

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	Obsolete
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I <sup>2</sup> C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dv32clc">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dv32clc</a>

- Write a 1 to the FCBEF bit in FSTAT to clear FCBEF and launch the command (including its address and data information).

A partial command sequence can be aborted manually by writing a 0 to FCBEF any time after the write to the memory array and before writing the 1 that clears FCBEF and launches the complete command. Aborting a command in this way sets the FACCERR access error flag which must be cleared before starting a new command.

A strictly monitored procedure must be obeyed or the command will not be accepted. This minimizes the possibility of any unintended changes to the memory contents. The command complete flag (FCCF) indicates when a command is complete. The command sequence must be completed by clearing FCBEF to launch the command. [Figure 4-2](#) is a flowchart for executing all of the commands except for burst programming and sector erase abort.

- Wait until the FCCF bit in FSTAT is set. As soon as FCCF= 1, the operation has completed successfully.

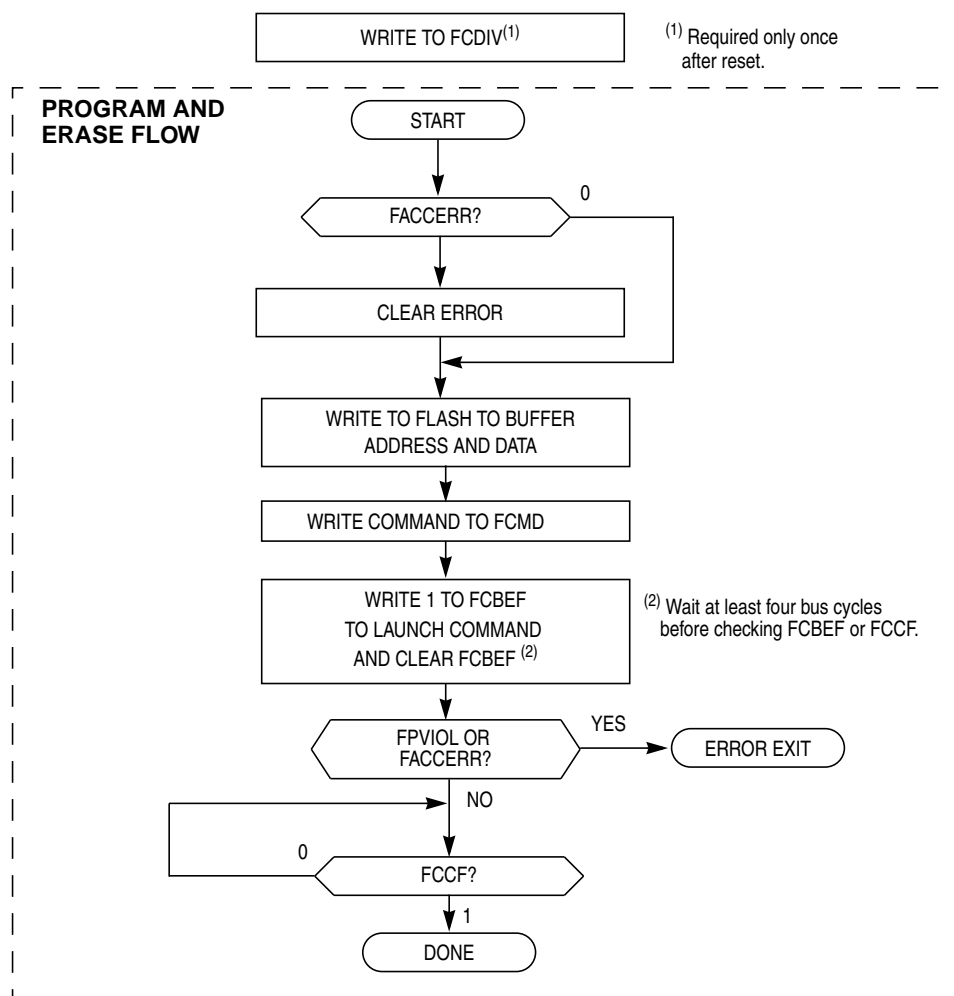


Figure 4-2. Program and Erase Flowchart

### 4.5.7 Block Protection

The block protection feature prevents the protected region of Flash from program or erase changes. Block protection is controlled through the Flash protection register (FPROT). The FPS bits determine the protected region of Flash. See [Section 4.5.10.4, “Flash Protection Register \(FPROT and NVPROT\).”](#)

After exit from reset, FPROT is loaded with the contents of the NVPROT location, which is in the nonvolatile register block of the Flash memory. Any FPROT write that attempts to decrease the size of the protected region will be ignored. Because NVPROT is within the last sector of Flash, if any amount of memory is protected, NVPROT is itself protected and cannot be unprotected (intentionally or unintentionally) by the application software. FPROT can be written through background debug commands, which provides a way to erase and reprogram protected Flash memory.

One use for block protection is to block protect an area of Flash memory for a bootloader program. this bootloader program can call a routine outside of Flash that can be used to sector erase the rest of the Flash memory and reprogram it. The bootloader is protected even if MCU power is lost during an erase and reprogram operation.

### 4.5.8 Vector Redirection

While any Flash is block protected, the reset and interrupt vectors will be protected. Vector redirection allows users to modify interrupt vector information without unprotecting bootloader and reset vector space. Vector redirection is enabled by programming the FNORED bit in the NVOPT register located at address 0xFFBF to 0. For redirection to occur, at least some portion of the Flash memory must be block protected by programming the NVPROT register located at address 0xFFBD. All interrupt vectors (memory locations 0xFFC0–0xFFFFD) are redirected, though the reset vector (0xFFFFE:0xFFFFF) is not.

For example, if 1536 bytes of Flash are protected, the protected address region is from 0xFA00 through 0xFFFFF. The interrupt vectors (0xFFC0–0xFFFFD) are redirected to the locations 0xF9C0–0xF9FD. If vector redirection is enabled and an interrupt occurs, the values in the locations 0xF9E0:0xF9E1 are used for the vector instead of the values in the locations 0xFFE0:0xFFE1. This allows the user to reprogram the unprotected portion of the Flash with new program code including new interrupt vector values while leaving the protected area, which includes the default vector locations, unchanged.

### 4.5.9 Security

The MC9S08DV60 Series includes circuitry to prevent unauthorized access to the contents of Flash and RAM memory. When security is engaged, Flash and RAM are considered secure resources. Direct-page registers, high-page registers, and the background debug controller are considered unsecured resources. Programs executing within secure memory have normal access to any MCU memory locations and resources. Attempts to access a secure memory location with a program executing from an unsecured memory space or through the background debug interface are blocked (writes are ignored and reads return all 0s).

Security is engaged or disengaged based on the state of two register bits (SEC[1:0]) in the FOPT register. During reset, the contents of the nonvolatile location NVOPT are copied from Flash into the working FOPT register in high-page register space. A user engages security by programming the NVOPT location, which can be performed at the same time the Flash memory is programmed. The 1:0 state disengages

In general, whenever a pin is shared with both an alternate digital function and an analog function, the analog function has priority such that if both the digital and analog functions are enabled, the analog function controls the pin.

It is a good programming practice to write to the port data register before changing the direction of a port pin to become an output. This ensures that the pin will not be driven momentarily with an old data value that happened to be in the port data register.

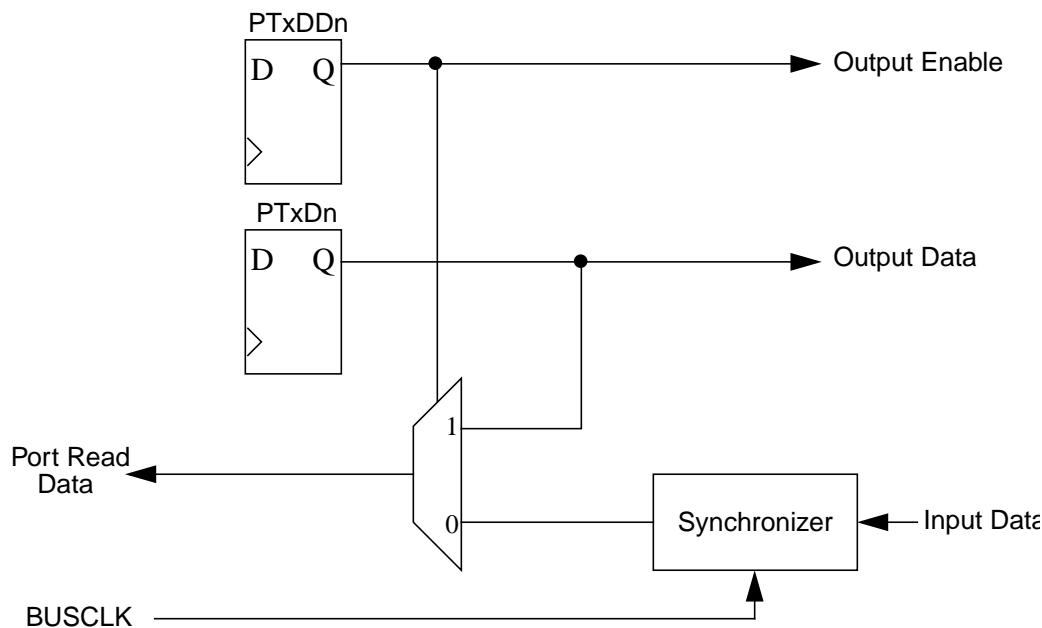


Figure 6-1. Parallel I/O Block Diagram

## 6.2 Pull-up, Slew Rate, and Drive Strength

Associated with the parallel I/O ports is a set of registers located in the high page register space that operate independently of the parallel I/O registers. These registers are used to control pull-ups, slew rate, and drive strength for the pins.

An internal pull-up device can be enabled for each port pin by setting the corresponding bit in the pull-up enable register (PTxPEN). The pull-up device is disabled if the pin is configured as an output by the parallel I/O control logic or any shared peripheral function regardless of the state of the corresponding pull-up enable register bit. The pull-up device is also disabled if the pin is controlled by an analog function.

Slew rate control can be enabled for each port pin by setting the corresponding bit in the slew rate control register (PTxSEn). When enabled, slew control limits the rate at which an output can transition in order to reduce EMC emissions. Slew rate control has no effect on pins that are configured as inputs.

### 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.

### 6.5.3.5 Port C Drive Strength Selection Register (PTCDS)

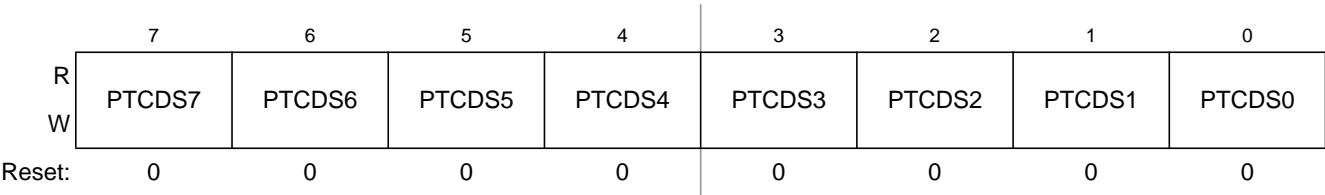


Figure 6-23. Drive Strength Selection for Port C Register (PTCDS)

Table 6-21. PTCDS Register Field Descriptions

Field	Description
7:0 PTCDS[7:0]	<b>Output Drive Strength Selection for Port C Bits</b> — Each of these control bits selects between low and high output drive for the associated PTC pin. For port C pins that are configured as inputs, these bits have no effect. 0 Low output drive strength selected for port C bit n. 1 High output drive strength selected for port C bit n.

### 6.5.4.5 Port D Drive Strength Selection Register (PTDDS)

	7	6	5	4	3	2	1	0
R	PTDDS7	PTDDS6	PTDDS5	PTDDS4	PTDDS3	PTDDS2	PTDDS1	PTDDS0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 6-28. Drive Strength Selection for Port D Register (PTDDS)

Table 6-26. PTDDS Register Field Descriptions

Field	Description
7:0 PTDDS[7:0]	<b>Output Drive Strength Selection for Port D Bits</b> — Each of these control bits selects between low and high output drive for the associated PTD pin. For port D pins that are configured as inputs, these bits have no effect. 0 Low output drive strength selected for port D bit n. 1 High output drive strength selected for port D bit n.

### 6.5.4.6 Port D Interrupt Status and Control Register (PTDSC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTDIF	0	PTDIE	PTDMOD
W						PTDACK		
Reset:	0	0	0	0	0	0	0	0

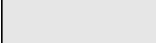
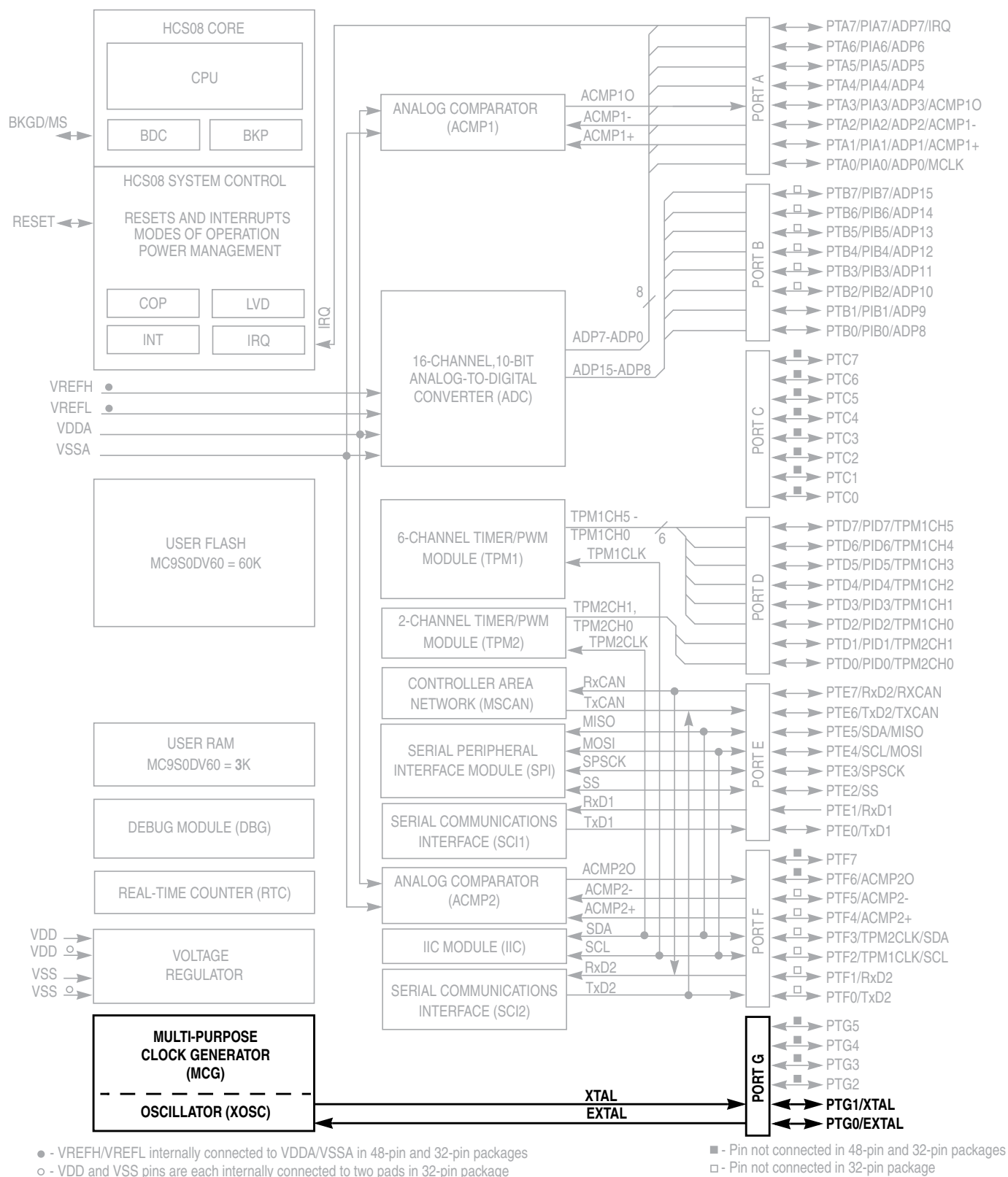
 = Unimplemented or Reserved

Figure 6-29. Port D Interrupt Status and Control Register (PTDSC)

Table 6-27. PTDSC Register Field Descriptions

Field	Description
3 PTDIF	<b>Port D Interrupt Flag</b> — PTDIF indicates when a port D interrupt is detected. Writes have no effect on PTDIF. 0 No port D interrupt detected. 1 Port D interrupt detected.
2 PTDACK	<b>Port D Interrupt Acknowledge</b> — Writing a 1 to PTDACK is part of the flag clearing mechanism. PTDACK always reads as 0.
1 PTDIE	<b>Port D Interrupt Enable</b> — PTDIE determines whether a port D interrupt is requested. 0 Port D interrupt request not enabled. 1 Port D interrupt request enabled.
0 PTDMOD	<b>Port A Detection Mode</b> — PTDMOD (along with the PTDES bits) controls the detection mode of the port D interrupt pins. 0 Port D pins detect edges only. 1 Port D pins detect both edges and levels.



**Figure 8-1. MC9S08DV60 Block Diagram**





### 11.4.3 General Call Address

General calls can be requested in 7-bit address or 10-bit address. If the GCAEN bit is set, the IIC matches the general call address as well as its own slave address. When the IIC responds to a general call, it acts as a slave-receiver and the IAAS bit is set after the address cycle. Software must read the IICD register after the first byte transfer to determine whether the address matches its own slave address or a general call. If the value is 00, the match is a general call. If the GCAEN bit is clear, the IIC ignores any data supplied from a general call address by not issuing an acknowledgement.

## 11.5 Resets

The IIC is disabled after reset. The IIC cannot cause an MCU reset.

## 11.6 Interrupts

The IIC generates a single interrupt.

An interrupt from the IIC is generated when any of the events in [Table 11-11](#) occur, provided the IICIE bit is set. The interrupt is driven by bit IICIF (of the IIC status register) and masked with bit IICIE (of the IIC control register). The IICIF bit must be cleared by software by writing a 1 to it in the interrupt routine. You can determine the interrupt type by reading the status register.

**Table 11-11. Interrupt Summary**

Interrupt Source	Status	Flag	Local Enable
Complete 1-byte transfer	TCF	IICIF	IICIE
Match of received calling address	IAAS	IICIF	IICIE
Arbitration Lost	ARBL	IICIF	IICIE

### 11.6.1 Byte Transfer Interrupt

The TCF (transfer complete flag) bit is set at the falling edge of the ninth clock to indicate the completion of byte transfer.

### 11.6.2 Address Detect Interrupt

When the calling address matches the programmed slave address (IIC address register) or when the GCAEN bit is set and a general call is received, the IAAS bit in the status register is set. The CPU is interrupted, provided the IICIE is set. The CPU must check the SRW bit and set its Tx mode accordingly.

### 11.6.3 Arbitration Lost Interrupt

The IIC is a true multi-master bus that allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, the relative priority of the contending masters is determined by a data arbitration procedure. The IIC module asserts this interrupt when it loses the data arbitration process and the ARBL bit in the status register is set.

## NOTE

The CANRFLG register is held in the reset state<sup>1</sup> when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable again as soon as the initialization mode is exited (INITRQ = 0 and INITAK = 0).

Read: Anytime

Write: Anytime when out of initialization mode, except RSTAT[1:0] and TSTAT[1:0] flags which are read-only; write of 1 clears flag; write of 0 is ignored.

**Table 12-9. CANRFLG Register Field Descriptions**

Field	Description
7 WUPIF	<b>Wake-Up Interrupt Flag</b> — If the MSCAN detects CAN bus activity while in sleep mode (see <a href="#">Section 12.5.5.4, “MSCAN Sleep Mode,”</a> ) and WUPE = 1 in CANTCTL0 (see <a href="#">Section 12.3.1, “MSCAN Control Register 0 (CANCTL0)”</a> ), the module will set WUPIF. If not masked, a wake-up interrupt is pending while this flag is set. 0 No wake-up activity observed while in sleep mode 1 MSCAN detected activity on the CAN bus and requested wake-up
6 CSCIF	<b>CAN Status Change Interrupt Flag</b> — This flag is set when the MSCAN changes its current CAN bus status due to the actual value of the transmit error counter (TEC) and the receive error counter (REC). An additional 4-bit (RSTAT[1:0], TSTAT[1:0]) status register, which is split into separate sections for TEC/REC, informs the system on the actual CAN bus status (see <a href="#">Section 12.3.5, “MSCAN Receiver Interrupt Enable Register (CANRIER)”</a> ). If not masked, an error interrupt is pending while this flag is set. CSCIF provides a blocking interrupt. That guarantees that the receiver/transmitter status bits (RSTAT/TSTAT) are only updated when no CAN status change interrupt is pending. If the TECs/RECs change their current value after the CSCIF is asserted, which would cause an additional state change in the RSTAT/TSTAT bits, these bits keep their status until the current CSCIF interrupt is cleared again. 0 No change in CAN bus status occurred since last interrupt 1 MSCAN changed current CAN bus status
5:4 RSTAT[1:0]	<b>Receiver Status Bits</b> — The values of the error counters control the actual CAN bus status of the MSCAN. As soon as the status change interrupt flag (CSCIF) is set, these bits indicate the appropriate receiver related CAN bus status of the MSCAN. The coding for the bits RSTAT1, RSTAT0 is: 00 RxOK: $0 \leq \text{receive error counter} \leq 96$ 01 RxWRN: $96 < \text{receive error counter} \leq 127$ 10 RxERR: $127 < \text{receive error counter}$ 11 Bus-off <sup>1</sup> : transmit error counter $> 255$
3:2 TSTAT[1:0]	<b>Transmitter Status Bits</b> — The values of the error counters control the actual CAN bus status of the MSCAN. As soon as the status change interrupt flag (CSCIF) is set, these bits indicate the appropriate transmitter related CAN bus status of the MSCAN. The coding for the bits TSTAT1, TSTAT0 is: 00 TxOK: $0 \leq \text{transmit error counter} \leq 96$ 01 TxWRN: $96 < \text{transmit error counter} \leq 127$ 10 TxERR: $127 < \text{transmit error counter} \leq 255$ 11 Bus-Off: transmit error counter $> 255$

1. The RSTAT[1:0], TSTAT[1:0] bits are not affected by initialization mode.

### 12.5.2.1 Message Transmit Background

Modern application layer software is built upon two fundamental assumptions:

- Any CAN node is able to send out a stream of scheduled messages without releasing the CAN bus between the two messages. Such nodes arbitrate for the CAN bus immediately after sending the previous message and only release the CAN bus in case of lost arbitration.
- The internal message queue within any CAN node is organized such that the highest priority message is sent out first, if more than one message is ready to be sent.

The behavior described in the bullets above cannot be achieved with a single transmit buffer. That buffer must be reloaded immediately after the previous message is sent. This loading process lasts a finite amount of time and must be completed within the inter-frame sequence (IFS) to be able to send an uninterrupted stream of messages. Even if this is feasible for limited CAN bus speeds, it requires that the CPU reacts with short latencies to the transmit interrupt.

A double buffer scheme de-couples the reloading of the transmit buffer from the actual message sending and, therefore, reduces the reactivity requirements of the CPU. Problems can arise if the sending of a message is finished while the CPU re-loads the second buffer. No buffer would then be ready for transmission, and the CAN bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN implements with the “local priority” concept described in [Section 12.5.2.2, “Transmit Structures.”](#)

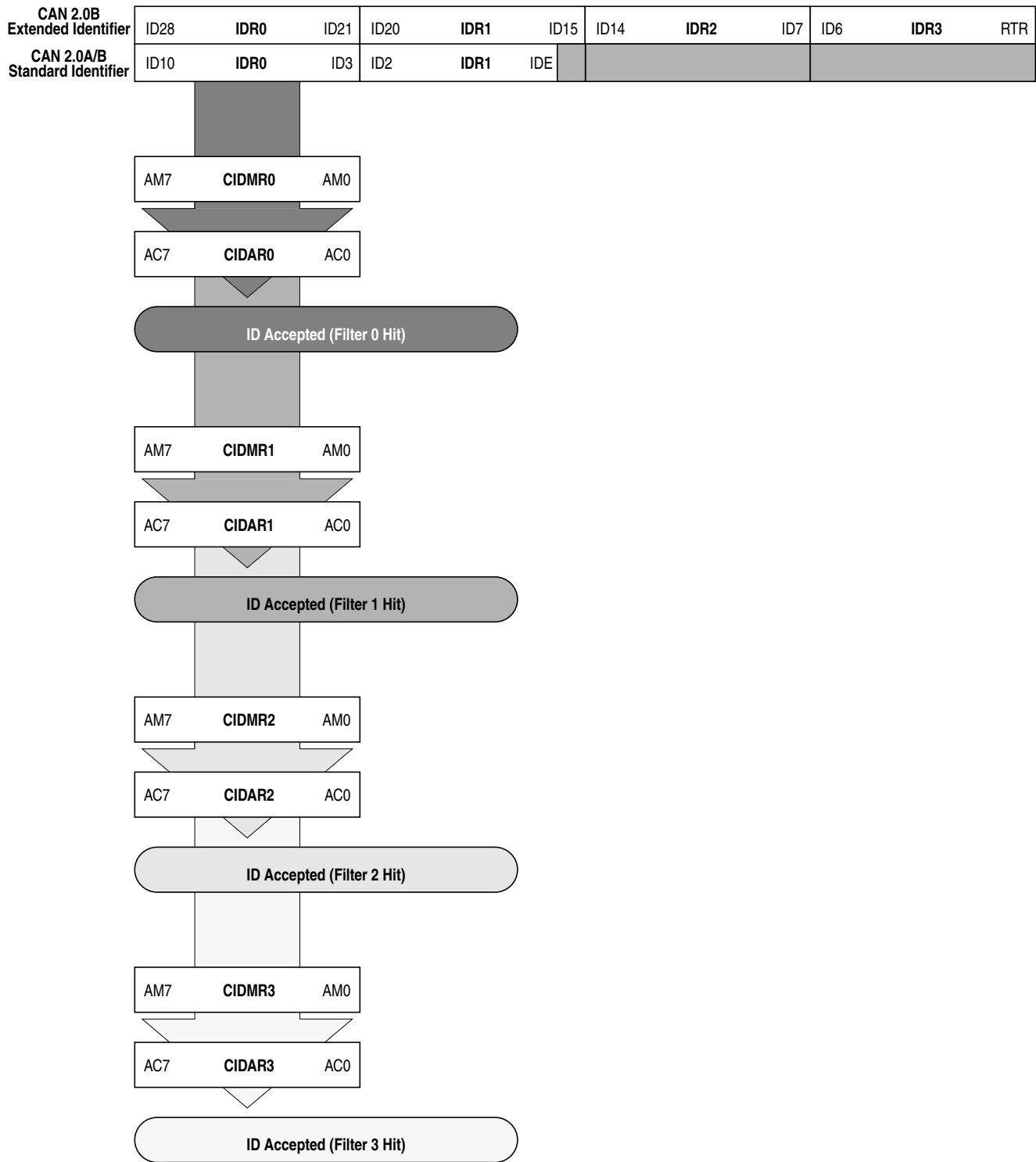
### 12.5.2.2 Transmit Structures

The MSCAN triple transmit buffer scheme optimizes real-time performance by allowing multiple messages to be set up in advance. The three buffers are arranged as shown in [Figure 12-38](#).

All three buffers have a 13-byte data structure similar to the outline of the receive buffers (see [Section 12.4, “Programmer’s Model of Message Storage”](#)). An additional [Section 12.4.5, “Transmit Buffer Priority Register \(TBPR\)”](#) contains an 8-bit local priority field (PRIO) (see [Section 12.4.5, “Transmit Buffer Priority Register \(TBPR\)”](#)). The remaining two bytes are used for time stamping of a message, if required (see [Section 12.4.6, “Time Stamp Register \(TSRH–TSRL\)”](#)).

To transmit a message, the CPU must identify an available transmit buffer, which is indicated by a set transmitter buffer empty (TXEx) flag (see [Section 12.3.6, “MSCAN Transmitter Flag Register \(CANTFLG\)”](#)). If a transmit buffer is available, the CPU must set a pointer to this buffer by writing to the CANTBSEL register (see [Section 12.3.10, “MSCAN Transmit Buffer Selection Register \(CANTBSEL\)”](#)). This makes the respective buffer accessible within the CANTXFG address space (see [Section 12.4, “Programmer’s Model of Message Storage”](#)). The algorithmic feature associated with the CANTBSEL register simplifies the transmit buffer selection. In addition, this scheme makes the handler software simpler because only one address area is applicable for the transmit process, and the required address space is minimized.

The CPU then stores the identifier, the control bits, and the data content into one of the transmit buffers. Finally, the buffer is flagged as ready for transmission by clearing the associated TXE flag.



**Figure 12-41. 8-bit Maskable Identifier Acceptance Filters**

MSCAN filter uses three sets of registers to provide the filter configuration. Firstly, the CANIDAC register determines the configuration of the banks into filter sizes and number of filters. Secondly, registers CANIDMR0/1/2/3 determine those bits on which the filter will operate by placing a '0' at the appropriate

**Table 12-34. Time Segment Syntax**

Syntax	Description
SYNC_SEG	System expects transitions to occur on the CAN bus during this period.
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node in receive mode samples the CAN bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

The synchronization jump width (see the Bosch CAN specification for details) can be programmed in a range of 1 to 4 time quanta by setting the SJW parameter.

The SYNC\_SEG, TSEG1, TSEG2, and SJW parameters are set by programming the MSCAN bus timing registers (CANBTR0, CANBTR1) (see [Section 12.3.3, “MSCAN Bus Timing Register 0 \(CANBTR0\)”](#) and [Section 12.3.4, “MSCAN Bus Timing Register 1 \(CANBTR1\)”](#)).

[Table 12-35](#) gives an overview of the CAN compliant segment settings and the related parameter values.

#### NOTE

It is the user's responsibility to ensure the bit time settings are in compliance with the CAN standard.

**Table 12-35. CAN Standard Compliant Bit Time Segment Settings**

Time Segment 1	TSEG1	Time Segment 2	TSEG2	Synchronization Jump Width	SJW
5 .. 10	4 .. 9	2	1	1 .. 2	0 .. 1
4 .. 11	3 .. 10	3	2	1 .. 3	0 .. 2
5 .. 12	4 .. 11	4	3	1 .. 4	0 .. 3
6 .. 13	5 .. 12	5	4	1 .. 4	0 .. 3
7 .. 14	6 .. 13	6	5	1 .. 4	0 .. 3
8 .. 15	7 .. 14	7	6	1 .. 4	0 .. 3
9 .. 16	8 .. 15	8	7	1 .. 4	0 .. 3

## 12.5.4 Modes of Operation

### 12.5.4.1 Normal Modes

The MSCAN module behaves as described within this specification in all normal system operation modes.

### 12.5.4.2 Special Modes

The MSCAN module behaves as described within this specification in all special system operation modes.



message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force RWU to 0 so all receivers wake up in time to look at the first character(s) of the next message.

#### 14.3.3.2.1 Idle-Line Wakeup

When WAKE = 0, the receiver is configured for idle-line wakeup. In this mode, RWU is cleared automatically when the receiver detects a full character time of the idle-line level. The M control bit selects 8-bit or 9-bit data mode that determines how many bit times of idle are needed to constitute a full character time (10 or 11 bit times because of the start and stop bits).

When RWU is one and RWUID is zero, the idle condition that wakes up the receiver does not set the IDLE flag. The receiver wakes up and waits for the first data character of the next message which will set the RDRF flag and generate an interrupt if enabled. When RWUID is one, any idle condition sets the IDLE flag and generates an interrupt if enabled, regardless of whether RWU is zero or one.

The idle-line type (ILT) control bit selects one of two ways to detect an idle line. When ILT = 0, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When ILT = 1, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

#### 14.3.3.2.2 Address-Mark Wakeup

When WAKE = 1, the receiver is configured for address-mark wakeup. In this mode, RWU is cleared automatically when the receiver detects a logic 1 in the most significant bit of a received character (eighth bit in M = 0 mode and ninth bit in M = 1 mode).

Address-mark wakeup allows messages to contain idle characters but requires that the MSB be reserved for use in address frames. The logic 1 MSB of an address frame clears the RWU bit before the stop bit is received and sets the RDRF flag. In this case the character with the MSB set is received even though the receiver was sleeping during most of this character time.

### 14.3.4 Interrupts and Status Flags

The SCI system has three separate interrupt vectors to reduce the amount of software needed to isolate the cause of the interrupt. One interrupt vector is associated with the transmitter for TDRE and TC events. Another interrupt vector is associated with the receiver for RDRF, IDLE, RXEDGIF and LBKDIF events, and a third vector is used for OR, NF, FE, and PF error conditions. Each of these ten interrupt sources can be separately masked by local interrupt enable masks. The flags can still be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The SCI transmitter has two status flags that optionally can generate hardware interrupt requests. Transmit data register empty (TDRE) indicates when there is room in the transmit data buffer to write another transmit character to SCIxD. If the transmit interrupt enable (TIE) bit is set, a hardware interrupt will be requested whenever TDRE = 1. Transmit complete (TC) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TxD at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (TCIE) bit is set, a hardware interrupt will be requested whenever TC = 1.

Table 16-5. TPMxCnSC Field Descriptions (continued)

Field	Description
4 MSnA	Mode select A for TPM channel n. When CPWMS=0 and MSnB=0, MSnA configures TPM channel n for input-capture mode or output compare mode. Refer to Table 16-6 for a summary of channel mode and setup controls. <b>Note:</b> If the associated port pin is not stable for at least two bus clock cycles before changing to input capture mode, it is possible to get an unexpected indication of an edge trigger.
3–2 ELSnB ELSnA	Edge/level select bits. Depending upon the operating mode for the timer channel as set by CPWMS:MSnB:MSnA and shown in Table 16-6, these bits select the polarity of the input edge that triggers an input capture event, select the level that will be driven in response to an output compare match, or select the polarity of the PWM output. Setting ELSnB:ELSnA to 0:0 configures the related timer pin as a general purpose I/O pin not related to any timer functions. This function is typically used to temporarily disable an input capture channel or to make the timer pin available as a general purpose I/O pin when the associated timer channel is set up as a software timer that does not require the use of a pin.

Table 16-6. Mode, Edge, and Level Selection

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	XX	00	Pin not used for TPM - revert to general purpose I/O or other peripheral control	
0	00	01	Input capture	Capture on rising edge only
		10		Capture on falling edge only
		11		Capture on rising or falling edge
	01	01	Output compare	Toggle output on compare
		10		Clear output on compare
		11		Set output on compare
	1X	10	Edge-aligned PWM	High-true pulses (clear output on compare)
		X1		Low-true pulses (set output on compare)
1	XX	10	Center-aligned PWM	High-true pulses (clear output on compare-up)
		X1		Low-true pulses (set output on compare-up)

### 16.3.5 TPM Channel Value Registers (TPMxCnVH:TPMxCnVL)

These read/write registers contain the captured TPM counter value of the input capture function or the output compare value for the output compare or PWM functions. The channel registers are cleared by reset.



BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the frozen TPM counter value.

- This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxSC, TPMxCNTH or TPMxCNTL. Instead, in these conditions the TPM v2 does not clear this read coherency mechanism.
3. Read of TPMxCnVH:L registers (Section 16.3.5, “TPM Channel Value Registers (TPMxCnVH:TPMxCnVL))
- In TPM v3, any read of TPMxCnVH:L registers during BDM mode returns the value of the TPMxCnVH:L register. In TPM v2, if only one byte of the TPMxCnVH:L registers was read before the BDM mode became active, then any read of TPMxCnVH:L registers during BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the value in the TPMxCnVH:L registers.
  - This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxCnSC. Instead, in this condition the TPM v2 does not clear this read coherency mechanism.
4. Write to TPMxCnVH:L registers
- Input Capture Mode (Section 16.4.2.1, “Input Capture Mode)
- In this mode the TPM v3 does not allow the writes to TPMxCnVH:L registers. Instead, the TPM v2 allows these writes.
- Output Compare Mode (Section 16.4.2.2, “Output Compare Mode)
- In this mode and if (CLKSB:CLKSA not = 0:0), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer at the next change of the TPM counter (end of the prescaler counting) after the second byte is written. Instead, the TPM v2 always updates these registers when their second byte is written.
- The following procedure can be used in the TPM v3 to verify if the TPMxCnVH:L registers were updated with the new value that was written to these registers (value in their write buffer).
- ```

...
write the new value to TPMxCnVH:L;
read TPMxCnVH and TPMxCnVL registers;
while (the read value of TPMxCnVH:L is different from the new value written to
TPMxCnVH:L)
begin
    read again TPMxCnVH and TPMxCnVL;
end
...

```
- In this point, the TPMxCnVH:L registers were updated, so the program can continue and, for example, write to TPMxC0SC without cancelling the previous write to TPMxCnVH:L registers.
- Edge-Aligned PWM (Section 16.4.2.3, “Edge-Aligned PWM Mode)
- In this mode and if (CLKSB:CLKSA not = 00), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer after that the both bytes were written and when the

# Appendix A

## Electrical Characteristics

### A.1 Introduction

This section contains the most accurate electrical and timing information for the MC9S08DV60 Series of microcontrollers available at the time of publication.

### A.2 Parameter Classification

The electrical parameters shown in this supplement are guaranteed by various methods. To give the customer a better understanding the following classification is used and the parameters are tagged accordingly in the tables where appropriate:

**Table A-1. Parameter Classifications**

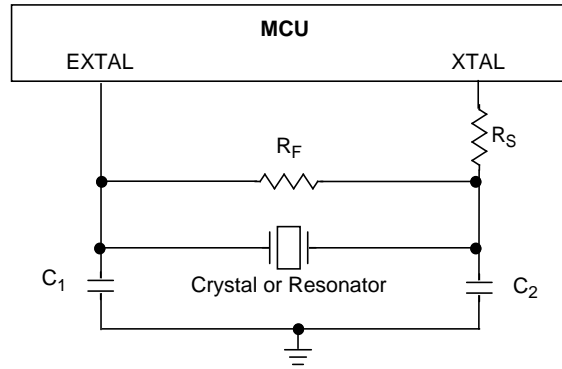
|          |                                                                                                                                                                                                                        |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>P</b> | Those parameters are guaranteed during production testing on each individual device.                                                                                                                                   |
| <b>C</b> | Those parameters are achieved by the design characterization by measuring a statistically relevant sample size across process variations.                                                                              |
| <b>T</b> | Those parameters are achieved by design characterization on a small sample size from typical devices under typical conditions unless otherwise noted. All values shown in the typical column are within this category. |
| <b>D</b> | Those parameters are derived mainly from simulations.                                                                                                                                                                  |

#### NOTE

The classification is shown in the column labeled “C” in the parameter tables where appropriate.

### A.3 Absolute Maximum Ratings

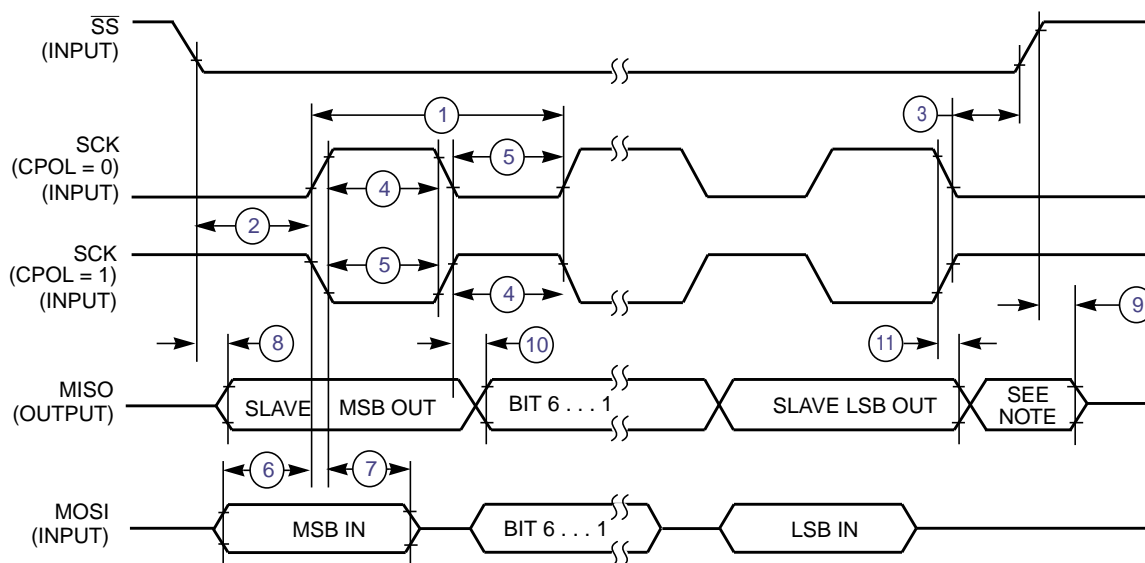
Absolute maximum ratings are stress ratings only, and functional operation at the maxima is not guaranteed. Stress beyond the limits specified in [Table A-2](#) may affect device reliability or cause permanent damage to the device. For functional operating conditions, refer to the remaining tables in this section.



## A.11 MCG Specifications

Table A-12. MCG Frequency Specifications (Temperature Range = –40 to 125°C Ambient)

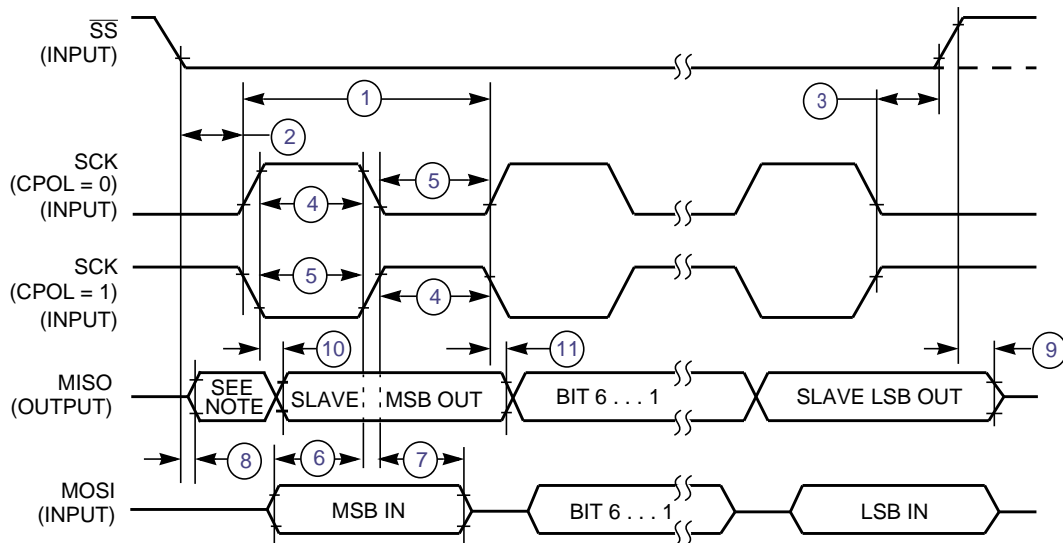
| Num | C | Rating                                                                                                                       | Symbol                   | Min   | Typical            | Max       | Unit        |
|-----|---|------------------------------------------------------------------------------------------------------------------------------|--------------------------|-------|--------------------|-----------|-------------|
| 1   | P | Internal reference frequency - factory trimmed at $V_{DD} = 5\text{ V}$ and temperature = 25 °C                              | $f_{int\_ft}$            | —     | 31.25              | —         | kHz         |
| 2   | P | Average internal reference frequency - untrimmed <sup>1</sup>                                                                | $f_{int\_ut}$            | 25    | 32.7               | 41.66     | kHz         |
| 3   | P | Average internal reference frequency - user trimmed                                                                          | $f_{int\_t}$             | 31.25 | —                  | 39.0625   | kHz         |
| 4   | D | Internal reference startup time                                                                                              | $t_{irefst}$             | —     | 60                 | 100       | us          |
| 5   | — | DCO output frequency range - untrimmed <sup>1</sup><br>value provided for reference: $f_{dco\_ut} = 1024 \times f_{int\_ut}$ | $f_{dco\_ut}$            | 25.6  | 33.48              | 42.66     | MHz         |
| 6   | P | DCO output frequency range - trimmed                                                                                         | $f_{dco\_t}$             | 32    | —                  | 40        | MHz         |
| 7   | C | Resolution of trimmed DCO output frequency at fixed voltage and temperature (using FTRIM)                                    | $\Delta f_{dco\_res\_t}$ | —     | $\pm 0.1$          | $\pm 0.2$ | % $f_{dco}$ |
| 8   | C | Resolution of trimmed DCO output frequency at fixed voltage and temperature (not using FTRIM)                                | $\Delta f_{dco\_res\_t}$ | —     | $\pm 0.2$          | $\pm 0.4$ | % $f_{dco}$ |
| 9   | P | Total deviation of trimmed DCO output frequency over voltage and temperature                                                 | $\Delta f_{dco\_t}$      | —     | + 0.5<br>-1.0      | $\pm 2$   | % $f_{dco}$ |
| 10  | C | Total deviation of trimmed DCO output frequency over fixed voltage and temperature range of 0 - 70 °C                        | $\Delta f_{dco\_t}$      | —     | $\pm 0.5$          | $\pm 1$   | % $f_{dco}$ |
| 11  | C | FLL acquisition time <sup>2</sup>                                                                                            | $t_{fl\_acquire}$        | —     | —                  | 1         | ms          |
| 12  | D | PLL acquisition time <sup>3</sup>                                                                                            | $t_{pll\_acquire}$       | —     | —                  | 1         | ms          |
| 13  | C | Long term Jitter of DCO output clock (averaged over 2ms interval) <sup>4</sup>                                               | $C_{jitter}$             | —     | 0.02               | 0.2       | % $f_{dco}$ |
| 14  | D | VCO operating frequency                                                                                                      | $f_{vco}$                | 7.0   | —                  | 55.0      | MHz         |
| 15  | D | PLL reference frequency range                                                                                                | $f_{pll\_ref}$           | 1.0   | —                  | 2.0       | MHz         |
| 16  | T | RMS frequency variation of a single clock cycle measured 2 ms after reference edge. <sup>5</sup>                             | $f_{pll\_cycjit\_2ms}$   | —     | 0.590 <sup>4</sup> | —         | % $f_{pll}$ |
| 17  | T | Maximum frequency variation averaged over 2 ms window.                                                                       | $f_{pll\_maxjit\_2ms}$   | —     | 0.001              | —         | % $f_{pll}$ |



NOTE:

1. Not defined but normally MSB of character just received

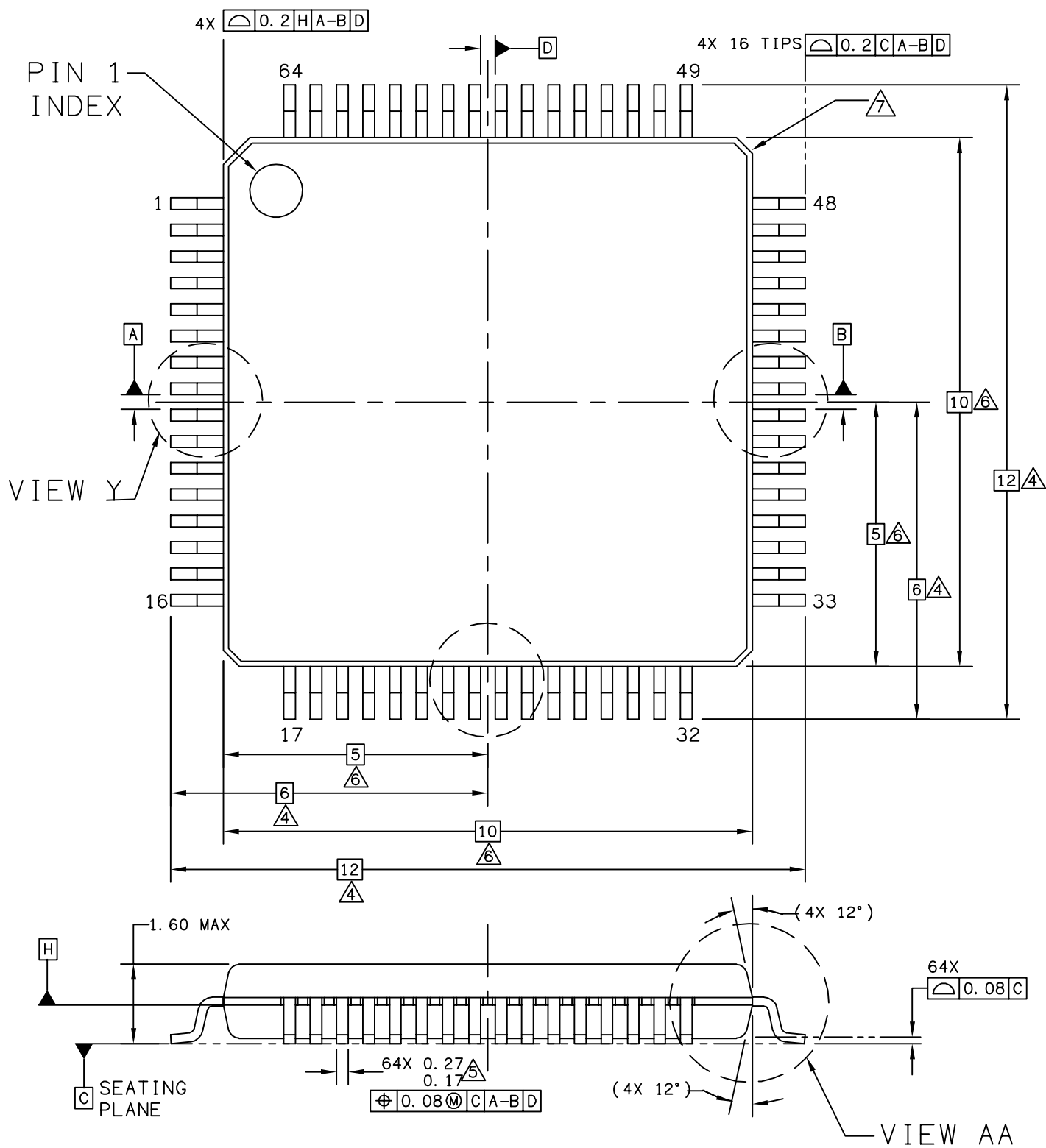
**Figure A-9. SPI Slave Timing (CPHA = 0)**



NOTE:

1. Not defined but normally LSB of character just received

**Figure A-10. SPI Slave Timing (CPHA = 1)**



|                                                                       |  |                    |                            |                            |             |
|-----------------------------------------------------------------------|--|--------------------|----------------------------|----------------------------|-------------|
| © FREESCALE SEMICONDUCTOR, INC.<br>ALL RIGHTS RESERVED.               |  | MECHANICAL OUTLINE |                            | PRINT VERSION NOT TO SCALE |             |
| TITLE:<br>64LD LQFP,<br>10 X 10 X 1.4 PKG,<br>0.5 PITCH, CASE OUTLINE |  |                    | DOCUMENT NO: 98ASS23234W   |                            | REV: E      |
|                                                                       |  |                    | CASE NUMBER: 840F-02       |                            | 11 AUG 2006 |
|                                                                       |  |                    | STANDARD: JEDEC MS-026 BCD |                            |             |