

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	8MHz
Connectivity	SCI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	39
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VQFN Exposed Pad
Supplier Device Package	48-QFN-EP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08re16cfde

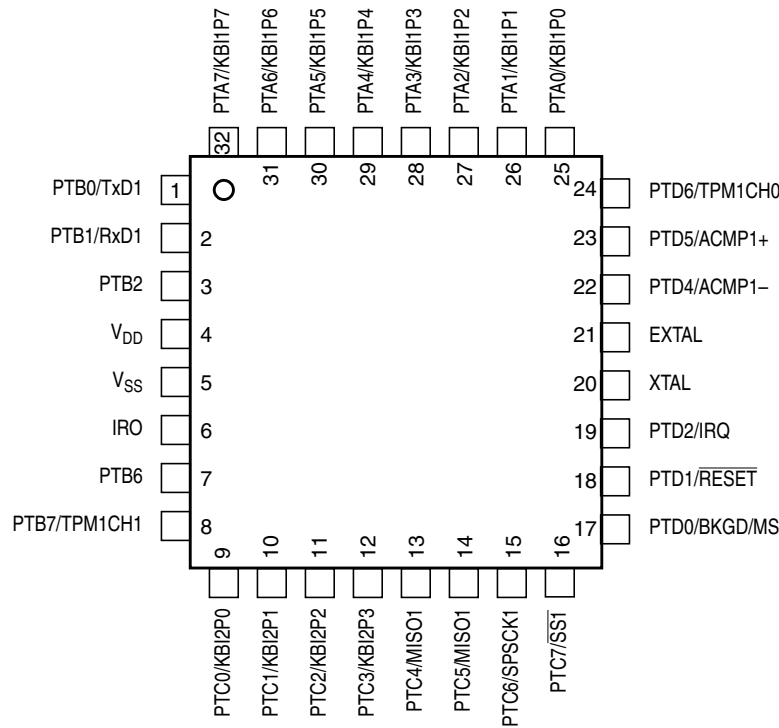


Figure 2-2. MC9S08RC/RD/RE/RG in 32-Pin LQFP Package

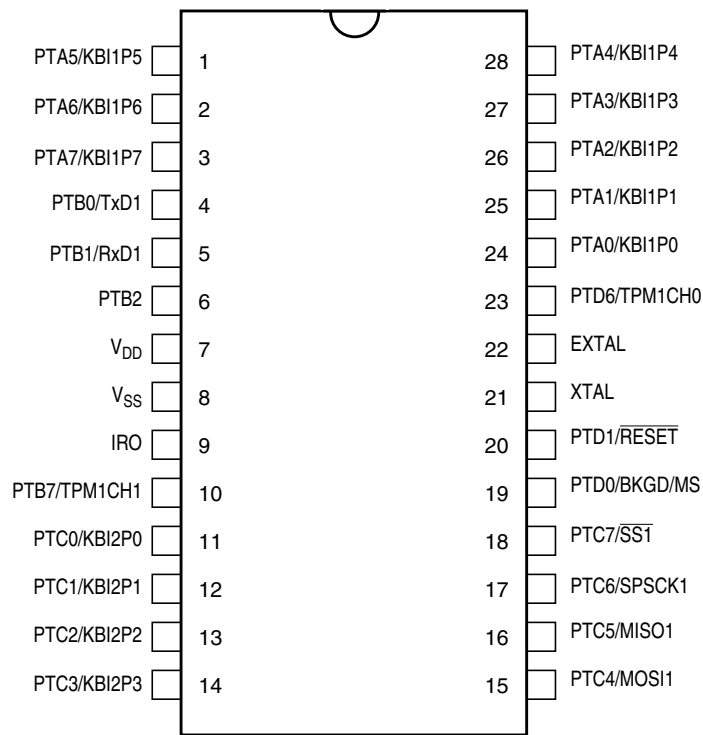


Figure 2-3. MC9S08RC/RD/RE/RG in 28-Pin SOIC Package and 28-Pin PDIP Package

3.6.4 Active BDM Enabled in Stop Mode

Entry into the active background mode from run mode is enabled if the ENBDM bit in BDCSCR is set. This register is described in the Development Support chapter of this data sheet. If ENBDM is set when the CPU executes a STOP instruction, the system clocks to the background debug logic remain active when the MCU enters stop mode so background debug communication is still possible. In addition, the voltage regulator does not enter its low-power standby state but maintains full internal regulation. The MCU cannot enter either stop1 mode or stop2 mode if ENBDM is set.

Most background commands are not available in stop mode. The memory-access-with-status commands do not allow memory access, but they report an error indicating that the MCU is in either stop or wait mode. The BACKGROUND command can be used to wake the MCU from stop and enter active background mode if the ENBDM bit is set. After active background mode is entered, all background commands are available. Table 3-2 summarizes the behavior of the MCU in stop when entry into the active background mode is enabled.

Table 3-2. BDM Enabled Stop Mode Behavior

Mode	PDC	PPDC	CPU, Digital Peripherals, FLASH	RAM	OSC	ACMP	Regulator	I/O Pins	RTI
Stop3	Don't care	Don't care	Standby	Standby	On	Standby	On	States held	Optionally on

3.6.5 LVD Reset Enabled

The LVD system is capable of generating either an interrupt or a reset when the supply voltage drops below the LVD voltage. If the LVD reset is enabled in stop by setting the LVDRE bit in SPMSC1 when the CPU executes a STOP instruction, then the voltage regulator remains active during stop mode. If the user attempts to enter either stop1 or stop2 with the LVD reset enabled (LVDRE = 1) the MCU will instead enter stop3. Table 3-3 summarizes the behavior of the MCU in stop when LVD reset is enabled.

Table 3-3. LVD Enabled Stop Mode Behavior

Mode	PDC	PPDC	CPU, Digital Peripherals, FLASH	RAM	OSC	ACMP	Regulator	I/O Pins	RTI
Stop3	Don't care	Don't care	Standby	Standby	On	Standby	On	States held	Optionally on

3.6.6 On-Chip Peripheral Modules in Stop Mode

When the MCU enters any stop mode, system clocks to the internal peripheral modules are stopped. Even in the exception case (ENBDM = 1), where clocks are kept alive to the background debug logic, clocks to the peripheral systems are halted to reduce power consumption.

4.6.5 FLASH Status Register (FSTAT)

Bits 3, 1, and 0 always read 0 and writes have no meaning or effect. The remaining five bits are status bits that can be read at any time. Writes to these bits have special meanings that are discussed in the bit descriptions.

	7	6	5	4	3	2	1	0
R	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
W								
Reset	1	1	0	0	0	0	0	0


 = Unimplemented or Reserved

Figure 4-9. FLASH Status Register (FSTAT)

Table 4-12. FSTAT Field Descriptions

Field	Description
7 FCBEF	FLASH Command Buffer Empty Flag — The FCBEF bit is used to launch commands. It also indicates that the command buffer is empty so that a new command sequence can be executed when performing burst programming. The FCBEF bit is cleared by writing a 1 to it or when a burst program command is transferred to the array for programming. Only burst program commands can be buffered. 0 Command buffer is full (not ready for additional commands). 1 A new burst program command may be written to the command buffer.
6 FCCF	FLASH Command Complete Flag — FCCF is set automatically when the command buffer is empty and no command is being processed. FCCF is cleared automatically when a new command is started (by writing 1 to FCBEF to register a command). Writing to FCCF has no meaning or effect. 0 Command in progress 1 All commands complete
5 FPVIOL	Protection Violation Flag — FPVIOL is set automatically when FCBEF is cleared to register a command that attempts to erase or program a location in a protected block (the erroneous command is ignored). FPVIOL is cleared by writing a 1 to FPVIOL. 0 No protection violation. 1 An attempt was made to erase or program a protected location.
4 FACCERR	Access Error Flag — FACCERR is set automatically when the proper command sequence is not followed exactly (the erroneous command is ignored), if a program or erase operation is attempted before the FCDIV register has been initialized, or if the MCU enters stop while a command was in progress. For a more detailed discussion of the exact actions that are considered access errors, see Section 4.4.5, "Access Errors." FACCERR is cleared by writing a 1 to FACCERR. Writing a 0 to FACCERR has no meaning or effect. 0 No access error. 1 An access error has occurred.
2 FBLANK	FLASH Verified as All Blank (erased) Flag — FBLANK is set automatically at the conclusion of a blank check command if the entire FLASH array was verified to be erased. FBLANK is cleared by clearing FCBEF to write a new valid command. Writing to FBLANK has no meaning or effect. 0 After a blank check command is completed and FCCF = 1, FBLANK = 0 indicates the FLASH array is not completely erased. 1 After a blank check command is completed and FCCF = 1, FBLANK = 1 indicates the FLASH array is completely erased (all \$FF).



as an edge on the IRQ pin or a timer-overflow event. The debug module can also generate an SWI under certain circumstances.

If an event occurs in an enabled interrupt source, an associated read-only status flag will become set. The CPU will not respond until and unless the local interrupt enable is a logic 1 to enable the interrupt. The I bit in the CCR is logic 0 to allow interrupts. The global interrupt mask (I bit) in the CCR is initially set after reset, which masks (prevents) all maskable interrupt sources. The user program initializes the stack pointer and performs other system setup before clearing the I bit to allow the CPU to respond to interrupts.

When the CPU receives a qualified interrupt request, it completes the current instruction before responding to the interrupt. The interrupt sequence uses the same cycle-by-cycle sequence as the SWI instruction and consists of:

- Saving the CPU registers on the stack
- Setting the I bit in the CCR to mask further interrupts
- Fetching the interrupt vector for the highest-priority interrupt that is currently pending
- Filling the instruction queue with the first three bytes of program information starting from the address fetched from the interrupt vector locations

While the CPU is responding to the interrupt, the I bit is automatically set to avoid the possibility of another interrupt interrupting the ISR itself (this is called nesting of interrupts). Normally, the I bit is restored to 0 when the CCR is restored from the value stacked on entry to the ISR. In rare cases, the I bit may be cleared inside an ISR (after clearing the status flag that generated the interrupt) so that other interrupts can be serviced without waiting for the first service routine to finish. This practice is not recommended for anyone other than the most experienced programmers because it can lead to subtle program errors that are difficult to debug.

The interrupt service routine ends with a return-from-interrupt (RTI) instruction that restores the CCR, A, X, and PC registers to their pre-interrupt values by reading the previously saved information off the stack.

NOTE

For compatibility with the M68HC08 Family, the H register is not automatically saved and restored. It is good programming practice to push H onto the stack at the start of the interrupt service routine (ISR) and restore it just before the RTI that is used to return from the ISR.

If two or more interrupts are pending when the I bit is cleared, the highest priority source is serviced first (see Table 5-1).

5.5.1 Interrupt Stack Frame

Figure 5-1 shows the contents and organization of a stack frame. Before the interrupt, the stack pointer (SP) points at the next available byte location on the stack. The current values of CPU registers are stored on the stack starting with the low-order byte of the program counter (PCL) and ending with the CCR. After stacking, the SP points at the next available location on the stack, which is the address that is one less than the address where the CCR was saved. The PC value that is stacked is the address of the instruction in the main program that would have executed next if the interrupt had not occurred.

5.6.1 Power-On Reset Operation

When power is initially applied to the MCU, or when the supply voltage drops below the V_{POR} level, the POR circuit will cause a reset condition. As the supply voltage rises, the LVD circuit will hold the chip in reset until the supply has risen above the V_{LVD} level. Both the POR bit and the LVD bit in SRS are set following a POR.

5.6.2 LVD Reset Operation

The LVD can be configured to generate a reset upon detection of a low voltage condition. This is done by setting LVDRE to 1. LVDRE is a write-once bit that is set following a POR and is unaffected by other resets. When LVDRE = 1, setting the SAFE bit has no effect. After an LVD reset has occurred, the LVD system will hold the MCU in reset until the supply voltage is above the V_{LVD} level. The LVD bit in the SRS register is set following either an LVD reset or POR.

5.6.3 LVD Interrupt and Safe State Operation

When the voltage on the supply pin V_{DD} drops below V_{LVD} and the LVD circuit is configured for interrupt operation (LVDIE is set and LVDRE is clear), an LVD interrupt will occur. The LVD trip point is set above the minimum voltage at which the MCU can reliably operate, but the supply voltage may still be dropping. It is recommended that the user place the MCU in the safe state as soon as possible following a LVD interrupt. For systems where the supply voltage may drop so rapidly that the MCU may not have time to service the LVD interrupt and enter the safe state, it is recommended that the LVD be configured to generate a reset. The safe state is entered by executing a STOP instruction with the SAFE bit in the system power management status and control 1 (SPMSC1) register set while in a low voltage condition (LVDF = 1).

After the LVD interrupt has occurred, the user may configure the system to block all interrupts, resets, or wakeups by writing a 1 to the SAFE bit. While SAFE = 1 and V_{DD} is below V_{REARM} all interrupts, resets, and wakeups are blocked. After V_{DD} is above V_{REARM} , the SAFE bit is ignored (the SAFE bit will still read a 1). After setting the SAFE bit, the MCU must be put into either the stop3 or stop2 mode before the supply voltage drops below the minimum operating voltage of the MCU. The supply voltage may now drop to a level just above the POR trip point and then restored to a level above V_{REARM} and the MCU state (in the case of stop3) and RAM contents will be preserved. When the supply voltage has been restored, interrupts, resets, and wakeups are then unblocked. When the MCU has recovered from stop mode, the SAFE bit should be cleared.

5.6.4 Low-Voltage Warning (LVW)

The LVD system has a low-voltage warning flag to indicate to the user that the supply voltage is approaching, but is still above, the low-voltage detect voltage. The LVW does not have an interrupt associated with it. However, the FLASH memory cannot be reliably programmed or erased below the V_{LVW} level, so the status of the LVWF bit in the system power management status and control 2 (SPMSC2) register must be checked before initiating any FLASH program or erase operation.

The TPM can be configured to use PTD6 as either an input capture, output compare, PWM, or external clock input pin. Refer to the Chapter 6, “Parallel Input/Output,” for more information about using PTD6 as a timer pin.

6.3.5 Port E

Port E	Bit 7	6	5	4	3	2	1	Bit 0
MCU Pin:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0

Figure 6-5. Port E Pin Names

Port E is an 8-bit general-purpose I/O port.

Port E pins are available as general-purpose I/O pins controlled by the port E data (PTED), data direction (PTEDD), and pullup enable (PTEPE) registers. Refer to Section 6.4, “Parallel I/O Controls,” for more information about general-purpose I/O control.

6.4 Parallel I/O Controls

Provided no on-chip peripheral is controlling a port pin, the pins operate as general-purpose I/O pins that are accessed and controlled by a data register (PTxD), a data direction register (PTxDD), and a pullup enable register (PTxPE) where x is A, B, C, D, or E.

Reads of the data register return the pin value (if PTxDDn = 0) or the contents of the port data register (if PTxDDn = 1). Writes to the port data register are latched into the port register whether the pin is controlled by an on-chip peripheral or the pin is configured as an input. If the corresponding pin is not controlled by a peripheral and is configured as an output, this level will be driven out the port pin.

6.4.1 Data Direction Control

The data direction control bits determine whether the pin output driver is enabled, and they control what is read for port data register reads. Each port pin has a data direction control bit. When PTxDDn = 0, the corresponding pin is an input and reads of PTxD return the pin value. When PTxDDn = 1, the corresponding pin is an output and reads of PTxD return the last value written to the port data register. When a peripheral module or system function is in control of a port pin, the data direction control still controls what is returned for reads of the port data register, even though the peripheral system has overriding control of the actual pin direction.

For the MC9S08RC/RD/RE/RG MCU, reads of PTD0/BKGD/MS and PTD1/ $\overline{\text{RESET}}$ will return the value on the output pin.

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

6.4.2 Internal Pullup Control

An internal pullup device can be enabled for each port pin that is configured as an input ($PTxDDn = 0$). The pullup device is available for a peripheral module to use, provided the peripheral is enabled and is an input function as long as the $PTxDDn = 0$.

NOTE

The voltage measured on the pulled up PTA0 pin will be less than V_{DD} . The internal gates connected to this pin are pulled all the way to V_{DD} . All other pins with enabled pullup resistors will have an unloaded measurement of V_{DD} .

6.5 Stop Modes

Depending on the stop mode, I/O functions differently as the result of executing a STOP instruction. An explanation of I/O behavior for the various stop modes follows:

- When the MCU enters stop1 mode, all internal registers, including general-purpose I/O control and data registers, are powered down. All of the general-purpose I/O pins assume their reset state: output buffers and pullups turned off. Upon exit from stop1, all I/O must be initialized as if the MCU had been reset.
- When the MCU enters stop2 mode, the internal registers are powered down as in stop1 but the I/O pin states are latched and held. For example, a port pin that is an output driving low continues to function as an output driving low even though its associated data direction and output data registers are powered down internally. Upon exit from stop2, the pins continue to hold their states until a 1 is written to the PPDACK bit. To avoid discontinuity in the pin state following exit from stop2, the user must restore the port control and data registers to the values they held before entering stop2. These values can be stored in RAM before entering stop2 because the RAM is maintained during stop2.
- In stop3 mode, all I/O is maintained because internal logic circuitry stays powered up. Upon recovery, normal I/O function is available to the user.

6.6 Parallel I/O Registers and Control Bits

This section provides information about all registers and control bits associated with the parallel I/O ports.

Refer to tables in the Memory chapter for the absolute address assignments for all parallel I/O registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

6.6.1 Port A Registers (PTAD, PTAPE, and PTADD)

Port A pins used as general-purpose I/O pins are controlled by the port A data (PTAD), data direction (PTADD), and pullup enable (PTAPE) registers.

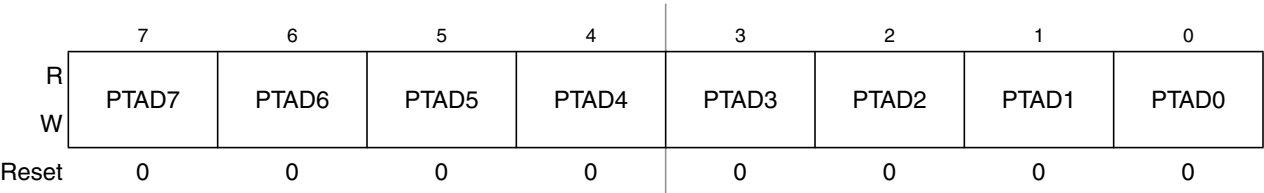


Figure 6-6. Port A Data Register (PTAD)

Table 6-1. PTAD Field Descriptions

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

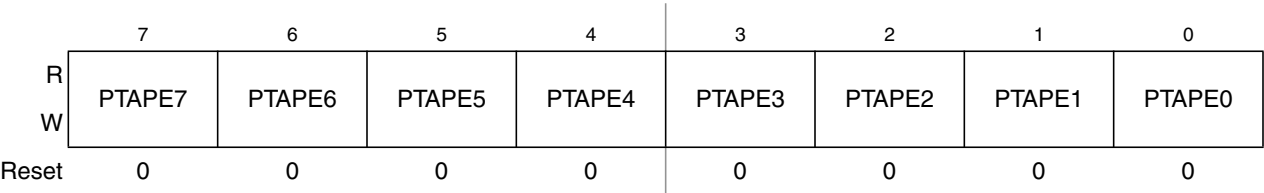


Figure 6-7. Pullup Enable for Port A (PTAPE)

Table 6-2. PTAPE Field Descriptions

Field	Description
7:0 PTAPE[7:0]	<p>Pullup Enable for Port A Bits — For port A pins that are inputs, these read/write control bits determine whether internal pullup devices are enabled provided the corresponding PTADDn is a logic 0. For port A pins that are configured as outputs, these bits are ignored and the internal pullup devices are disabled. When any of bits 7 through 4 of port A are enabled as KBI inputs and are configured to detect rising edges/high levels, the pullup enable bits enable pulldown rather than pullup devices.</p> <p>0 Internal pullup device disabled.</p> <p>1 Internal pullup device enabled.</p>

6.6.3 Port C Registers (PTCD, PTCPE, and PTCDD)

Port C pins used as general-purpose I/O pins are controlled by the port C data (PTCD), data direction (PTCDD), and pullup enable (PTCPE) registers.

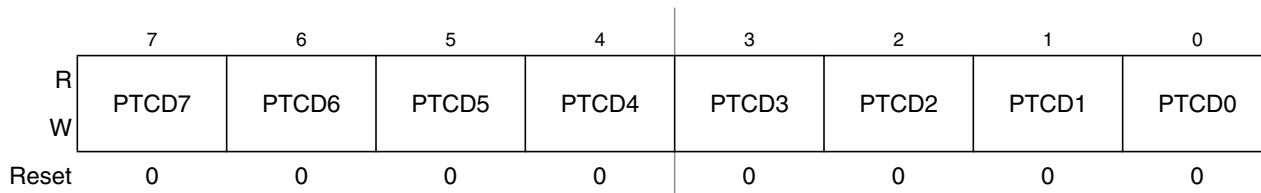


Figure 6-12. Port C Data Register (PTCD)

Table 6-7. PTCD Field Descriptions

Field	Description
7:0 PTCD[7:0]	Port C Data Register Bits — For port C pins that are inputs, reads return the logic level on the pin. For port C 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 C pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTCD 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.

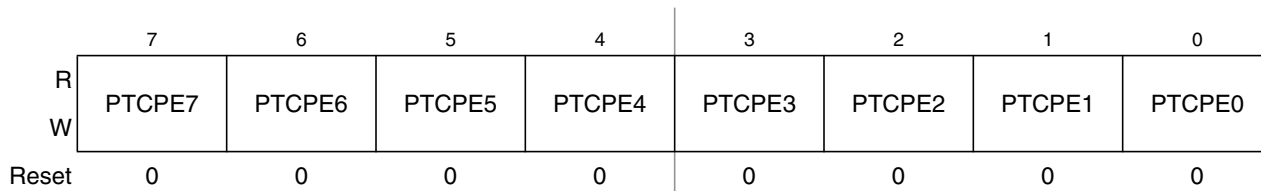


Figure 6-13. Pullup Enable for Port C (PTCPE)

Table 6-8. PTCPE Field Descriptions

Field	Description
7:0 PTCPE[7:0]	Pullup Enable for Port C Bits — For port C pins that are inputs, these read/write control bits determine whether internal pullup devices are enabled. For port C pins that are configured as outputs, these bits are ignored and the internal pullup devices are disabled. 0 Internal pullup device disabled. 1 Internal pullup device enabled.

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.

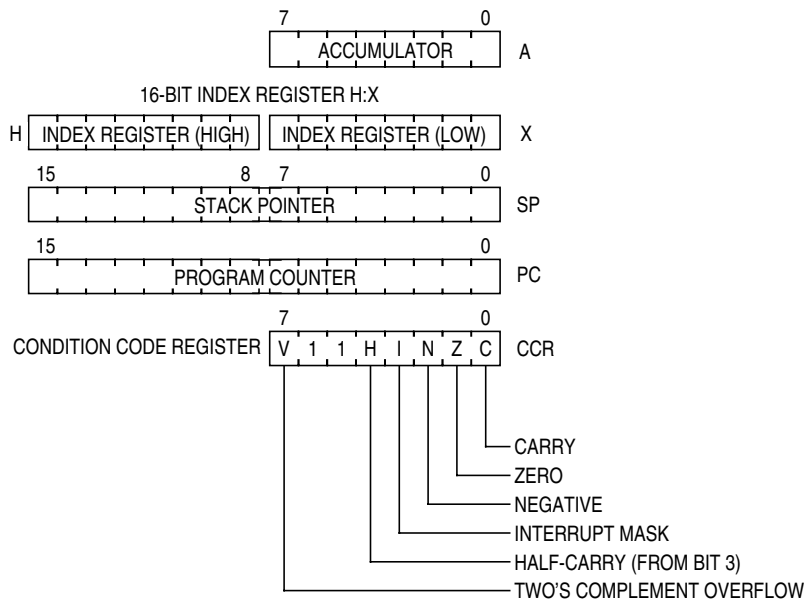


Figure 7-1. CPU Registers

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 7-1. CCR Register Field Descriptions

Field	Description
7 V	Two's Complement Overflow Flag — The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag. 0 No overflow 1 Overflow
4 H	Half-Carry Flag — The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C condition code bits to automatically add a correction value to the result from a previous ADD or ADC on BCD operands to correct the result to a valid BCD value. 0 No carry between bits 3 and 4 1 Carry between bits 3 and 4
3 I	Interrupt Mask Bit — When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the first instruction of the interrupt service routine is executed. Interrupts are not recognized at the instruction boundary after any instruction that clears I (CLI or TAP). This ensures that the next instruction after a CLI or TAP will always be executed without the possibility of an intervening interrupt, provided I was set. 0 Interrupts enabled 1 Interrupts disabled
2 N	Negative Flag — The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result. Simply loading or storing an 8-bit or 16-bit value causes N to be set if the most significant bit of the loaded or stored value was 1. 0 Non-negative result 1 Negative result
1 Z	Zero Flag — The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of 0x00 or 0x0000. Simply loading or storing an 8-bit or 16-bit value causes Z to be set if the loaded or stored value was all 0s. 0 Non-zero result 1 Zero result
0 C	Carry/Borrow Flag — The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag. 0 No carry out of bit 7 1 Carry out of bit 7

7.3 Addressing Modes

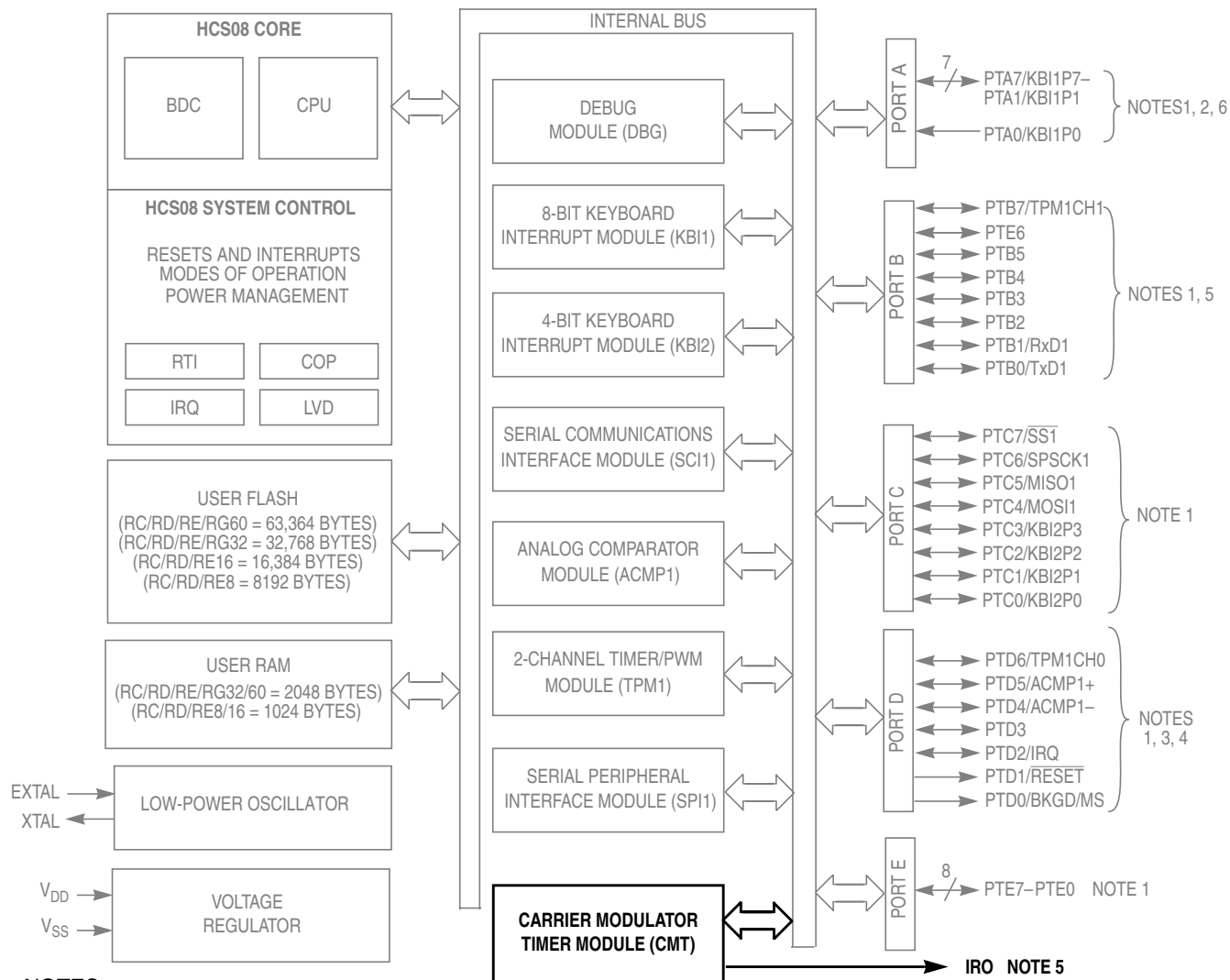
Addressing modes define the way the CPU accesses operands and data. In the HCS08, all memory, status and control registers, and input/output (I/O) ports share a single 64-Kbyte linear address space so a 16-bit binary address can uniquely identify any memory location. This arrangement means that the same instructions that access variables in RAM can also be used to access I/O and control registers or nonvolatile program space.

Some instructions use more than one addressing mode. For instance, move instructions use one addressing mode to specify the source operand and a second addressing mode to specify the destination address. Instructions such as BRCLR, BRSET, CBEQ, and DBNZ use one addressing mode to specify the location

Chapter 8

Carrier Modulator Timer (S08CMTV1)

8.1 Introduction



NOTES:

1. Port pins are software configurable with pullup device if input port
2. PTA0 does not have a clamp diode to VDD. PTA0 should not be driven above VDD. Also, PTA0 does not pullup to VDD when internal pullup is enabled.
3. IRQ pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1)
4. The RESET pin contains integrated pullup device enabled if reset enabled (RSTPE = 1)
5. High current drive
6. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown enabled when KBI is enabled (KBIPEn = 1) and rising edge is selected (KBEDGn = 1).

Figure 8-1. MC9S08RC/RD/RE/RG Block Diagram

The duty cycle of the carrier signal is controlled by varying the ratio of high time to low + high time. As the input clock period is fixed, the duty cycle resolution will be proportional to the number of counts required to generate the desired carrier period.

$$\text{Duty Cycle} = \frac{\text{Highcount}}{\text{Highcount} + \text{Lowcount}} \quad \text{Eqn. 8-4}$$

8.5.2 Modulator

The modulator has three main modes of operation:

- Gate the carrier onto the modulator output (time mode)
- Control the logic level of the modulator output (baseband mode)
- Count carrier periods and instruct the carrier generator to alternate between two carrier frequencies whenever a modulation period (mark + space counts) expires (FSK mode)

The modulator includes a 17-bit down counter with underflow detection. The counter is loaded from the 16-bit modulation mark period buffer registers, CMTCMD1 and CMTCMD2. The most significant bit is loaded with a logic zero and serves as a sign bit. When the counter holds a positive value, the modulator gate is open and the carrier signal is driven to the transmitter block.

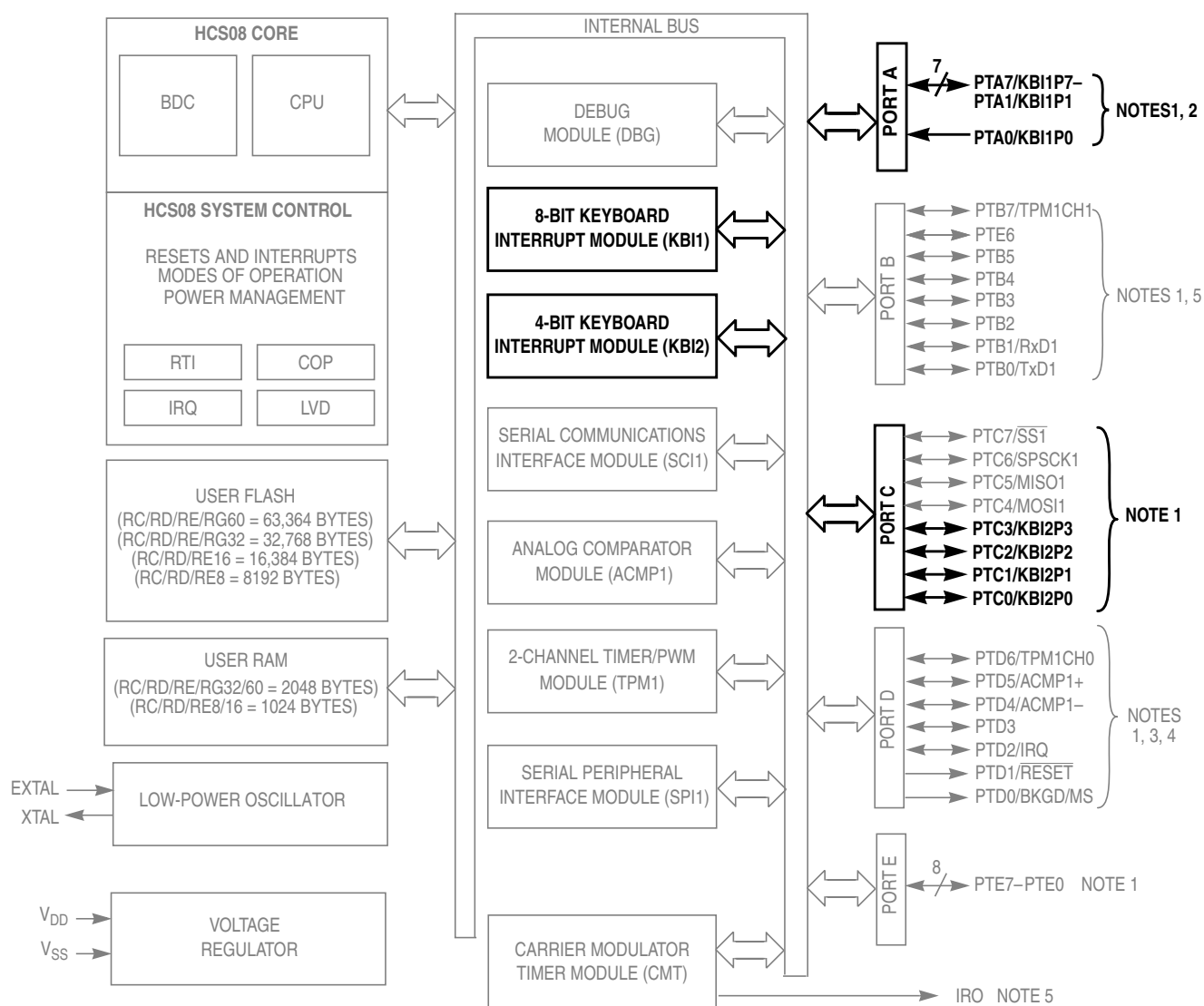
When the counter underflows, the modulator gate is closed and a 16-bit comparator is enabled that compares the logical complement of the value of the down-counter with the contents of the modulation space period register (which has been loaded from the registers CMTCMD3 and CMTCMD4).

When a match is obtained the cycle repeats by opening the modulator gate, reloading the counter with the contents of CMTCMD1 and CMTCMD2, and reloading the modulation space period register with the contents of CMTCMD3 and CMTCMD4.

If the contents of the modulation space period register are all zeroes, the match will be immediate and no space period will be generated (for instance, for FSK protocols that require successive bursts of different frequencies).

The MCGEN bit in the CMTMSC register must be set to enable the modulator timer.

Keyboard Interrupt (S08KBIV1)



NOTES:

7. Port pins are software configurable with pullup device if input port
8. PTA0 does not have a clamp diode to V_{DD}. PTA0 should not be driven above V_{DD}. Also, PTA0 does not pullup to V_{DD} when internal pullup is enabled.
9. IRQ pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1)
10. The RESET pin contains integrated pullup device enabled if reset enabled (RSTPE = 1)
11. High current drive
12. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown enabled when KBI is enabled (KBIPEn = 1) and rising edge is selected (KBEDGn = 1).

Figure 9-1. MC9S08RC/RD/RE/RG Block Diagram

10.7.4 Timer Channel n Status and Control Register (TPM1CnSC)

TPM1CnSC contains the channel interrupt status flag and control bits that are used to configure the interrupt enable, channel configuration, and pin function.

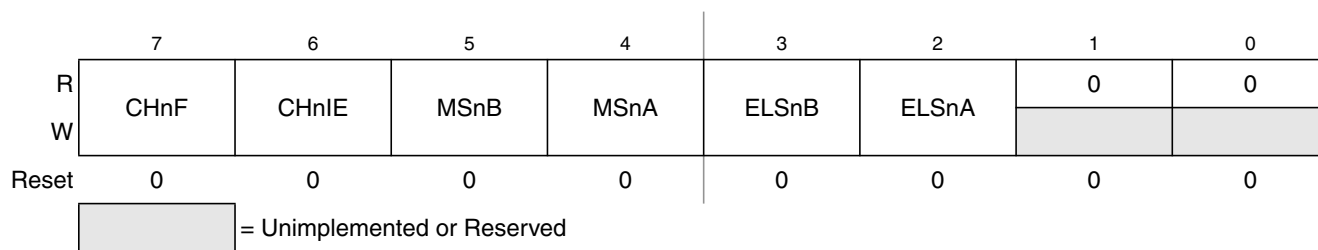


Figure 10-10. Timer Channel n Status and Control Register (TPM1CnSC)

Table 10-4. TPM1CnSC Register Field Descriptions

Field	Description
7 CHnF	<p>Channel n Flag — When channel n is configured for input capture, this flag bit is set when an active edge occurs on the channel n pin. When channel n is an output compare or edge-aligned PWM channel, CHnF is set when the value in the TPM counter registers matches the value in the TPM channel n value registers. This flag is seldom used with center-aligned PWMs because it is set every time the counter matches the channel value register, which correspond to both edges of the active duty cycle period.</p> <p>A corresponding interrupt is requested when CHnF is set and interrupts are enabled (CHnIE = 1). Clear CHnF by reading TPM1CnSC while CHnF is set and then writing a 0 to CHnF. If another interrupt request occurs before the clearing sequence is complete, the sequence is reset so CHnF would remain set after the clear sequence was completed for the earlier CHnF. This is done so a CHnF interrupt request cannot be lost by clearing a previous CHnF. Reset clears CHnF. Writing a 1 to CHnF has no effect.</p> <p>0 No input capture or output compare event occurred on channel n 1 Input capture or output compare event occurred on channel n</p>
6 CHnIE	<p>Channel n Interrupt Enable — This read/write bit enables interrupts from channel n. Reset clears CHnIE.</p> <p>0 Channel n interrupt requests disabled (use software polling) 1 Channel n interrupt requests enabled</p>
5 MSnB	<p>Mode Select B for TPM Channel n — When CPWMS = 0, MSnB = 1 configures TPM channel n for edge-aligned PWM mode. For a summary of channel mode and setup controls, refer to Table 10-5.</p>
4 MSnA	<p>Mode Select A for TPM Channel n — When CPWMS = 0 and MSnB = 0, MSnA configures TPM channel n for input capture mode or output compare mode. Refer to Table 10-5 for a summary of channel mode and setup controls.</p>
3:2 ELSn[B:A]	<p>Edge/Level Select Bits — Depending on the operating mode for the timer channel as set by CPWMS:MSnB:MSnA and shown in Table 10-5, these bits select the polarity of the input edge that triggers an input capture event, select the level that will be driven in response to an output compare match, or select the polarity of the PWM output.</p> <p>Setting ELSnB:ELSnA to 0:0 configures the related timer pin as a general-purpose I/O pin unrelated to any timer channel functions. This function is typically used to temporarily disable an input capture channel or to make the timer pin available as a general-purpose I/O pin when the associated timer channel is set up as a software timer that does not require the use of a pin. This is also the setting required for channel 0 when the TPM1CH0 pin is used as an external clock input.</p>

Table 12-7. SCI1C3 Register Field Descriptions (continued)

Field	Description
3 ORIE	Overrun Interrupt Enable — This bit enables the overrun flag (OR) to generate hardware interrupt requests. 0 OR interrupts disabled (use polling). 1 Hardware interrupt requested when OR = 1.
2 NEIE	Noise Error Interrupt Enable — This bit enables the noise flag (NF) to generate hardware interrupt requests. 0 NF interrupts disabled (use polling). 1 Hardware interrupt requested when NF = 1.
1 FEIE	Framing Error Interrupt Enable — This bit enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled (use polling). 1 Hardware interrupt requested when FE = 1.
0 PEIE	Parity Error Interrupt Enable — This bit enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled (use polling). 1 Hardware interrupt requested when PF = 1.

12.2.7 SCI Data Register (SCI1D)

This register is actually two separate registers. Reads return the contents of the read-only receive data buffer and writes go to the write-only transmit data buffer. Reads and writes of this register are also involved in the automatic flag clearing mechanisms for the SCI status flags.

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Figure 12-10. SCI Data Register (SCI1D)

Figure 15-4 shows the host receiving a logic 0 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target MCU. The host initiates the bit time but the target HCS08 finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 BDC clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 cycles after starting the bit time.

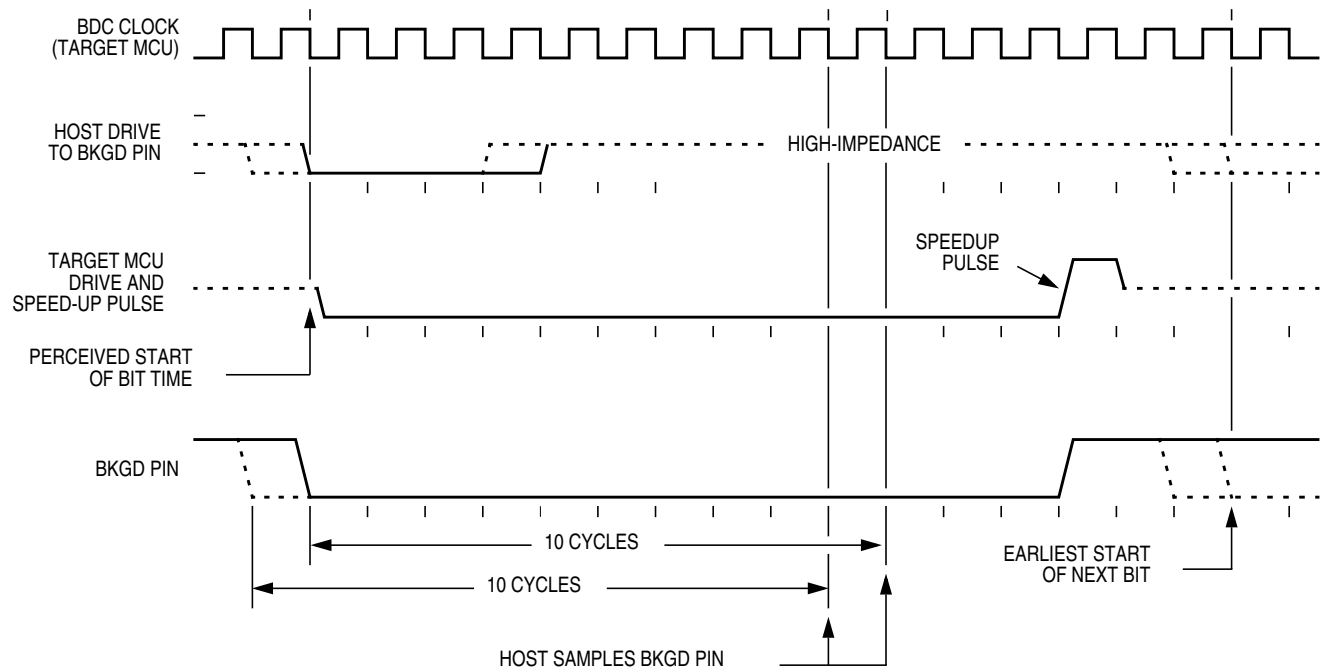


Figure 15-4. BDM Target-to-Host Serial Bit Timing (Logic 0)

