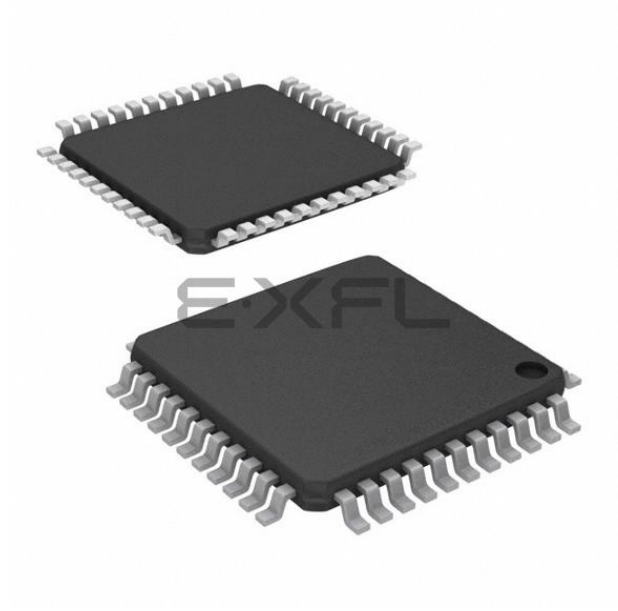


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, LINbus, SCI, SPI, USB
Peripherals	LVD, POR, PWM, WDT
Number of I/O	33
Program Memory Size	8KB (8K 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 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08jm8cld

```
LDHX    #RamLast+1    ;point one past RAM
TXS                   ;SP<-(H:X-1)
```

When security is enabled, the RAM is considered a secure memory resource and is not accessible through BDM or through code executing from non-secure memory. See Section 4.6, “Security,” for a detailed description of the security feature.

4.4 USB RAM

USB RAM is discussed in detail in Chapter 17, “Universal Serial Bus Device Controller (S08USBV1).”

4.5 Flash

Flash memory is used for program storage. In-circuit programming allows the operating program to be loaded into the flash memory after final assembly of the application product. It is possible to program the entire array through the single-wire background debug interface. Because no special voltages are needed for flash erase and programming operations, in-application programming is also possible through other software-controlled communication paths. For a more detailed discussion of in-circuit and in-application programming, refer to the *HCS08 Family Reference Manual, Volume I,* Freescale Semiconductor document order number HCS08RMv1.

4.5.1 Features

Features of the flash memory include:

- Flash size
  - MC9S08JM16 — 16, 384 bytes (32 pages of 512 bytes each)
  - MC9S08JM8 — 8,192 bytes (16 pages of 512 bytes each)
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 100,000 program/erase cycles at typical voltage and temperature
- Flexible block protection
- Security feature for flash and RAM
- Auto power-down for low-frequency read accesses

4.5.2 Program and Erase Times

Before any program or erase command can be accepted, the flash clock divider register (FCDIV) must be written to set the internal clock for the flash module to a frequency (f_{FCLK}) between 150 kHz and 200 kHz (see Section 4.7.1, “Flash Clock Divider Register (FCDIV).”) This register can be written only once, so normally this write is done during reset initialization. FCDIV cannot be written if the access error flag, FACCERR in FSTAT, is set. The user must ensure that FACCERR is not set before writing to the FCDIV register. One period of the resulting clock ($1/f_{FCLK}$) is used by the command processor to time program

5.6.3 LVD Interrupt Operation

When a low voltage condition is detected and the LVD circuit is configured for interrupt operation (LVDE set, LVDIE set, and LVDRE clear), then LVDF will be set and an LVD interrupt will occur.

5.6.4 Low-Voltage Warning (LVW)

The LVD system has a low voltage warning flag to indicate the user that the supply voltage is approaching, but is still above, the LVD voltage. The LVW does not have an interrupt associated with it. There are two user selectable trip voltages for the LVW, one high (V_{LVWH}) and one low (V_{LVWL}). The trip voltage is selected by LVWV in SPMSC2.

5.7 Reset, Interrupt, and System Control Registers and Control Bits

One 8-bit register in the direct page register space and eight 8-bit registers in the high-page register space are related to reset and interrupt systems.

Refer to the direct-page register summary in Chapter 4, "Memory," of this data sheet for the absolute address assignments for all 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.

Some control bits in the SOPT1 and SPMSC2 registers are related to modes of operation. Although brief descriptions of these bits are provided here, the related functions are discussed in greater detail in Chapter 3, "Modes of Operation."

5.7.1 Interrupt Pin Request Status and Control Register (IRQSC)

This direct-page register includes status and control bits, which are used to configure the IRQ function, report status, and acknowledge IRQ events.

	7	6	5	4	3	2	1	0
R	0	IRQPDD	IRQEDG	IRQPE	IRQF	0	IRQIE	IRQMOD
W						IRQACK		
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-2. Interrupt Request Status and Control Register (IRQSC)

	7	6	5	4	3	2	1	0
R W	PTEDS7	PTEDS6	PTEDS5	PTEDS4	PTEDS3	PTEDS2	PTEDS1	PTEDS0
Reset	0	0	0	0	0	0	0	0

Figure 6-26. Output Drive Strength Selection for Port E (PTEDS)

Table 6-25. PTEDS Register Field Descriptions

Field	Description
7:0 PTEDS[7:0]	**Output Drive Strength Selection for Port E Bits** — Each of these control bits selects between low and high output drive for the associated PTE pin. 0 Low output drive enabled for port E bit n. 1 High output drive enabled for port E bit n.

6.5.11 Port F I/O Registers (PTFD and PTFDD)

Port F parallel I/O function is controlled by the registers listed below.

	7	6	5	4	3	2	1	0
R W		PTFD6	PTFD5	PTFD4			PTFD1	PTFD0
Reset	0	0	0	0	0	0	0	0

Figure 6-27. Port F Data Register (PTFD)

Table 6-26. PTFD Register Field Descriptions

Field	Description
6:4, 1:0 PTFD [6:4, 1:0]	**Port F Data Register Bits**— For port F pins that are inputs, reads return the logic level on the pin. For port F 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 F pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTFD 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 pullups disabled.

	7	6	5	4	3	2	1	0
R			PTGDS5	PTGDS4	PTGDS3	PTGDS2	PTGDS1	PTGDS0
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-36. Output Drive Strength Selection for Port G (PTGDS)

Table 6-35. PTGDS Register Field Descriptions

Field	Description
5:0 PTGDSn	**Output Drive Strength Selection for Port G Bits** — Each of these control bits selects between low and high output drive for the associated PTG pin. 0 Low output drive enabled for port G bit n. 1 High output drive enabled for port G bit n.

7.2 Programmer's Model and CPU Registers

Figure 7-1 shows the five CPU registers. CPU registers are not part of the memory map.

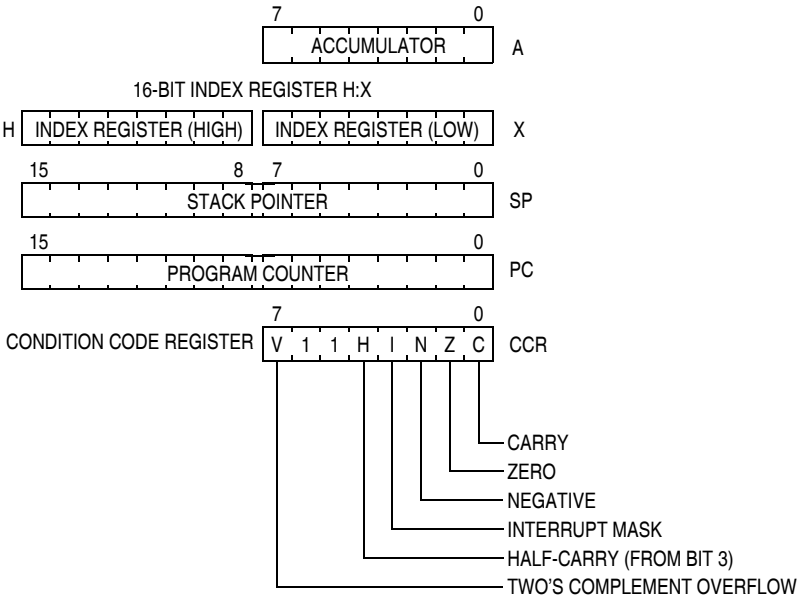


Figure 7-1. CPU Registers

7.2.1 Accumulator (A)

The A accumulator is a general-purpose 8-bit register. One operand input to the arithmetic logic unit (ALU) is connected to the accumulator and the ALU results are often stored into the A accumulator after arithmetic and logical operations. The accumulator can be loaded from memory using various addressing modes to specify the address where the loaded data comes from, or the contents of A can be stored to memory using various addressing modes to specify the address where data from A will be stored.

Reset has no effect on the contents of the A accumulator.

7.2.2 Index Register (H:X)

This 16-bit register is actually two separate 8-bit registers (H and X), which often work together as a 16-bit address pointer where H holds the upper byte of an address and X holds the lower byte of the address. All indexed addressing mode instructions use the full 16-bit value in H:X as an index reference pointer; however, for compatibility with the earlier M68HC05 Family, some instructions operate only on the low-order 8-bit half (X).

Many instructions treat X as a second general-purpose 8-bit register that can be used to hold 8-bit data values. X can be cleared, incremented, decremented, complemented, negated, shifted, or rotated. Transfer instructions allow data to be transferred from A or transferred to A where arithmetic and logical operations can then be performed.

For compatibility with the earlier M68HC05 Family, H is forced to 0x00 during reset. Reset has no effect on the contents of X.

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)

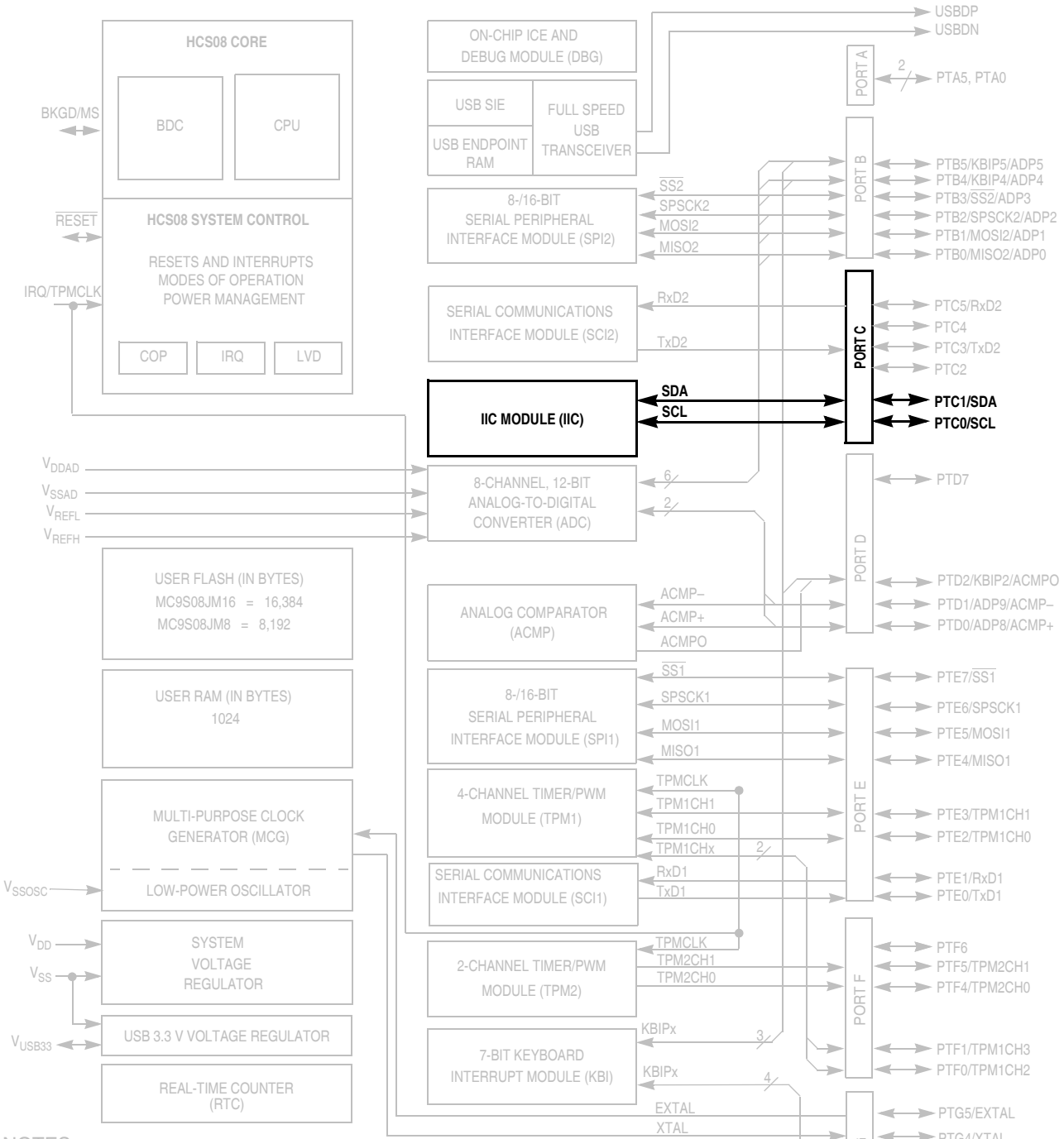
The pin control registers disable the I/O port control of MCU pins used as analog inputs. APCTL1 is used to control the pins associated with channels 0–7 of the ADC module.

	7	6	5	4	3	2	1	0
R W	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
Reset:	0	0	0	0	0	0	0	0

Figure 10-10. Pin Control 1 Register (APCTL1)

Table 10-10. APCTL1 Register Field Descriptions

Field	Description
7 ADPC7	ADC Pin Control 7. ADPC7 controls the pin associated with channel AD7. 0 AD7 pin I/O control enabled 1 AD7 pin I/O control disabled
6 ADPC6	ADC Pin Control 6. ADPC6 controls the pin associated with channel AD6. 0 AD6 pin I/O control enabled 1 AD6 pin I/O control disabled



NOTES:
1. Port pins are software configurable with pullup device if input port.
2. Pin contains software configurable pullup/pulldown device if IRQ is enabled (IRQPE = 1). Pulldown is enabled if rising edge detect is selected (IRQEDG = 1)
3. IRQ does not have a clamp diode to V_{DD}. IRQ must not be driven above V_{DD}.
4. Pin contains integrated pullup device.
5. When pin functions as KBI (KBIPEn = 1) and associated pin is configured to enable the pullup device, KBEDGn can be used to reconfigure the pullup as a pulldown device.

Figure 11-1. MC9S08JM16 Series Block Diagram Highlighting the IIC Block and Pins

In FLL engaged internal mode, the MCGOUT clock is derived from the FLL clock, which is controlled by the internal reference clock. The FLL clock frequency locks to 1024 times the reference frequency, as selected by the RDIV bits. The MCGLCLK is derived from the FLL and the PLL is disabled in a low power state.

12.4.1.2 FLL Engaged External (FEE)

The FLL engaged external (FEE) mode is entered when all the following conditions occur:

- CLKS bits are written to 00
- IREFS bit is written to 0
- PLLS bit is written to 0
- RDIV bits are written to divide reference clock to be within the range of 31.25 kHz to 39.0625 kHz

In FLL engaged external mode, the MCGOUT clock is derived from the FLL clock which is controlled by 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 FLL clock frequency locks to 1024 times the reference frequency, as selected by the RDIV bits. The MCGLCLK is derived from the FLL and the PLL is disabled in a low power state.

12.4.1.3 FLL Bypassed Internal (FBI)

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

The FLL bypassed internal 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
- RDIV bits are written to 000. Since the internal reference clock frequency must already be in the range of 31.25 kHz to 39.0625 kHz after it is trimmed, no further frequency divide is necessary.
- LP bit is written to 0

In FLL bypassed internal mode, the MCGOUT clock is derived from the internal reference clock. The FLL clock is controlled by the internal reference clock, and the FLL clock frequency locks to 1024 times the reference frequency, as selected by the RDIV bits. The MCGLCLK is derived from the FLL and the PLL is disabled in a low power state.

12.4.1.4 FLL Bypassed External (FBE)

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

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

- BDIV=00 (divide by 1), RDIV < 010

BDIV=01 (divide by 2), RDIV < 011

12.5 Initialization / Application Information

This section describes how to initialize and configure the MCG module in application. The following sections include examples on how to initialize the MCG and properly switch between the various available modes.

12.5.1 MCG Module Initialization Sequence

The MCG comes out of reset configured for FEI mode with the BDIV set for divide-by-2. The internal reference will stabilize in t_{irefst} microseconds before the FLL can acquire lock. As soon as the internal reference is stable, the FLL will acquire lock in t_{fll_lock} milliseconds.

Upon POR, the internal reference will require trimming to guarantee an accurate clock. Freescale recommends using FLASH location 0xFFAE for storing the fine trim bit, FTRIM in the MCGSC register, and 0xFFAF for storing the 8-bit trim value in the MCGTRM register. The MCU will not automatically copy the values in these FLASH locations to the respective registers. Therefore, user code must copy these values from FLASH to the registers.

NOTE

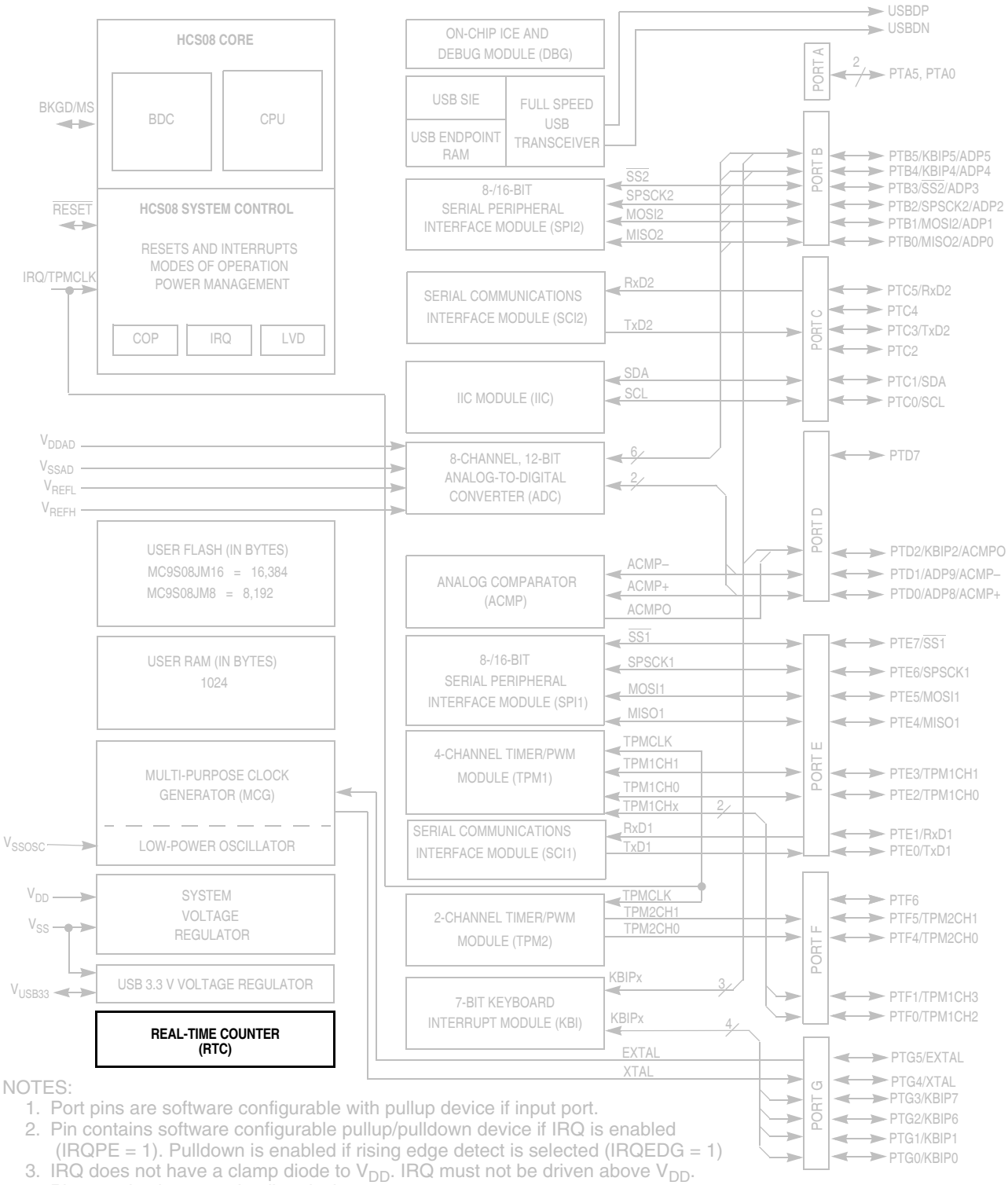
The BDIV value must not be changed to divide-by-1 without first trimming the internal reference. Failure to do so could result in the MCU running out of specification.

12.5.1.1 Initializing the MCG

Because the MCG comes out of reset in FEI mode, the only MCG modes which can be directly switched to upon reset are FEE, FBE, and FBI modes (see Figure 12-8). Reaching any of the other modes requires first configuring the MCG for one of these three initial modes. Care must be taken to check relevant status bits in the MCGSC register reflecting all configuration changes within each mode.

To change from FEI mode to FEE or FBE modes, follow this procedure:

1. Enable the external clock source by setting the appropriate bits in MCGC2.
2. Write to MCGC1 to select the clock mode.
   - If entering FEE, set RDIV appropriately, clear the IREFS bit to switch to the external reference, and leave the CLKS bits at %00 so that the output of the FLL is selected as the system clock source.
   - If entering FBE, clear the IREFS bit to switch to the external reference and change the CLKS bits to %10 so that the external reference clock is selected as the system clock source. The RDIV bits must also be set appropriately here according to the external reference frequency because although the FLL is bypassed, it is still on in FBE mode.
   - The internal reference can optionally be kept running by setting the IRCLKEN bit. This is useful if the application will switch back and forth between internal and external modes. For



NOTES:
1. Port pins are software configurable with pullup device if input port.
2. Pin contains software configurable pullup/pulldown device if IRQ is enabled (IRQPE = 1). Pulldown is enabled if rising edge detect is selected (IRQEDG = 1)
3. IRQ does not have a clamp diode to V_{DD}. IRQ must not be driven above V_{DD}.
4. Pin contains integrated pullup device.
5. When pin functions as KBI (KBIPEn = 1) and associated pin is configured to enable the pullup device, KBEDGn can be used to reconfigure the pullup as a pulldown device.

Figure 13-1. MC9S08JM16 Series Block Diagram Highlighting RTC Block

Table 14-6. SCIxS2 Field Descriptions (continued)

Field	Description
1 LBKDE	**LIN Break Detection Enable**— LBKDE is used to select a longer break character detection length. While LBKDE is set, framing error (FE) and receive data register full (RDRF) flags are prevented from setting. 0 Break character is detected at length of 10 bit times (11 if M = 1). 1 Break character is detected at length of 11 bit times (12 if M = 1).
0 RAF	**Receiver Active Flag** — RAF is set when the SCI receiver detects the beginning of a valid start bit, and RAF is cleared automatically when the receiver detects an idle line. This status flag can be used to check whether an SCI character is being received before instructing the MCU to go to stop mode. 0 SCI receiver idle waiting for a start bit. 1 SCI receiver active (RxD input not idle).

[1] Setting RXINV inverts the RxD input for all cases: data bits, start and stop bits, break, and idle.

When using an internal oscillator in a LIN system, it is necessary to raise the break detection threshold by one bit time. Under the worst case timing conditions allowed in LIN, it is possible that a 0x00 data character can appear to be 10.26 bit times long at a slave which is running 14% faster than the master. This would trigger normal break detection circuitry which is designed to detect a 10 bit break symbol. When the LBKDE bit is set, framing errors are inhibited and the break detection threshold changes from 10 bits to 11 bits, preventing false detection of a 0x00 data character as a LIN break symbol.

14.2.6 SCI Control Register 3 (SCIxC3)

	7	6	5	4	3	2	1	0
R	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-10. SCI Control Register 3 (SCIxC3)

Table 14-7. SCIxC3 Field Descriptions

Field	Description
7 R8	**Ninth Data Bit for Receiver** — When the SCI is configured for 9-bit data (M = 1), R8 can be thought of as a ninth receive data bit to the left of the MSB of the buffered data in the SCIxD register. When reading 9-bit data, read R8 before reading SCIxD because reading SCIxD completes automatic flag clearing sequences which could allow R8 and SCIxD to be overwritten with new data.
6 T8	**Ninth Data Bit for Transmitter** — When the SCI is configured for 9-bit data (M = 1), T8 may be thought of as a ninth transmit data bit to the left of the MSB of the data in the SCIxD register. When writing 9-bit data, the entire 9-bit value is transferred to the SCI shift register after SCIxD is written so T8 must be written (if it needs to change from its previous value) before SCIxD is written. If T8 does not need to change in the new value (such as when it is used to generate mark or space parity), it need not be written each time SCIxD is written.
5 TXDIR	**TxD Pin Direction in Single-Wire Mode** — When the SCI is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this bit determines the direction of data at the TxD pin. 0 TxD pin is an input in single-wire mode. 1 TxD pin is an output in single-wire mode.

When the TPM is configured for center-aligned PWM (and ELSnB:ELSnA not = 0:0), the data direction for all channels in this TPM are overridden, the TPMxCHn pins are forced to be outputs controlled by the TPM, and the ELSnA bits control the polarity of each TPMxCHn output. If ELSnB:ELSnA=1:0, the corresponding TPMxCHn pin is cleared when the timer counter is counting up, and the channel value register matches the timer counter; the TPMxCHn pin is set when the timer counter is counting down, and the channel value register matches the timer counter. If ELSnA=1, the corresponding TPMxCHn pin is set when the timer counter is counting up and the channel value register matches the timer counter; the TPMxCHn pin is cleared when the timer counter is counting down and the channel value register matches the timer counter.

TPMxMODH:TPMxMODL = 0x0008
TPMxMODH:TPMxMODL = 0x0005

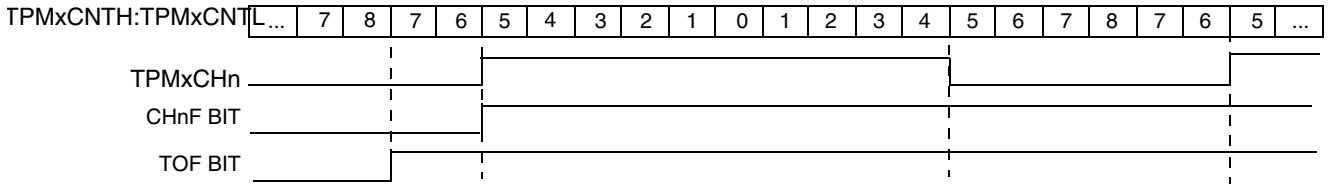


Figure 16-5. High-True Pulse of a Center-Aligned PWM

TPMxMODH:TPMxMODL = 0x0008
TPMxMODH:TPMxMODL = 0x0005

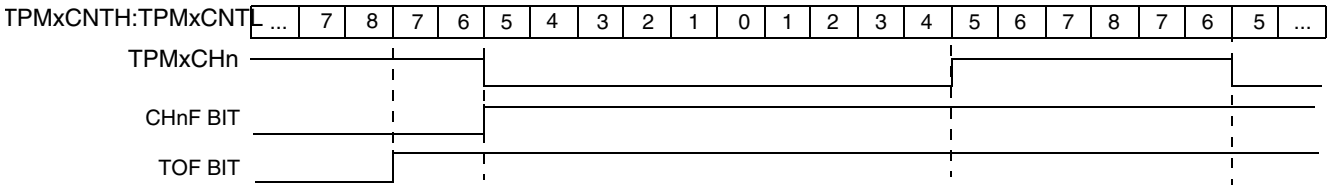


Figure 16-6. Low-True Pulse of a Center-Aligned PWM

Table 17-7. REV Field Descriptions

Field	Description
8–0 REV[7:0]	**Revision** — Revision number of the USB module.

17.3.5 Interrupt Status Register (INTSTAT)

The INTSTAT contains bits for each of the interrupt source within the USB module. Each of these bits is qualified with its respective interrupt enable bits (see the interrupt enable register). All bits of the register are logically OR'ed together to form a single interrupt source for the microcontroller. Once an interrupt bit has been set, it may only be cleared by writing a 1 to the respective interrupt bit. This register will contain the value of 0x00 after a reset.

	7	6	5	4	3	2	1	0
R	STALLF	0	RESUMEF	SLEEPF	TOKDNEF	SOFTOKF	ERRORF	USBRSTF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 17-8. Interrupt Status Register (INTSTAT)

Table 17-9. INTSTAT Field Descriptions

Field	Description
7 STALLF	**Stall Flag** — The stall interrupt is used in device mode. In device mode the stall flag is asserted when a STALL handshake is sent by the serial interface engine (SIE). 0 A STALL handshake has not been sent 1 A STALL handshake has been sent
5 RESUMEF	**Resume Flag** — This bit is set 2.5 μs after clocks to the USB module have restarted following resume signaling. It can be used to indicate remote wakeup signaling on the USB bus. This interrupt is enabled only when the USB module is about to enter suspend mode (usually when SLEEPF interrupt detected). 0 No RESUME observed 1 RESUME detected (K-state is observed on the USBDP/USBDN signals for 2.5 μs)
4 SLEEPF	**Sleep Flag** — This bit is set if the USB module has detected a constant idle on the USB bus for 3 ms, indicating that the USB module will go into suspend mode. The sleep timer is reset by activity on the USB bus. 0 No constant idle state of 3 ms has been detected on the USB bus 1 A constant idle state of 3 ms has been detected on the USB bus
3 TOKDNEF	**Token Complete Flag** — This bit is set when the current transaction is completed. The firmware must immediately read the STAT register to determine the endpoint and BD information. Clearing this bit (by setting it to 1) causes the STAT register to be cleared or the STAT FIFO holding register to be loaded into the STAT register. 0 No tokens being processed are complete 1 Current token being processed is complete
2 SOFTOKF	**SOF Token Flag** — This bit is set if the USB module has received a start of frame (SOF) token. 0 The USB module has not received an SOF token 1 The USB module has received an SOF token

Table 17-11. ERRSTAT Field Descriptions (continued)

Field	Description
3 DFN8F	**Data Field Error Flag** — The data field received was not an interval of 8 bits. The USB Specification specifies that the data field must be an integer number of bytes. If the data field was not an integer number of bytes, this bit will be set. 0 The data field was an integer number of bytes 1 The data field was not an integer number of bytes
2 CRC16F	**CRC16 Error Flag** — The CRC16 failed. If set, the data packet was rejected due to a CRC16 error. 0 No CRC16 error detected 1 CRC16 error detected
1 CRC5F	**CRC5 Error Flag** — This bit will detect a CRC5 error in the token packets generated by the host. If set, the token packet was rejected due to a CRC5 error. 0 No CRC5 error detected 1 CRC5 error detected, and the token packet was rejected.
0 PIDERRF	**PID Error Flag** — The PID check failed. 0 No PID check error detected 1 PID check error detected

17.3.8 Error Interrupt Enable Register (ERRENB)

	7	6	5	4	3	2	1	0
R	BTSERR	0	BUFERR	BTOERR	DFN8	CRC16	CRC5	PIDERR
W								
Reset	0	0	0	0	0	0	0	0

Figure 17-11. Error Interrupt Enable Register (ERRENB)

Table 17-12. ERRSTAT Field Descriptions

Field	Description
7 BTSERR	**BTSERR Interrupt Enable** — Setting this bit will enable BTSERR interrupts. 0 Interrupt disabled 1 Interrupt enabled
5 BUFERR	**BUFERR Interrupt Enable** — Setting this bit will enable BUFERR interrupts. 0 Interrupt disabled 1 Interrupt enabled
4 BTOERR	**BTOERR Interrupt Enable** — Setting this bit will enable BTOERR interrupts. 0 Interrupt disabled 1 Interrupt enabled
3 DFN8	**DFN8 Interrupt Enable** — Setting this bit will enable DFN8 interrupts. 0 Interrupt disabled 1 Interrupt enabled
2 CRC16	**CRC16 Interrupt Enable** — Setting this bit will enable CRC16 interrupts. 0 Interrupt disabled 1 Interrupt enabled

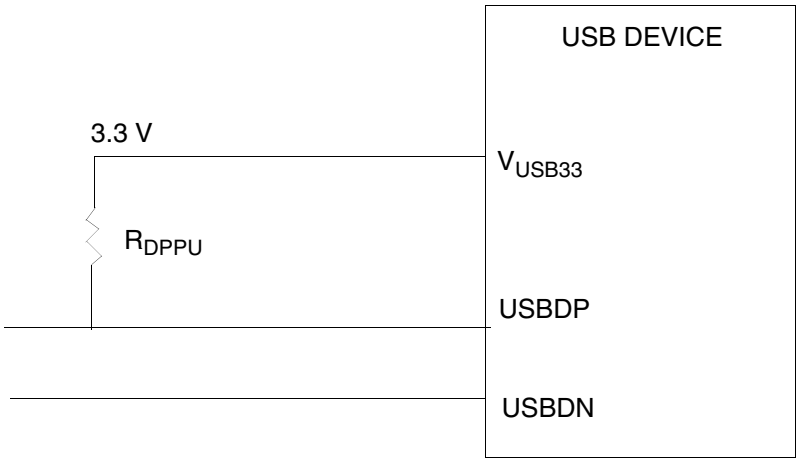


Figure 17-18. USBDP/USBDN Pullup Resistor Configuration for USB module

17.4.1.7 USB Powering and USBDP Pullup Enable Options

The USB module provides a single-chip solution for USB device applications that are self-powered or bus-powered. The USB device needs to know when it has a valid USB connection in order to enable or disable the pullup resistor on the USBDP line. For the USB module on this device, the pullup on USBDP is only applied when a valid VBUS connection is sensed, as required by the USB specification.

In bus-powered applications, system power must be derived from VBUS. Because VBUS is only available when a valid USB connection from host to device is made, the VBUS sensing is built-in, and the USBDP pullup can be enabled accordingly.

With self-powered applications, determining when a valid USB connection is made is different from that of bus-powered applications. In self-powered applications, VBUS sensing must be built into the application. For instance, a KBI pin interrupt can be utilized (if available). When a valid VBUS connection is made, the KBI interrupt can notify the application that a valid USB connection is available, and the internal pullup resistor can be enabled using the USBPU bit. If an external pullup resistor is used instead of the internal one, the VBUS sensing mechanism must be included in the system design.

Table 17-20 summarizes the differences in enabling the USBDP pullup for different USB power modes.

Table 17-20. USBDP Pullup Enable for Different USB Power Modes

Power	USBDP Pullup	Pullup Enable
Bus Power (Built-in VBUS sense)	Internal	Set USBPU bit
	External	Build into application
Self Power (Build VBUS sense into application)	Internal	Set USBPU bit
	External	Build into application

18.3.6 Hardware Breakpoints

The BRKEN control bit in the DBGC register may be set to 1 to allow any of the trigger conditions described in Section 18.3.5, “Trigger Modes,” to be used to generate a hardware breakpoint request to the CPU. TAG in DBGC 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.

18.4 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.

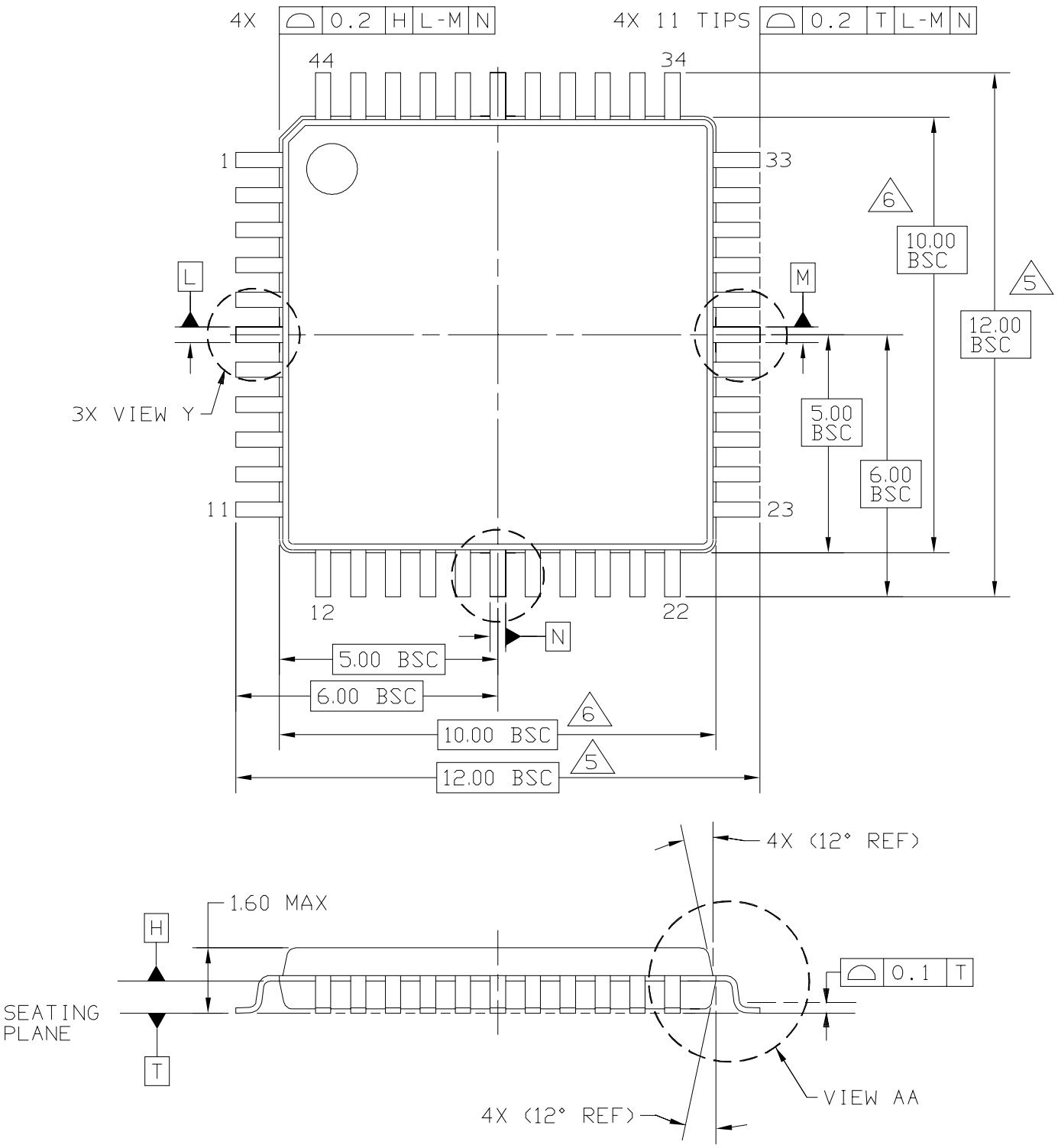
18.4.1 BDC Registers and Control Bits

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.



	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 44 LD TQFP, 10 X 10 PKG, 0.8 PITCH, 1.4 THICK	DOCUMENT NO: 98ASS23225W	REV: C
	CASE NUMBER: 824D-02	19 MAY 2005
	STANDARD: JEDEC MS-026-BCB	