



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	40MHz
Connectivity	I ² C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	42-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08gt32cb

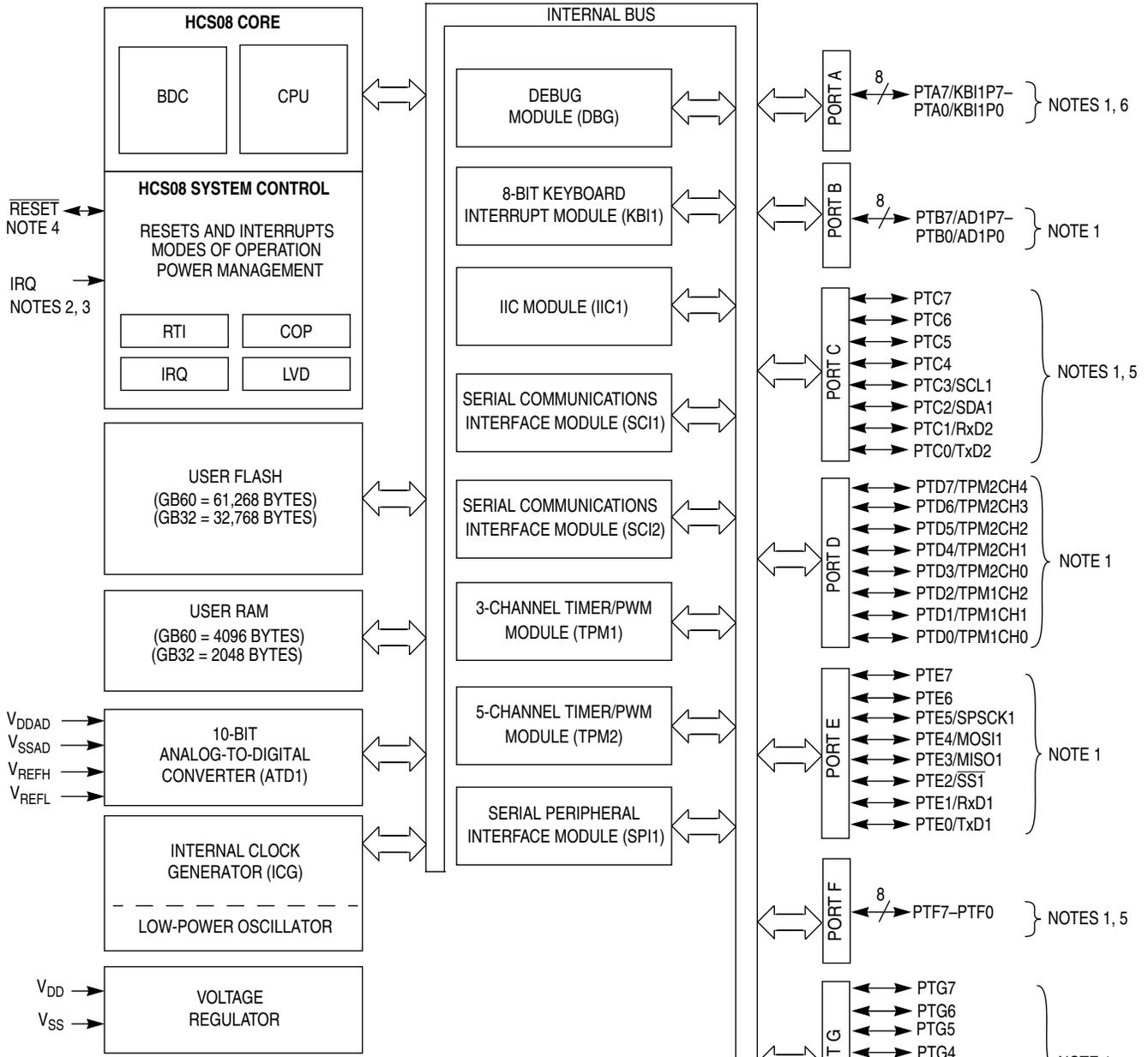
Section Number	Title	Page
14.2.1.2	ATD Reference Pins — V_{REFH} , V_{REFL}	223
14.2.1.3	ATD Supply Pins — V_{DDAD} , V_{SSAD}	223
14.3	Functional Description	223
14.3.1	Mode Control	223
14.3.2	Sample and Hold	224
14.3.3	Analog Input Multiplexer	226
14.3.4	ATD Module Accuracy Definitions	226
14.4	Resets	229
14.5	Interrupts	229
14.6	ATD Registers and Control Bits	229
14.6.1	ATD Control (ATDC)	230
14.6.2	ATD Status and Control (ATD1SC)	232
14.6.3	ATD Result Data (ATD1RH, ATD1RL)	234
14.6.4	ATD Pin Enable (ATD1PE)	234

Chapter 15 Development Support

15.1	Introduction	235
15.2	Features	236
15.3	Background Debug Controller (BDC)	237
15.3.1	BKGD Pin Description	237
15.3.2	Communication Details	238
15.3.3	BDC Commands	242
15.3.4	BDC Hardware Breakpoint	244
15.4	On-Chip Debug System (DBG)	245
15.4.1	Comparators A and B	245
15.4.2	Bus Capture Information and FIFO Operation	245
15.4.3	Change-of-Flow Information	246
15.4.4	Tag vs. Force Breakpoints and Triggers	246
15.4.5	Trigger Modes	247
15.4.6	Hardware Breakpoints	249
15.5	Registers and Control Bits	249
15.5.1	BDC Registers and Control Bits	249
15.5.1.1	BDC Status and Control Register (BDCSCR)	250
15.5.1.2	BDC Breakpoint Match Register (BDCBKPT)	251
15.5.2	System Background Debug Force Reset Register (SBDFR)	251
15.5.3	DBG Registers and Control Bits	252
15.5.3.1	Debug Comparator A High Register (DBGCAH)	252
15.5.3.2	Debug Comparator A Low Register (DBGCAL)	252
15.5.3.3	Debug Comparator B High Register (DBGCBH)	252
15.5.3.4	Debug Comparator B Low Register (DBGCBL)	252

1.3 MCU Block Diagrams

These block diagrams show the structure of the MC9S08GB/GT MCUs.



- NOTES:
1. Port pins are software configurable with pullup device if input port.
 2. Pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1).
 3. IRQ does not have a clamp diode to V_{DD}. IRQ should not be driven above V_{DD}.
 4. Pin contains integrated pullup device.
 5. High current drive
 6. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown available when KBI enabled (KBIPn = 1).

Figure 1-1. MC9S08GBxx Block Diagram

2.3.1 Power

V_{DD} and V_{SS} are the primary power supply pins for the MCU. This voltage source supplies power to all I/O buffer circuitry and to an internal voltage regulator. The internal voltage regulator provides regulated lower-voltage source to the CPU and other internal circuitry of the MCU.

Typically, application systems have two separate capacitors across the power pins. In this case, there should be a bulk electrolytic capacitor, such as a 10- μ F tantalum capacitor, to provide bulk charge storage for the overall system and a 0.1- μ F ceramic bypass capacitor located as close to the MCU power pins as practical to suppress high-frequency noise.

V_{DDAD} and V_{SSAD} are the analog power supply pins for the MCU. This voltage source supplies power to the ATD. A 0.1- μ F ceramic bypass capacitor should be located as close to the MCU power pins as practical to suppress high-frequency noise.

2.3.2 Oscillator

Out of reset, the MCU uses an internally generated clock (self-clocked mode — $f_{\text{Self_reset}}$) that is approximately equivalent to an 8-MHz crystal rate. This frequency source is used during reset startup and can be enabled as the clock source for stop recovery to avoid the need for a long crystal startup delay. This MCU also contains a trimmable internal clock generator (ICG) module that can be used to run the MCU. For more information on the ICG, see [Chapter 7, “Internal Clock Generator \(ICG\) Module.”](#)

The oscillator in this MCU is a Pierce oscillator that can accommodate a crystal or ceramic resonator in either of two frequency ranges selected by the RANGE bit in the ICGC1 register. Rather than a crystal or ceramic resonator, an external oscillator can be connected to the EXTAL input pin, and the XTAL output pin can be used as general I/O.

Refer to [Figure 2-5](#) for the following discussion. R_S (when used) and R_F should be low-inductance resistors such as carbon composition resistors. Wire-wound resistors, and some metal film resistors, have too much inductance. C1 and C2 normally should be high-quality ceramic capacitors that are specifically designed for high-frequency applications.

R_F is used to provide a bias path to keep the EXTAL input in its linear range during crystal startup and its value is not generally critical. Typical systems use 1 M Ω to 10 M Ω . Higher values are sensitive to humidity and lower values reduce gain and (in extreme cases) could prevent startup.

C1 and C2 are typically in the 5-pF to 25-pF range and are chosen to match the requirements of a specific crystal or resonator. Be sure to take into account printed circuit board (PCB) capacitance and MCU pin capacitance when sizing C1 and C2. The crystal manufacturer typically specifies a load capacitance which is the series combination of C1 and C2 which are usually the same size. As a first-order approximation, use 10 pF as an estimate of combined pin and PCB capacitance for each oscillator pin (EXTAL and XTAL).

2.3.3 Reset

$\overline{\text{RESET}}$ is a dedicated pin with a pullup device built in. It has input hysteresis, a high current output driver, and no output slew rate control. Internal power-on reset and low-voltage reset circuitry typically make external reset circuitry unnecessary. This pin is normally connected to the standard 6-pin background

Exit from stop2 is performed by asserting either of the wake-up pins: $\overline{\text{RESET}}$ or IRQ, or by an RTI interrupt. IRQ is always an active low input when the MCU is in stop2, regardless of how it was configured before entering stop2.

Upon wake-up from stop2 mode, the MCU will start up as from a power-on reset (POR) except pin states remain latched. The CPU will take the reset vector. The system and all peripherals will be in their default reset states and must be initialized.

After waking up from stop2, the PPDF bit in SPMSC2 is set. This flag may be used to direct user code to go to a stop2 recovery routine. PPDF remains set and the I/O pin states remain latched until a 1 is written to PPDACK in SPMSC2.

To maintain I/O state for pins that were configured as general-purpose I/O, the user must restore the contents of the I/O port registers, which have been saved in RAM, to the port registers before writing to the PPDACK bit. If the port registers are not restored from RAM before writing to PPDACK, then the register bits will assume their reset states when the I/O pin latches are opened and the I/O pins will switch to their reset states.

For pins that were configured as peripheral I/O, the user must reconfigure the peripheral module that interfaces to the pin before writing to the PPDACK bit. If the peripheral module is not enabled before writing to PPDACK, the pins will be controlled by their associated port control registers when the I/O latches are opened.

3.6.3 Stop3 Mode

Upon entering the stop3 mode, all of the clocks in the MCU, including the oscillator itself, are halted. The ICG is turned off, the ATD is disabled, and the voltage regulator is put in standby. The states of all of the internal registers and logic, as well as the RAM content, are maintained. The I/O pin states are not latched at the pin as in stop2. Instead they are maintained by virtue of the states of the internal logic driving the pins being maintained.

Exit from stop3 is performed by asserting $\overline{\text{RESET}}$, an asynchronous interrupt pin, or through the real-time interrupt. The asynchronous interrupt pins are the IRQ or KBI pins.

If stop3 is exited by means of the $\overline{\text{RESET}}$ pin, then the MCU will be reset and operation will resume after taking the reset vector. Exit by means of an asynchronous interrupt or the real-time interrupt will result in the MCU taking the appropriate interrupt vector.

A separate self-clocked source (≈ 1 kHz) for the real-time interrupt allows a wakeup from stop2 or stop3 mode with no external components. When RTIS2:RTIS1:RTIS0 = 0:0:0, the real-time interrupt function and this 1-kHz source are disabled. Power consumption is lower when the 1-kHz source is disabled, but in that case the real-time interrupt cannot wake the MCU from stop.

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 [Chapter 15, “Development Support,”](#) section 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,

4.4.3 Program and Erase Command Execution

The steps for executing any of the commands are listed below. The FCDIV register must be initialized and any error flags cleared before beginning command execution. The command execution steps are:

1. Write a data value to an address in the FLASH array. The address and data information from this write is latched into the FLASH interface. This write is a required first step in any command sequence. For erase and blank check commands, the value of the data is not important. For page erase commands, the address may be any address in the 512-byte page of FLASH to be erased. For mass erase and blank check commands, the address can be any address in the FLASH memory. Whole pages of 512 bytes are the smallest blocks of FLASH that may be erased. In the 60K version, there are two instances where the size of a block that is accessible to the user is less than 512 bytes: the first page following RAM, and the first page following the high page registers. These pages are overlapped by the RAM and high page registers, respectively.

NOTE

Do not program any byte in the FLASH more than once after a successful erase operation. Reprogramming bits in a byte which is already programmed is not allowed without first erasing the page in which the byte resides or mass erasing the entire FLASH memory. Programming without first erasing may disturb data stored in the FLASH.

2. Write the command code for the desired command to FCMD. The five valid commands are blank check (\$05), byte program (\$20), burst program (\$25), page erase (\$40), and mass erase (\$41). The command code is latched into the command buffer.
3. Write a 1 to the FCBEF bit in FSTAT to clear FCBEF and launch the command (including its address and data information).

A partial command sequence can be aborted manually by writing a 0 to FCBEF any time after the write to the memory array and before writing the 1 that clears FCBEF and launches the complete command. Aborting a command in this way sets the FACCERR access error flag which must be cleared before starting a new command.

A strictly monitored procedure must be adhered to, or the command will not be accepted. This minimizes the possibility of any unintended change to the FLASH memory contents. The command complete flag (FCCF) indicates when a command is complete. The command sequence must be completed by clearing FCBEF to launch the command. [Figure 4-2](#) is a flowchart for executing all of the commands except for burst programming. The FCDIV register must be initialized before using any FLASH commands. This must be done only once following a reset.

Table 4-6. FLASH Clock Divider Settings

f_{Bus}	PRDIV8 (Binary)	DIV5:DIV0 (Decimal)	f_{CLK}	Program/Erase Timing Pulse (5 μs Min, 6.7 μs Max)
20 MHz	1	12	192.3 kHz	5.2 μs
10 MHz	0	49	200 kHz	5 μs
8 MHz	0	39	200 kHz	5 μs
4 MHz	0	19	200 kHz	5 μs
2 MHz	0	9	200 kHz	5 μs
1 MHz	0	4	200 kHz	5 μs
200 kHz	0	0	200 kHz	5 μs
150 kHz	0	0	150 kHz	6.7 μs

4.6.2 FLASH Options Register (FOPT and NVOPT)

During reset, the contents of the nonvolatile location NVOPT are copied from FLASH into FOPT. Bits 5 through 2 are not used and always read 0. This register may be read at any time, but writes have no meaning or effect. To change the value in this register, erase and reprogram the NVOPT location in FLASH memory as usual and then issue a new MCU reset.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
Write:								
Reset:	This register is loaded from nonvolatile location NVOPT during reset.							
	 = Unimplemented or Reserved							

Figure 4-5. FLASH Options Register (FOPT)

KEYEN — Backdoor Key Mechanism Enable

When this bit is 0, the backdoor key mechanism cannot be used to disengage security. The backdoor key mechanism is accessible only from user (secured) firmware. BDM commands cannot be used to write key comparison values that would unlock the backdoor key. For more detailed information about the backdoor key mechanism, refer to [Section 4.5, “Security.”](#)

1 = If user firmware writes an 8-byte value that matches the nonvolatile backdoor key (NVBACKKEY through NVBACKKEY+7, in that order), security is temporarily disengaged until the next MCU reset.

0 = No backdoor key access allowed.

FNORED — Vector Redirection Disable

When this bit is 1, vector redirection is disabled.

1 = Vector redirection disabled.

0 = Vector redirection enabled.

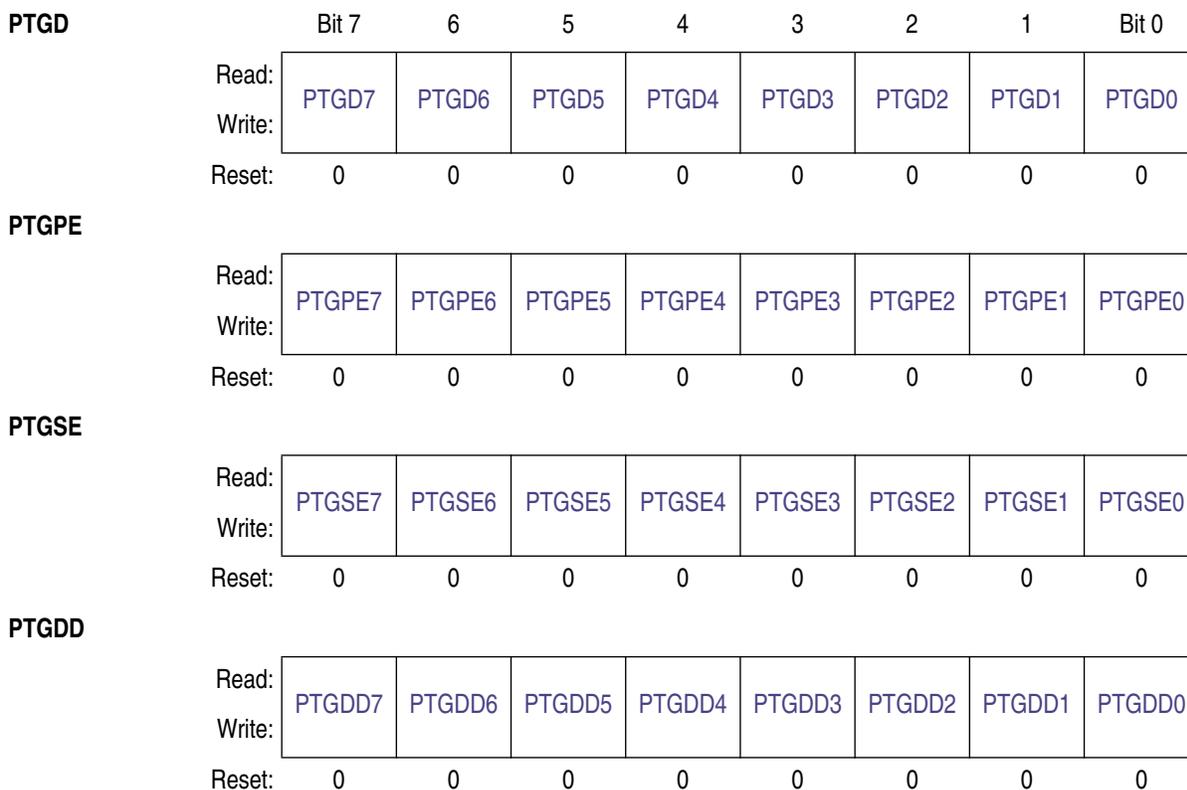


Figure 6-15. Port G Registers

PTGD_n — Port PTG Data Register Bit *n* (*n* = 0–7)

For port G pins that are inputs, reads return the logic level on the pin. For port G 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 G pins that are configured as outputs, the logic level is driven out the corresponding MCU pin.

Reset forces PTGD 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.

PTGPE_n — Pullup Enable for Port G Bit *n* (*n* = 0–7)

For port G pins that are inputs, these read/write control bits determine whether internal pullup devices are enabled. For port G pins that are configured as outputs, these bits are ignored and the internal pullup devices are disabled.

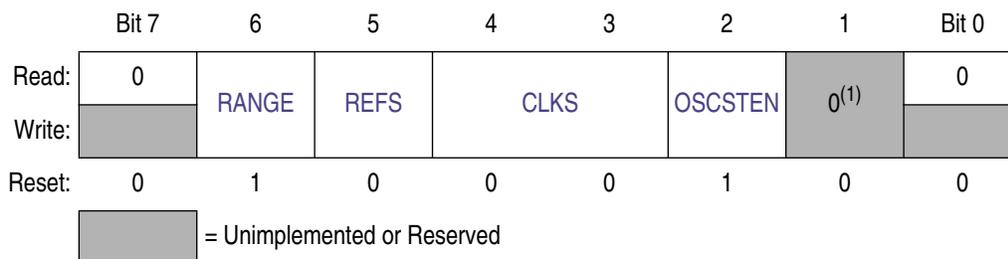
- 1 = Internal pullup device enabled.
- 0 = Internal pullup device disabled.

PTGSE_n — Slew Rate Control Enable for Port G Bit *n* (*n* = 0–7)

For port G pins that are outputs, these read/write control bits determine whether the slew rate controlled outputs are enabled. For port G pins that are configured as inputs, these bits are ignored.

- 1 = Slew rate control enabled.
- 0 = Slew rate control disabled.

7.5.1 ICG Control Register 1 (ICGC1)



1. This bit is reserved for Freescale Semiconductor internal use only. Any write operations to this register should write a 0 to this bit.

Figure 7-12. ICG Control Register 1 (ICGC1)

RANGE — Frequency Range Select

The RANGE bit controls the oscillator, reference divider, and FLL loop prescaler multiplication factor (P). It selects one of two reference frequency ranges for the ICG. The RANGE bit is write-once after a reset. The RANGE bit only has an effect in FLL engaged external and FLL bypassed external modes.

1 = Oscillator configured for high frequency range. FLL loop prescale factor P is 1.

0 = Oscillator configured for low frequency range. FLL loop prescale factor P is 64.

REFS — External Reference Select

The REFS bit controls the external reference clock source for ICGERCLK. The REFS bit is write-once after a reset.

1 = Oscillator using crystal or resonator requested.

0 = External clock requested.

CLKS — Clock Mode Select

The CLKS bits control the clock mode according to [Figure 7-13](#). If FLL bypassed external is requested, it will not be selected until ERCS = 1. If the ICG enters off mode, the CLKS bits will remain unchanged. Writes to the CLKS bits will not take effect if a previous write is not complete.

Figure 7-13. CLKS Clock Select

CLKS[1:0]	Clock Mode
00	Self-clocked
01	FLL engaged, internal reference
10	FLL bypassed, external reference
11	FLL engaged, external reference

The CLKS bits are writable at any time, unless the first write after a reset was CLKS = 0X, the CLKS bits cannot be written to 1X until after the next reset (because the EXTAL pin was not reserved).

8.4 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 of an operand for a test and then use relative addressing mode to specify the branch destination address when the tested condition is true. For BRCLR, BRSET, CBEQ, and DBNZ, the addressing mode listed in the instruction set tables is the addressing mode needed to access the operand to be tested, and relative addressing mode is implied for the branch destination.

8.4.1 Inherent Addressing Mode (INH)

In this addressing mode, operands needed to complete the instruction (if any) are located within CPU registers so the CPU does not need to access memory to get any operands.

8.4.2 Relative Addressing Mode (REL)

Relative addressing mode is used to specify the destination location for branch instructions. A signed 8-bit offset value is located in the memory location immediately following the opcode. During execution, if the branch condition is true, the signed offset is sign-extended to a 16-bit value and is added to the current contents of the program counter, which causes program execution to continue at the branch destination address.

8.4.3 Immediate Addressing Mode (IMM)

In immediate addressing mode, the operand needed to complete the instruction is included in the object code immediately following the instruction opcode in memory. In the case of a 16-bit immediate operand, the high-order byte is located in the next memory location after the opcode, and the low-order byte is located in the next memory location after that.

8.4.4 Direct Addressing Mode (DIR)

In direct addressing mode, the instruction includes the low-order eight bits of an address in the direct page (\$0000–\$00FF). During execution a 16-bit address is formed by concatenating an implied \$00 for the high-order half of the address and the direct address from the instruction to get the 16-bit address where the desired operand is located. This is faster and more memory efficient than specifying a complete 16-bit address for the operand.

10.3 TPM Block Diagram

The TPM uses one input/output (I/O) pin per channel, TPMxCHn where x is the TPM number (for example, 1 or 2) and n is the channel number (for example, 0–4). The TPM shares its I/O pins with general-purpose I/O port pins (refer to the [Pins and Connections](#) chapter for more information).

Figure 10-2 shows the structure of a TPM. Some MCUs include more than one TPM, with various numbers of channels.

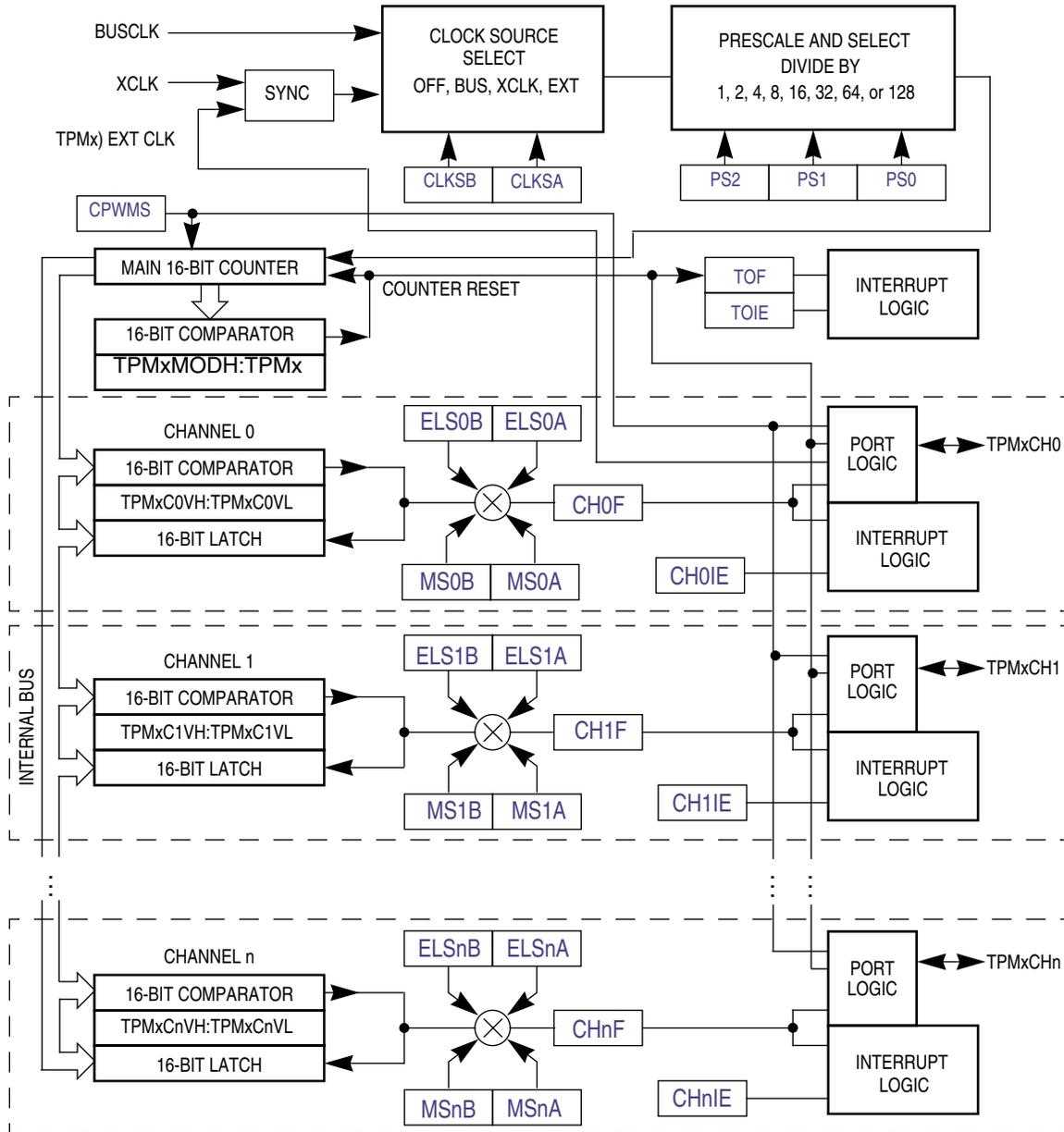


Figure 10-2. TPM Block Diagram

The central component of the TPM is the 16-bit counter that can operate as a free-running counter, a modulo counter, or an up-/down-counter when the TPM is configured for center-aligned PWM. The TPM

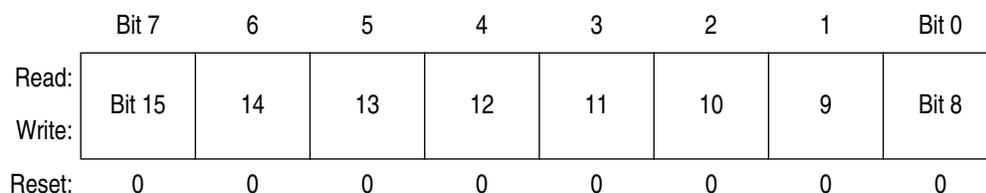


Figure 10-11. Timer x Channel Value Register High (TPMxCnVH)

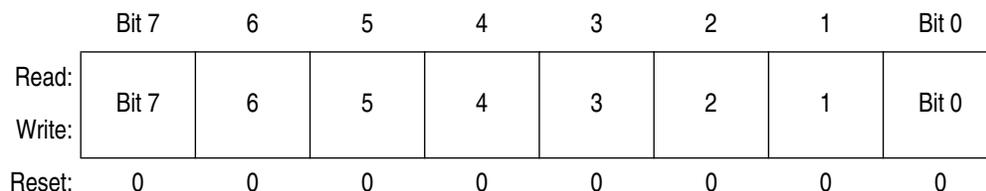


Figure 10-12. Timer x Channel Value Register Low (TPMxCnVL)

In input capture mode, reading either byte (TPMxCnVH or TPMxCnVL) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. This latching mechanism also resets (becomes unlatched) when the TPMxCnSC register is written.

In output compare or PWM modes, writing to either byte (TPMxCnVH or TPMxCnVL) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent 16-bit value into the timer channel value registers. This latching mechanism may be manually reset by writing to the TPMxCnSC register.

This latching mechanism allows coherent 16-bit writes in either order, which is friendly to various compiler implementations.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF) status flag is set. If RDRF was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the SCI receiver is double-buffered, the program has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full ($RDRF = 1$), it gets the data from the receive data register by reading SCIxD. The RDRF flag is cleared automatically by a 2-step sequence (which is normally satisfied in the course of the user's program that handles receive data). Refer to [Section 11.7, "Interrupts and Status Flags,"](#) for more details about flag clearing.

11.6.2 Data Sampling Technique

The SCI receiver uses a $16\times$ baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD1 serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The $16\times$ baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for that bit, the noise flag (NF) will be set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is re-synchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is filled with three logic 1 samples so that a new start bit can be detected almost immediately.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

11.6.3 Receiver Wakeup Operation

Receiver wakeup is a hardware mechanism that allows an SCI receiver to ignore the characters in a message that is intended for a different SCI receiver. In such a system, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up (RWU) control bit in SCIxC2. When $RWU = 1$, it

RE — Receiver Enable

When the SCI receiver is off, the RxD1 pin reverts to being a general-purpose port I/O pin.

- 1 = Receiver on.
- 0 = Receiver off.

RWU — Receiver Wakeup Control

This bit can be written to 1 to place the SCI receiver in a standby state where it waits for automatic hardware detection of a selected wakeup condition. The wakeup condition is either an idle line between messages (WAKE = 0, idle-line wakeup), or a logic 1 in the most significant data bit in a character (WAKE = 1, address-mark wakeup). Application software sets RWU and (normally) a selected hardware condition automatically clears RWU. Refer to [Section 11.6.3, “Receiver Wakeup Operation,”](#) for more details.

- 1 = SCI receiver in standby waiting for wakeup condition.
- 0 = Normal SCI receiver operation.

SBK — Send Break

Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 bit times of logic 0 are queued as long as SBK = 1. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. Refer to [Section 11.5.2, “Send Break and Queued Idle,”](#) for more details.

- 1 = Queue break character(s) to be sent.
- 0 = Normal transmitter operation.

11.10.4 SCI x Status Register 1 (SCIxS1)

This register has eight read-only status flags. Writes have no effect. Special software sequences (that do not involve writing to this register) are used to clear these status flags.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write:								
Reset:	1	1	0	0	0	0	0	0

 = Unimplemented or Reserved

Figure 11-9. SCI x Status Register 1 (SCIxS1)

TDRE — Transmit Data Register Empty Flag

TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SCIxS1 with TDRE = 1 and then write to the SCI data register (SCIxD).

- 1 = Transmit data register (buffer) empty.
- 0 = Transmit data register (buffer) full.

The most common uses of the SPI system include connecting simple shift registers for adding input or output ports or connecting small peripheral devices such as serial A/D or D/A converters. Although [Figure 12-2](#) shows a system where data is exchanged between two MCUs, many practical systems involve simpler connections where data is unidirectionally transferred from the master MCU to a slave or from a slave to the master MCU.

12.2.2 SPI Module Block Diagram

[Figure 12-3](#) is a block diagram of the SPI module. The central element of the SPI is the SPI shift register. Data is written to the double-buffered transmitter (write to SPI1D) and gets transferred to the SPI shift register at the start of a data transfer. After shifting in a byte of data, the data is transferred into the double-buffered receiver where it can be read (read from SPI1D). Pin multiplexing logic controls connections between MCU pins and the SPI module.

When the SPI is configured as a master, the clock output is routed to the SPSCCK1 pin, the shifter output is routed to MOSI1, and the shifter input is routed from the MISO1 pin.

When the SPI is configured as a slave, the SPSCCK1 pin is routed to the clock input of the SPI, the shifter output is routed to MISO1, and the shifter input is routed from the MOSI1 pin.

In the external SPI system, simply connect all SPSCCK pins to each other, all MISO pins together, and all MOSI pins together. Peripheral devices often use slightly different names for these pins.

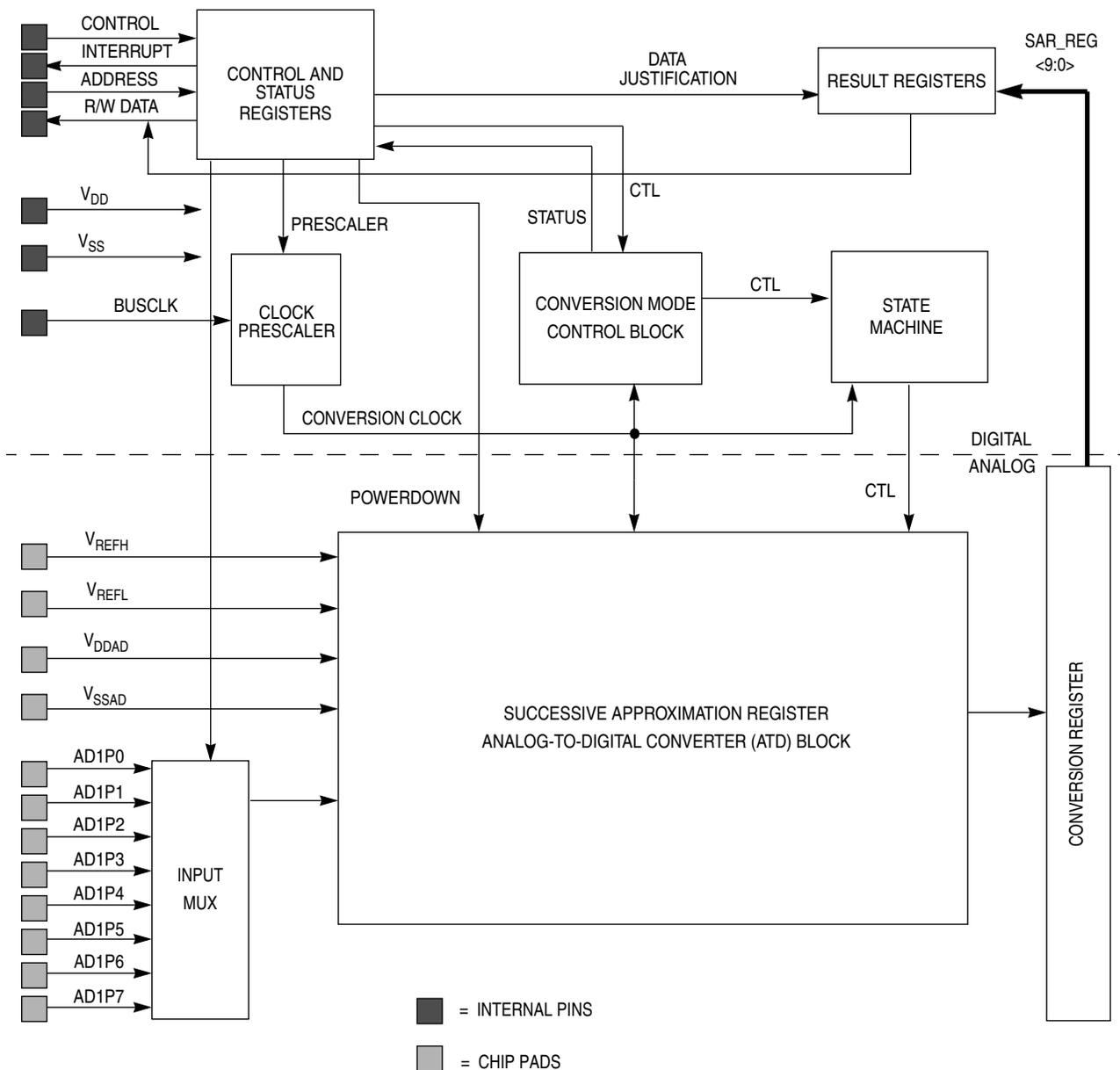
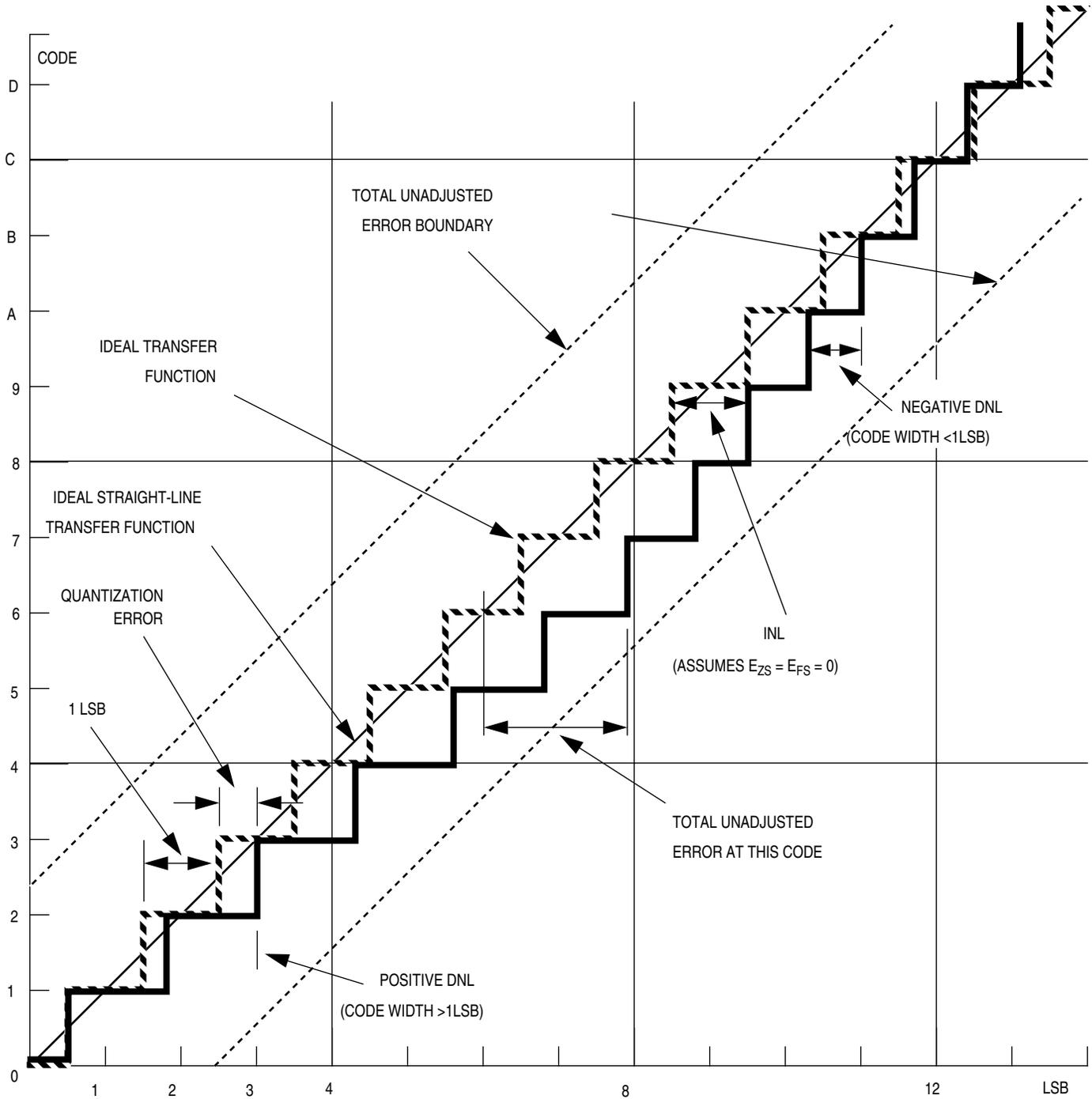


Figure 14-2. ATD Block Diagram

14.2 Signal Description

14.2.1 Overview

The ATD supports eight input channels and requires 4 supply/reference/ground pins. These pins are listed in Table 14-1.



NOTES: Graph is for example only and may not represent actual performance

Figure 14-4. ATD Accuracy Definitions

Figure 15-3 shows the host receiving a logic 1 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 perceived start of the bit time in the target MCU. The host holds the BKGD pin low long enough for the target to recognize it (at least two target BDC cycles). The host must release the low drive before the target MCU drives a brief active-high speedup pulse seven cycles after the perceived start of the bit time. The host should sample the bit level about 10 cycles after it started the bit time.

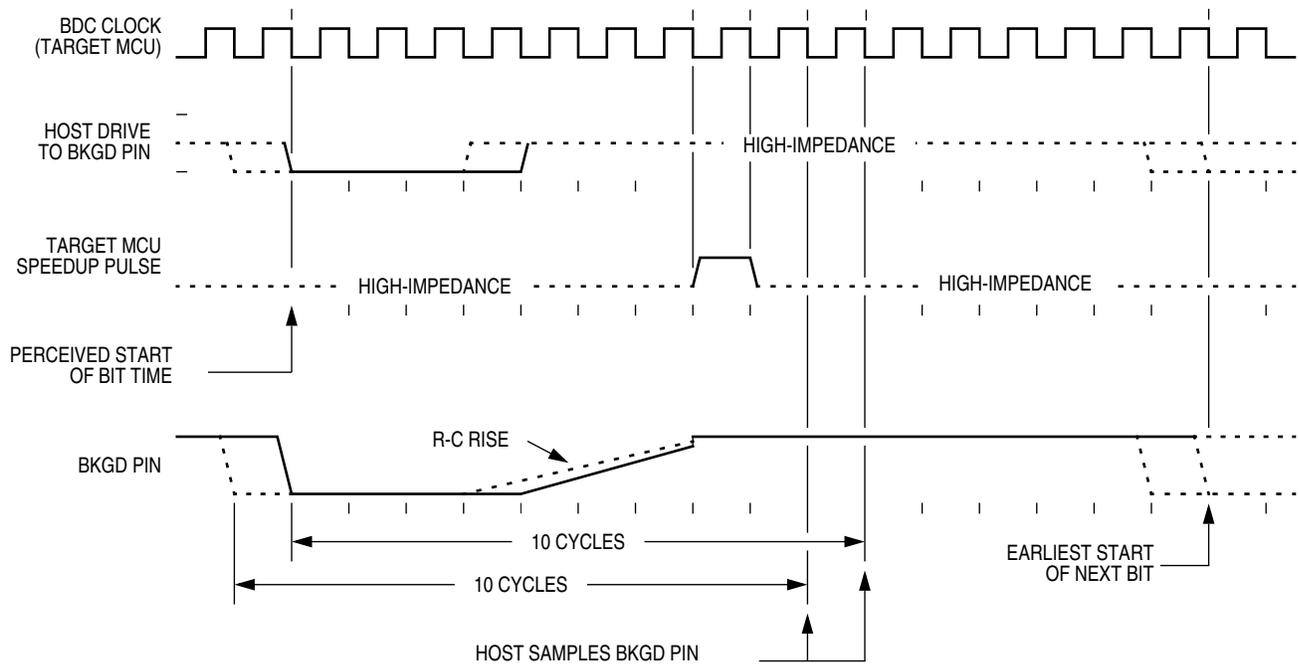


Figure 15-3. BDC Target-to-Host Serial Bit Timing (Logic 1)

A force-type breakpoint waits for the current instruction to finish and then acts upon the breakpoint request. The usual action in response to a breakpoint is to go to active background mode rather than continuing to the next instruction in the user application program.

The tag vs. force terminology is used in two contexts within the debug module. The first context refers to breakpoint requests from the debug module to the CPU. The second refers to match signals from the comparators to the debugger control logic. When a tag-type break request is sent to the CPU, a signal is entered into the instruction queue along with the opcode so that if/when this opcode ever executes, the CPU will effectively replace the tagged opcode with a BGND opcode so the CPU goes to active background mode rather than executing the tagged instruction. When the TRGSEL control bit in the DBGTC register is set to select tag-type operation, the output from comparator A or B is qualified by a block of logic in the debug module that tracks opcodes and only produces a trigger to the debugger if the opcode at the compare address is actually executed. There is separate opcode tracking logic for each comparator so more than one compare event can be tracked through the instruction queue at a time.

15.4.5 Trigger Modes

The trigger mode controls the overall behavior of a debug run. The 4-bit TRG field in the DBGTC register selects one of nine trigger modes. When TRGSEL = 1 in the DBGTC register, the output of the comparator must propagate through an opcode tracking circuit before triggering FIFO actions. The BEGIN bit in DBGTC chooses whether the FIFO begins storing data when the qualified trigger is detected (begin trace), or the FIFO stores data in a circular fashion from the time it is armed until the qualified trigger is detected (end trigger).

A debug run is started by writing a 1 to the ARM bit in the DBGTC register, which sets the ARMF flag and clears the AF and BF flags and the CNT bits in DBGS. A begin-trace debug run ends when the FIFO gets full. An end-trace run ends when the selected trigger event occurs. Any debug run can be stopped manually by writing a 0 to the ARM bit or DBGEN bit in DBGTC.

In all trigger modes except event-only modes, the FIFO stores change-of-flow addresses. In event-only trigger modes, the FIFO stores data in the low-order eight bits of the FIFO.

The BEGIN control bit is ignored in event-only trigger modes and all such debug runs are begin type traces. When TRGSEL = 1 to select opcode fetch triggers, it is not necessary to use R/W in comparisons because opcode tags would only apply to opcode fetches that are always read cycles. It would also be unusual to specify TRGSEL = 1 while using a full mode trigger because the opcode value is normally known at a particular address.

The following trigger mode descriptions only state the primary comparator conditions that lead to a trigger. Either comparator can usually be further qualified with R/W by setting RWAEN (RWBEN) and the corresponding RWA (RWB) value to be matched against R/W. The signal from the comparator with optional R/W qualification is used to request a CPU breakpoint if BRKEN = 1 and TAG determines whether the CPU request will be a tag request or a force request.

ARMF — Arm Flag

While DBGEN = 1, this status bit is a read-only image of the ARM bit in DBGCR. This bit is set by writing 1 to the ARM control bit in DBGCR (while DBGEN = 1) and is automatically cleared at the end of a debug run. A debug run is completed when the FIFO is full (begin trace) or when a trigger event is detected (end trace). A debug run can also be ended manually by writing 0 to the ARM or DBGEN bits in DBGCR.

1 = Debugger armed.

0 = Debugger not armed.

CNT3:CNT2:CNT1:CNT0 — FIFO Valid Count

These bits are cleared at the start of a debug run and indicate the number of words of valid data in the FIFO at the end of a debug run. The value in CNT does not decrement as data is read out of the FIFO. The external debug host is responsible for keeping track of the count as information is read out of the FIFO.

Table 15-3. CNT Status Bits

CNT[3:0]	Valid Words in FIFO
0000	No valid data
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8

Table A-5. Supply Current Characteristics (continued)

Parameter	Symbol	V _{DD} (V)	Typical ¹	Max ²	Temp. (°C)
LVI adder to stop3 (LVDSE = LVDE = 1)		3	70 μA		55 70 85
		2	60 μA		55 70 85

- ¹ Typicals are measured at 25°C. See [Figure A-6](#) through [Figure A-9](#) for typical curves across voltage/temperature.
- ² Values given here are preliminary estimates prior to completing characterization.
- ³ All modules except ATD active, ICG configured for FBE, and does not include any dc loads on port pins
- ⁴ Values are characterized but not tested on every part.
- ⁵ Every unit tested to this parameter. All other values in the Max column are guaranteed by characterization.
- ⁶ Most customers are expected to find that auto-wakeup from stop2 or stop3 can be used instead of the higher current wait mode. Wait mode typical is 560 μA at 3 V and 422 μA at 2V with f_{BUS} = 1 MHz.

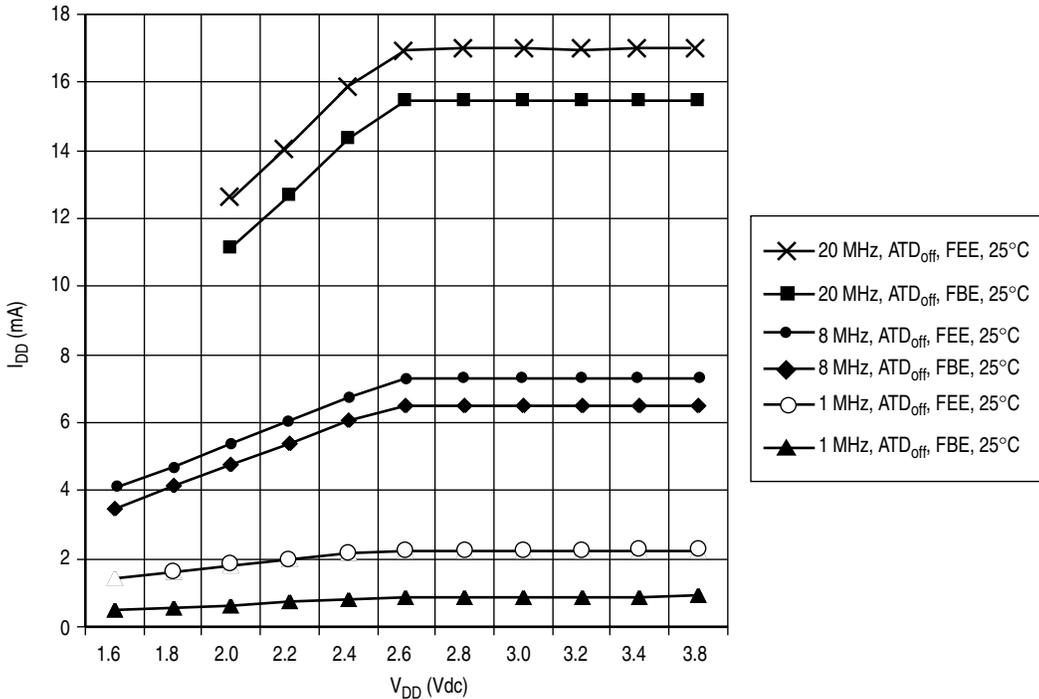


Figure A-6. Typical Run I_{DD} for FBE and FEE Modes, I_{DD} vs V_{DD}