

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	48KB (48K 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/mc9s08dv48clc">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dv48clc</a>

### NOTE

The FCBEF flag will not set after launching the sector erase abort command. If an attempt is made to start a new command write sequence with a sector erase abort operation active, the FACCERR flag in the FSTAT register will be set. A new command write sequence may be started after clearing the ACCERR flag, if set.

### NOTE

The sector erase abort command should be used sparingly since a sector erase operation that is aborted counts as a complete program/erase cycle.

## 4.5.6 Access Errors

An access error occurs whenever the command execution protocol is violated.

Any of the following specific actions will cause the access error flag (FACCERR) in FSTAT to be set. FACCERR must be cleared by writing a 1 to FACCERR in FSTAT before any command can be processed.

- Writing to a Flash address before the internal Flash clock frequency has been set by writing to the FCDIV register.
- Writing to a Flash address while FCBEF is not set. (A new command cannot be started until the command buffer is empty.)
- Writing a second time to a Flash address before launching the previous command. (There is only one write to Flash for every command.)
- Writing a second time to FCMD before launching the previous command. (There is only one write to FCMD for every command.)
- Writing to any Flash control register other than FCMD after writing to a Flash address.
- Writing any command code other than the six allowed codes (0x05, 0x20, 0x25, 0x40, 0x41, or 0x47) to FCMD.
- Writing any Flash control register other than to write to FSTAT (to clear FCBEF and launch the command) after writing the command to FCMD.
- The MCU enters stop mode while a program or erase command is in progress. (The command is aborted.)
- Writing the byte program, burst program, sector erase or sector erase abort command code (0x20, 0x25, 0x40, or 0x47) with a background debug command while the MCU is secured. (The background debug controller can do blank check and mass erase commands only when the MCU is secure.)
- Writing 0 to FCBEF to cancel a partial command.

## 5.8.8 System Power Management Status and Control 2 Register (SPMSC2)

This register is used to report the status of the low-voltage warning function, and to configure the stop mode behavior of the MCU. This register should be written during the user's reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

	7	6	5	4	3	2	1	0
R	0	0	LVDV <sup>1</sup>	LVWV	PPDF	0	0	PPDC <sup>2</sup>
W						PPDACK		
Power-on Reset:	0	0	0	0	0	0	0	0
LVD Reset:	0	0	u	u	0	0	0	0
Any other Reset:	0	0	u	u	0	0	0	0

= Unimplemented or Reserved
 u = Unaffected by reset

<sup>1</sup> This bit can be written only one time after power-on reset. Additional writes are ignored.

<sup>2</sup> This bit can be written only one time after reset. Additional writes are ignored.

**Figure 5-10. System Power Management Status and Control 2 Register (SPMSC2)**

**Table 5-11. SPMSC2 Register Field Descriptions**

Field	Description
5 LVDV	<b>Low-Voltage Detect Voltage Select</b> — This write-once bit selects the low-voltage detect (LVD) trip point setting. It also selects the warning voltage range. See <a href="#">Table 5-12</a> .
4 LVWV	<b>Low-Voltage Warning Voltage Select</b> — This bit selects the low-voltage warning (LVW) trip point voltage. See <a href="#">Table 5-12</a> .
3 PPDF	<b>Partial Power Down Flag</b> — This read-only status bit indicates that the MCU has recovered from stop2 mode. 0 MCU has not recovered from stop2 mode. 1 MCU recovered from stop2 mode.
2 PPDACK	<b>Partial Power Down Acknowledge</b> — Writing a 1 to PPDACK clears the PPDF bit.
0 PPDC	<b>Partial Power Down Control</b> — This write-once bit controls whether stop2 or stop3 mode is selected. 0 Stop3 mode enabled. 1 Stop2, partial power down, mode enabled.

**Table 5-12. LVD and LVW Trip Point Typical Values<sup>1</sup>**

LVDV:LVWV	LVW Trip Point	LVD Trip Point
0:0	$V_{LVW0} = 2.74 \text{ V}$	$V_{LVD0} = 2.56 \text{ V}$
0:1	$V_{LVW1} = 2.92 \text{ V}$	
1:0	$V_{LVW2} = 4.3 \text{ V}$	$V_{LVD1} = 4.0 \text{ V}$
1:1	$V_{LVW3} = 4.6 \text{ V}$	

<sup>1</sup> See Electrical Characteristics appendix for minimum and maximum values.

### 6.5.1.7 Port A Interrupt Pin Select Register (PTAPS)

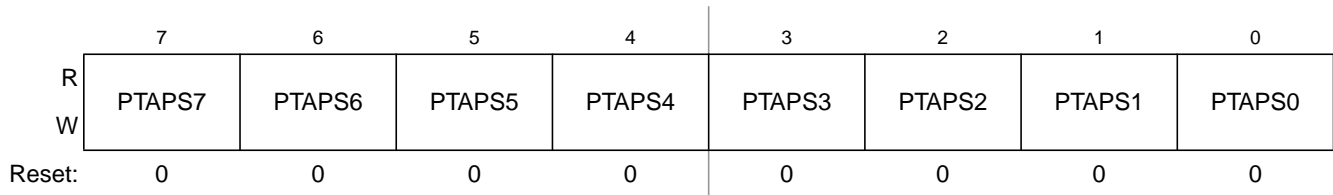


Figure 6-9. Port A Interrupt Pin Select Register (PTAPS)

Table 6-7. PTAPS Register Field Descriptions

Field	Description
7:0 PTAPS[7:0]	<b>Port A Interrupt Pin Selects</b> — Each of the PTAPSn bits enable the corresponding port A interrupt pin. 0 Pin not enabled as interrupt. 1 Pin enabled as interrupt.

### 6.5.1.8 Port A Interrupt Edge Select Register (PTAES)

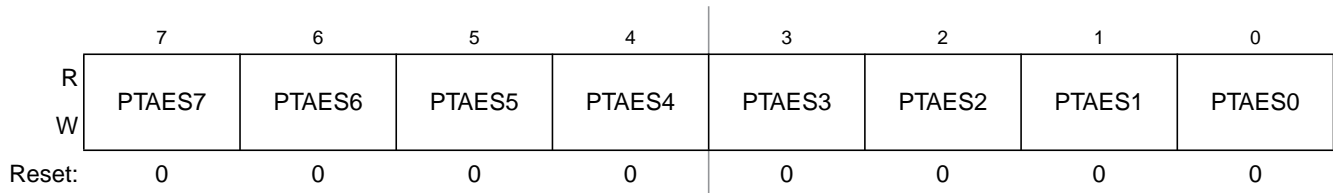


Figure 6-10. Port A Edge Select Register (PTAES)

Table 6-8. PTAES Register Field Descriptions

Field	Description
7:0 PTAES[7:0]	<b>Port A Edge Selects</b> — Each of the PTAESn bits serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 0 A pull-up device is connected to the associated pin and detects falling edge/low level for interrupt generation. 1 A pull-down device is connected to the associated pin and detects rising edge/high level for interrupt generation.

### 6.5.4.7 Port D Interrupt Pin Select Register (PTDPS)

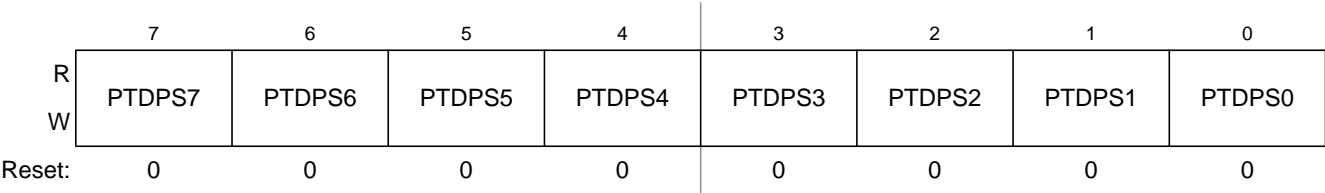


Figure 6-30. Port D Interrupt Pin Select Register (PTDPS)

Table 6-28. PTDPS Register Field Descriptions

Field	Description
7:0 PTDPS[7:0]	<b>Port D Interrupt Pin Selects</b> — Each of the PTDPSn bits enable the corresponding port D interrupt pin. 0 Pin not enabled as interrupt. 1 Pin enabled as interrupt.

### 6.5.4.8 Port D Interrupt Edge Select Register (PTDES)

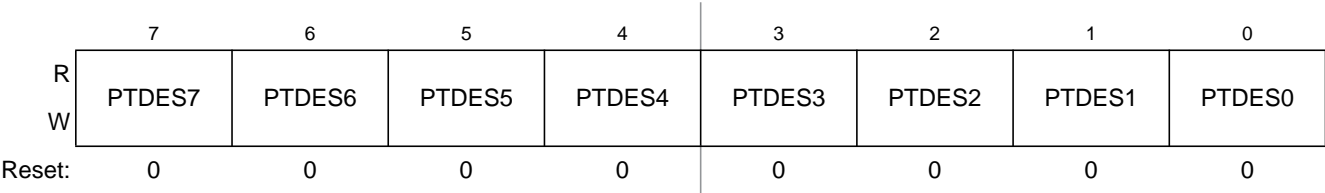


Figure 6-31. Port D Edge Select Register (PTDES)

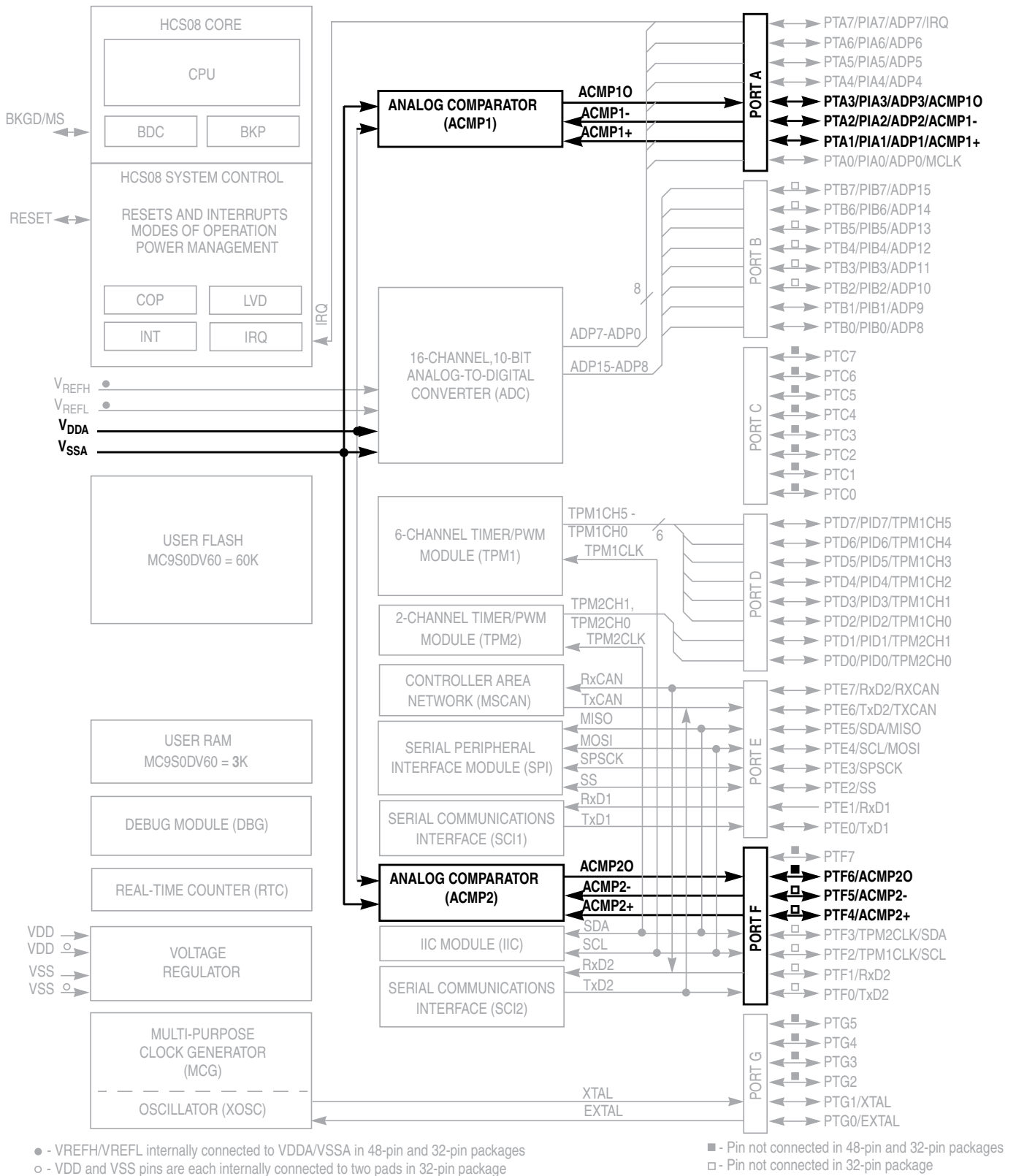
Table 6-29. PTDES Register Field Descriptions

Field	Description
7:0 PTDES[7:0]	<b>Port D Edge Selects</b> — Each of the PTDESn bits serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 0 A pull-up device is connected to the associated pin and detects falling edge/low level for interrupt generation. 1 A pull-down device is connected to the associated pin and detects rising edge/high level for interrupt generation.

### 8.5.2.3 Example #3: Moving from BLPI to FEE Mode: External Crystal = 4 MHz, Bus Frequency = 16 MHz

In this example, the MCG will move through the proper operational modes from BLPI mode at a 16 kHz bus frequency running off of the internal reference clock (see previous example) to FEE mode using a 4 MHz crystal configured for a 16 MHz bus frequency. First, the code sequence will be described. Then a flowchart will be included which illustrates the sequence.

1. First, BLPI must transition to FBI mode.
  - a) MCGC2 = 0x00 (%00000000)
    - LP (bit 3) in MCGSC is 0
  - b) Optionally, loop until LOCK (bit 6) in the MCGSC is set, indicating that the FLL has acquired lock. Although the FLL is bypassed in FBI mode, it is still enabled and running.
2. Next, FBI will transition to FEE mode.
  - a) MCGC2 = 0x36 (%00110110)
    - RANGE (bit 5) set to 1 because the frequency of 4 MHz is within the high frequency range
    - HGO (bit 4) set to 1 to configure external oscillator for high gain operation
    - EREFS (bit 2) set to 1, because a crystal is being used
    - ERCLKEN (bit 1) set to 1 to ensure the external reference clock is active
  - b) Loop until OSCINIT (bit 1) in MCGSC is 1, indicating the crystal selected by the EREFS bit has been initialized.
  - c) MCGC1 = 0x38 (%00111000)
    - CLKS (bits 7 and 6) set to %00 in order to select the output of the FLL as system clock source
    - RDIV (bits 5-3) set to %111, or divide-by-128 because  $4 \text{ MHz} / 128 = 31.25 \text{ kHz}$  which is in the 31.25 kHz to 39.0625 kHz range required by the FLL
    - IREFS (bit 1) cleared to 0, selecting the external reference clock
  - d) Loop until IREFST (bit 4) in MCGSC is 0, indicating the external reference clock is the current source for the reference clock
  - e) Optionally, loop until LOCK (bit 6) in the MCGSC is set, indicating that the FLL has reacquired lock.
  - f) Loop until CLKST (bits 3 and 2) in MCGSC are %00, indicating that the output of the FLL is selected to feed MCGOUT



**Figure 9-1. MC9S08DV60 Block Diagram**

### 9.1.4 Block Diagram

The block diagram for the analog comparator module is shown [Figure 9-2](#).

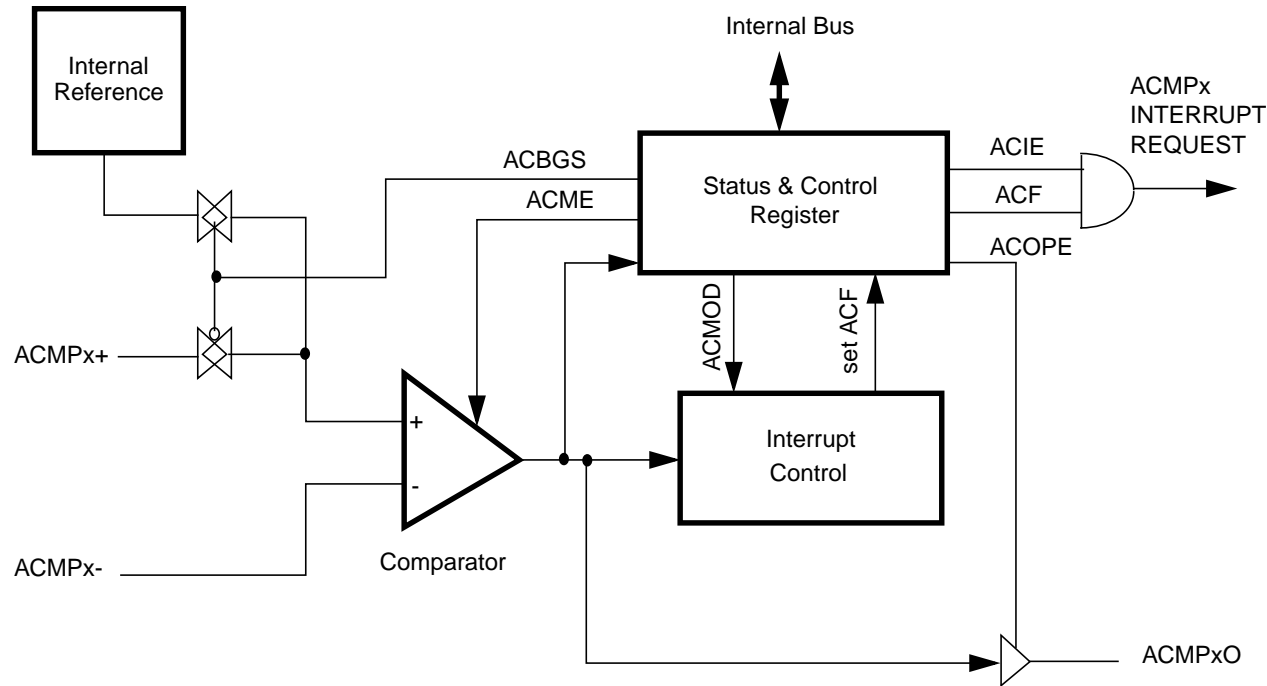


Figure 9-2. Analog Comparator (ACMP) Block Diagram

## 9.2 External Signal Description

The ACMP has two analog input pins, ACMPx+ and ACMPx– and one digital output pin ACMPxO. Each of these pins can accept an input voltage that varies across the full operating voltage range of the MCU. As shown in [Figure 9-2](#), the ACMPx– pin is connected to the inverting input of the comparator, and the ACMPx+ pin is connected to the comparator non-inverting input if ACBGS is a 0. As shown in [Figure 9-2](#), the ACMPxO pin can be enabled to drive an external pin.

The signal properties of ACMP are shown in [Table 9-1](#).

Table 9-1. Signal Properties

Signal	Function	I/O
ACMPx–	Inverting analog input to the ACMP. (Minus input)	I
ACMPx+	Non-inverting analog input to the ACMP. (Positive input)	I
ACMPxO	Digital output of the ACMP.	O



**Table 10-6. ADCCFG Register Field Descriptions (continued)**

Field	Description
3:2 MODE	Conversion Mode Selection. MODE bits are used to select between 12-, 10-, or 8-bit operation. See <a href="#">Table 10-8</a> .
1:0 ADICLK	Input Clock Select. ADICLK bits select the input clock source to generate the internal clock ADCK. See <a href="#">Table 10-9</a> .

**Table 10-7. Clock Divide Select**

ADIV	Divide Ratio	Clock Rate
00	1	Input clock
01	2	Input clock ÷ 2
10	4	Input clock ÷ 4
11	8	Input clock ÷ 8

**Table 10-8. Conversion Modes**

MODE	Mode Description
00	8-bit conversion (N=8)
01	12-bit conversion (N=12)
10	10-bit conversion (N=10)
11	Reserved

**Table 10-9. Input Clock Select**

ADICLK	Selected Clock Source
00	Bus clock
01	Bus clock divided by 2
10	Alternate clock (ALTCLK)
11	Asynchronous clock (ADACK)

### 10.3.8 Pin Control 1 Register (APCTL1)

The pin control registers disable the I/O port control of MCU pins used as analog inputs. APCTL1 is

### 11.3.3 IIC Control Register (IICC1)

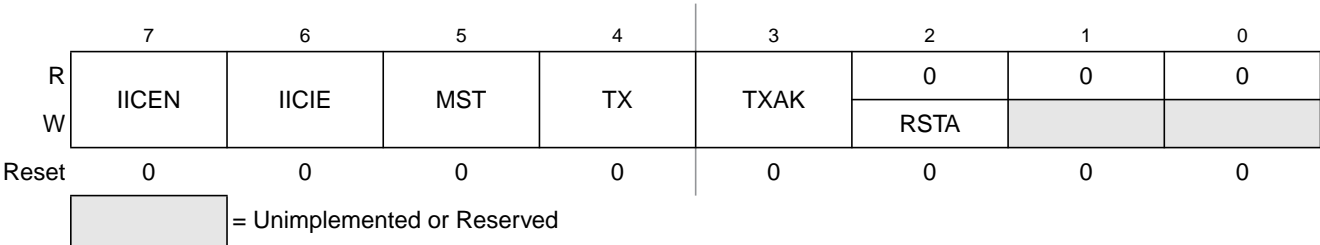


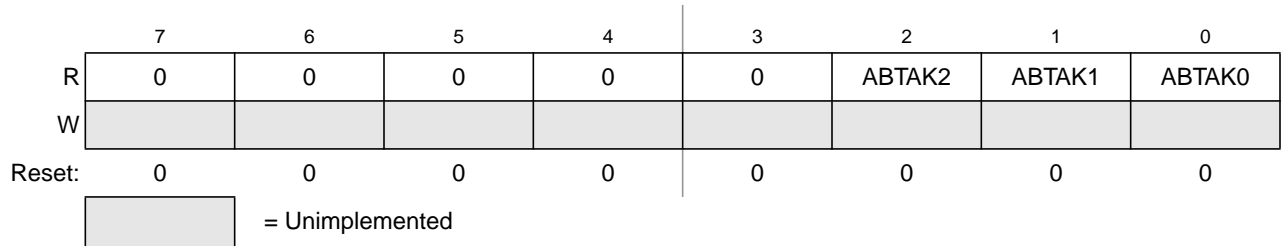
Figure 11-5. IIC Control Register (IICC1)

Table 11-5. IICC1 Field Descriptions

Field	Description
7 IICEN	<b>IIC Enable.</b> The IICEN bit determines whether the IIC module is enabled. 0 IIC is not enabled 1 IIC is enabled
6 IICIE	<b>IIC Interrupt Enable.</b> The IICIE bit determines whether an IIC interrupt is requested. 0 IIC interrupt request not enabled 1 IIC interrupt request enabled
5 MST	<b>Master Mode Select.</b> The MST bit changes from a 0 to a 1 when a start signal is generated on the bus and master mode is selected. When this bit changes from a 1 to a 0 a stop signal is generated and the mode of operation changes from master to slave. 0 Slave mode 1 Master mode
4 TX	<b>Transmit Mode Select.</b> The TX bit selects the direction of master and slave transfers. In master mode, this bit should be set according to the type of transfer required. Therefore, for address cycles, this bit is always high. When addressed as a slave, this bit should be set by software according to the SRW bit in the status register. 0 Receive 1 Transmit
3 TXAK	<b>Transmit Acknowledge Enable.</b> This bit specifies the value driven onto the SDA during data acknowledge cycles for master and slave receivers. 0 An acknowledge signal is sent out to the bus after receiving one data byte 1 No acknowledge signal response is sent
2 RSTA	<b>Repeat start.</b> Writing a 1 to this bit generates a repeated start condition provided it is the current master. This bit is always read as cleared. Attempting a repeat at the wrong time results in loss of arbitration.

### 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

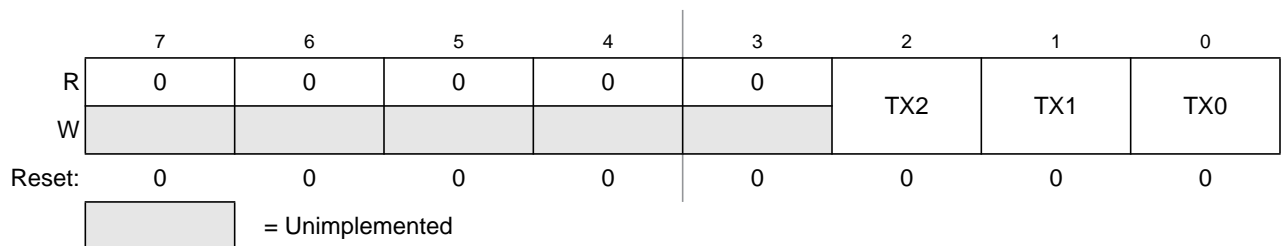
Write: Unimplemented for ABTAKx flags

**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)**

**Table 12-24. Message Buffer Organization**

Offset Address	Register	Access
0x00X0	Identifier Register 0	
0x00X1	Identifier Register 1	
0x00X2	Identifier Register 2	
0x00X3	Identifier Register 3	
0x00X4	Data Segment Register 0	
0x00X5	Data Segment Register 1	
0x00X6	Data Segment Register 2	
0x00X7	Data Segment Register 3	
0x00X8	Data Segment Register 4	
0x00X9	Data Segment Register 5	
0x00XA	Data Segment Register 6	
0x00XB	Data Segment Register 7	
0x00XC	Data Length Register	
0x00XD	Transmit Buffer Priority Register <sup>1</sup>	
0x00XE	Time Stamp Register (High Byte) <sup>2</sup>	
0x00XF	Time Stamp Register (Low Byte) <sup>3</sup>	

<sup>1</sup> Not applicable for receive buffers

<sup>2</sup> Read-only for CPU

<sup>3</sup> Read-only for CPU

Figure 12-23 shows the common 13-byte data structure of receive and transmit buffers for extended identifiers. The mapping of standard identifiers into the IDR registers is shown in Figure 12-24.

All bits of the receive and transmit buffers are 'x' out of reset because of RAM-based implementation<sup>1</sup>. All reserved or unused bits of the receive and transmit buffers always read 'x'.


1. Exception: The transmit priority registers are 0 out of reset.

Section 12.3.10, “MSCAN Transmit Buffer Selection Register (CANTBSEL)”). For receive buffers, only when RXF flag is set (see Section 12.3.4.1, “MSCAN Receiver Flag Register (CANRFLG)”).

Write: For transmit buffers, anytime when TXEx flag is set (see Section 12.3.6, “MSCAN Transmitter Flag Register (CANTFLG)”) and the corresponding transmit buffer is selected in CANTBSEL (see Section 12.3.10, “MSCAN Transmit Buffer Selection Register (CANTBSEL)”). Unimplemented for receive buffers.

Reset: Undefined (0x00XX) because of RAM-based implementation

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
IDR0	R W	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
IDR1	R W	ID2	ID1	ID0	RTR <sup>1</sup>	IDE <sup>2</sup>			
IDR2	R W								
IDR3	R W								

 = Unused, always read 'x'

**Figure 12-24. Receive/Transmit Message Buffer — Standard Identifier Mapping**

<sup>1</sup> The position of RTR differs between extended and standard identifier mapping.  
<sup>2</sup> IDE is 0.

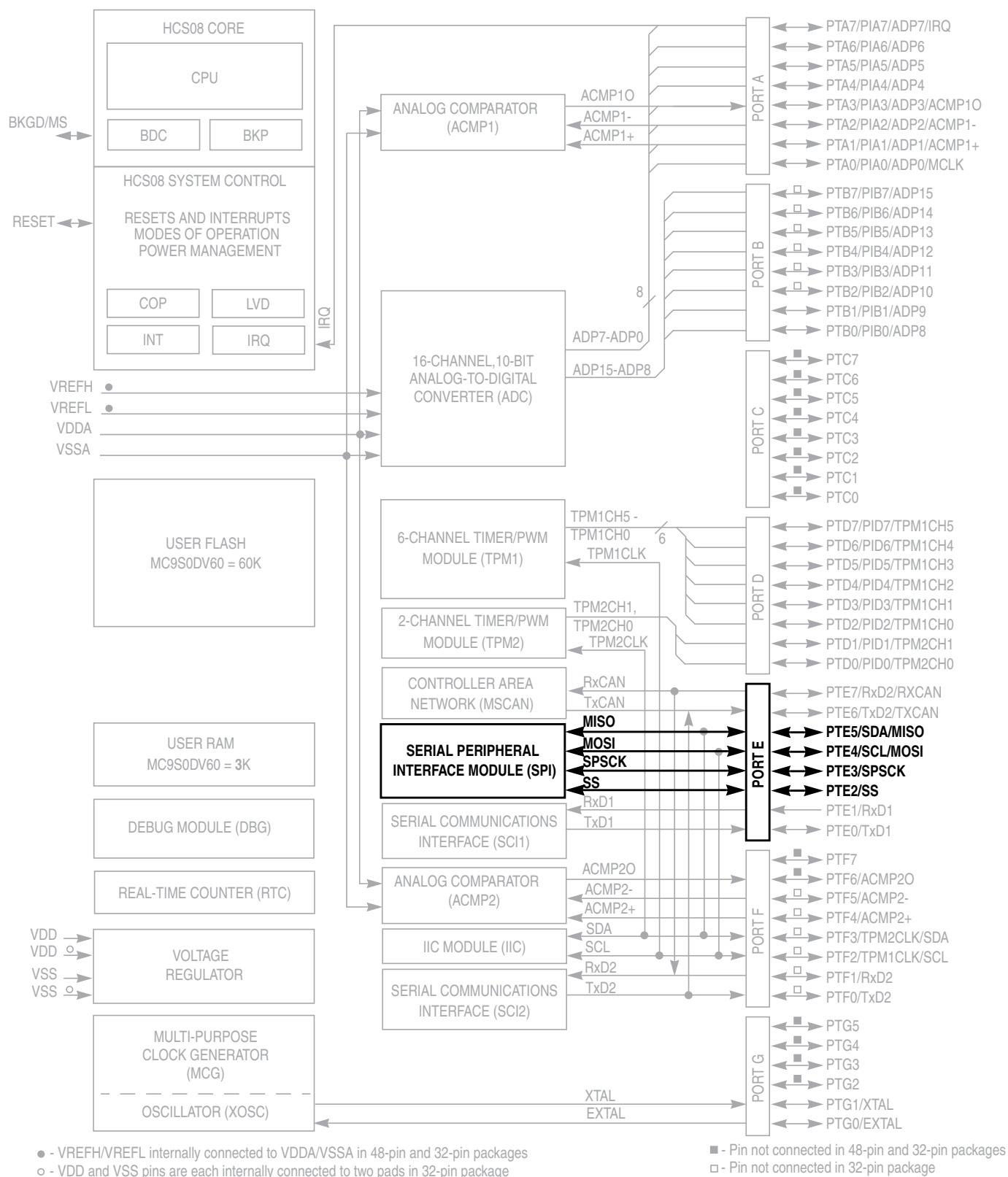
### 12.4.1 Identifier Registers (IDR0–IDR3)

The identifier registers for an extended format identifier consist of a total of 32 bits; ID[28:0], SRR, IDE, and RTR bits. The identifier registers for a standard format identifier consist of a total of 13 bits; ID[10:0], RTR, and IDE bits.

#### 12.4.1.1 IDR0–IDR3 for Extended Identifier Mapping

	7	6	5	4	3	2	1	0
R W	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
Reset:	x	x	x	x	x	x	x	x

**Figure 12-25. Identifier Register 0 (IDR0) — Extended Identifier Mapping**



**Figure 13-1. MC9S08DV60 Block Diagram**

**Table 13-5. SPI Baud Rate Prescaler Divisor**

SPPR2:SPPR1:SPPR0	Prescaler Divisor
0:0:0	1
0:0:1	2
0:1:0	3
0:1:1	4
1:0:0	5
1:0:1	6
1:1:0	7
1:1:1	8

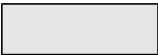
**Table 13-6. SPI Baud Rate Divisor**

SPR2:SPR1:SPR0	Rate Divisor
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

### 13.4.4 SPI Status Register (SPIS)

This register has three read-only status bits. Bits 6, 3, 2, 1, and 0 are not implemented and always read 0. Writes have no meaning or effect.

	7	6	5	4	3	2	1	0
R	SPRF	0	SPTEF	MODF	0	0	0	0
W								
Reset	0	0	1	0	0	0	0	0

 = Unimplemented or Reserved

**Figure 13-8. SPI Status Register (SPIS)**

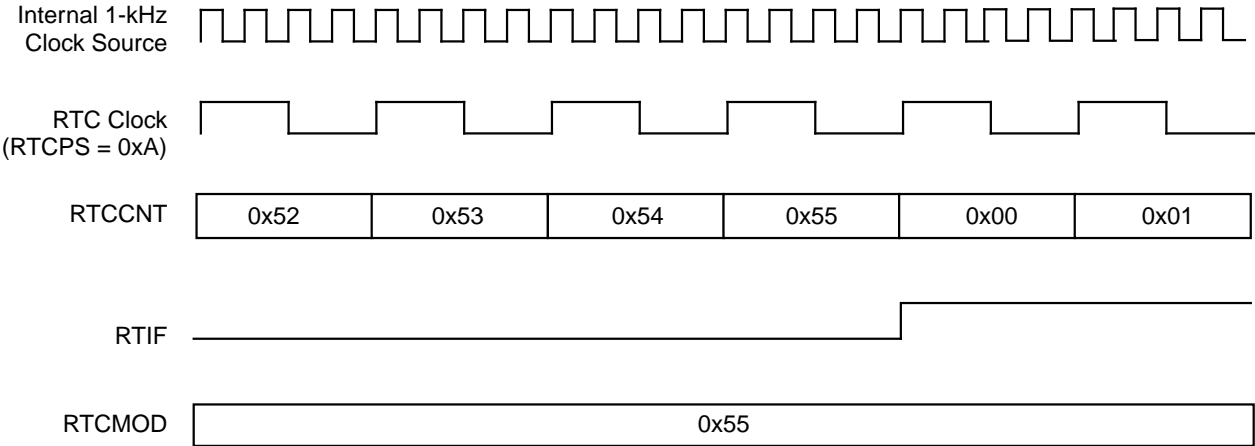


Figure 15-6. RTC Counter Overflow Example

In the example of Figure 15-6, the selected clock source is the 1-kHz internal oscillator clock source. The prescaler (RTCPs) is set to 0xA or divide-by-4. The modulo value in the RTCMOD register is set to 0x55. When the counter, RTCCNT, reaches the modulo value of 0x55, the counter overflows to 0x00 and continues counting. The real-time interrupt flag, RTIF, sets when the counter value changes from 0x55 to 0x00. A real-time interrupt is generated when RTIF is set, if RTIE is set.

### 15.5 Initialization/Application Information

This section provides example code to give some basic direction to a user on how to initialize and configure the RTC module. The example software is implemented in C language.

The example below shows how to implement time of day with the RTC using the 1-kHz clock source to achieve the lowest possible power consumption. Because the 1-kHz clock source is not as accurate as a crystal, software can be added for any adjustments. For accuracy without adjustments at the expense of additional power consumption, the external clock (ERCLK) or the internal clock (IRCLK) can be selected with appropriate prescaler and modulo values.

```

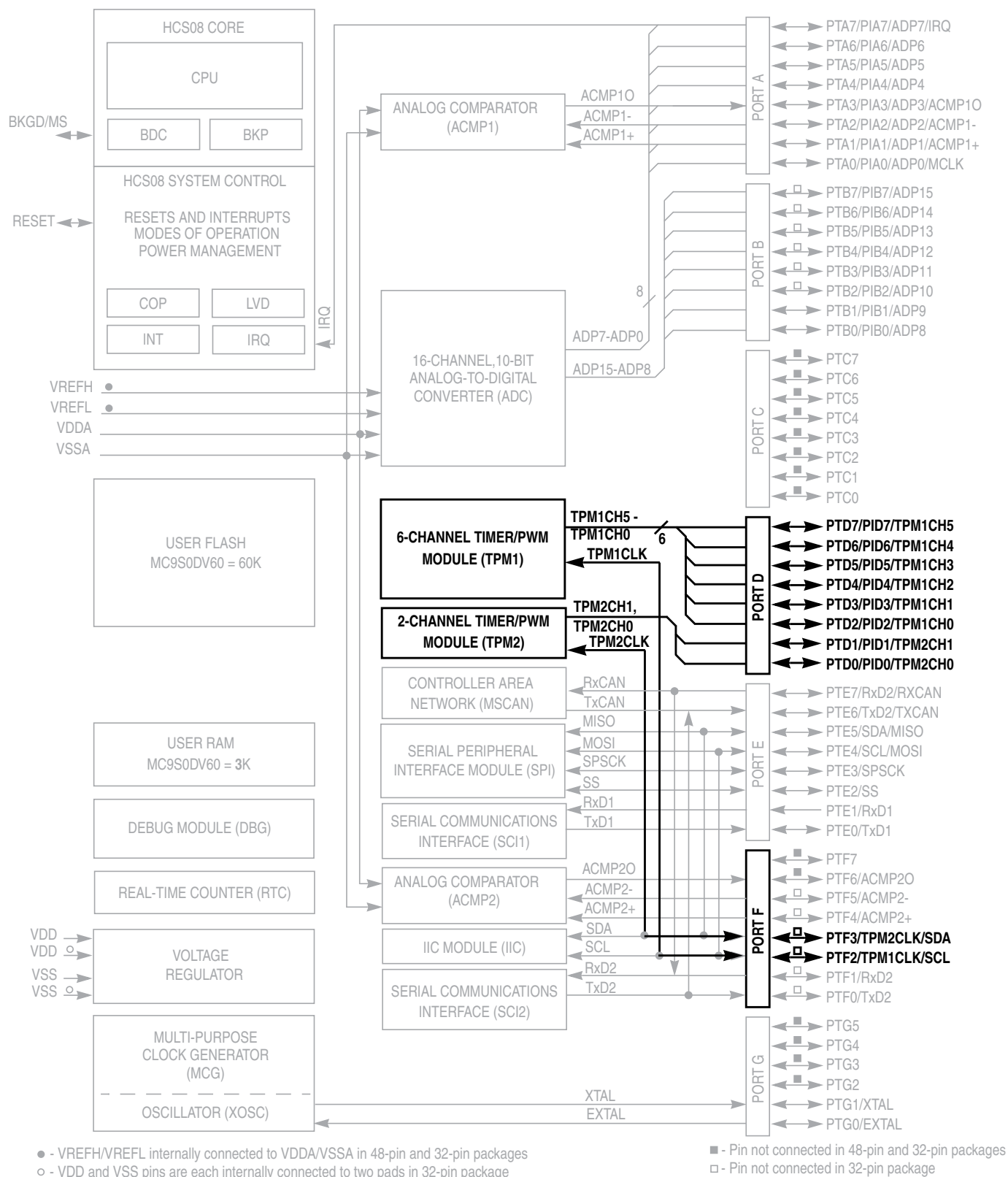
/* Initialize the elapsed time counters */
Seconds = 0;
Minutes = 0;
Hours = 0;
Days=0;

/* Configure RTC to interrupt every 1 second from 1-kHz clock source */
RTCMOD.byte = 0x00;
RTCSC.byte = 0x1F;

/*****
Function Name : RTC_ISR
Notes : Interrupt service routine for RTC module.
*****/

```





**Figure 16-1. MC9S08DV60 Block Diagram**

### 16.6.2.1.2 Center-Aligned PWM Case

When CPWMS=1, TOF gets set when the timer counter changes direction from up-counting to down-counting at the end of the terminal count (the value in the modulo register). In this case the TOF corresponds to the end of a PWM period.

### 16.6.2.2 Channel Event Interrupt Description

The meaning of channel interrupts depends on the channel's current mode (input-capture, output-compare, edge-aligned PWM, or center-aligned PWM).

#### 16.6.2.2.1 Input Capture Events

When a channel is configured as an input capture channel, the ELSnB:ELSnA control bits select no edge (off), rising edges, falling edges or any edge as the edge which triggers an input capture event. When the selected edge is detected, the interrupt flag is set. The flag is cleared by the two-step sequence described in [Section 16.6.2, "Description of Interrupt Operation."](#)

#### 16.6.2.2.2 Output Compare Events

When a channel is configured as an output compare channel, the interrupt flag is set each time the main timer counter matches the 16-bit value in the channel value register. The flag is cleared by the two-step sequence described [Section 16.6.2, "Description of Interrupt Operation."](#)

#### 16.6.2.2.3 PWM End-of-Duty-Cycle Events

For channels configured for PWM operation there are two possibilities. When the channel is configured for edge-aligned PWM, the channel flag gets set when the timer counter matches the channel value register which marks the end of the active duty cycle period. When the channel is configured for center-aligned PWM, the timer count matches the channel value register twice during each PWM cycle. In this CPWM case, the channel flag is set at the start and at the end of the active duty cycle period which are the times when the timer counter matches the channel value register. The flag is cleared by the two-step sequence described [Section 16.6.2, "Description of Interrupt Operation."](#)

## 16.7 The Differences from TPM v2 to TPM v3

1. Write to TPMxCNTH:L registers ([Section 16.3.2, "TPM-Counter Registers \(TPMxCNTH:TPMxCNTL\)"](#)) [SE110-TPM case 7]  
Any write to TPMxCNTH or TPMxCNTL registers in TPM v3 clears the TPM counter (TPMxCNTH:L) and the prescaler counter. Instead, in the TPM v2 only the TPM counter is cleared in this case.
2. Read of TPMxCNTH:L registers ([Section 16.3.2, "TPM-Counter Registers \(TPMxCNTH:TPMxCNTL\)"](#))  
— In TPM v3, any read of TPMxCNTH:L registers during BDM mode returns the value of the TPM counter that is frozen. In TPM v2, if only one byte of the TPMxCNTH:L registers was read before the BDM mode became active, then any read of TPMxCNTH:L registers during

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.

- Non-intrusive commands can be executed at any time even while the user's program is running. Non-intrusive commands allow a user to read or write MCU memory locations or access status and control registers within the background debug controller.

Typically, a relatively simple interface pod is used to translate commands from a host computer into commands for the custom serial interface to the single-wire background debug system. Depending on the development tool vendor, this interface pod may use a standard RS-232 serial port, a parallel printer port, or some other type of communications such as a universal serial bus (USB) to communicate between the host PC and the pod. The pod typically connects to the target system with ground, the BKGD pin,  $\overline{\text{RESET}}$ , and sometimes  $V_{DD}$ . An open-drain connection to reset allows the host to force a target system reset, which is useful to regain control of a lost target system or to control startup of a target system before the on-chip nonvolatile memory has been programmed. Sometimes  $V_{DD}$  can be used to allow the pod to use power from the target system to avoid the need for a separate power supply. However, if the pod is powered separately, it can be connected to a running target system without forcing a target system reset or otherwise disturbing the running application program.

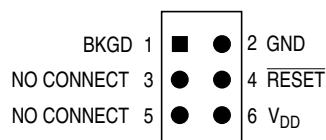


Figure 17-1. BDM Tool Connector

## 17.2.1 BKGD Pin Description

BKGD is the single-wire background debug interface pin. The primary function of this pin is for bidirectional serial communication of active background mode commands and data. During reset, this pin is used to select between starting in active background mode or starting the user's application program. This pin is also used to request a timed sync response pulse to allow a host development tool to determine the correct clock frequency for background debug serial communications.

BDC serial communications use a custom serial protocol first introduced on the M68HC12 Family of microcontrollers. This protocol assumes the host knows the communication clock rate that is determined by the target BDC clock rate. All communication is initiated and controlled by the host that drives a high-to-low edge to signal the beginning of each bit time. Commands and data are sent most significant bit first (MSB first). For a detailed description of the communications protocol, refer to [Section 17.2.2, "Communication Details."](#)

If a host is attempting to communicate with a target MCU that has an unknown BDC clock rate, a SYNC command may be sent to the target MCU to request a timed sync response signal from which the host can determine the correct communication speed.

BKGD is a pseudo-open-drain pin and there is an on-chip pullup so no external pullup resistor is required. Unlike typical open-drain pins, the external RC time constant on this pin, which is influenced by external capacitance, plays almost no role in signal rise time. The custom protocol provides for brief, actively driven speedup pulses to force rapid rise times on this pin without risking harmful drive level conflicts. Refer to [Section 17.2.2, "Communication Details,"](#) for more detail.

### 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.