



Welcome to [E-XFL.COM](https://www.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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08dv60aclh

Section Number	Title	Page
----------------	-------	------

Chapter 16

Timer Pulse-Width Modulator (S08TPMV3)

16.1	Introduction	317
16.1.1	Features	319
16.1.2	Modes of Operation	319
16.1.3	Block Diagram	320
16.2	Signal Description	322
16.2.1	Detailed Signal Descriptions	322
16.3	Register Definition	326
16.3.1	TPM Status and Control Register (TPMxSC)	326
16.3.2	TPM-Counter Registers (TPMxCNTH:TPMxCNTL)	327
16.3.3	TPM Counter Modulo Registers (TPMxMODH:TPMxMODL)	328
16.3.4	TPM Channel n Status and Control Register (TPMxCnSC)	329
16.3.5	TPM Channel Value Registers (TPMxCnVH:TPMxCnVL)	330
16.4	Functional Description	332
16.4.1	Counter	332
16.4.2	Channel Mode Selection	334
16.5	Reset Overview	337
16.5.1	General	337
16.5.2	Description of Reset Operation	337
16.6	Interrupts	337
16.6.1	General	337
16.6.2	Description of Interrupt Operation	338
16.7	The Differences from TPM v2 to TPM v3.....	339

Chapter 17

Development Support

17.1	Introduction	345
17.1.1	Forcing Active Background	345
17.1.2	Features	346
17.2	Background Debug Controller (BDC)	346
17.2.1	BKGD Pin Description	347
17.2.2	Communication Details	348
17.2.3	BDC Commands	352
17.2.4	BDC Hardware Breakpoint	354
17.3	On-Chip Debug System (DBG)	355
17.3.1	Comparators A and B	355
17.3.2	Bus Capture Information and FIFO Operation	355
17.3.3	Change-of-Flow Information	356
17.3.4	Tag vs. Force Breakpoints and Triggers	356
17.3.5	Trigger Modes	357
17.3.6	Hardware Breakpoints	359



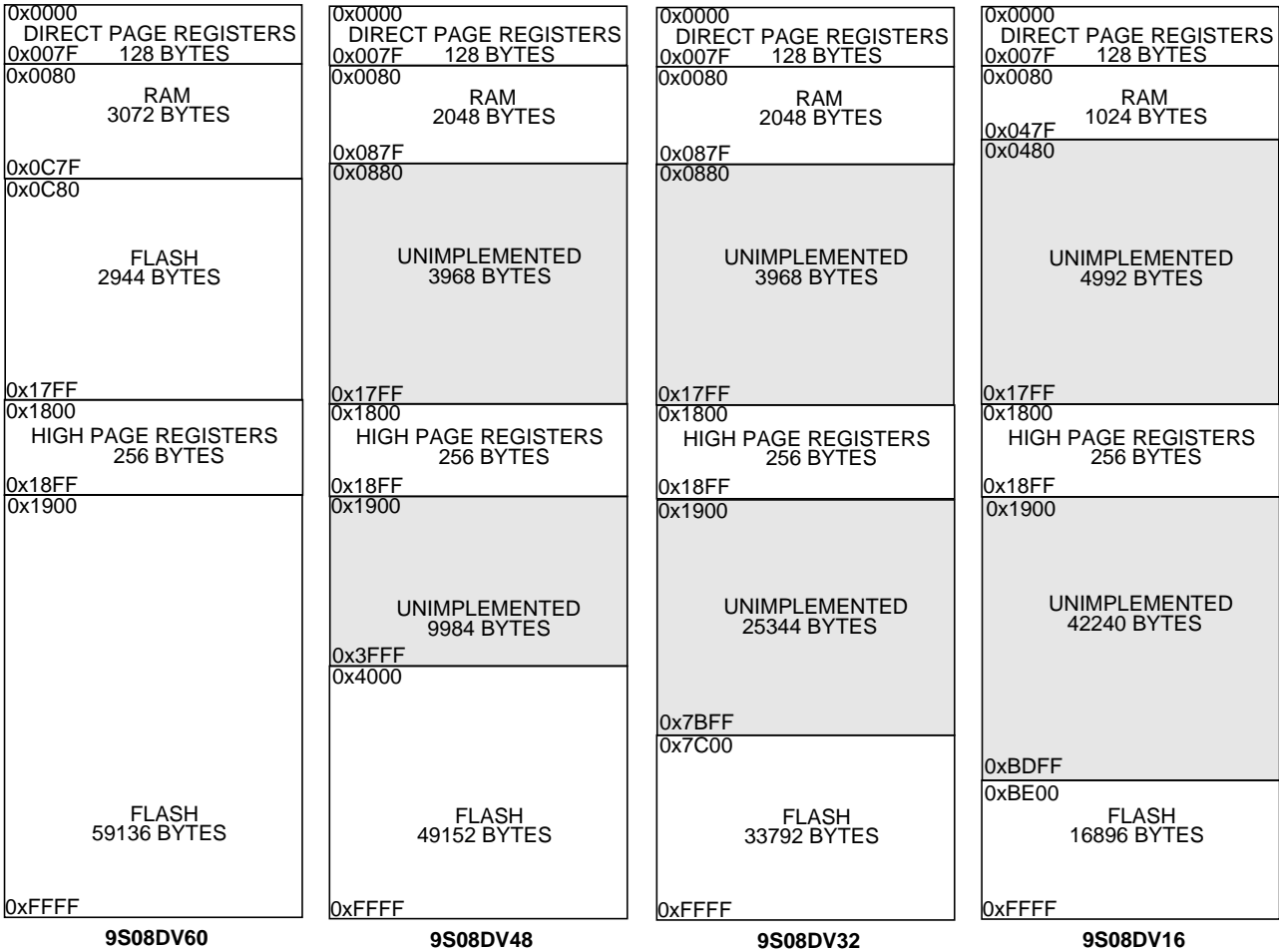


Figure 4-1. MC9S08DV60 Series Memory Map

4.2 Reset and Interrupt Vector Assignments

Table 4-1 shows address assignments for reset and interrupt vectors. The vector names shown in this table are the labels used in the MC9S08DV60 Series equate file provided by Freescale Semiconductor.

Table 4-1. Reset and Interrupt Vectors

Address (High/Low)	Vector	Vector Name
0xFFC0:0xFFC1	ACMP2	Vacmp2
0xFFC2:0xFFC3	ACMP1	Vacmp1
0xFFC4:0xFFC5	MSCAN Transmit	Vcantx
0xFFC6:0xFFC7	MSCAN Receive	Vcanrx
0xFFC8:0xFFC9	MSCAN errors	Vcanerr
0xFFCA:0xFFCB	MSCAN wake up	Vcanwu
0xFFCC:0xFFCD	RTC	Vrtc

5.4 Computer Operating Properly (COP) Watchdog

The COP watchdog is intended to force a system reset when the application software fails to execute as expected. To prevent a system reset from the COP timer (when it is enabled), application software must reset the COP counter periodically. If the application program gets lost and fails to reset the COP counter before it times out, a system reset is generated to force the system back to a known starting point.

After any reset, the COP watchdog is enabled (see [Section 5.8.4, “System Options Register 1 \(SOPT1\),”](#) for additional information). If the COP watchdog is not used in an application, it can be disabled by clearing COPT bits in SOPT1.

The COP counter is reset by writing 0x55 and 0xAA (in this order) to the address of SRS during the selected timeout period. Writes do not affect the data in the read-only SRS. As soon as the write sequence is done, the COP timeout period is restarted. If the program fails to do this during the time-out period, the MCU will reset. Also, if any value other than 0x55 or 0xAA is written to SRS, the MCU is immediately reset.

The COPCLKS bit in SOPT2 (see [Section 5.8.5, “System Options Register 2 \(SOPT2\),”](#) for additional information) selects the clock source used for the COP timer. The clock source options are either the bus clock or an internal 1-kHz clock source. With each clock source, there are three associated time-outs controlled by the COPT bits in SOPT1. [Table 5-6](#) summarizes the control functions of the COPCLKS and COPT bits. The COP watchdog defaults to operation from the 1-kHz clock source and the longest time-out (2^{10} cycles).

When the bus clock source is selected, windowed COP operation is available by setting COPW in the SOPT2 register. In this mode, writes to the SRS register to clear the COP timer must occur in the last 25% of the selected timeout period. A premature write immediately resets the MCU. When the 1-kHz clock source is selected, windowed COP operation is not available.

The COP counter is initialized by the first writes to the SOPT1 and SOPT2 registers and after any system reset. Subsequent writes to SOPT1 and SOPT2 have no effect on COP operation. Even if the application will use the reset default settings of COPT, COPCLKS, and COPW bits, the user should write to the write-once SOPT1 and SOPT2 registers during reset initialization to lock in the settings. This will prevent accidental changes if the application program gets lost.

The write to SRS that services (clears) the COP counter should not be placed in an interrupt service routine (ISR) because the ISR could continue to be executed periodically even if the main application program fails.

If the bus clock source is selected, the COP counter does not increment while the MCU is in background debug mode or while the system is in stop mode. The COP counter resumes when the MCU exits background debug mode or stop mode.

If the 1-kHz clock source is selected, the COP counter is re-initialized to zero upon entry to either background debug mode or stop mode and begins from zero upon exit from background debug mode or stop mode.

6.5.6.3 Port F Pull Enable Register (PTFPE)

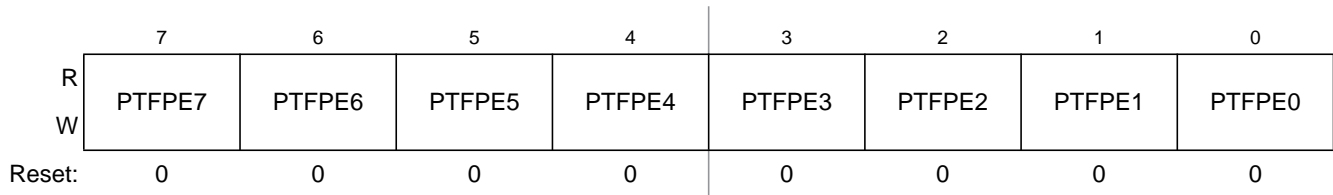


Figure 6-39. Internal Pull Enable for Port F Register (PTFPE)

Table 6-37. PTFPE Register Field Descriptions

Field	Description
7:0 PTFPE[7:0]	Internal Pull Enable for Port F Bits — Each of these control bits determines if the internal pull-up device is enabled for the associated PTF pin. For port F pins that are configured as outputs, these bits have no effect and the internal pull devices are disabled. 0 Internal pull-up device disabled for port F bit n. 1 Internal pull-up device enabled for port F bit n.

NOTE

Pull-down devices only apply when using pin interrupt functions, when corresponding edge select and pin select functions are configured.

6.5.6.4 Port F Slew Rate Enable Register (PTFSE)

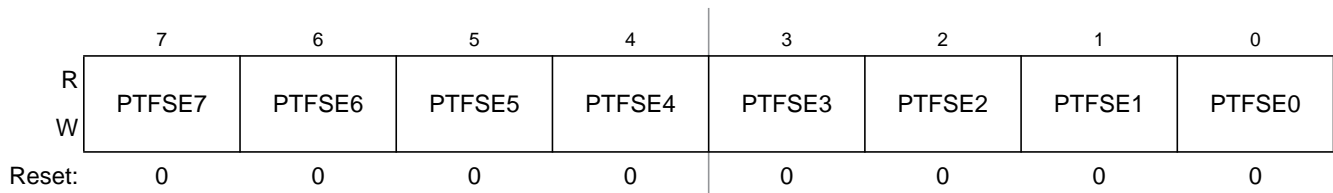


Figure 6-40. Slew Rate Enable for Port F Register (PTFSE)

Table 6-38. PTFSE Register Field Descriptions

Field	Description
7:0 PTFSE[7:0]	Output Slew Rate Enable for Port F Bits — Each of these control bits determines if the output slew rate control is enabled for the associated PTF pin. For port F pins that are configured as inputs, these bits have no effect. 0 Output slew rate control disabled for port F bit n. 1 Output slew rate control enabled for port F bit n.

Note: Slew rate reset default values may differ between engineering samples and final production parts. Always initialize slew rate control to the desired value to ensure correct operation.

7.4.5 BGND Instruction

The BGND instruction is new to the HCS08 compared to the M68HC08. BGND would not be used in normal user programs because it forces the CPU to stop processing user instructions and enter the active background mode. The only way to resume execution of the user program is through reset or by a host debug system issuing a GO, TRACE1, or TAGGO serial command through the background debug interface.

Software-based breakpoints can be set by replacing an opcode at the desired breakpoint address with the BGND opcode. When the program reaches this breakpoint address, the CPU is forced to active background mode rather than continuing the user program.

Chapter 10

Analog-to-Digital Converter (S08ADC12V1)

10.1 Introduction

The 12-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

NOTE

MC9S08DV60 Series devices operate at a higher voltage range (2.7 V to 5.5 V) and do not include stop1 mode. Please ignore references to stop1.

10.1.1 Analog Power and Ground Signal Names

References to V_{DDAD} and V_{SSAD} in this chapter correspond to signals V_{DDA} and V_{SSA} , respectively.

10.1.2 Channel Assignments

NOTE

The ADC channel assignments for the MC9S08DV60 Series devices are shown in [Table 10-1](#). Reserved channels convert to an unknown value.

This chapter shows bits for all S08ADC12V1 channels. MC9S08DV60 Series MCUs do not use all of these channels. All bits corresponding to channels that are not available on a device are reserved.

In 10-bit mode, the ADCCVH register holds the upper two bits of the 10-bit compare value (ADCV[9:8]). These bits are compared to the upper two bits of the result following a conversion in 10-bit mode when the compare function is enabled.

In 8-bit mode, ADCCVH is not used during compare.

10.3.6 Compare Value Low Register (ADCCVL)

This register holds the lower 8 bits of the 12-bit or 10-bit compare value or all 8 bits of the 8-bit compare value. When the compare function is enabled, bits ADCV[7:0] are compared to the lower 8 bits of the result following a conversion in 12-bit, 10-bit or 8-bit mode.

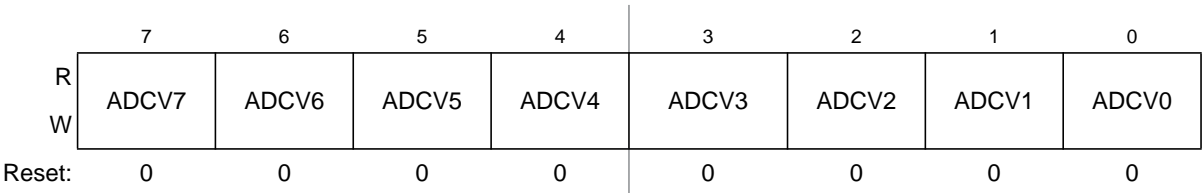


Figure 10-8. Compare Value Low Register (ADCCVL)

10.3.7 Configuration Register (ADCCFG)

ADCCFG selects the mode of operation, clock source, clock divide, and configures for low power and long sample time.

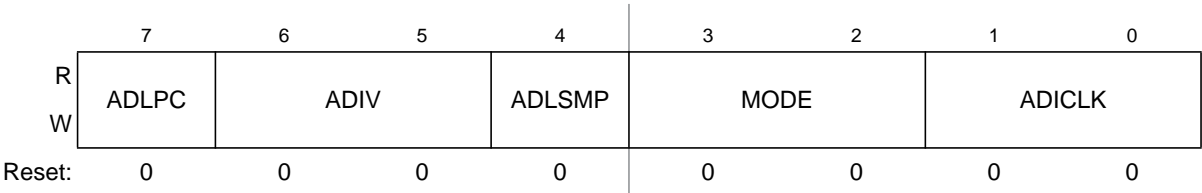


Figure 10-9. Configuration Register (ADCCFG)

Table 10-6. ADCCFG Register Field Descriptions

Field	Description
7 ADLPC	Low-Power Configuration. ADLPC controls the speed and power configuration of the successive approximation converter. This optimizes power consumption when higher sample rates are not required. 0 High speed configuration 1 Low power configuration: The power is reduced at the expense of maximum clock speed.
6:5 ADIV	Clock Divide Select. ADIV selects the divide ratio used by the ADC to generate the internal clock ADCK. Table 10-7 shows the available clock configurations.
4 ADLSMP	Long Sample Time Configuration. ADLSMP selects between long and short sample time. This adjusts the sample period to allow higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption when continuous conversions are enabled if high conversion rates are not required. 0 Short sample time 1 Long sample time

11.3.4 IIC Status Register (IICS)

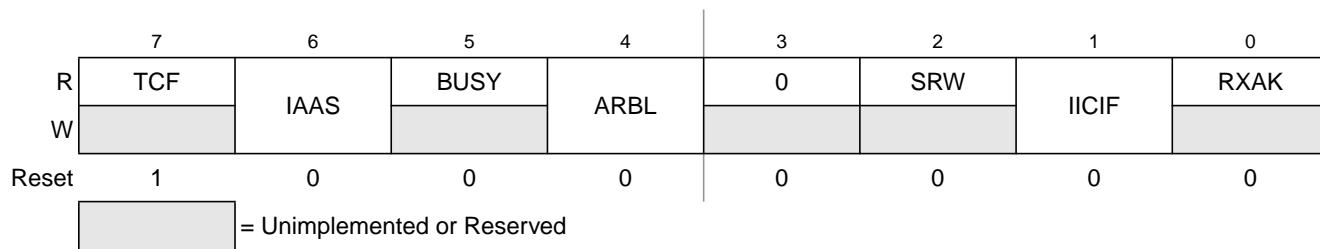


Figure 11-6. IIC Status Register (IICS)

Table 11-6. IICS Field Descriptions

Field	Description
7 TCF	Transfer Complete Flag. This bit is set on the completion of a byte transfer. This bit is only valid during or immediately following a transfer to the IIC module or from the IIC module. The TCF bit is cleared by reading the IICD register in receive mode or writing to the IICD in transmit mode. 0 Transfer in progress 1 Transfer complete
6 IAAS	Addressed as a Slave. The IAAS bit is set when the calling address matches the programmed slave address or when the GCAEN bit is set and a general call is received. Writing the IICC register clears this bit. 0 Not addressed 1 Addressed as a slave
5 BUSY	Bus Busy. The BUSY bit indicates the status of the bus regardless of slave or master mode. The BUSY bit is set when a start signal is detected and cleared when a stop signal is detected. 0 Bus is idle 1 Bus is busy
4 ARBL	Arbitration Lost. This bit is set by hardware when the arbitration procedure is lost. The ARBL bit must be cleared by software by writing a 1 to it. 0 Standard bus operation 1 Loss of arbitration
2 SRW	Slave Read/Write. When addressed as a slave, the SRW bit indicates the value of the R/W command bit of the calling address sent to the master. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IICIF	IIC Interrupt Flag. The IICIF bit is set when an interrupt is pending. This bit must be cleared by software, by writing a 1 to it in the interrupt routine. One of the following events can set the IICIF bit: <ul style="list-style-type: none"> One byte transfer completes Match of slave address to calling address Arbitration lost 0 No interrupt pending 1 Interrupt pending
0 RXAK	Receive Acknowledge. When the RXAK bit is low, it indicates an acknowledge signal has been received after the completion of one byte of data transmission on the bus. If the RXAK bit is high it means that no acknowledge signal is detected. 0 Acknowledge received 1 No acknowledge received

Table 12-1. CANCTL0 Register Field Descriptions (continued)

Field	Description
1 SLPRQ ⁵	<p>Sleep Mode Request — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see Section 12.5.5.4, “MSCAN Sleep Mode”). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPK = 1 (see Section 12.3.2, “MSCAN Control Register 1 (CANCTL1)”). SLPRQ cannot be set while the WUPIF flag is set (see Section 12.3.4.1, “MSCAN Receiver Flag Register (CANRFLG)”). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself.</p> <p>0 Running — The MSCAN functions normally</p> <p>1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle</p>
0 INITRQ ^{6,7}	<p>Initialization Mode Request — When this bit is set by the CPU, the MSCAN skips to initialization mode (see Section 12.5.5.5, “MSCAN Initialization Mode”). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (Section 12.3.2, “MSCAN Control Register 1 (CANCTL1)”).</p> <p>The following registers enter their hard reset state and restore their default values: CANCTL0⁸, CANRFLG⁹, CANRIER¹⁰, CANTFLG, CANTIER, CANTARQ, CANTAAR, and CANTBSEL.</p> <p>The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode.</p> <p>When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits.</p> <p>Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0.</p> <p>0 Normal operation</p> <p>1 MSCAN in initialization mode</p>

¹ The MSCAN must be in normal mode for this bit to become set.

² See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.

³ In order to protect from accidentally violating the CAN protocol, the TXCAN pin is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see [Section 12.5.5.2, “Operation in Wait Mode”](#) and [Section 12.5.5.3, “Operation in Stop Mode”](#)).

⁴ The CPU has to make sure that the WUPE bit and the WUPIE wake-up interrupt enable bit (see [Section 12.3.5, “MSCAN Receiver Interrupt Enable Register \(CANRIER\)”](#)) is enabled, if the recovery mechanism from stop or wait is required.

⁵ The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPK = 1).

⁶ The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).

⁷ In order to protect from accidentally violating the CAN protocol, the TXCAN pin is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPK = 1) before requesting initialization mode.

⁸ Not including WUPE, INITRQ, and SLPRQ.

⁹ TSTAT1 and TSTAT0 are not affected by initialization mode.

¹⁰ RSTAT1 and RSTAT0 are not affected by initialization mode.

Table 13-3. SPIC2 Register Field Descriptions

Field	Description
4 MODFEN	Master Mode-Fault Function Enable — When the SPI is configured for slave mode, this bit has no meaning or effect. (The \overline{SS} pin is the slave select input.) In master mode, this bit determines how the \overline{SS} pin is used (refer to Table 13-2 for more details). 0 Mode fault function disabled, master \overline{SS} pin reverts to general-purpose I/O not controlled by SPI 1 Mode fault function enabled, master \overline{SS} pin acts as the mode fault input or the slave select output
3 BIDIROE	Bidirectional Mode Output Enable — When bidirectional mode is enabled by SPI pin control 0 (SPC0) = 1, BIDIROE determines whether the SPI data output driver is enabled to the single bidirectional SPI I/O pin. Depending on whether the SPI is configured as a master or a slave, it uses either the MOSI (MOMI) or MISO (SISO) pin, respectively, as the single SPI data I/O pin. When SPC0 = 0, BIDIROE has no meaning or effect. 0 Output driver disabled so SPI data I/O pin acts as an input 1 SPI I/O pin enabled as an output
1 SPISWAI	SPI Stop in Wait Mode 0 SPI clocks continue to operate in wait mode 1 SPI clocks stop when the MCU enters wait mode
0 SPC0	SPI Pin Control 0 — The SPC0 bit chooses single-wire bidirectional mode. If MSTR = 0 (slave mode), the SPI uses the MISO (SISO) pin for bidirectional SPI data transfers. If MSTR = 1 (master mode), the SPI uses the MOSI (MOMI) pin for bidirectional SPI data transfers. When SPC0 = 1, BIDIROE is used to enable or disable the output driver for the single bidirectional SPI I/O pin. 0 SPI uses separate pins for data input and data output 1 SPI configured for single-wire bidirectional operation

13.4.3 SPI Baud Rate Register (SPIBR)

This register is used to set the prescaler and bit rate divisor for an SPI master. This register may be read or written at any time.

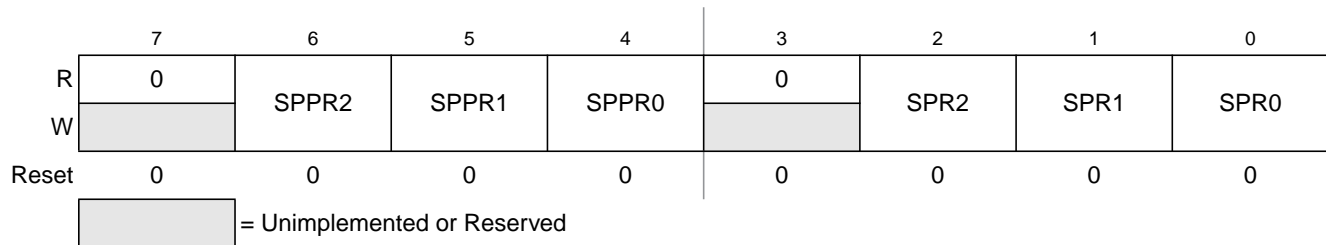


Figure 13-7. SPI Baud Rate Register (SPIBR)

Table 13-4. SPIBR Register Field Descriptions

Field	Description
6:4 SPPR[2:0]	SPI Baud Rate Prescale Divisor — This 3-bit field selects one of eight divisors for the SPI baud rate prescaler as shown in Table 13-5. The input to this prescaler is the bus rate clock (BUSCLK). The output of this prescaler drives the input of the SPI baud rate divider (see Figure 13-4).
2:0 SPR[2:0]	SPI Baud Rate Divisor — This 3-bit field selects one of eight divisors for the SPI baud rate divider as shown in Table 13-6. The input to this divider comes from the SPI baud rate prescaler (see Figure 13-4). The output of this divider is the SPI bit rate clock for master mode.

14.2 Register Definition

The SCI has eight 8-bit registers to control baud rate, select SCI options, report SCI status, and for transmit/receive data.

Refer to the direct-page register summary in the [Memory](#) chapter of this data sheet for the absolute address assignments for all SCI registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

14.2.1 SCI Baud Rate Registers (SCIxBDH, SCIxBDL)

This pair of registers controls the prescale divisor for SCI baud rate generation. To update the 13-bit baud rate setting [SBR12:SBR0], first write to SCIxBDH to buffer the high half of the new value and then write to SCIxBDL. The working value in SCIxBDH does not change until SCIxBDL is written.

SCIxBDL is reset to a non-zero value, so after reset the baud rate generator remains disabled until the first time the receiver or transmitter is enabled (RE or TE bits in SCIxC2 are written to 1).

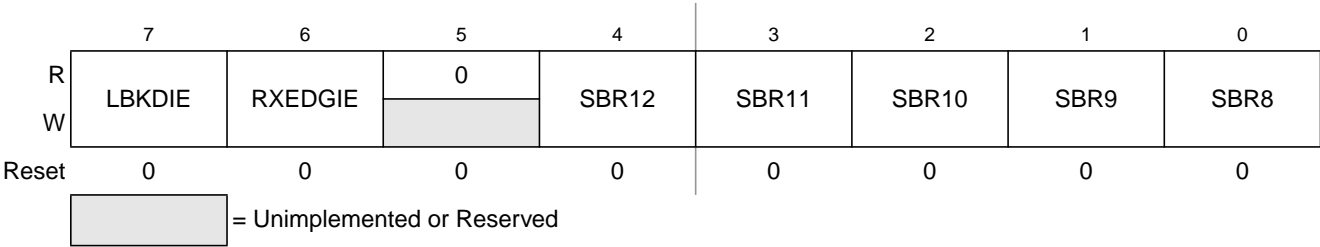


Figure 14-4. SCI Baud Rate Register (SCIxBDH)

Table 14-2. SCIxBDH Field Descriptions

Field	Description
7 LBKDIE	LIN Break Detect Interrupt Enable (for LBKDIF) 0 Hardware interrupts from LBKDIF disabled (use polling). 1 Hardware interrupt requested when LBKDIF flag is 1.
6 RXEDGIE	RxD Input Active Edge Interrupt Enable (for RXEDGIF) 0 Hardware interrupts from RXEDGIF disabled (use polling). 1 Hardware interrupt requested when RXEDGIF flag is 1.
4:0 SBR[12:8]	Baud Rate Modulo Divisor — The 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = $BUSCLK/(16 \times BR)$. See also BR bits in Table 14-3 .

Instead of hardware interrupts, software polling may be used to monitor the TDRE and TC status flags if the corresponding TIE or TCIE local interrupt masks are 0s.

When a program detects that the receive data register is full ($RDRF = 1$), it gets the data from the receive data register by reading `SCIxD`. The `RDRF` flag is cleared by reading `SCIxS1` while $RDRF = 1$ and then reading `SCIxD`.

When polling is used, this sequence is naturally satisfied in the normal course of the user program. If hardware interrupts are used, `SCIxS1` must be read in the interrupt service routine (ISR). Normally, this is done in the ISR anyway to check for receive errors, so the sequence is automatically satisfied.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the `RxD` line remains idle for an extended period of time. IDLE is cleared by reading `SCIxS1` while $IDLE = 1$ and then reading `SCIxD`. After IDLE has been cleared, it cannot become set again until the receiver has received at least one new character and has set `RDRF`.

If the associated error was detected in the received character that caused `RDRF` to be set, the error flags — noise flag (NF), framing error (FE), and parity error flag (PF) — get set at the same time as `RDRF`. These flags are not set in overrun cases.

If `RDRF` was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (OR) flag gets set instead the data along with any associated NF, FE, or PF condition is lost.

At any time, an active edge on the `RxD` serial data input pin causes the `RXEDGIF` flag to set. The `RXEDGIF` flag is cleared by writing a “1” to it. This function does depend on the receiver being enabled ($RE = 1$).

14.3.5 Additional SCI Functions

The following sections describe additional SCI functions.

14.3.5.1 8- and 9-Bit Data Modes

The SCI system (transmitter and receiver) can be configured to operate in 9-bit data mode by setting the `M` control bit in `SCIxC1`. In 9-bit mode, there is a ninth data bit to the left of the MSB of the SCI data register. For the transmit data buffer, this bit is stored in `T8` in `SCIxC3`. For the receiver, the ninth bit is held in `R8` in `SCIxC3`.

For coherent writes to the transmit data buffer, write to the `T8` bit before writing to `SCIxD`.

If the bit value to be transmitted as the ninth bit of a new character is the same as for the previous character, it is not necessary to write to `T8` again. When data is transferred from the transmit data buffer to the transmit shifter, the value in `T8` is copied at the same time data is transferred from `SCIxD` to the shifter.

9-bit data mode typically is used in conjunction with parity to allow eight bits of data plus the parity in the ninth bit. Or it is used with address-mark wakeup so the ninth data bit can serve as the wakeup bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

15.3.2 RTC Counter Register (RTCCNT)

RTCCNT is the read-only value of the current RTC count of the 8-bit counter.

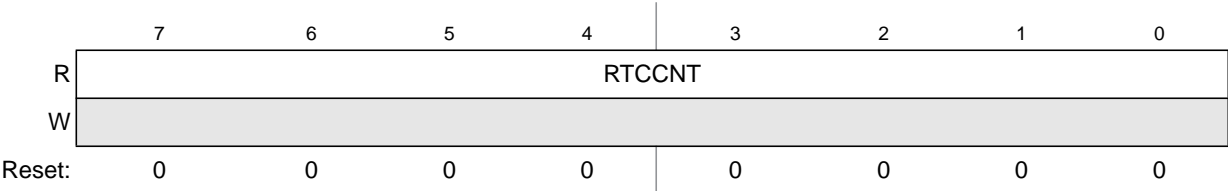


Figure 15-4. RTC Counter Register (RTCCNT)

Table 15-4. RTCCNT Field Descriptions

Field	Description
7:0 RTCCNT	RTC Count. These eight read-only bits contain the current value of the 8-bit counter. Writes have no effect to this register. Reset, writing to RTCMOD, or writing different values to RTCLKS and RTCPS clear the count to 0x00.

15.3.3 RTC Modulo Register (RTCMOD)

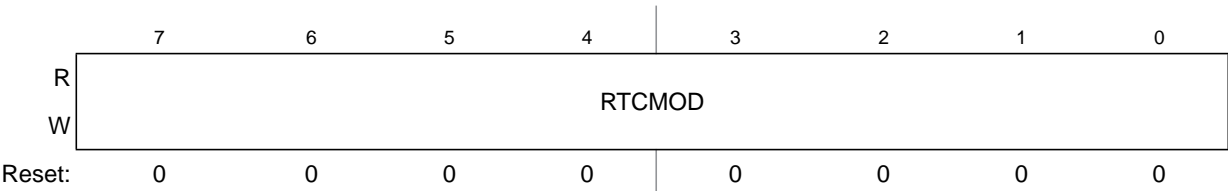


Figure 15-5. RTC Modulo Register (RTCMOD)

Table 15-5. RTCMOD Field Descriptions

Field	Description
7:0 RTCMOD	RTC Modulo. These eight read/write bits contain the modulo value used to reset the count to 0x00 upon a compare match and set the RTIF status bit. A value of 0x00 sets the RTIF bit on each rising edge of the prescaler output. Writing to RTCMOD resets the prescaler and the RTCCNT counters to 0x00. Reset sets the modulo to 0x00.

15.4 Functional Description

The RTC is composed of a main 8-bit up-counter with an 8-bit modulo register, a clock source selector, and a prescaler block with binary-based and decimal-based selectable values. The module also contains software selectable interrupt logic.

After any MCU reset, the counter is stopped and reset to 0x00, the modulus register is set to 0x00, and the prescaler is off. The 1-kHz internal oscillator clock is selected as the default clock source. To start the prescaler, write any value other than zero to the prescaler select bits (RTCPS).

Three clock sources are software selectable: the low power oscillator clock (LPO), the external clock (ERCLK), and the internal clock (IRCLK). The RTC clock select bits (RTCLKS) select the desired clock source. If a different value is written to RTCLKS, the prescaler and RTCCNT counters are reset to 0x00.

Figure 17-4 shows the host receiving a logic 0 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target MCU. The host initiates the bit time but the target HCS08 finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 BDC clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 cycles after starting the bit time.

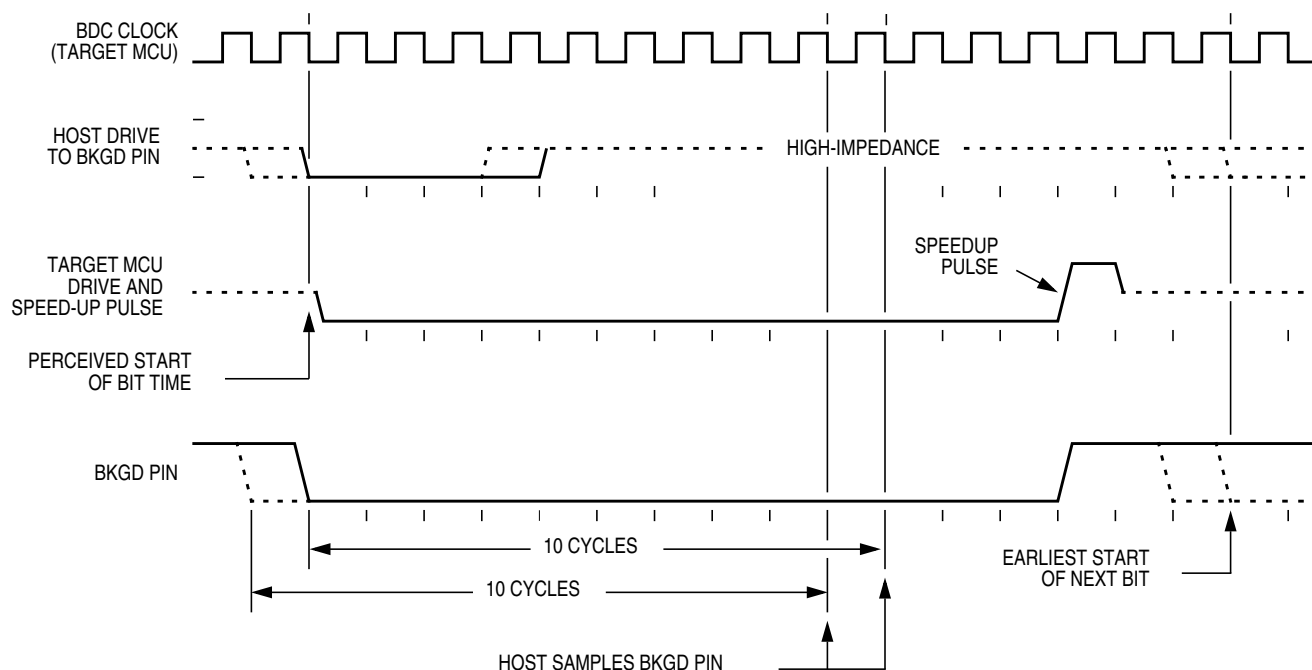


Figure 17-4. BDM Target-to-Host Serial Bit Timing (Logic 0)

17.3.6 Hardware Breakpoints

The BRKEN control bit in the DBGCR register may be set to 1 to allow any of the trigger conditions described in [Section 17.3.5, “Trigger Modes,”](#) to be used to generate a hardware breakpoint request to the CPU. TAG in DBGCR controls whether the breakpoint request will be treated as a tag-type breakpoint or a force-type breakpoint. A tag breakpoint causes the current opcode to be marked as it enters the instruction queue. If a tagged opcode reaches the end of the pipe, the CPU executes a BGND instruction to go to active background mode rather than executing the tagged opcode. A force-type breakpoint causes the CPU to finish the current instruction and then go to active background mode.

If the background mode has not been enabled (ENBDM = 1) by a serial WRITE_CONTROL command through the BKGD pin, the CPU will execute an SWI instruction instead of going to active background mode.

17.4 Register Definition

This section contains the descriptions of the BDC and DBG registers and control bits.

Refer to the high-page register summary in the device overview chapter of this data sheet for the absolute address assignments for all DBG registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

17.4.1 BDC Registers and Control Bits

The BDC has two registers:

- The BDC status and control register (BDCSCR) is an 8-bit register containing control and status bits for the background debug controller.
- The BDC breakpoint match register (BDCBKPT) holds a 16-bit breakpoint match address.

These registers are accessed with dedicated serial BDC commands and are not located in the memory space of the target MCU (so they do not have addresses and cannot be accessed by user programs).

Some of the bits in the BDCSCR have write limitations; otherwise, these registers may be read or written at any time. For example, the ENBDM control bit may not be written while the MCU is in active background mode. (This prevents the ambiguous condition of the control bit forbidding active background mode while the MCU is already in active background mode.) Also, the four status bits (BDMACT, WS, WSF, and DVF) are read-only status indicators and can never be written by the WRITE_CONTROL serial BDC command. The clock switch (CLKSW) control bit may be read or written at any time.

Table 17-2. BDCSCR Register Field Descriptions (continued)

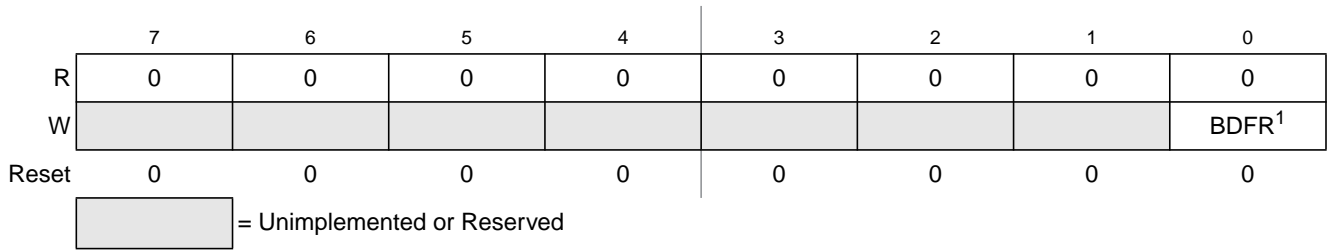
Field	Description
2 WS	Wait or Stop Status — When the target CPU is in wait or stop mode, most BDC commands cannot function. However, the BACKGROUND command can be used to force the target CPU out of wait or stop and into active background mode where all BDC commands work. Whenever the host forces the target MCU into active background mode, the host should issue a READ_STATUS command to check that BDMACT = 1 before attempting other BDC commands. 0 Target CPU is running user application code or in active background mode (was not in wait or stop mode when background became active) 1 Target CPU is in wait or stop mode, or a BACKGROUND command was used to change from wait or stop to active background mode
1 WSF	Wait or Stop Failure Status — This status bit is set if a memory access command failed due to the target CPU executing a wait or stop instruction at or about the same time. The usual recovery strategy is to issue a BACKGROUND command to get out of wait or stop mode into active background mode, repeat the command that failed, then return to the user program. (Typically, the host would restore CPU registers and stack values and re-execute the wait or stop instruction.) 0 Memory access did not conflict with a wait or stop instruction 1 Memory access command failed because the CPU entered wait or stop mode
0 DVF	Data Valid Failure Status — This status bit is not used in the MC9S08DV60 Series because it does not have any slow access memory. 0 Memory access did not conflict with a slow memory access 1 Memory access command failed because CPU was not finished with a slow memory access

17.4.1.2 BDC Breakpoint Match Register (BDCBKPT)

This 16-bit register holds the address for the hardware breakpoint in the BDC. The BKPTEN and FTS control bits in BDCSCR are used to enable and configure the breakpoint logic. Dedicated serial BDC commands (READ_BKPT and WRITE_BKPT) are used to read and write the BDCBKPT register but is not accessible to user programs because it is not located in the normal memory map of the MCU. Breakpoints are normally set while the target MCU is in active background mode before running the user application program. For additional information about setup and use of the hardware breakpoint logic in the BDC, refer to [Section 17.2.4, “BDC Hardware Breakpoint.”](#)

17.4.2 System Background Debug Force Reset Register (SBDFFR)

This register contains a single write-only control bit. A serial background mode command such as WRITE_BYTE must be used to write to SBDFFR. Attempts to write this register from a user program are ignored. Reads always return 0x00.



¹ BDFR is writable only through serial background mode debug commands, not from user programs.

Figure 17-6. System Background Debug Force Reset Register (SBDFR)

Table 17-3. SBDFR Register Field Description

Field	Description
0 BDFR	Background Debug Force Reset — A serial active background mode command such as WRITE_BYTE allows an external debug host to force a target system reset. Writing 1 to this bit forces an MCU reset. This bit cannot be written from a user program.

17.4.3 DBG Registers and Control Bits

The debug module includes nine bytes of register space for three 16-bit registers and three 8-bit control and status registers. These registers are located in the high register space of the normal memory map so they are accessible to normal application programs. These registers are rarely if ever accessed by normal user application programs with the possible exception of a ROM patching mechanism that uses the breakpoint logic.

17.4.3.1 Debug Comparator A High Register (DBGCAH)

This register contains compare value bits for the high-order eight bits of comparator A. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

17.4.3.2 Debug Comparator A Low Register (DBGCAL)

This register contains compare value bits for the low-order eight bits of comparator A. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

17.4.3.3 Debug Comparator B High Register (DBGCBH)

This register contains compare value bits for the high-order eight bits of comparator B. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

17.4.3.4 Debug Comparator B Low Register (DBGCBL)

This register contains compare value bits for the low-order eight bits of comparator B. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

A.12 AC Characteristics

This section describes ac timing characteristics for each peripheral system.

A.12.1 Control Timing

Table A-13. Control Timing

Nu m	C	Rating	Symbol	Min	Typical ¹	Max	Unit
1	D/ P	Bus frequency ($t_{cyc} = 1/f_{Bus}$)	f_{Bus}	dc	—	20	MHz
2	T	Internal low-power oscillator period	t_{LPO}	—	1500	—	μs
3	D	External reset pulse width ²	t_{extrst}	$1.5 \times t_{cyc}$		—	ns
4	D	Reset low drive ³	t_{rstdrv}	$34 \times t_{cyc}$		—	ns
5	D	Active background debug mode latch setup time	t_{MSSU}	25		—	ns
6	D	Active background debug mode latch hold time	t_{MSH}	25		—	ns
7	D	IRQ/PIAx/ PIBx/PIDx pulse width Asynchronous path ² Synchronous path ³	t_{LIH}, t_{IHIL}	100 $1.5 t_{cyc}$	—	—	ns
8	T	Port rise and fall time — Low output drive ($PTxDS = 0$) (load = 50 pF) ⁴ Slew rate control disabled ($PTxSE = 0$) Slew rate control enabled ($PTxSE = 1$)	t_{Rise}, t_{Fall}	— —	40 75		ns
		Port rise and fall time — High output drive ($PTxDS = 1$) (load = 50 pF) ⁴ Slew rate control disabled ($PTxSE = 0$) Slew rate control enabled ($PTxSE = 1$)	t_{Rise}, t_{Fall}	— —	11 35		ns

¹ Typical data was characterized at 5.0 V, 25°C unless otherwise stated.

² This is the shortest pulse that is guaranteed to be recognized as a reset pin request. Shorter pulses are not guaranteed to override reset requests from internal sources.

³ When any reset is initiated, internal circuitry drives the \overline{RESET} pin low for about 34 cycles of t_{cyc} . After POR reset, the bus clock frequency changes to the untrimmed DCO frequency ($f_{reset} = (f_{dco_ut})/4$) because TRIM is reset to 0x80 and FTRIM is reset to 0; and there is an extra divide-by-two because BDIV is reset to 0:1. After other resets, trim stays at the pre-reset value.

⁴ Timing is shown with respect to 20% V_{DD} and 80% V_{DD} levels. Temperature range –40°C to 125°C.

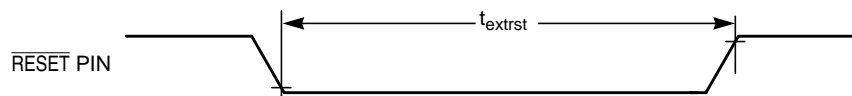


Figure A-2. Reset Timing

A.13 Flash

This section provides details about program/erase times and program-erase endurance for the Flash memory.

Program and erase operations do not require any special power sources other than the normal V_{DD} supply. For more detailed information about program/erase operations, see [Chapter 4, “Memory.”](#)

Table A-17. Flash Characteristics

Num	C	Rating	Symbol	Min	Typical	Max	Unit
1	—	Supply voltage for program/erase	$V_{\text{prog/erase}}$	2.7		5.5	V
2	—	Supply voltage for read operation $0 < f_{\text{Bus}} < 8 \text{ MHz}$ $0 < f_{\text{Bus}} < 20 \text{ MHz}$	V_{Read}	2.7		5.5	V
3	—	Internal FCLK frequency ¹	f_{FCLK}	150		200	kHz
4	—	Internal FCLK period (1/FCLK)	t_{Fcyc}	5		6.67	μs
5	—	Byte program time (random location) ⁽²⁾	t_{prog}	9			t_{Fcyc}
6	—	Byte program time (burst mode) ⁽²⁾	t_{Burst}	4			t_{Fcyc}
7	—	Page erase time ²	t_{Page}	4000			t_{Fcyc}
8	—	Mass erase time ⁽²⁾	t_{Mass}	20,000			t_{Fcyc}
9	C	Flash Program/erase endurance ³ T_L to $T_H = -40^\circ\text{C}$ to $+125^\circ\text{C}$ $T = 25^\circ\text{C}$	n_{FLPE}	10,000 —	— 100,000	— —	cycles
10	C	Data retention ⁴	$t_{\text{D_ret}}$	15	100	—	years

¹ The frequency of this clock is controlled by a software setting.

² These values are hardware state machine controlled. User code does not need to count cycles. This information supplied for calculating approximate time to program and erase.

³ **Typical endurance** for Flash is based on the intrinsic bit cell performance. For additional information on how Freescale Semiconductor defines typical endurance, please refer to Engineering Bulletin EB619, *Typical Endurance for Nonvolatile Memory*.

⁴ **Typical data retention** values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Freescale Semiconductor defines typical data retention, please refer to Engineering Bulletin EB618, *Typical Data Retention for Nonvolatile Memory*.

A.14 EMC Performance

Electromagnetic compatibility (EMC) performance is highly dependant on the environment in which the MCU resides. Board design and layout, circuit topology choices, location and characteristics of external components as well as MCU software operation all play a significant role in EMC performance. The system designer should consult Freescale applications notes such as AN2321, AN1050, AN1263, AN2764, and AN1259 for advice and guidance specifically targeted at optimizing EMC performance.