



Welcome to [E-XFL.COM](#)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	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	39
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K 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	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dv16clf">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dv16clf</a>



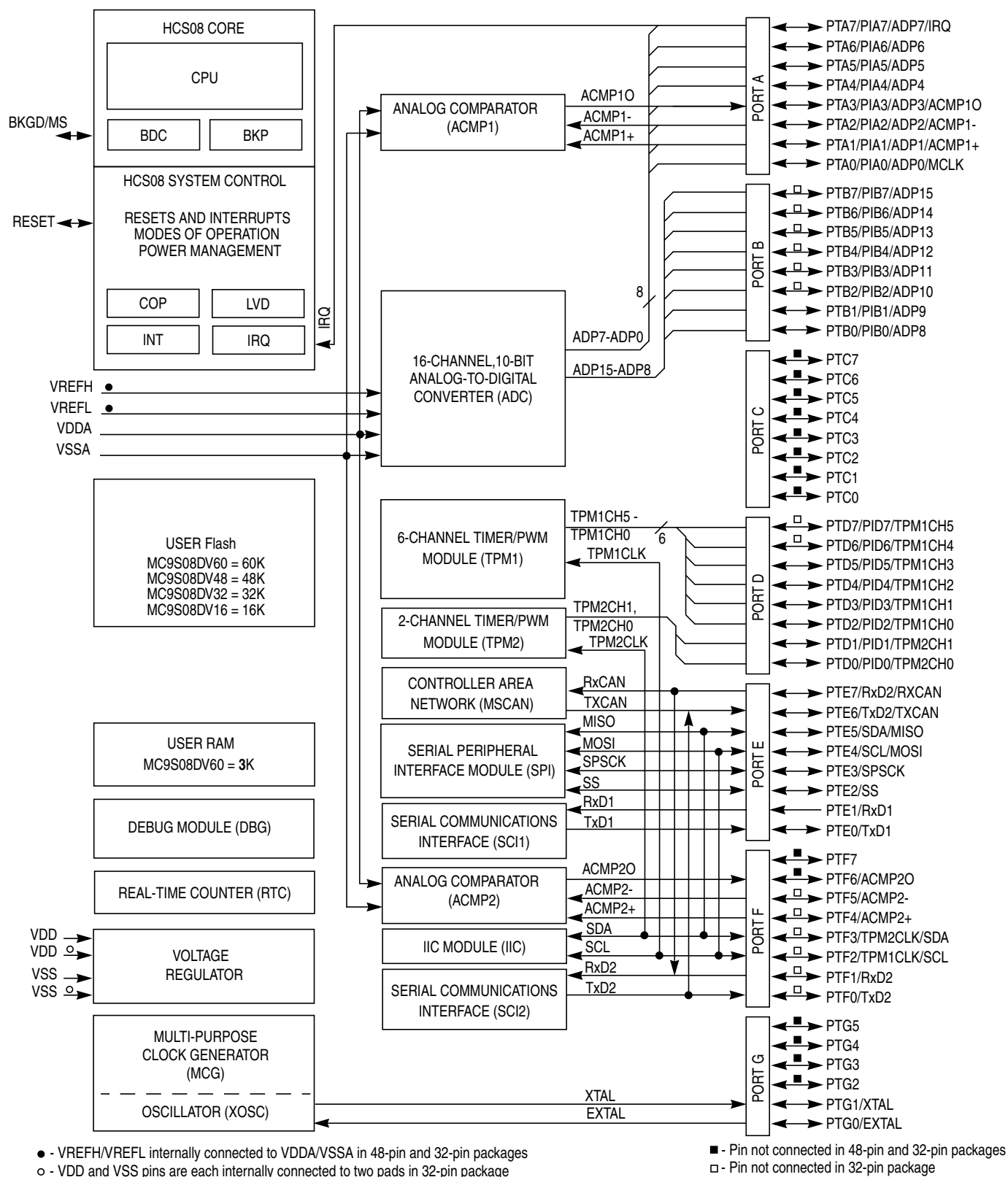


Figure 1-1. MC9S08DV60 Block Diagram

### 4.5.5 Sector Erase Abort

The sector erase abort operation will terminate the active sector erase operation so that other sectors are available for read and program operations without waiting for the sector erase operation to complete.

The sector erase abort command write sequence is as follows:

1. Write to any Flash address to start the command write sequence for the sector erase abort command. The address and data written are ignored.
2. Write the sector erase abort command, 0x47, to the FCMD register.
3. Clear the FCBEF flag in the FSTAT register by writing a 1 to FCBEF to launch the sector erase abort command.

If the sector erase abort command is launched resulting in the early termination of an active sector erase operation, the FACCERR flag will set once the operation completes as indicated by the FCCF flag being set. The FACCERR flag sets to inform the user that the Flash sector may not be fully erased and a new sector erase command must be launched before programming any location in that specific sector.

If the sector erase abort command is launched but the active sector erase operation completes normally, the FACCERR flag will not set upon completion of the operation as indicated by the FCCF flag being set. Therefore, if the FACCERR flag is not set after the sector erase abort command has completed, a sector being erased when the abort command was launched will be fully erased.

A flowchart to execute the sector erase abort operation is shown in [Figure 4-4](#).

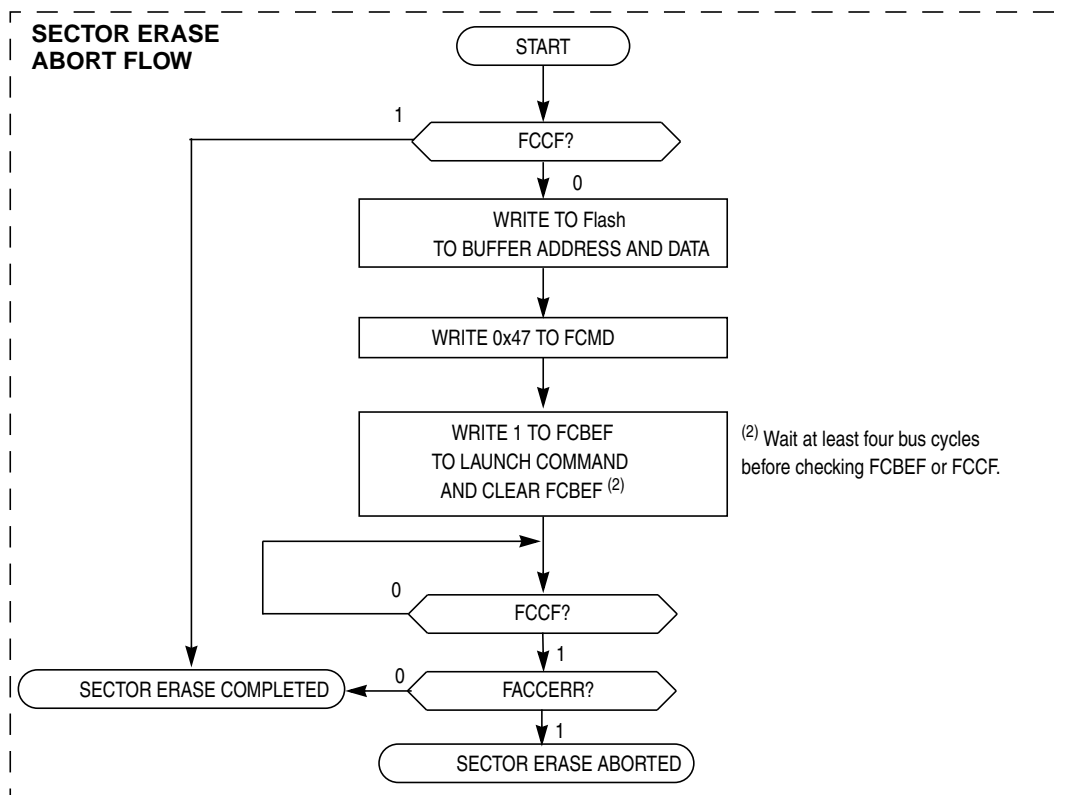


Figure 4-4. Sector Erase Abort Flowchart

Table 4-10. Security States<sup>1</sup>

SEC[1:0]	Description
0:0	secure
0:1	secure
1:0	unsecured
1:1	secure

<sup>1</sup> SEC changes to 1:0 after successful backdoor key entry or a successful blank check of Flash.

### 4.5.10.3 Flash Configuration Register (FCNFG)

	7	6	5	4	3	2	1	0
R	0	Reserved	KEYACC	Reserved <sup>1</sup>	0	0	0	1
W								
Reset	0	0	0	1	0	0	0	1

= Unimplemented or Reserved

Figure 4-7. Flash Configuration Register (FCNFG)

<sup>1</sup> User must write a 1 to this bit. Failing to do so may result in unexpected behavior.

Table 4-11. FCNFG Register Field Descriptions

Field	Description
5 KEYACC	<b>Enable Writing of Access Key</b> — This bit enables writing of the backdoor comparison key. For more detailed information about the backdoor key mechanism, refer to <a href="#">Section 4.5.9, “Security.”</a> 0 Writes to 0xFFB0–0xFFB7 are interpreted as the start of a Flash programming or erase command. 1 Writes to NVBACKKEY (0xFFB0–0xFFB7) are interpreted as comparison key writes.

### 4.5.10.4 Flash Protection Register (FPROT and NVPROT)

The FPROT register defines which Flash and EEPROM sectors are protected against program and erase operations.

During the reset sequence, the FPROT register is loaded from the nonvolatile location NVPROT. To change the protection that will be loaded during the reset sequence, the sector containing NVPROT must be unprotected and erased, then NVPROT can be reprogrammed.

FPROT bits are readable at any time and writable as long as the size of the protected region is being increased. Any write to FPROT that attempts to decrease the size of the protected memory will be ignored.

Trying to alter data in any protected area will result in a protection violation error and the FPVIOL flag will be set in the FSTAT register. Mass erase is not possible if any one of the sectors is protected.

# 6.5.1 Port A Registers

Port A is controlled by the registers listed below.

## 6.5.1.1 Port A Data Register (PTAD)

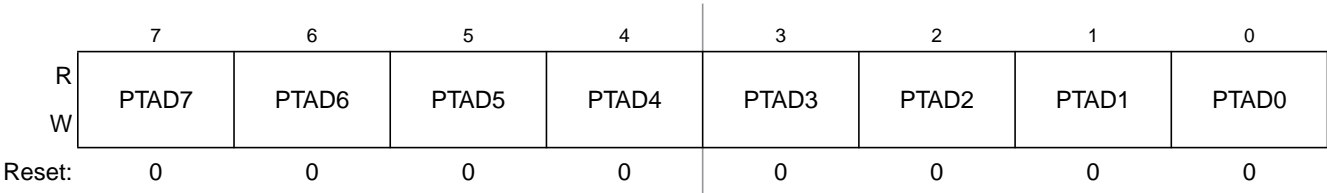


Figure 6-3. Port A Data Register (PTAD)

Table 6-1. PTAD Register Field Descriptions

Field	Description
7:0 PTAD[7:0]	<b>Port A Data Register Bits</b> — For port A pins that are inputs, reads return the logic level on the pin. For port A pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port A pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTAD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pull-ups/pull-downs disabled.

## 6.5.1.2 Port A Data Direction Register (PTADD)

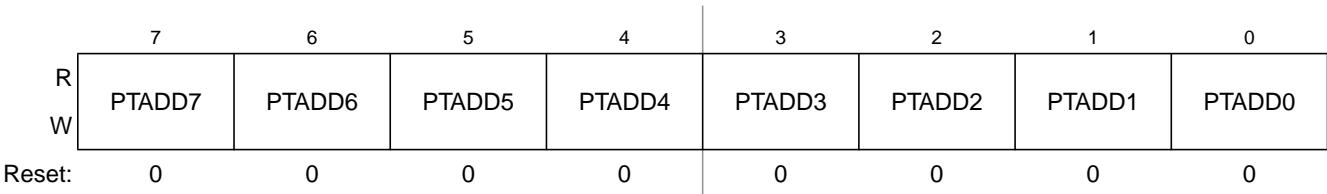


Figure 6-4. Port A Data Direction Register (PTADD)

Table 6-2. PTADD Register Field Descriptions

Field	Description
7:0 PTADD[7:0]	<b>Data Direction for Port A Bits</b> — These read/write bits control the direction of port A pins and what is read for PTAD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port A bit n and PTAD reads return the contents of PTADn.

### 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

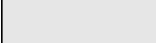
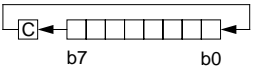
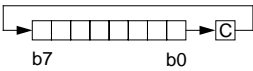
 = Unimplemented or Reserved

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.

Table 7-2. Instruction Set Summary (Sheet 6 of 9)

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V 1 1 H	I N Z C
MOV <i>opr8a,opr8a</i> MOV <i>opr8a,X+</i> MOV <i>#opr8i,opr8a</i> MOV <i>,X+,opr8a</i>	Move $(M)_{\text{destination}} \leftarrow (M)_{\text{source}}$ In IX+/DIR and DIR/IX+ Modes, $H:X \leftarrow (H:X) + \$0001$	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E dd dd 5E dd 6E ii dd 7E dd	5 5 4 5	rwppp rwwpp pwpp rwwpp	0 1 1 -	- $\uparrow$ $\downarrow$ -
MUL	Unsigned multiply $X:A \leftarrow (X) \times (A)$	INH	42	5	ffffp	- 1 1 0	- - - 0
NEG <i>opr8a</i> NEGA NEGX NEG <i>opr8,X</i> NEG <i>,X</i> NEG <i>opr8,SP</i>	Negate $M \leftarrow -(M) = \$00 - (M)$ (Two's Complement) $A \leftarrow -(A) = \$00 - (A)$ $X \leftarrow -(X) = \$00 - (X)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$	DIR INH INH IX1 IX SP1	30 dd 40 50 60 ff 70 9E 60 ff	5 1 1 5 4 6	rwwpp p p rwwpp rwwp prwwpp	$\uparrow$ 1 1 -	- $\uparrow$ $\downarrow$ $\uparrow$ $\downarrow$
NOP	No Operation — Uses 1 Bus Cycle	INH	9D	1	p	- 1 1 -	- - - -
NSA	Nibble Swap Accumulator $A \leftarrow (A[3:0]:A[7:4])$	INH	62	1	p	- 1 1 -	- - - -
ORA <i>#opr8i</i> ORA <i>opr8a</i> ORA <i>opr16a</i> ORA <i>opr8,X</i> ORA <i>opr8,X</i> ORA <i>,X</i> ORA <i>opr8,SP</i> ORA <i>opr8,SP</i>	Inclusive OR Accumulator and Memory $A \leftarrow (A)   (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA ii BA dd CA hh ll DA ee ff EA ff FA 9E DA ee ff 9E EA ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 -	- $\uparrow$ $\downarrow$ -
PSHA	Push Accumulator onto Stack $\text{Push } (A); SP \leftarrow (SP) - \$0001$	INH	87	2	sp	- 1 1 -	- - - -
PSHH	Push H (Index Register High) onto Stack $\text{Push } (H); SP \leftarrow (SP) - \$0001$	INH	8B	2	sp	- 1 1 -	- - - -
PSHX	Push X (Index Register Low) onto Stack $\text{Push } (X); SP \leftarrow (SP) - \$0001$	INH	89	2	sp	- 1 1 -	- - - -
PULA	Pull Accumulator from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (A)$	INH	86	3	ufp	- 1 1 -	- - - -
PULH	Pull H (Index Register High) from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (H)$	INH	8A	3	ufp	- 1 1 -	- - - -
PULX	Pull X (Index Register Low) from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (X)$	INH	88	3	ufp	- 1 1 -	- - - -
ROL <i>opr8a</i> ROLA ROLX ROL <i>opr8,X</i> ROL <i>,X</i> ROL <i>opr8,SP</i>	Rotate Left through Carry 	DIR INH INH IX1 IX SP1	39 dd 49 59 69 ff 79 9E 69 ff	5 1 1 5 4 6	rwwpp p p rwwpp rwwp prwwpp	$\uparrow$ 1 1 -	- $\uparrow$ $\downarrow$ $\uparrow$ $\downarrow$
ROR <i>opr8a</i> RORA RORX ROR <i>opr8,X</i> ROR <i>,X</i> ROR <i>opr8,SP</i>	Rotate Right through Carry 	DIR INH INH IX1 IX SP1	36 dd 46 56 66 ff 76 9E 66 ff	5 1 1 5 4 6	rwwpp p p rwwpp rwwp prwwpp	$\uparrow$ 1 1 -	- $\uparrow$ $\downarrow$ $\uparrow$ $\downarrow$



multiplication factor, as selected by the VDIV bits, times the reference frequency, as selected by the RDIV bits. If BDM is enabled then the MCGLCLK is derived from the DCO (open-loop mode) divided by two. If BDM is not enabled then the FLL is disabled in a low power state.

#### 8.4.1.6 PLL Bypassed External (PBE)

In PLL bypassed external (PBE) mode, the MCGOUT clock is derived from the external reference clock and the PLL is operational but its output clock is not used. This mode is useful to allow the PLL to acquire its target frequency while the MCGOUT clock is driven from the external reference clock.

The PLL bypassed external mode is entered when all the following conditions occur:

- CLKS bits are written to 10
- IREFS bit is written to 0
- PLLS bit is written to 1
- RDIV bits are written to divide reference clock to be within the range of 1 MHz to 2 MHz
- LP bit is written to 0

In PLL bypassed external mode, the MCGOUT clock is derived from the external reference clock. The external reference clock which is enabled can be an external crystal/resonator or it can be another external clock source. The PLL clock frequency locks to a multiplication factor, as selected by the VDIV bits, times the reference frequency, as selected by the RDIV bits. If BDM is enabled then the MCGLCLK is derived from the DCO (open-loop mode) divided by two. If BDM is not enabled then the FLL is disabled in a low power state.

#### 8.4.1.7 Bypassed Low Power Internal (BLPI)

The bypassed low power internal (BLPI) mode is entered when all the following conditions occur:

- CLKS bits are written to 01
- IREFS bit is written to 1
- PLLS bit is written to 0
- LP bit is written to 1
- BDM mode is not active

In bypassed low power internal mode, the MCGOUT clock is derived from the internal reference clock.

The PLL and the FLL are disabled at all times in BLPI mode and the MCGLCLK will not be available for BDC communications. If the BDM becomes active the mode will switch to FLL bypassed internal (FBI) mode.

#### 8.4.1.8 Bypassed Low Power External (BLPE)

The bypassed low power external (BLPE) mode is entered when all the following conditions occur:

- CLKS bits are written to 10
- IREFS bit is written to 0
- PLLS bit is written to 0 or 1

### 10.2.1 Analog Power ( $V_{DDAD}$ )

The ADC analog portion uses  $V_{DDAD}$  as its power connection. In some packages,  $V_{DDAD}$  is connected internally to  $V_{DD}$ . If externally available, connect the  $V_{DDAD}$  pin to the same voltage potential as  $V_{DD}$ . External filtering may be necessary to ensure clean  $V_{DDAD}$  for good results.

### 10.2.2 Analog Ground ( $V_{SSAD}$ )

The ADC analog portion uses  $V_{SSAD}$  as its ground connection. In some packages,  $V_{SSAD}$  is connected internally to  $V_{SS}$ . If externally available, connect the  $V_{SSAD}$  pin to the same voltage potential as  $V_{SS}$ .

### 10.2.3 Voltage Reference High ( $V_{REFH}$ )

$V_{REFH}$  is the high reference voltage for the converter. In some packages,  $V_{REFH}$  is connected internally to  $V_{DDAD}$ . If externally available,  $V_{REFH}$  may be connected to the same potential as  $V_{DDAD}$  or may be driven by an external source between the minimum  $V_{DDAD}$  spec and the  $V_{DDAD}$  potential ( $V_{REFH}$  must never exceed  $V_{DDAD}$ ).

### 10.2.4 Voltage Reference Low ( $V_{REFL}$ )

$V_{REFL}$  is the low-reference voltage for the converter. In some packages,  $V_{REFL}$  is connected internally to  $V_{SSAD}$ . If externally available, connect the  $V_{REFL}$  pin to the same voltage potential as  $V_{SSAD}$ .

### 10.2.5 Analog Channel Inputs (ADx)

The ADC module supports up to 28 separate analog inputs. An input is selected for conversion through the ADCH channel select bits.

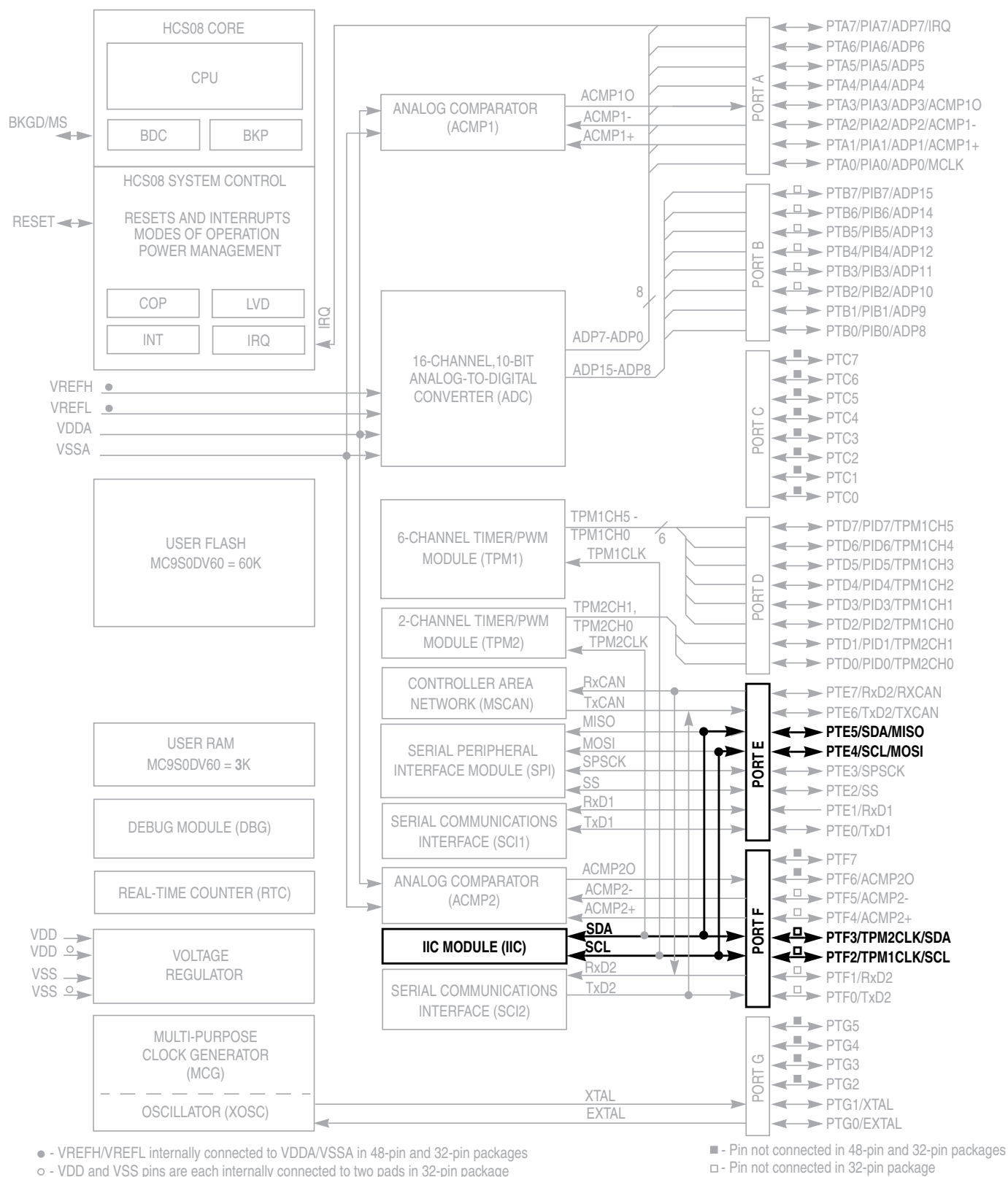
## 10.3 Register Definition

These memory-mapped registers control and monitor operation of the ADC:

- Status and control register, ADCSC1
- Status and control register, ADCSC2
- Data result registers, ADCRH and ADCRL
- Compare value registers, ADCCVH and ADCCVL
- Configuration register, ADCCFG
- Pin control registers, APCTL1, APCTL2, APCTL3

### 10.3.1 Status and Control Register 1 (ADCSC1)

This section describes the function of the ADC status and control register (ADCSC1). Writing ADCSC1 aborts the current conversion and initiates a new conversion (if the ADCH bits are equal to a value other than all 1s).



**Figure 11-1. MC9S08DV60 Block Diagram**

### 11.3.5 IIC Data I/O Register (IICD)

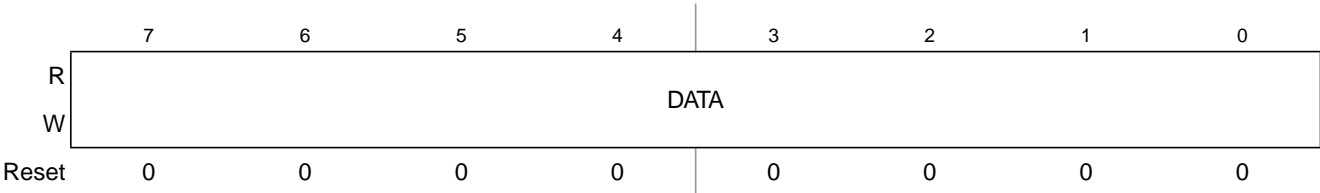


Figure 11-7. IIC Data I/O Register (IICD)

Table 11-7. IICD Field Descriptions

Field	Description
7–0 DATA	<b>Data</b> — In master transmit mode, when data is written to the IICD, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates receiving of the next byte of data.

#### NOTE

When transitioning out of master receive mode, the IIC mode should be switched before reading the IICD register to prevent an inadvertent initiation of a master receive data transfer.

In slave mode, the same functions are available after an address match has occurred.

The TX bit in IICC must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For instance, if the IIC is configured for master transmit but a master receive is desired, reading the IICD does not initiate the receive.

Reading the IICD returns the last byte received while the IIC is configured in master receive or slave receive modes. The IICD does not reflect every byte transmitted on the IIC bus, nor can software verify that a byte has been written to the IICD correctly by reading it back.

In master transmit mode, the first byte of data written to IICD following assertion of MST is used for the address transfer and should comprise of the calling address (in bit 7 to bit 1) concatenated with the required  $R/\overline{W}$  bit (in position bit 0).

### 11.3.6 IIC Control Register 2 (IICC2)

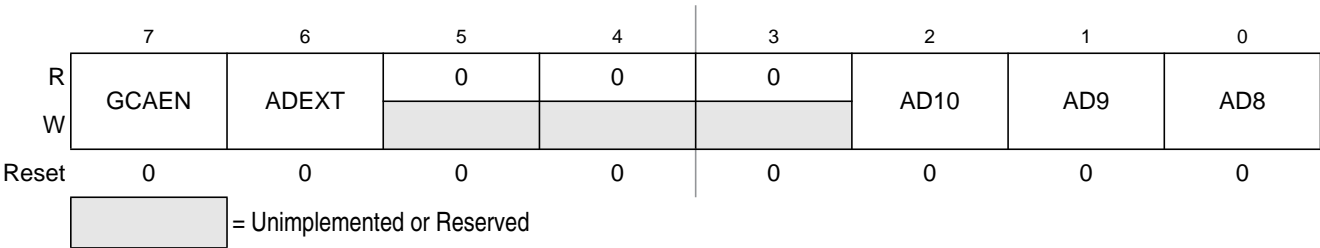


Figure 11-8. IIC Control Register (IICC2)

# 11.7 Initialization/Application Information

## Module Initialization (Slave)

1. Write: IICC2
  - to enable or disable general call
  - to select 10-bit or 7-bit addressing mode
2. Write: IICA
  - to set the slave address
3. Write: IICC1
  - to enable IIC and interrupts
4. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
5. Initialize RAM variables used to achieve the routine shown in [Figure 11-12](#)

## Module Initialization (Master)

1. Write: IICF
  - to set the IIC baud rate (example provided in this chapter)
2. Write: IICC1
  - to enable IIC and interrupts
3. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
4. Initialize RAM variables used to achieve the routine shown in [Figure 11-12](#)
5. Write: IICC1
  - to enable TX
6. Write: IICC1
  - to enable MST (master mode)
7. Write: IICD
  - with the address of the target slave. (The lsb of this byte determines whether the communication is master receive or transmit.)

## Module Use

The routine shown in [Figure 11-12](#) can handle both master and slave IIC operations. For slave operation, an incoming IIC message that contains the proper address begins IIC communication. For master operation, communication must be initiated by writing to the IICD register.

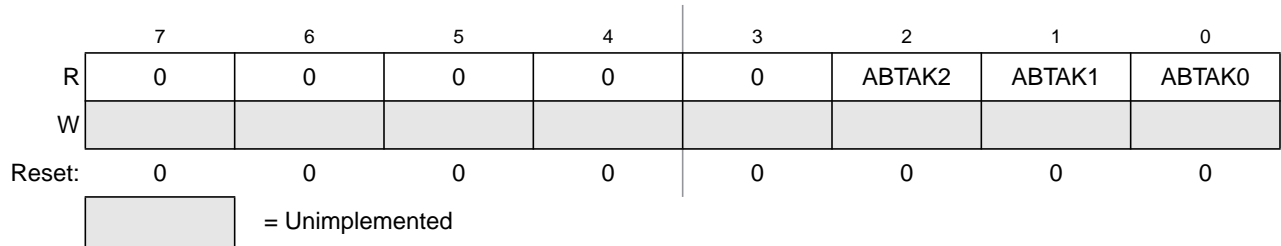
## Register Model

IICA	AD[7:1]							0
When addressed as a slave (in slave mode), the module responds to this address								
IICF	MULT		ICR					
Baud rate = BUSCLK / (2 x MULT x (SCL DIVIDER))								
IICC1	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
Module configuration								
IICS	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
Module status flags								
IICD	DATA							
Data register; Write to transmit IIC data read to read IIC data								
IICC2	GCAEN	ADEXT	0	0	0	AD10	AD9	AD8
Address configuration								

Figure 11-11. IIC Module Quick Start

### 12.3.9 MSCAN Transmitter Message Abort Acknowledge Register (CANTAACK)

The CANTAACK register indicates the successful abort of messages queued for transmission, if requested by the appropriate bits in the CANTARQ register.



**Figure 12-13. MSCAN Transmitter Message Abort Acknowledge Register (CANTAACK)**

#### NOTE

The CANTAACK register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1).

Read: Anytime

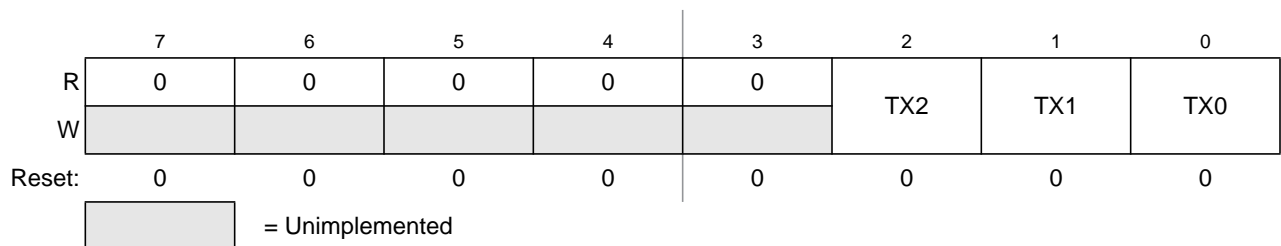
Write: Unimplemented for ABTAKx flags

**Table 12-14. CANTAACK Register Field Descriptions**

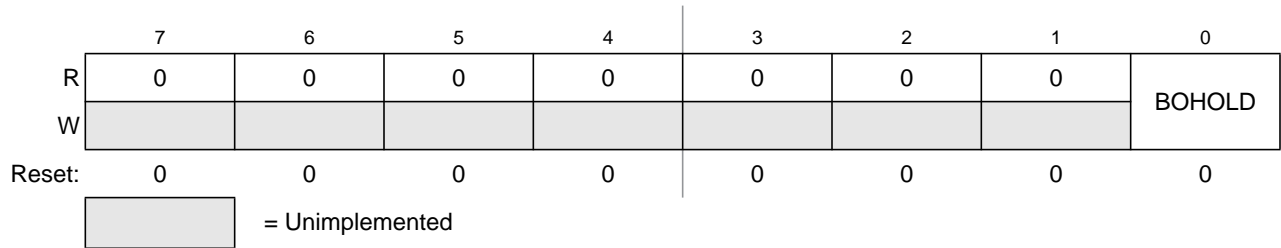
Field	Description
2:0 ABTAK[2:0]	<b>Abort Acknowledge</b> — This flag acknowledges that a message was aborted due to a pending transmission abort request from the CPU. After a particular message buffer is flagged empty, this flag can be used by the application software to identify whether the message was aborted successfully or was sent anyway. The ABTAKx flag is cleared whenever the corresponding TXE flag is cleared. 0 The message was not aborted. 1 The message was aborted.

### 12.3.10 MSCAN Transmit Buffer Selection Register (CANTBSEL)

The CANTBSEL selections of the actual transmit message buffer, which is accessible in the CANTXFG register space.



**Figure 12-14. MSCAN Transmit Buffer Selection Register (CANTBSEL)**



**Figure 12-16. MSCAN Miscellaneous Register (CANMISC)**

Read: Anytime  
 Write: Anytime; write of '1' clears flag; write of '0' ignored

**Table 12-19. CANMISC Register Field Descriptions**

Field	Description
0 BOHOLD	<b>Bus-off State Hold Until User Request</b> — If BORM is set in <a href="#">Section 12.3.2, “MSCAN Control Register 1 (CANCTL1)”</a> , this bit indicates whether the module has entered the bus-off state. Clearing this bit requests the recovery from bus-off. Refer to <a href="#">Section 12.6.2, “Bus-Off Recovery,”</a> for details. 0 Module is not bus-off or recovery has been requested by user in bus-off state 1 Module is bus-off and holds this state until user request

### 12.3.13 MSCAN Receive Error Counter (CANRXERR)

This register reflects the status of the MSCAN receive error counter.



**Figure 12-17. MSCAN Receive Error Counter (CANRXERR)**

Read: Only when in sleep mode (SLPRQ = 1 and SLPK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)

Write: Unimplemented

#### NOTE

Reading this register when in any other mode other than sleep or initialization mode may return an incorrect value. For MCUs with dual CPUs, this may result in a CPU fault condition.

Writing to this register when in special modes can alter the MSCAN functionality.

bit position in the filter register. Finally, registers CANIDAR0/1/2/3 determine the value of those bits determined by CANIDMR0/1/2/3.

For instance in the case of the filter value of:

0001x1001x0

The CANIDMR0/1/2/3 register would be configured as:

00001000010

and so all message identifier bits except bit 1 and bit 6 would be compared against the CANIDAR0/1/2/3 registers. These would be configured as:

00010100100

In this case bits 1 and 6 are set to '0', but since they are ignored it is equally valid to set them to '1'.

### 12.5.3.1 Identifier Acceptance Filters example

As described above, filters work by comparisons to individual bits in the CAN message identifier field. The filter will check each one of the eleven bits of a standard CAN message identifier. Suppose a filter value of 0001x1001x0. In this simple example, there are only three possible CAN messages.

Filter value: 0001x1001x0

Message 1: 00011100110

Message 2: 00110100110

Message 3: 00010100100

Message 2 will be rejected since its third most significant bit is not '0' - 001. The filter is simply a convenient way of defining the set of messages that the CPU must receive. For full 29-bits of an extended CAN message identifier, the filter identifies two sets of messages: one set that it receives and one set that it rejects. Alternatively, the filter may be split into two. This allows the MSCAN to examine only the first 16 bits of a message identifier, but allows two separate filters to perform the checking. See the example below:

Filter value A: 0001x1001x0

Filter value B: 00x101x01x0

Message 1: 00011100110

Message 2: 00110100110

Message 3: 00010100100

MSCAN will accept all three messages. Filter A will accept messages 1 and 3 as before and filter B will accept message 2. In practice, it is unimportant which filter accepts the message - messages accepted by either will be placed in the input buffer. A message may be accepted by more than one filter.

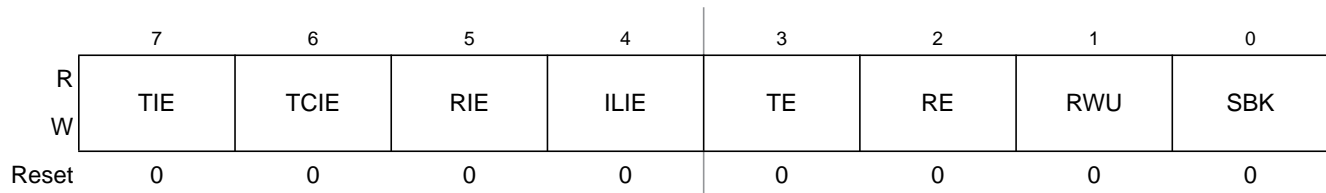


**Table 14-4. SC1xC1 Field Descriptions (continued)**

Field	Description
3 WAKE	<b>Receiver Wakeup Method Select</b> — Refer to <a href="#">Section 14.3.3.2, “Receiver Wakeup Operation”</a> for more information. 0 Idle-line wakeup. 1 Address-mark wakeup.
2 ILT	<b>Idle Line Type Select</b> — Setting this bit to 1 ensures that the stop bit and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of logic high level needed by the idle line detection logic. Refer to <a href="#">Section 14.3.3.2.1, “Idle-Line Wakeup”</a> for more information. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	<b>Parity Enable</b> — Enables hardware parity generation and checking. When parity is enabled, the most significant bit (MSB) of the data character (eighth or ninth data bit) is treated as the parity bit. 0 No hardware parity generation or checking. 1 Parity enabled.
0 PT	<b>Parity Type</b> — Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even. 0 Even parity. 1 Odd parity.

### 14.2.3 SCI Control Register 2 (SC1xC2)

This register can be read or written at any time.



**Figure 14-7. SCI Control Register 2 (SC1xC2)**

**Table 14-5. SC1xC2 Field Descriptions**

Field	Description
7 TIE	<b>Transmit Interrupt Enable (for TDRE)</b> 0 Hardware interrupts from TDRE disabled (use polling). 1 Hardware interrupt requested when TDRE flag is 1.
6 TCIE	<b>Transmission Complete Interrupt Enable (for TC)</b> 0 Hardware interrupts from TC disabled (use polling). 1 Hardware interrupt requested when TC flag is 1.
5 RIE	<b>Receiver Interrupt Enable (for RDRF)</b> 0 Hardware interrupts from RDRF disabled (use polling). 1 Hardware interrupt requested when RDRF flag is 1.
4 ILIE	<b>Idle Line Interrupt Enable (for IDLE)</b> 0 Hardware interrupts from IDLE disabled (use polling). 1 Hardware interrupt requested when IDLE flag is 1.

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.

TPM counter changes from (TPMxMODH:L - 1) to (TPMxMODH:L). If the TPM counter is a free-running counter, then this update is made when the TPM counter changes from \$FFFE to \$FFFF. Instead, the TPM v2 makes this update after that the both bytes were written and when the TPM counter changes from TPMxMODH:L to \$0000.

— Center-Aligned PWM (Section 16.4.2.4, “Center-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 TPM counter changes from (TPMxMODH:L - 1) to (TPMxMODH:L). If the TPM counter is a free-running counter, then this update is made when the TPM counter changes from \$FFFE to \$FFFF. Instead, the TPM v2 makes this update after that the both bytes were written and when the TPM counter changes from TPMxMODH:L to (TPMxMODH:L - 1).

5. Center-Aligned PWM (Section 16.4.2.4, “Center-Aligned PWM Mode)

— TPMxCnVH:L = TPMxMODH:L [SE110-TPM case 1]

In this case, the TPM v3 produces 100% duty cycle. Instead, the TPM v2 produces 0% duty cycle.

— TPMxCnVH:L = (TPMxMODH:L - 1) [SE110-TPM case 2]

In this case, the TPM v3 produces almost 100% duty cycle. Instead, the TPM v2 produces 0% duty cycle.

— TPMxCnVH:L is changed from 0x0000 to a non-zero value [SE110-TPM case 3 and 5]

In this case, the TPM v3 waits for the start of a new PWM period to begin using the new duty cycle setting. Instead, the TPM v2 changes the channel output at the middle of the current PWM period (when the count reaches 0x0000).

— TPMxCnVH:L is changed from a non-zero value to 0x0000 [SE110-TPM case 4]

In this case, the TPM v3 finishes the current PWM period using the old duty cycle setting. Instead, the TPM v2 finishes the current PWM period using the new duty cycle setting.

6. Write to TPMxMODH:L registers in BDM mode (Section 16.3.3, “TPM Counter Modulo Registers (TPMxMODH:TPMxMODL))

In the TPM v3 a write to TPMxSC register in BDM mode clears the write coherency mechanism of TPMxMODH:L registers. Instead, in the TPM v2 this coherency mechanism is not cleared when there is a write to TPMxSC register.

7. Update of EPWM signal when CLKSB:CLKSA = 00

In the TPM v3 if CLKSB:CLKSA = 00, then the EPWM signal in the channel output is not update (it is frozen while CLKSB:CLKSA = 00). Instead, in the TPM v2 the EPWM signal is updated at the next rising edge of bus clock after a write to TPMxCnSC register.

The Figure 0-1 and Figure 0-2 show when the EPWM signals generated by TPM v2 and TPM v3 after the reset (CLKSB:CLKSA = 00) and if there is a write to TPMxCnSC register.

Table A-7. Supply Current Characteristics (continued)

Num	C	Parameter	Symbol	V <sub>DD</sub> (V)	Typical <sup>1</sup>	Max <sup>2</sup>	Unit
6	C	RTC adder to stop2 or stop3 <sup>5</sup> , 25°C		5	300	—	nA
				3	300	—	nA
7	C	LVD adder to stop3 (LVDE = LVDSE = 1)		5	110	—	μA
				3	90	—	μA
8	C	Adder to stop3 for oscillator enabled <sup>6</sup> (IRCLKEN = 1 and IREFSTEN = 1 or ERCLKEN = 1 and EREFSTEN = 1)		5	5	—	μA
				3	5	—	μA

<sup>1</sup> Typicals are measured at 25°C, unless otherwise noted.

<sup>2</sup> Maximum values in this column apply for the full operating temperature range of the device unless otherwise noted.

<sup>3</sup> All modules except ADC active, MCG configured for FBE, and does not include any dc loads on port pins

<sup>4</sup> Stop currents are tested in production for 25°C on all parts. Tests at other temperatures depend upon the part number suffix and maturity of the product. Freescale may eliminate a test insertion at a particular temperature from the production test flow once sufficient data has been collected and is approved.

<sup>5</sup> Most customers are expected to find that auto-wakeup from stop2 or stop3 can be used instead of the higher current wait mode.

<sup>6</sup> Values given under the following conditions: low range operation (RANGE = 0), low power mode (HGO = 0).

## A.8 Analog Comparator (ACMP) Electricals

Table A-8. Analog Comparator Electrical Specifications

Num	C	Rating	Symbol	Min	Typical	Max	Unit
1	—	Supply voltage	V <sub>DD</sub>	2.7	—	5.5	V
2	D	Supply current (active)	I <sub>DDAC</sub>	—	20	35	μA
3	D	Analog input voltage	V <sub>AIN</sub>	V <sub>SS</sub> – 0.3	—	V <sub>DD</sub>	V
4	D	Analog input offset voltage	V <sub>AIO</sub>		20	40	mV
5	D	Analog Comparator hysteresis	V <sub>H</sub>	3.0	6.0	20.0	mV
6	D	Analog input leakage current	I <sub>ALKG</sub>	--	--	1.0	μA
7	D	Analog Comparator initialization delay	t <sub>AINIT</sub>	—	—	1.0	μs

## A.9 ADC Characteristics

Table A-9. 12-bit ADC Operating Conditions

Characteristic	Conditions	Symb	Min	Typ <sup>1</sup>	Max	Unit	Comment
Supply voltage	Absolute	V <sub>DDAD</sub>	2.7	—	5.5	V	
	Delta to V <sub>DD</sub> (V <sub>DD</sub> –V <sub>DDAD</sub> ) <sup>2</sup>	ΔV <sub>DDAD</sub>	-100	0	+100	mV	
Ground voltage	Delta to V <sub>SS</sub> (V <sub>SS</sub> –V <sub>SSAD</sub> ) <sup>2</sup>	ΔV <sub>SSAD</sub>	-100	0	+100	mV	



NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
3. DATUMS A, B AND D TO BE DETERMINED AT DATUM PLANE H.

△4. DIMENSIONS TO BE DETERMINED AT SEATING PLANE C.

△5. THIS DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE UPPER LIMIT BY MORE THAN 0.08 mm AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD SHALL NOT BE LESS THAN 0.07 mm.

△6. THIS DIMENSION DOES NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. THIS DIMENSION IS MAXIMUM PLASTIC BODY SIZE DIMENSION INCLUDING MOLD MISMATCH.

△7. EXACT SHAPE OF EACH CORNER IS OPTIONAL.

△8. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.1 mm AND 0.25 mm FROM THE LEAD TIP.

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