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, I²C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	87
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08dv128mll

Document Number: MC9S08DZ128AD
Rev.1, 07/2011

Addendum Rev.1 to Rev. 1 of the MC9S08DZ128 Series Data Sheet

This addendum identifies changes to Rev. 1 of the MC9S08DZ128 Series Data Sheet. The changes described in this addendum have not been implemented in the specified pages.

1 Pin Availability by Package Pin-Count

Location:	Table 2-1, Page 34

Pin assignments for rows numbered 9— 15 in table 2-1 required updating. The correct information should be:

Pin Number			<-- Lowest **Priority** --> Highest			
100	**64**	**48**	**Port Pin/Interrupt**		**Alt 1**	**Alt 2**
9	7	4				V_{DD}
10	8	5				V_{SS}
11	9	6	PTG0		EXTAL	
12	10	7	PTG1		XTAL	
13	11	8				$\overline{\text{RESET}}$
14	—	—	PTJ2	PIJ2	TPM3CH2	
15	—	—	PTJ3	PIJ3	TPM3CH3	

2 Edge-Aligned PWM Mode

Location:	Section 16.4.2.3, Page 373

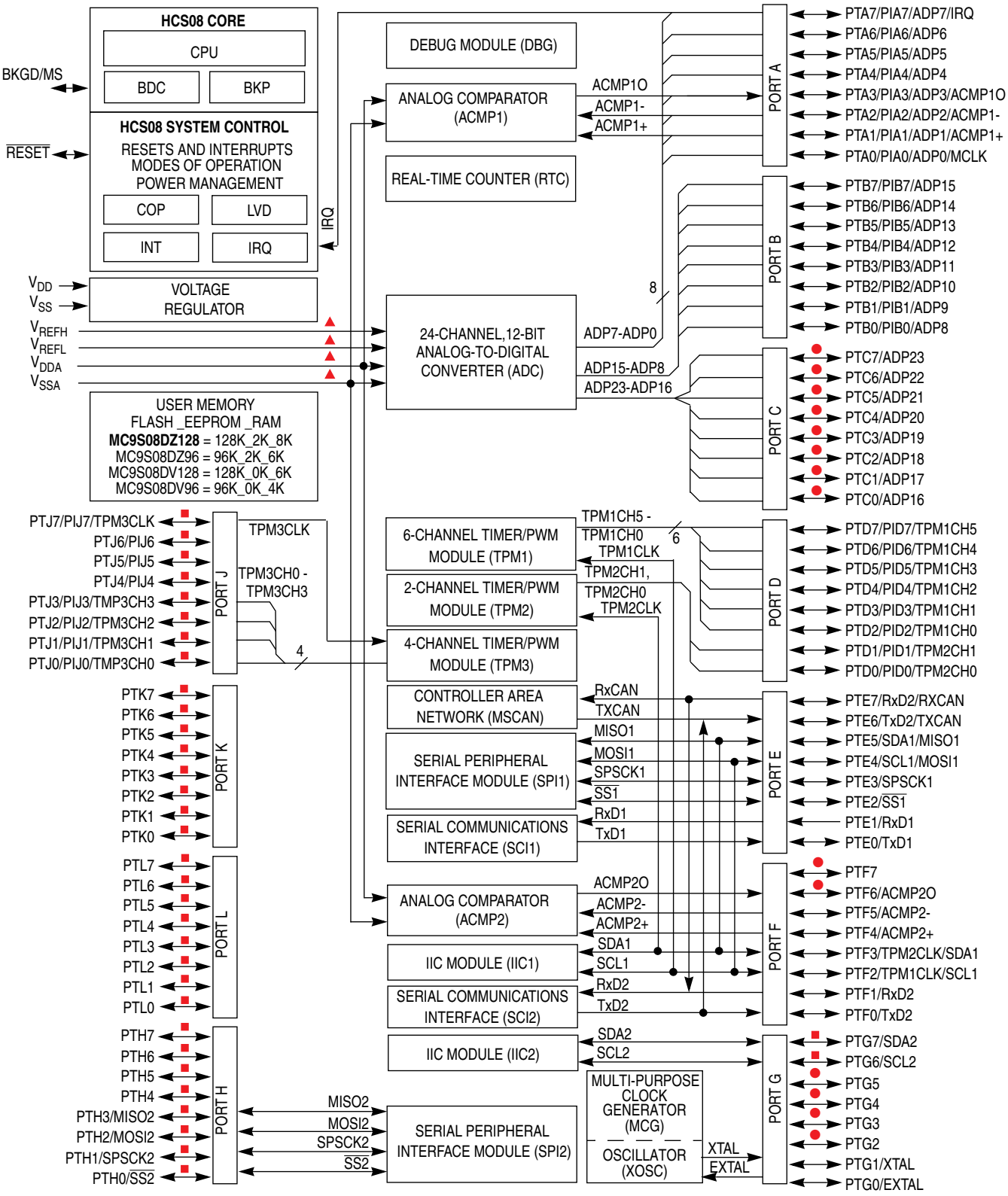
Table 1-2 provides the functional version of the on-chip modules.

Table 1-2. Module Versions

Module		Version
Central Processor Unit	(CPU)	5
Analog Comparator (5V)	(ACMP_5V)	3
Analog-to-Digital Converter	(ADC)	1
Debug Module	(DBG)	3
Inter-Integrated Circuit	(IIC)	2
Multi-Purpose Clock Generator	(MCG)	2
Freescale's Controller Area Network	(MSCAN)	1
Serial Peripheral Interface	(SPI)	3
Serial Communications Interface	(SCI)	4
Real-Time Counter	(RTC)	1
Timer Pulse Width Modulator	(TPM)	3

1.2 MCU Block Diagram

Figure 1-1 is the MC9S08DZ128 Series system-level block diagram.



■ - Pin not connected in 64-pin and 48-pin packages ● - Pin not available in the 48-pin package

▲ - In 48-pin package, VDDA and VREFH are internally connected to each other and VSSA and VREFL are internally connected to each other.

Figure 1-1. MC9S08DZ128 Block Diagram

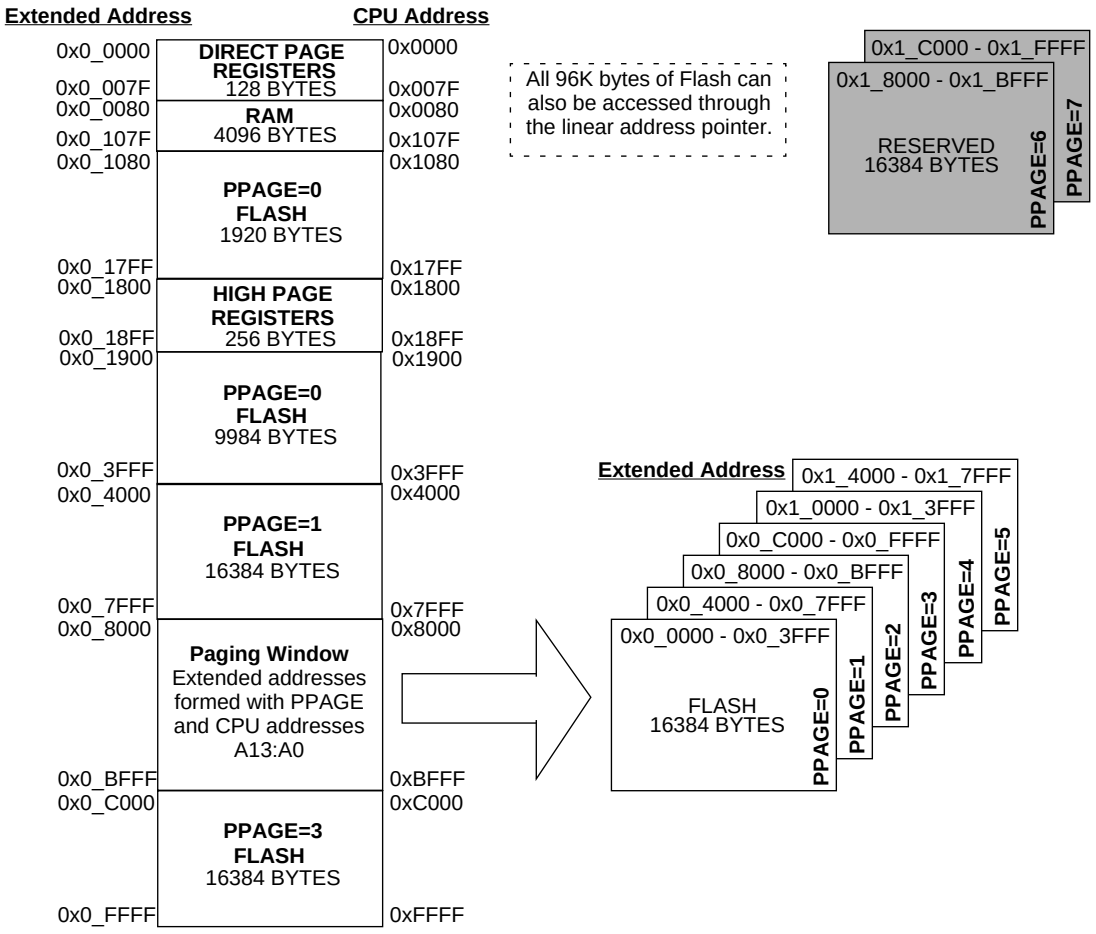


Figure 4-4. MC9S08DV96 Memory Map

4.2 Reset and Interrupt Vector Assignments

Table 4-1 shows address assignments for reset and interrupt vectors. The vector names shown in this table are the labels used in the MC9S08DZ128 Series equate file provided by Freescale Semiconductor.

Table 4-1. Reset and Interrupt Vectors

Address (High:Low)	Vector	Vector Name
0xFF80:0xFF81 - 0xFF8E:0xFF8F	Reserved	Reserved
0xFF90:0xFF91	Port J	Vportj
0xFF92:0xFF93	IIC2	Viic2
0xFF94:0xFF95	SPI2	Vspi2
0xFF96:0xFF97	TPM3 Overflow	Vtpm3ovf
0xFF98:0xFF99	TPM3 Channel 3	Vtpm3ch3
0xFF9A:0xFF9B	TPM3 Channel 2	Vtpm3ch2

Table 5-1. Vector Summary[1]

Vector Number	Address (High/Low)	Vector Name	Module	Source	Enable	Description
56-63	0xFF80/0xFF81 - 0xFF8E/0xFF8F	Reserved				
55	0xFF90/0xFF91	Vportj	Port J	PTJIF	PTJIE	Port J Pins
54	0xFF92/0xFF93	Viic1	IIC2	IICIS	IICIE	IIC2 control
53	0xFF94/0xFF95	Vspi2	SPI2	SPIF, MODF, SPTEF	SPIE, SPIE, SPTIE	SPI2
52	0xFF96/0xFF97	Vtpm3ovf	TPM3	TOF	TOIE	TPM3 overflow
51	0xFF98/0xFF99	Vtpm3ch3	TPM3	CH1F	CH3IE	TPM3 channel 3
50	0xFF9A/0xFF9B	Vtpm3ch2	TPM3	CH0F	CH2IE	TPM3 channel 2
49	0xFF9C/0xFF9D	Vtpm3ch1	TPM3	CH1F	CH1IE	TPM3 channel 1
48	0xFF9E/0xFF9F	Vtpm3ch0	TPM3	CH0F	CH0IE	TPM3 channel 0
N/A	0xFFA0 - 0xFFBF	Non-vector space				
31	0xFFC0/0xFFC1	Vacmp2	ACMP2	ACF	ACIE	Analog comparator 2
30	0xFFC2/0xFFC3	Vacmp1	ACMP1	ACF	ACIE	Analog comparator 1
29	0xFFC4/0xFFC5	Vcantx	MSCAN	TXE[2:0]	TXEIE[2:0]	CAN transmit
28	0xFFC6/0xFFC7	Vcanrx	MSCAN	RXF	RXFIE	CAN receive
27	0xFFC8/0xFFC9	Vcanerr	MSCAN	CSCIF, OVRIF	CSCIE, OVRIE	CAN errors
26	0xFFCA/0xFFCB	Vcanwu	MSCAN	WUPIF	WUPIE	CAN wake-up
25	0xFFCC/0xFFCD	Vrtc	RTC	RTIF	RTIE	Real-time interrupt
24	0xFFCE/0xFFCF	Viic1	IIC1	IICIS	IICIE	IIC1 control
23	0xFFD0/0xFFD1	Vadc	ADC	COCOF	AIEN	ADC
22	0xFFD2/0xFFD3	Vport	Port A,B,D	PTAIF, PTBIF, PTDIF	PTAIE, PTBIE, PTDIE	Port A, B and D Pins
21	0xFFD4/0xFFD5	Vsci2tx	SCI2	TDRE, TC	TIE, TCIE	SCI2 transmit
20	0xFFD6/0xFFD7	Vsci2rx	SCI2	IDLE, LBKDIF, RDRF, RXEDGIF	ILIE, LBKDIE, RIE, RXEDGIE	SCI2 receive
19	0xFFD8/0xFFD9	Vsci2err	SCI2	OR, NF FE, PF	ORIE, NFIE, FEIE, PFIE	SCI2 error
18	0xFFDA/0xFFDB	Vsci1tx	SCI1	TDRE, TC	TIE, TCIE	SCI1 transmit
17	0xFFDC/0xFFDD	Vsci1rx	SCI1	IDLE, LBKDIF, RDRF, RXEDGIF	ILIE, LBKDIE, RIE, RXEDGIE	SCI1 receive
16	0xFFDE/0xFFDF	Vsci1err	SCI1	OR, NF, FE, PF	ORIE, NFIE, FEIE, PFIE	SCI1 error
15	0xFFE0/0xFFE1	Vspi1	SPI1	SPIF, MODF, SPTEF	SPIE, SPIE, SPTIE	SPI1
14	0xFFE2/0xFFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 overflow
13	0xFFE4/0xFFE5	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 channel 1
12	0xFFE6/0xFFE7	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 channel 0
11	0xFFE8/0xFFE9	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 overflow
10	0xFFEA/0xFFEB	Vtpm1ch5	TPM1	CH5F	CH5IE	TPM1 channel 5
9	0xFFEC/0xFFED	Vtpm1ch4	TPM1	CH4F	CH4IE	TPM1 channel 4
8	0xFFEE/0xFFEF	Vtpm1ch3	TPM1	CH3F	CH3IE	TPM1 channel 3
7	0xFFF0/0xFFF1	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1 channel 2

PTxSC provided all enabled port inputs are at their deasserted levels. PTxIF will remain set if any enabled port pin is asserted while attempting to clear by writing a 1 to PTxACK.

6.3.3 Pull-up/Pull-down Resistors

The port interrupt pins can be configured to use an internal pull-up/pull-down resistor using the associated I/O port pull-up enable register. If an internal resistor is enabled, the PTxES register is used to select whether the resistor is a pull-up (PTxESn = 0) or a pull-down (PTxESn = 1).

6.3.4 Pin Interrupt Initialization

When an interrupt pin is first enabled, it is possible to get a false interrupt flag. To prevent a false interrupt request during pin interrupt initialization, the user should do the following:

1. Mask interrupts by clearing PTxIE in PTxSC.
2. Select the pin polarity by setting the appropriate PTxESn bits in PTxES.
3. If using internal pull-up/pull-down device, configure the associated pull enable bits in PTxPE.
4. Enable the interrupt pins by setting the appropriate PTxPSn bits in PTxPS.
5. Write to PTxACK in PTxSC to clear any false interrupts.
6. Set PTxIE in PTxSC to enable interrupts.

6.4 Pin Behavior in Stop Modes

Pin behavior following execution of a STOP instruction depends on the stop mode that is entered. An explanation of pin behavior for the various stop modes follows:

- Stop2 mode is a partial power-down mode, whereby I/O latches are maintained in their state as before the STOP instruction was executed. CPU register status and the state of I/O registers should be saved in RAM before the STOP instruction is executed to place the MCU in stop2 mode. Upon recovery from stop2 mode, before accessing any I/O, the user should examine the state of the PPDF bit in the SPMSC2 register. If the PPDF bit is 0, I/O must be initialized as if a power on reset had occurred. If the PPDF bit is 1, peripherals may require initialization to be restored to their pre-stop condition. This can be done using data previously stored in RAM if it was saved before the STOP instruction was executed. The user must then write a 1 to the PPDACK bit in the SPMSC2 register. Access to I/O is now permitted again in the user application program.
- In stop3 mode, all I/O is maintained because internal logic circuity stays powered up. Upon recovery, normal I/O function is available to the user.

6.5 Parallel I/O and Pin Control Registers

This section provides information about the registers associated with the parallel I/O ports. The data and data direction registers are located in page zero of the memory map. The pull up, slew rate, drive strength, and interrupt control registers are located in the high page section of the memory map.

Refer to tables in Chapter 4, “Memory,” for the absolute address assignments for all parallel I/O and their pin control registers. This section refers to registers and control bits only by their names. A Freescale

6.5.8.3 Port H Pull Enable Register (PTHPE)

	7	6	5	4	3	2	1	0
R W	PTHPE7	PTHPE6	PTHPE5	PTHPE4	PTHPE3	PTHPE2	PTHPE1	PTHPE0
Reset:	0	0	0	0	0	0	0	0

Figure 6-49. Internal Pull Enable for Port H Register (PTHPE)

Table 6-47. PTHPE Register Field Descriptions

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

NOTE

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

6.5.8.4 Port H Slew Rate Enable Register (PTHSE)

	7	6	5	4	3	2	1	0
R W	PTHSE7	PTHSE6	PTHSE5	PTHSE4	PTHSE3	PTHSE2	PTHSE1	PTHSE0
Reset:	0	0	0	0	0	0	0	0

Figure 6-50. Slew Rate Enable for Port H Register (PTHSE)

Table 6-48. PTHSE Register Field Descriptions

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

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

6.5.9.3 Port J Pull Enable Register (PTJPE)

	7	6	5	4	3	2	1	0
R W	PTJPE7	PTJPE6	PTJPE5	PTJPE4	PTJPE3	PTJPE2	PTJPE1	PTJPE0
Reset:	0	0	0	0	0	0	0	0

Figure 6-54. Internal Pull Enable for Port J Register (PTJPE)

Table 6-52. PTJPE Register Field Descriptions

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

NOTE

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

6.5.9.4 Port J Slew Rate Enable Register (PTJSE)

	7	6	5	4	3	2	1	0
R W	PTJSE7	PTJSE6	PTJSE5	PTJSE4	PTJSE3	PTJSE2	PTJSE1	PTJSE0
Reset:	0	0	0	0	0	0	0	0

Figure 6-55. Slew Rate Enable for Port J Register (PTJSE)

Table 6-53. PTJSE Register Field Descriptions

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

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

6.5.11.3 Port L Pull Enable Register (PTLPE)

	7	6	5	4	3	2	1	0
R W	PTLPE7	PTLPE6	PTLPE5	PTLPE4	PTLPE3	PTLPE2	PTLPE1	PTLPE0
Reset:	0	0	0	0	0	0	0	0

Figure 6-67. Internal Pull Enable for Port L Register (PTLPE)

Table 6-65. PTLPE Register Field Descriptions

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

NOTE

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

6.5.11.4 Port L Slew Rate Enable Register (PTLSE)

	7	6	5	4	3	2	1	0
R W	PTLSE7	PTLSE6	PTLSE5	PTLSE4	PTLSE3	PTLSE2	PTLSE1	PTLSE0
Reset:	0	0	0	0	0	0	0	0

Figure 6-68. Slew Rate Enable for Port L Register (PTLSE)

Table 6-66. PTLSE Register Field Descriptions

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

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

The RTC instruction is used to terminate subroutines invoked by a CALL instruction. RTC unstacks the PPAGE value and the return address, the queue is refilled, and execution resumes with the next instruction after the corresponding CALL.

The actual sequence of operations that occur during execution of RTC is:

1. The return value of the 8-bit PPAGE register is pulled from the stack.
2. The 16-bit return address is pulled from the stack and loaded into the PC.
3. The return PPAGE value is written to the PPAGE register.
4. The queue is refilled and execution begins at the new address.

Since the return operation is implemented as a single uninterruptable CPU instruction, the RTC can be executed from anywhere in memory, including from a different page of extended memory in the overlay window.

The CALL and RTC instructions behave like JSR and RTS, except they have slightly longer execution times. Since extra execution cycles are required, routinely substituting CALL/RTC for JSR/RTS is not recommended. JSR and RTS can be used to access subroutines that are located outside the program overlay window or on the same memory page. However, if a subroutine can be called from other pages, it must be terminated with an RTC. In this case, since RTC unstacks the PPAGE value as well as the return address, all accesses to the subroutine, even those made from the same page, must use CALL instructions.

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 external reference frequency, as selected by the RDIV, RANGE and DIV32 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.

In this mode, the DRST bit reads 0 regardless of whether the DRS bit is set to 1 or 0.

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.

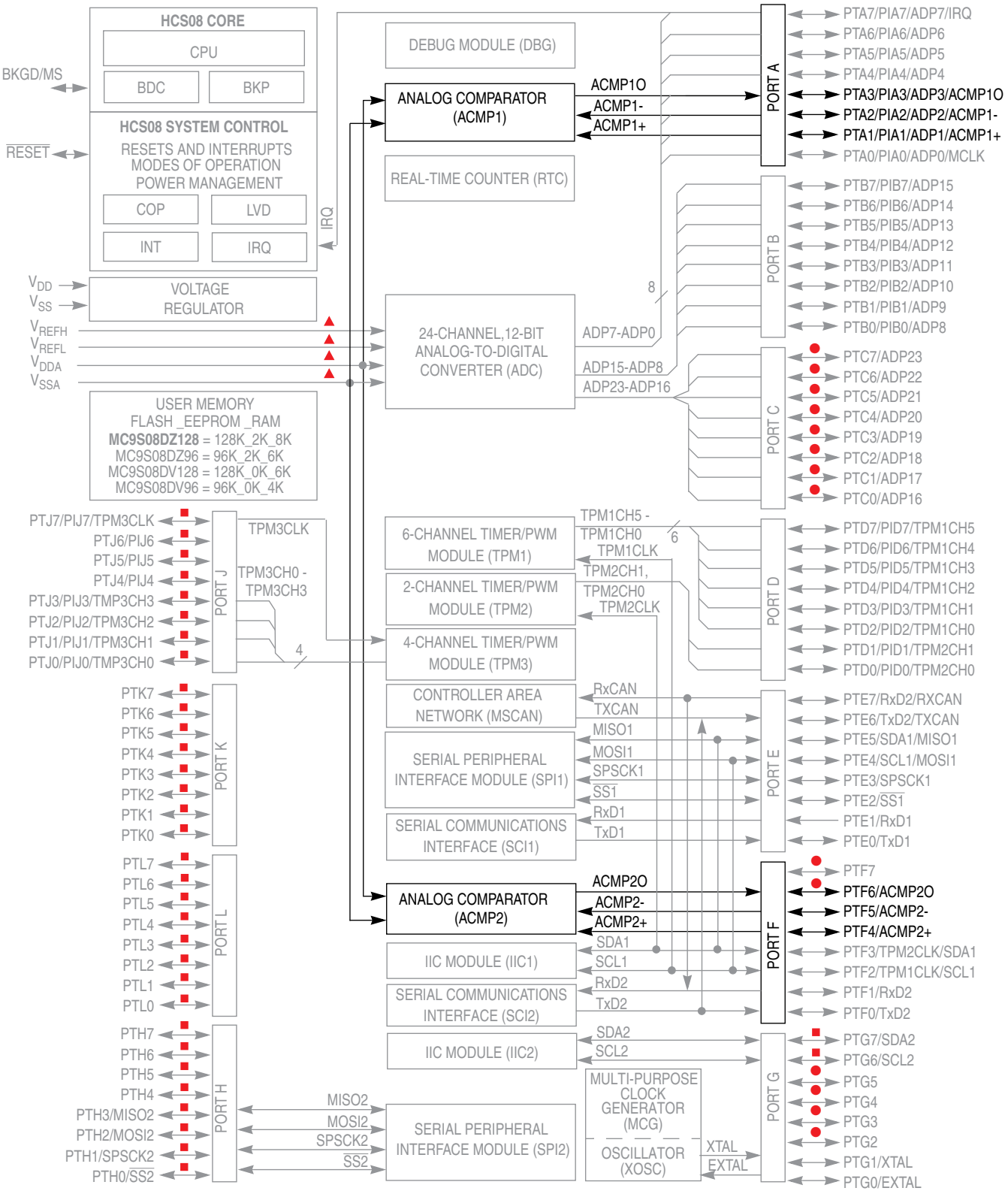
The PLL and the FLL are disabled at all times in BLPI mode and the MCGLCLK will not be available for BDC communications If the BDM becomes active the mode will switch to FLL bypassed internal (FBI) mode.

In this mode, the DRST bit reads 0 regardless of whether the DRS bit is set to 1 or 0.

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
- LP bit is written to 1



■ - Pin not connected in 64-pin and 48-pin packages ● - Pin not available in the 48-pin package

▲ - In 48-pin package, VDDA and VREFH are internally connected to each other and VSSA and VREFL are internally connected to each other.

Figure 9-1. MC9S08DZ128 Block Diagram with ACMP Highlighted

12.5.5.5 MSCAN Initialization Mode

In initialization mode, any on-going transmission or reception is immediately aborted and synchronization to the CAN bus is lost, potentially causing CAN protocol violations. To protect the CAN bus system from fatal consequences of violations, the MSCAN immediately drives the TXCAN pin into a recessive state.

NOTE

The user is responsible for ensuring that the MSCAN is not active when initialization mode is entered. The recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before setting the INITRQ bit in the CANCTL0 register. Otherwise, the abort of an on-going message can cause an error condition and can impact other CAN bus devices.

In initialization mode, the MSCAN is stopped. However, interface registers remain accessible. This mode is used to reset the CANCTL0, CANRFLG, CANRIER, CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL registers to their default values. In addition, the MSCAN enables the configuration of the CANBTR0, CANBTR1 bit timing registers; CANIDAC; and the CANIDAR, CANIDMR message filters. See Section 12.3.1, "MSCAN Control Register 0 (CANCTL0)," for a detailed description of the initialization mode.

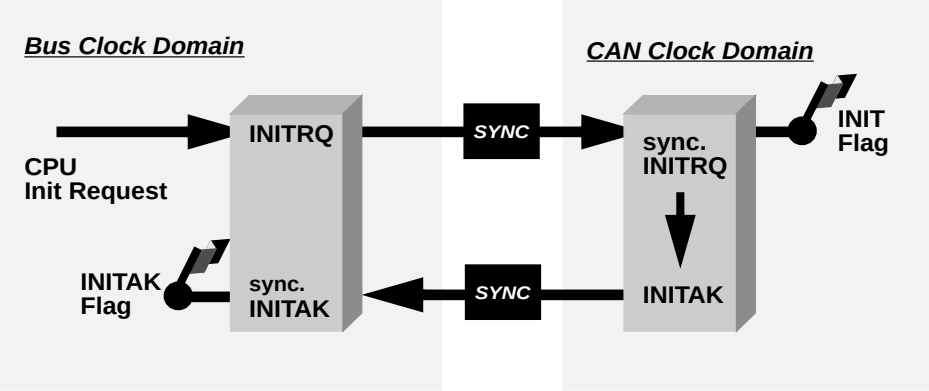


Figure 12-46. Initialization Request/Acknowledge Cycle

Due to independent clock domains within the MSCAN, INITRQ must be synchronized to all domains by using a special handshake mechanism. This handshake causes additional synchronization delay (see Section Figure 12-46., "Initialization Request/Acknowledge Cycle").

If there is no message transfer ongoing on the CAN bus, the minimum delay will be two additional bus clocks and three additional CAN clocks. When all parts of the MSCAN are in initialization mode, the INITAK flag is set. The application software must use INITAK as a handshake indication for the request (INITRQ) to go into initialization mode.

NOTE

The CPU cannot clear INITRQ before initialization mode (INITRQ = 1 and INITAK = 1) is active.

12.5.7.1 Description of Interrupt Operation

The MSCAN supports four interrupt vectors (see Table 12-37), any of which can be individually masked (for details see sections from Section 12.3.5, "MSCAN Receiver Interrupt Enable Register (CANRIER)," to Section 12.3.7, "MSCAN Transmitter Interrupt Enable Register (CANTIER)").

NOTE

The dedicated interrupt vector addresses are defined in the Resets and Interrupts chapter.

Table 12-37. Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Wake-Up Interrupt (WUPIF)	I bit	CANRIER (WUPIE)
Error Interrupts Interrupt (CSCIF, OVRIF)	I bit	CANRIER (CSCIE, OVRIE)
Receive Interrupt (RXF)	I bit	CANRIER (RXFIE)
Transmit Interrupts (TXE[2:0])	I bit	CANTIER (TXEIE[2:0])

12.5.7.2 Transmit Interrupt

At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. The TXEx flag of the empty message buffer is set.

12.5.7.3 Receive Interrupt

A message is successfully received and shifted into the foreground buffer (RxFG) of the receiver FIFO. This interrupt is generated immediately after receiving the EOF symbol. The RXF flag is set. If there are multiple messages in the receiver FIFO, the RXF flag is set as soon as the next message is shifted to the foreground buffer.

12.5.7.4 Wake-Up Interrupt

A wake-up interrupt is generated if activity on the CAN bus occurs during MSCAN internal sleep mode. WUPE (see Section 12.3.1, "MSCAN Control Register 0 (CANCTL0)") must be enabled.

12.5.7.5 Error Interrupt

An error interrupt is generated if an overrun of the receiver FIFO, error, warning, or bus-off condition occurrs. Section 12.3.4.1, "MSCAN Receiver Flag Register (CANRFLG) indicates one of the following conditions:

- **Overrun** — An overrun condition of the receiver FIFO as described in Section 12.5.2.3, "Receive Structures," occurred.
- **CAN Status Change** — The actual value of the transmit and receive error counters control the CAN bus state of the MSCAN. As soon as the error counters skip into a critical range (Tx/Rx-warning, Tx/Rx-error, bus-off) the MSCAN flags an error condition. The status change, which caused the error condition, is indicated by the TSTAT and RSTAT flags (see

pin from a master and the MISO waveform applies to the MISO output from a slave. The $\overline{SS}$ OUT waveform applies to the slave select output from a master (provided MODFEN and SSOE = 1). The master $\overline{SS}$ output goes to active low one-half SPSCK cycle before the start of the transfer and goes back high at the end of the eighth bit time of the transfer. The $\overline{SS}$ IN waveform applies to the slave select input of a slave.

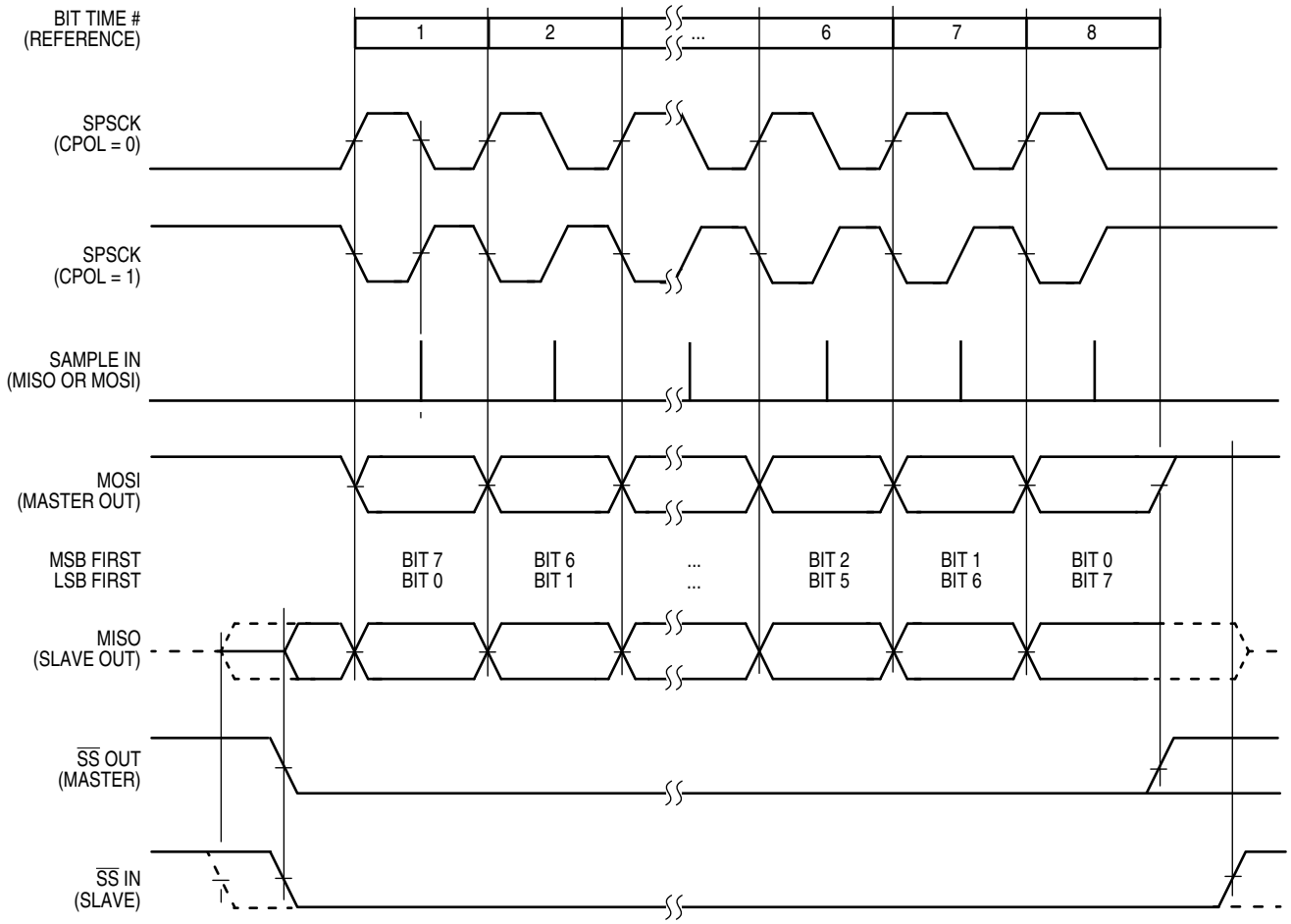


Figure 13-10. SPI Clock Formats (CPHA = 1)

When CPHA = 1, the slave begins to drive its MISO output when $\overline{SS}$ goes to active low, but the data is not defined until the first SPSCK edge. The first SPSCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the third SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled, and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When CHPA = 1, the slave's $\overline{SS}$ input is not required to go to its inactive high level between transfers.

Figure 13-11 shows the clock formats when CPHA = 0. At the top of the figure, the eight bit times are shown for reference with bit 1 starting as the slave is selected ($\overline{SS}$ IN goes low), and bit 8 ends at the last SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting

14.1.4 Block Diagram

Figure 14-2 shows the transmitter portion of the SCI.

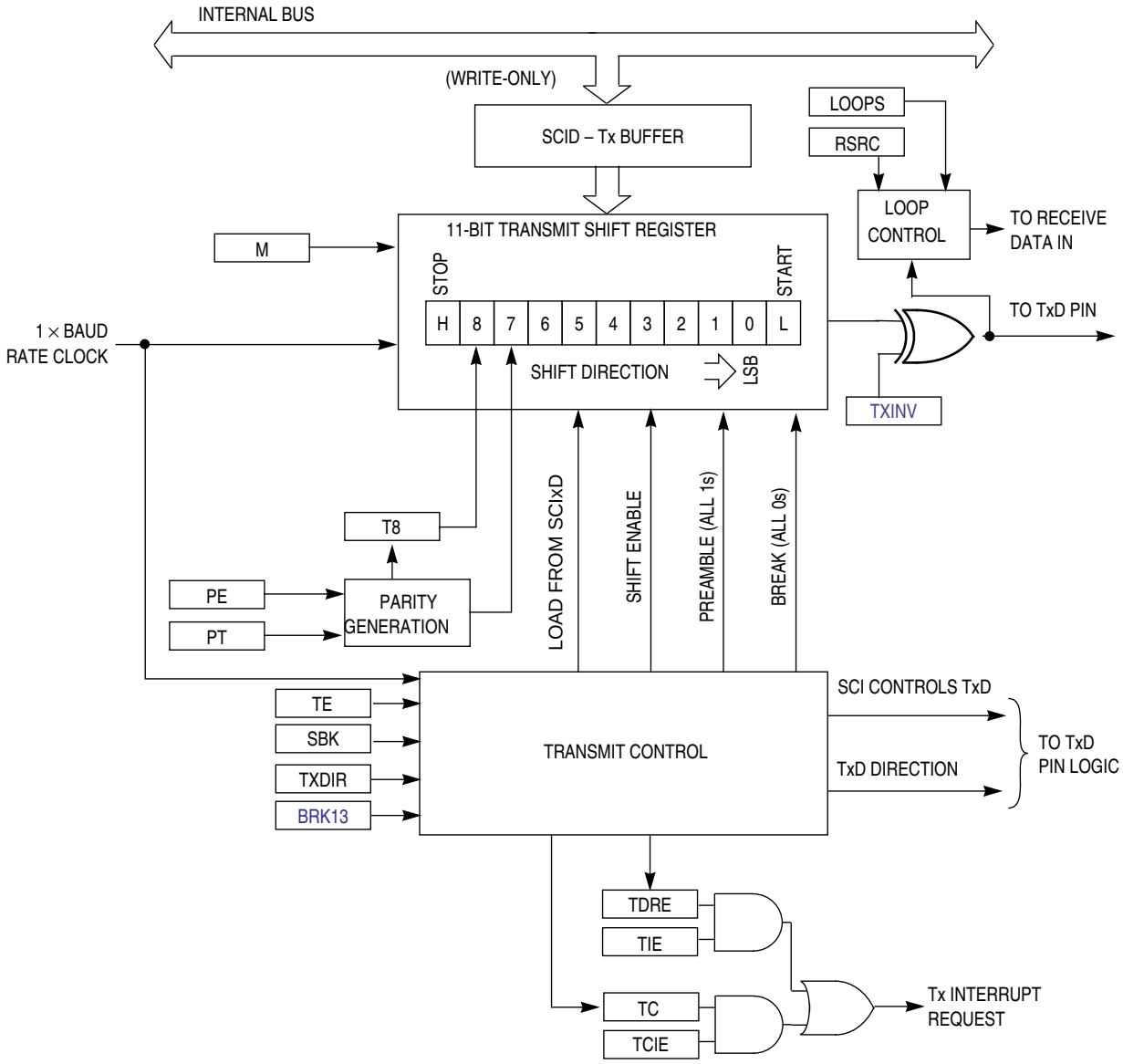


Figure 14-2. SCI Transmitter Block Diagram

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

14.3.3.2.1 Idle-Line Wakeup

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

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

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

14.3.3.2.2 Address-Mark Wakeup

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

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

14.3.4 Interrupts and Status Flags

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

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

The following sections describe the main counter and each of the timer operating modes (input capture, output compare, edge-aligned PWM, and center-aligned PWM). Because details of pin operation and interrupt activity depend upon the operating mode, these topics will be covered in the associated mode explanation sections.

16.4.1 Counter

All timer functions are based on the main 16-bit counter (TPMxCNTH:TPMxCNTL). This section discusses selection of the clock source, end-of-count overflow, up-counting vs. up/down counting, and manual counter reset.

16.4.1.1 Counter Clock Source

The 2-bit field, CLKSB:CLKSA, in the timer status and control register (TPMxSC) selects one of three possible clock sources or OFF (which effectively disables the TPM). See Table 16-3. After any MCU reset, CLKSB:CLKSA=0:0 so no clock source is selected, and the TPM is in a very low power state. These control bits may be read or written at any time and disabling the timer (writing 00 to the CLKSB:CLKSA field) does not affect the values in the counter or other timer registers.

Chapter 17
Development Support

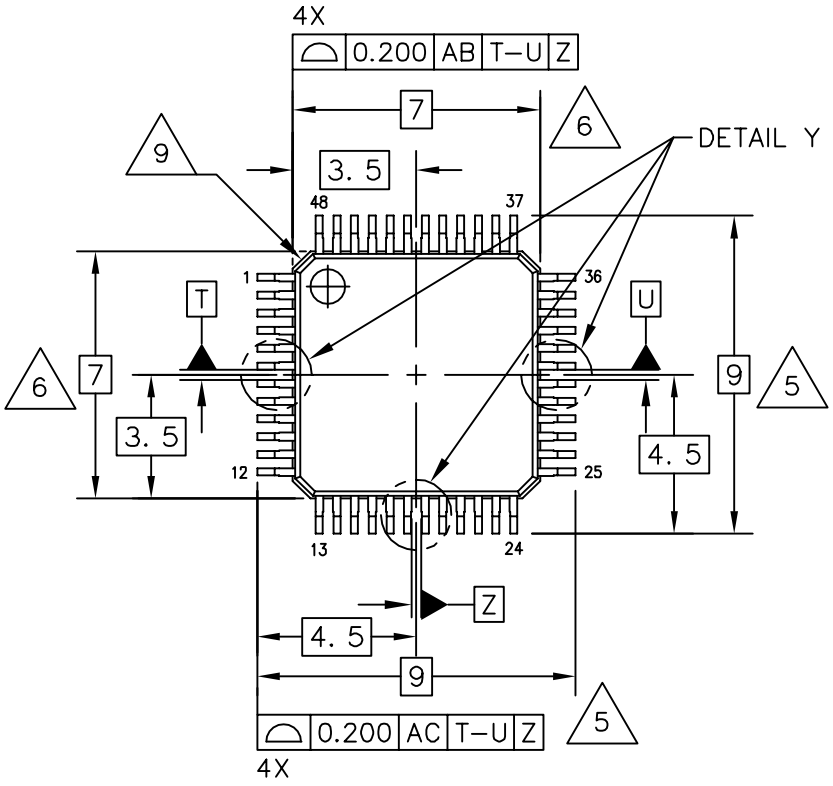
17.1 Introduction

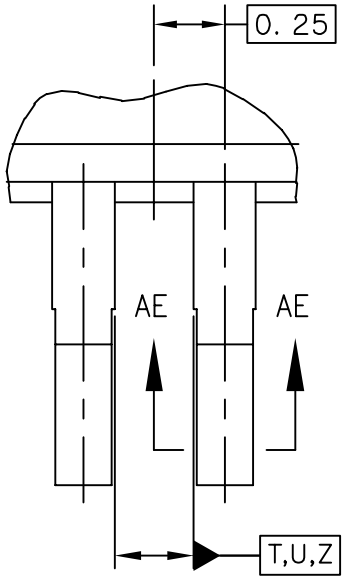
Development support systems in the HCS08 include the background debug controller (BDC) and the on-chip debug module (DBG). The BDC provides a single-wire debug interface to the target MCU that provides a convenient interface for programming the on-chip FLASH and other nonvolatile memories. The BDC is also the primary debug interface for development and allows non-intrusive access to memory data and traditional debug features such as CPU register modify, breakpoints, and single instruction trace commands.

In the HCS08 Family, address and data bus signals are not available on external pins (not even in test modes). Debug is done through commands fed into the target MCU via the single-wire background debug interface. The debug module provides a means to selectively trigger and capture bus information so an external development system can reconstruct what happened inside the MCU on a cycle-by-cycle basis without having external access to the address and data signals.

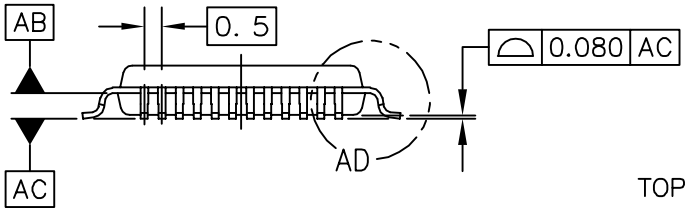
17.1.1 Forcing Active Background

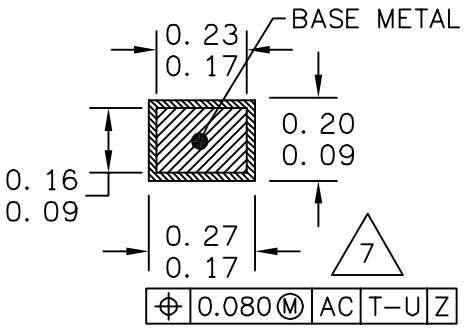
The method for forcing active background mode depends on the specific HCS08 derivative. For the MC9S08DZ128 Series, you can force active background after a power-on reset by holding the BKGD pin low as the device exits the reset condition. You can also force active background by driving BKGD low immediately after a serial background command that writes a one to the BDFR bit in the SBDFR register. Other causes of reset including an external pin reset or an internally generated error reset ignore the state of the BKGD pin and reset into normal user mode. If no debug pod is connected to the BKGD pin, the MCU will always reset into normal operating mode.



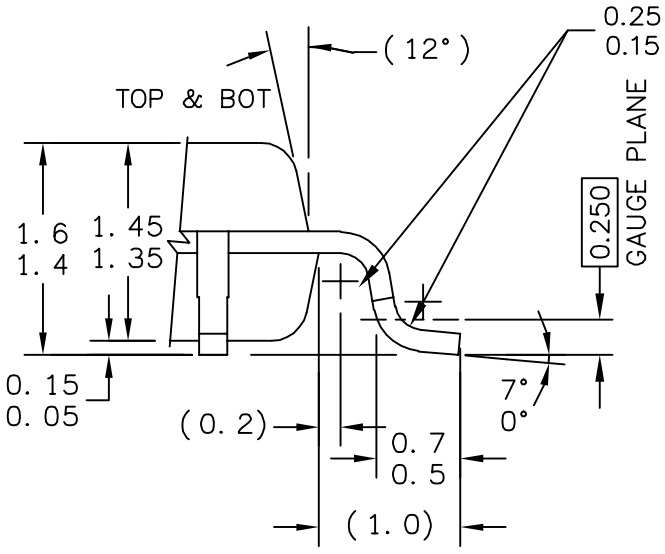


DETAIL Y





SECTION AE–AE



DETAIL AD

	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: LQFP, 48 LEAD, 0.50 PITCH (7.0 X 7.0 X 1.4)	DOCUMENT NO: 98ASH00962A	REV: G
	CASE NUMBER: 932-03	14 APR 2005
	STANDARD: JEDEC MS-026-BBC	