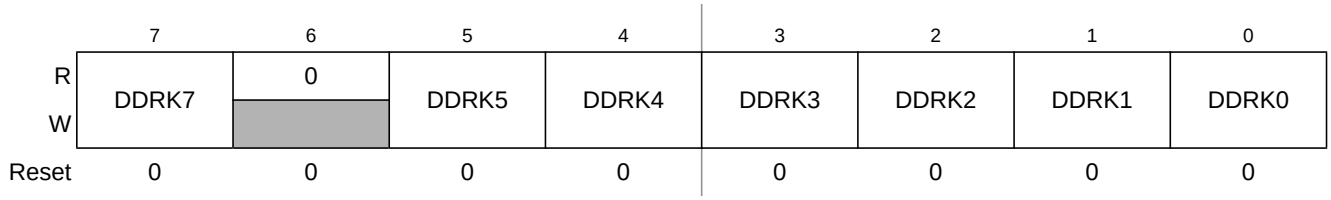

Figure 2-15. Port K Data Direction Register (DDRK)

1 Read: Anytime, the data source depends on the data direction value
Write: Anytime

2.4.2 Registers

A set of configuration registers is common to all ports with exception of the ATD port (Table 2-71). All registers can be written at any time, however a specific configuration might not become active.

For example selecting a pull-up device: This device does not become active while the port is used as a push-pull output.

Table 2-71. Register availability per port[1]

Port	Data	Input	Data Direction	Reduced Drive	Pull Enable	Polarity Select	Wired-Or Mode	Interrupt Enable	Interrupt Flag	Routing
A	yes	-	yes	yes	yes	-	-	-	-	-
B	yes	-	yes			-	-	-	-	-
E	yes	-	yes			-	-	-	-	-
K	yes	-	yes			-	-	-	-	-
T	yes	yes	yes	yes	yes	yes	-	-	-	yes
S	yes	yes	yes	yes	yes	yes	yes	-	-	-
M	yes	yes	yes	yes	yes	yes	yes	-	-	yes
P	yes	yes	yes	yes	yes	yes	-	yes	yes	-
H	yes	yes	yes	yes	yes	yes	-	yes	yes	-
J	yes	yes	yes	yes	yes	yes	-	yes	yes	-
AD	yes	-	yes	yes	yes	-	-	-	-	-

1 Each cell represents one register with individual configuration bits

2.4.2.1 Data register (PORTx, PTx)

This register holds the value driven out to the pin if the pin is used as a general purpose I/O.

Writing to this register has only an effect on the pin if the pin is used as general purpose output. When reading this address, the buffered state of the pin is returned if the associated data direction register bit is set to “0”.

If the data direction register bits are set to logic level “1”, the contents of the data register is returned. This is independent of any other configuration (Figure 2-73).

2.4.2.2 Input register (PTIx)

This is a read-only register and always returns the buffered state of the pin (Figure 2-73).

2.4.2.3 Data direction register (DDRx)

This register defines whether the pin is used as a input or an output.

If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 2-73).

Independent of the pin usage with a peripheral module this register determines the source of data when reading the associated data register address (2.4.2.1/2-121).

Chapter 6
S12X Debug (S12XDBGV3) Module

Table 6-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V03.20	14 Sep 2007	6.3.2.7/6-205	- Clarified reserved State Sequencer encodings.
V03.21	23 Oct 2007	6.4.2.2/6-218 6.4.2.4/6-219	- Added single databyte comparison limitation information - Added statement about interrupt vector fetches whilst tagging.
V03.22	12 Nov 2007	6.4.5.2/6-223 6.4.5.5/6-227	- Removed LOOP1 tracing restriction NOTE. - Added pin reset effect NOTE.
V03.23	13 Nov 2007	General	- Text readability improved, typo removed.
V03.24	04 Jan 2008	6.4.5.3/6-225	- Corrected bit name.
V03.25	14 May 2008	General	- Updated Revision History Table format. Corrected other paragraph formats.
V03.26	12 Sep 2012	General	- Added missing full stops. Removed redundant quotation marks.

6.1 Introduction

The S12XDBG module provides an on-chip trace buffer with flexible triggering capability to allow non-intrusive debug of application software. The S12XDBG module is optimized for the S12X 16-bit architecture and allows debugging of CPU12X module operations.

Typically the S12XDBG module is used in conjunction with the S12XBDM module, whereby the user configures the S12XDBG module for a debugging session over the BDM interface. Once configured the S12XDBG module is armed and the device leaves BDM Mode returning control to the user program, which is then monitored by the S12XDBG module. Alternatively the S12XDBG module can be configured over a serial interface using SWI routines.

6.1.1 Glossary

Table 6-2. Glossary Of Terms

Term	Definition
COF	Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt
BDM	Background Debug Mode
DUG	Device User Guide, describing the features of the device into which the DBG is integrated
WORD	16-bit data entity

Table 6-4. DBGC1 Field Descriptions (continued)

Field	Description
3 DBGBRK	**S12XDBG Breakpoint Enable Bit** — The DBGBRK bit controls whether the debugger will request a breakpoint to S12XCPU upon reaching the state sequencer Final State. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately. Please refer to Section 6.4.7 for further details. 0 No breakpoint on trigger. 1 Breakpoint on trigger
1–0 COMRV	**Comparator Register Visibility Bits** — These bits determine which bank of comparator register is visible in the 8-byte window of the S12XDBG module address map, located between 0x0028 to 0x002F. Furthermore these bits determine which register is visible at the address 0x0027. See Table 6-5.

Table 6-5. COMRV Encoding

COMRV	Visible Comparator	Visible Register at 0x0027
00	Comparator A	DBGSCR1
01	Comparator B	DBGSCR2
10	Comparator C	DBGSCR3
11	Comparator D	DBGMFR

6.3.2.2 Debug Status Register (DBGSR)

Address: 0x0021

	7	6	5	4	3	2	1	0
R	TBF	0	0	0	0	SSF2	SSF1	SSF0
W								
Reset	—	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 6-4. Debug Status Register (DBGSR)

Read: Anytime

Write: Never

Table 6-6. DBGSR Field Descriptions

Field	Description
7 TBF	**Trace Buffer Full** — 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.
2–0 SSF[2:0]	**State Sequencer Flag Bits** — 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 6-7.

```
SUB_1    BRN     *                       ; JMP Destination address TRACE BUFFER ENTRY 1
                                         ; RTI Destination address TRACE BUFFER ENTRY 3
         NOP                             ;
ADDR1    DBNE    A,PART5                 ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR  LDAB    #$F0                    ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
         STAB    VAR_C1
         RTI                             ;
```

The execution flow taking into account the IRQ is as follows

```
         LDX     #SUB_1
MARK1    JMP     0,X                     ;
IRQ_ISR  LDAB    #$F0                    ;
         STAB    VAR_C1
         RTI                             ;
SUB_1    BRN     *
         NOP                             ;
ADDR1    DBNE    A,PART5                 ;
```

6.4.5.2.2 Loop1 Mode

Loop1 Mode, similarly to Normal Mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 Mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the S12XDBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 Mode only inhibits consecutive duplicate source address entries that would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user's code that the S12XDBG module is designed to help find.

6.4.5.2.3 Detail Mode

In Detail Mode, address and data for all memory and register accesses is stored in the trace buffer. This mode also features information byte entries to the trace buffer, for each address byte entry. The information byte indicates the size of access (word or byte) and the type of access (read or write).

When tracing CPU12X activity in Detail Mode, all cycles are traced except those when the CPU12X is either in a free or opcode fetch cycle, the address range can be limited to a range specified by the TRANGE bits in DBGTCR. This function uses comparators C and D to define an address range inside which CPU12X activity should be traced (see Table 6-40). Thus the traced CPU12X activity can be restricted to particular register range accesses.

6.4.5.2.4 Pure PC Mode

In Pure PC Mode, tracing from the CPU the PC addresses of all executed opcodes, including illegal opcodes, are stored.

Chapter 11
Freescale's Scalable Controller Area Network (S12MSCANV3)

Table 11-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V03.11	31 Mar 2009		• Orthographic corrections
V03.12	09 Aug 2010	Table 11-37	• Added 'Bosch CAN 2.0A/B' to bit time settings table
V03.13	03 Mar 2011	Figure 11-4 Table 11-3	• Corrected CANE write restrictions • Removed footnote from RXFRM bit

11.1 Introduction

Freescale's scalable controller area network (S12MSCANV3) definition is based on the MSCAN12 definition, which is the specific implementation of the MSCAN concept targeted for the M68HC12 microcontroller family.

The module is a communication controller implementing the CAN 2.0A/B protocol as defined in the Bosch specification dated September 1991. For users to fully understand the MSCAN specification, it is recommended that the Bosch specification be read first to familiarize the reader with the terms and concepts contained within this document.

Though not exclusively intended for automotive applications, CAN protocol is designed to meet the specific requirements of a vehicle serial data bus: real-time processing, reliable operation in the EMI environment of a vehicle, cost-effectiveness, and required bandwidth.

MSCAN uses an advanced buffer arrangement resulting in predictable real-time behavior and simplified application software.

11.1.1 Glossary

Table 11-2. Terminology

ACK	Acknowledge of CAN message
CAN	Controller Area Network
CRC	Cyclic Redundancy Code
EOF	End of Frame
FIFO	First-In-First-Out Memory
IFS	Inter-Frame Sequence
SOF	Start of Frame
CPU bus	CPU related read/write data bus
CAN bus	CAN protocol related serial bus
oscillator clock	Direct clock from external oscillator
bus clock	CPU bus related clock
CAN clock	CAN protocol related clock

11.1.2 Block Diagram

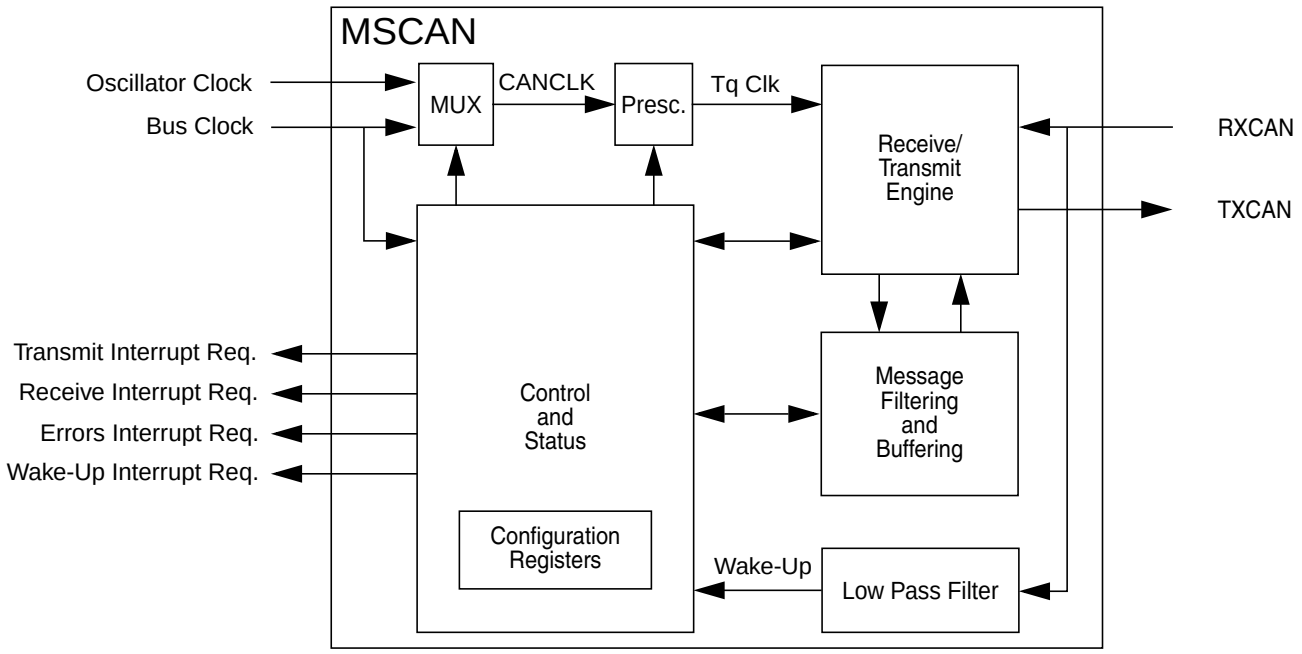

Figure 11-1. MSCAN Block Diagram

[1] The address bit identifies the frame as an address character. See Section 14.4.6.6, “Receiver Wakeup”.

14.4.4 Baud Rate Generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and the transmitter. The value from 0 to 8191 written to the SBR12:SBR0 bits determines the bus clock divisor. The SBR bits are in the SCI baud rate registers (SCIBDH and SCIBDL). The baud rate clock is synchronized with the bus clock and drives the receiver. The baud rate clock divided by 16 drives the transmitter. The receiver has an acquisition rate of 16 samples per bit time.

Baud rate generation is subject to one source of error:

- Integer division of the bus clock may not give the exact target frequency.

Table 14-16 lists some examples of achieving target baud rates with a bus clock frequency of 25 MHz.

When IREN = 0 then,

SCI baud rate = SCI bus clock / (16 * SCIBR[12:0])

Table 14-16. Baud Rates (Example: Bus Clock = 25 MHz)

Bits SBR[12:0]	Receiver Clock (Hz)	Transmitter Clock (Hz)	Target Baud Rate	Error (%)
41	609,756.1	38,109.8	38,400	.76
81	308,642.0	19,290.1	19,200	.47
163	153,374.2	9585.9	9,600	.16
326	76,687.1	4792.9	4,800	.15
651	38,402.5	2400.2	2,400	.01
1302	19,201.2	1200.1	1,200	.01
2604	9600.6	600.0	600	.00
5208	4800.0	300.0	300	.00

14.4.6.5.2 Fast Data Tolerance

Figure 14-29 shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

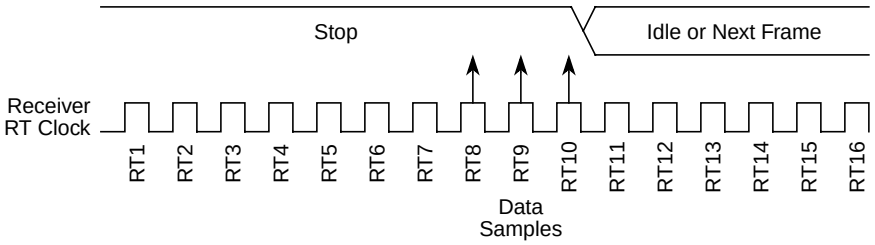

Figure 14-29. Fast Data

For an 8-bit data character, it takes the receiver 9 bit times x 16 RTr cycles + 10 RTr cycles = 154 RTr cycles to finish data sampling of the stop bit.

With the misaligned character shown in Figure 14-29, the receiver counts 154 RTr cycles at the point when the count of the transmitting device is 10 bit times x 16 RTt cycles = 160 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is:

((160 – 154) / 160) x 100 = 3.75%

For a 9-bit data character, it takes the receiver 10 bit times x 16 RTr cycles + 10 RTr cycles = 170 RTr cycles to finish data sampling of the stop bit.

With the misaligned character shown in Figure 14-29, the receiver counts 170 RTr cycles at the point when the count of the transmitting device is 11 bit times x 16 RTt cycles = 176 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

((176 – 170) /176) x 100 = 3.40%

14.4.6.6 Receiver Wakeup

To enable the SCI to ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCI control register 2 (SCICR2) puts the receiver into standby state during which receiver interrupts are disabled.The SCI will still load the receive data into the SCIDRH/L registers, but it will not set the RDRF flag.

The transmitting device can address messages to selected receivers by including addressing information in the initial frame or frames of each message.

The WAKE bit in SCI control register 1 (SCICR1) determines how the SCI is brought out of the standby state to process an incoming message. The WAKE bit enables either idle line wakeup or address mark wakeup.

Table 15-7. Example SPI Baud Rate Selection (25 MHz Bus Clock) (Sheet 2 of 3)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	1	1	0	0	64	390.63 kbit/s
0	0	1	1	0	1	128	195.31 kbit/s
0	0	1	1	1	0	256	97.66 kbit/s
0	0	1	1	1	1	512	48.83 kbit/s
0	1	0	0	0	0	6	4.16667 Mbit/s
0	1	0	0	0	1	12	2.08333 Mbit/s
0	1	0	0	1	0	24	1.04167 Mbit/s
0	1	0	0	1	1	48	520.83 kbit/s
0	1	0	1	0	0	96	260.42 kbit/s
0	1	0	1	0	1	192	130.21 kbit/s
0	1	0	1	1	0	384	65.10 kbit/s
0	1	0	1	1	1	768	32.55 kbit/s
0	1	1	0	0	0	8	3.125 Mbit/s
0	1	1	0	0	1	16	1.5625 Mbit/s
0	1	1	0	1	0	32	781.25 kbit/s
0	1	1	0	1	1	64	390.63 kbit/s
0	1	1	1	0	0	128	195.31 kbit/s
0	1	1	1	0	1	256	97.66 kbit/s
0	1	1	1	1	0	512	48.83 kbit/s
0	1	1	1	1	1	1024	24.41 kbit/s
1	0	0	0	0	0	10	2.5 Mbit/s
1	0	0	0	0	1	20	1.25 Mbit/s
1	0	0	0	1	0	40	625 kbit/s
1	0	0	0	1	1	80	312.5 kbit/s
1	0	0	1	0	0	160	156.25 kbit/s
1	0	0	1	0	1	320	78.13 kbit/s
1	0	0	1	1	0	640	39.06 kbit/s
1	0	0	1	1	1	1280	19.53 kbit/s
1	0	1	0	0	0	12	2.08333 Mbit/s
1	0	1	0	0	1	24	1.04167 Mbit/s
1	0	1	0	1	0	48	520.83 kbit/s
1	0	1	0	1	1	96	260.42 kbit/s
1	0	1	1	0	0	192	130.21 kbit/s
1	0	1	1	0	1	384	65.10 kbit/s
1	0	1	1	1	0	768	32.55 kbit/s
1	0	1	1	1	1	1536	16.28 kbit/s
1	1	0	0	0	0	14	1.78571 Mbit/s
1	1	0	0	0	1	28	892.86 kbit/s
1	1	0	0	1	0	56	446.43 kbit/s
1	1	0	0	1	1	112	223.21 kbit/s
1	1	0	1	0	0	224	111.61 kbit/s
1	1	0	1	0	1	448	55.80 kbit/s

16.3.2.8 Timer Control Register 1/Timer Control Register 2 (TCTL1/TCTL2)

Module Base + 0x0008

	7	6	5	4	3	2	1	0
R W	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
Reset	0	0	0	0	0	0	0	0

Figure 16-14. Timer Control Register 1 (TCTL1)

Module Base + 0x0009

	7	6	5	4	3	2	1	0
R W	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
Reset	0	0	0	0	0	0	0	0

Figure 16-15. Timer Control Register 2 (TCTL2)

Read: Anytime

Write: Anytime

Table 16-8. TCTL1/TCTL2 Field Descriptions

Field	Description
7:0 OMx	**Output Mode** — These eight pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare. When either OMx or OLx is 1, the pin associated with OCx becomes an output tied to OCx. **Note:** To enable output action by OMx bits on timer port, the corresponding bit in OC7M should be cleared. For an output line to be driven by an OCx the OCPDx must be cleared.
7:0 OLx	**Output Level** — These eight pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare. When either OMx or OLx is 1, the pin associated with OCx becomes an output tied to OCx. **Note:** To enable output action by OLx bits on timer port, the corresponding bit in OC7M should be cleared. For an output line to be driven by an OCx the OCPDx must be cleared.

Table 16-9. Compare Result Output Action

OMx	OLx	Action
0	0	No output compare action on the timer output signal
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

16.3.2.15 16-Bit Pulse Accumulator Control Register (PACTL)

Module Base + 0x0020

	7	6	5	4	3	2	1	0
R	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
W								
Reset	0	0	0	0	0	0	0	0

Unimplemented or Reserved

Figure 16-24. 16-Bit Pulse Accumulator Control Register (PACTL)

When PAEN is set, the PACT is enabled.The PACT shares the input pin with IOC7.

Read: Any time

Write: Any time

Table 16-18. PACTL Field Descriptions

Field	Description
6 PAEN	**Pulse Accumulator System Enable** — PAEN is independent from TEN. With timer disabled, the pulse accumulator can function unless pulse accumulator is disabled. 0 16-Bit Pulse Accumulator system disabled. 1 Pulse Accumulator system enabled.
5 PAMOD	**Pulse Accumulator Mode** — This bit is active only when the Pulse Accumulator is enabled (PAEN = 1). See Table 16-19. 0 Event counter mode. 1 Gated time accumulation mode.
4 PEDGE	**Pulse Accumulator Edge Control** — This bit is active only when the Pulse Accumulator is enabled (PAEN = 1). For PAMOD bit = 0 (event counter mode). See Table 16-19. 0 Falling edges on IOC7 pin cause the count to be incremented. 1 Rising edges on IOC7 pin cause the count to be incremented. For PAMOD bit = 1 (gated time accumulation mode). 0 IOC7 input pin high enables M (bus clock) divided by 64 clock to Pulse Accumulator and the trailing falling edge on IOC7 sets the PAIF flag. 1 IOC7 input pin low enables M (bus clock) divided by 64 clock to Pulse Accumulator and the trailing rising edge on IOC7 sets the PAIF flag.
3:2 CLK[1:0]	**Clock Select Bits —** Refer to Table 16-20.
1 PAOVI	**Pulse Accumulator Overflow Interrupt Enable** 0 Interrupt inhibited. 1 Interrupt requested if PAOVF is set.
0 PAI	**Pulse Accumulator Input Interrupt Enable** 0 Interrupt inhibited. 1 Interrupt requested if PAIF is set.

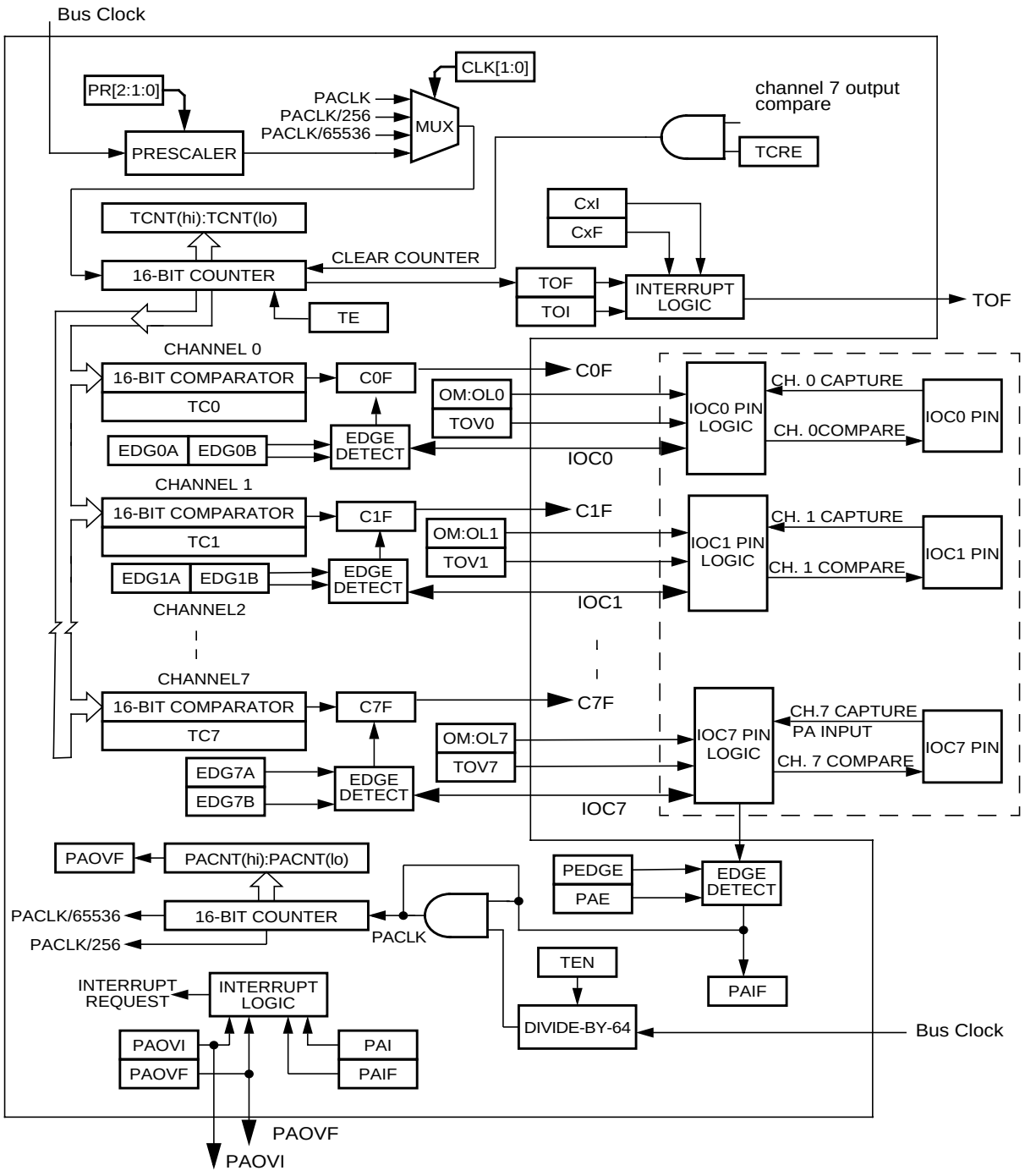

Figure 16-30. Detailed Timer Block Diagram

16.4.1 Prescaler

The prescaler divides the bus clock by 1,2,4,8,16,32,64 or 128. The prescaler select bits, PR[2:0], select the prescaler divisor. PR[2:0] are in timer system control register 2 (TSCR2).

17.3.2.2 Control Register (VREGCTRL)

The VREGCTRL register allows the configuration of the VREG_3V3 low-voltage detect features.

0x02F1

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	LVDS	LVIE	LVIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 17-2. Control Register (VREGCTRL)

Table 17-5. VREGCTRL Field Descriptions

Field	Description
2 LVDS	**Low-Voltage Detect Status Bit** — This read-only status bit reflects the input voltage. Writes have no effect. 0 Input voltage V_{DDA} is above level V_{LVID} or RPM or shutdown mode. 1 Input voltage V_{DDA} is below level V_{LVIA} and FPM.
1 LVIE	**Low-Voltage Interrupt Enable Bit** 0 Interrupt request is disabled. 1 Interrupt will be requested whenever LVIF is set.
0 LVIF	**Low-Voltage Interrupt Flag** — LVIF is set to 1 when LVDS status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LVIE = 1), LVIF causes an interrupt request. 0 No change in LVDS bit. 1 LVDS bit has changed. **Note:** On entering the Reduced Power Mode the LVIF is not cleared by the VREG_3V3.

keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 18-10) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash block 0 will not be available for read access and will return invalid data.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 18.3.2.2), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 18.4.2.11
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x7F_FF00–0x7F_FF07 in the Flash configuration field.

The security as defined in the Flash security byte (0x7F_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x7F_FF00–0x7F_FF07 are unaffected by the Verify Backdoor Access Key command sequence. After the next reset of the MCU, the security state of the Flash module is determined by the Flash security byte (0x7F_FF0F). The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

18.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

The MCU can be unsecured in special single chip mode by erasing the P-Flash and D-Flash memory by one of the following methods:

- Reset the MCU into special single chip mode, delay while the erase test is performed by the BDM, send BDM commands to disable protection in the P-Flash and D-Flash memory, and execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory.
- Reset the MCU into special expanded wide mode, disable protection in the P-Flash and D-Flash memory and run code from external memory to execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory.

After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode. The BDM will execute the Erase Verify All Blocks command write sequence to verify that the P-Flash and D-Flash memory is erased. If the P-Flash and D-Flash memory are verified as

Table A-1. Absolute Maximum Ratings[1]

Num	Rating	Symbol	Min	Max	Unit
1	I/O, regulator and analog supply voltage	V_{DD35}	–0.3	6.0	V
2	Digital logic supply voltage[2]	V_{DD}	–0.3	2.16	V
3	PLL supply voltage[2]	V_{DDPLL}	–0.3	2.16	V
4	NVM supply voltage[2]	V_{DDF}	–0.3	3.6	V
5	Voltage difference V_{DDX} to V_{DDA}	Δ_{VDDX}	–6.0	0.3	V
6	Voltage difference V_{SSX} to V_{SSA}	Δ_{VSSX}	–0.3	0.3	V
7	Digital I/O input voltage	V_{IN}	–0.3	6.0	V
8	Analog reference	V_{RH}, V_{RL}	–0.3	6.0	V
9	EXTAL, XTAL	V_{ILV}	–0.3	2.16	V
11	Instantaneous maximum current Single pin limit for all digital I/O pins[3]	I_D	–25	+25	mA
12	Instantaneous maximum current Single pin limit for EXTAL, XTAL[4]	I_{DL}	–25	+25	mA
14	Maximum current Single pin limit for power supply pins	I_{DV}	–100	+100	mA
15	Storage temperature range	T_{stg}	–65	155	°C

1 Beyond absolute maximum ratings device might be damaged.
2 The device contains an internal voltage regulator to generate the logic and PLL supply out of the I/O supply. The absolute maximum ratings apply when the device is powered from an external source.
3 All digital I/O pins are internally clamped to V_{SSX} and V_{DDX}, or V_{SSA} and V_{DDA}.
4 Those pins are internally clamped to V_{SSPLL} and V_{DDPLL}.

A.1.6 ESD Protection and Latch-up Immunity

All ESD testing is in conformity with CDF-AEC-Q100 stress test qualification for automotive grade integrated circuits. During the device qualification ESD stresses were performed for the Human Body Model (HBM) and the Charge Device Model.

A device will be defined as a failure if after exposure to ESD pulses the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature followed by hot temperature, unless specified otherwise in the device specification.

A.3.1.8 Erase P-Flash Block (FCMD=0x09)

Erasing a 256K NVM block takes

$$t_{mass} \approx 100100 \cdot \frac{1}{f_{NVMOP}} + 70000 \cdot \frac{1}{f_{NVMBUS}}$$

Erasing a 128K NVM block takes

$$t_{mass} \approx 100100 \cdot \frac{1}{f_{NVMOP}} + 35000 \cdot \frac{1}{f_{NVMBUS}}$$

A.3.1.9 Erase P-Flash Sector (FCMD=0x0A)

The typical time to erase a1024-byte P-Flash sector can be calculated using

$$t_{era} = \left(20020 \cdot \frac{1}{f_{NVMOP}}\right) + \left(700 \cdot \frac{1}{f_{NVMBUS}}\right)$$

The maximum time to erase a1024-byte P-Flash sector can be calculated using

$$t_{era} = \left(20020 \cdot \frac{1}{f_{NVMOP}}\right) + \left(1100 \cdot \frac{1}{f_{NVMBUS}}\right)$$

A.3.1.10 Unsecure Flash (FCMD=0x0B)

The maximum time for unsecuring the flash is given by

$$t_{uns} = \left(100100 \cdot \frac{1}{f_{NVMOP}} + 70000 \cdot \frac{1}{f_{NVMBUS}}\right)$$

A.3.1.11 Verify Backdoor Access Key (FCMD=0x0C)

The maximum verify backdoor access key time is given by

$$t = 400 \cdot \frac{1}{f_{NVMBUS}}$$

A.3.1.12 Set User Margin Level (FCMD=0x0D)

The maximum set user margin level time is given by

$$t = 350 \cdot \frac{1}{f_{NVMBUS}}$$

A.3.1.13 Set Field Margin Level (FCMD=0x0E)

The maximum set field margin level time is given by