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	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08gt8amfbe

Table 2-2. Signal Properties (continued)

Pin Name	Dir	High Current Pin	Output Slew [1]	Pull-Up[2]	Comments
PTB3/ADP3	I/O	N	SWC	SWC	
PTB4/ADP4	I/O	N	SWC	SWC	Not available on 32-pin pkg
PTB5/ADP5	I/O	N	SWC	SWC	Not available on 32-pin pkg
PTB6/ADP6	I/O	N	SWC	SWC	Not available on 32-pin pkg
PTB7/ADP7	I/O	N	SWC	SWC	Not available on 32-pin pkg
PTC0/TxD2	I/O	Y	SWC	SWC	When pin is configured for SCI function, pin is configured for partial output drive.
PTC1/RxD2	I/O	Y	SWC	SWC	
PTC2/SDA	I/O	Y	SWC	SWC	
PTC3/SCL	I/O	Y	SWC	SWC	
PTC4	I/O	Y	SWC	SWC	Not available on 32-pin pkg
PTC5	I/O	Y	SWC	SWC	Not available on 32-pin or 42-pin pkg
PTC6	I/O	Y	SWC	SWC	Not available on 32-pin or 42-pin pkg
PTC7	I/O	Y	SWC	SWC	Not available on 32-pin, 42- or 44-pin pkg
PTD0/TPM1CLK/TPM1CH0	I/O	N	SWC	SWC	
PTD1/TPM1CH1	I/O	N	SWC	SWC	
PTD2/TPM1CH2	I/O	N	SWC	SWC	Not available on 32-pin, 42- or 44-pin pkg
PTD3/TPM2CLK/TPM2CH0	I/O	N	SWC	SWC	
PTD4/TPM2CH1	I/O	N	SWC	SWC	Not available on 32-pin pkg
PTE0/TxD1	I/O	N	SWC	SWC	
PTE1/RxD1	I/O	N	SWC	SWC	
PTE2/$\overline{SS}$	I/O	N	SWC	SWC	
PTE3/MISO	I/O	N	SWC	SWC	
PTE4/MOSI	I/O	N	SWC	SWC	
PTE5/SPSCK	I/O	N	SWC	SWC	
PTG0/BKGD/MS	O	N	SWC	SWC	Pullup enabled and slew rate disabled when BDM function enabled.
PTG1/XTAL	I/O	N	SWC	SWC	Pullup and slew rate disabled when XTAL pin function.
PTG2/EXTAL	I/O	N	SWC	SWC	Pullup and slew rate disabled when EXTAL pin function.
PTG3	I/O	N	SWC	SWC	Not available on 32-pin, 42-, or 44-pin pkg

1 SWC is software controlled slew rate, the register is associated with the respective port.

2 SWC is software controlled pullup resistor, the register is associated with the respective port.

4.2 Register Addresses and Bit Assignments

The registers in the MC9S08GT16A/GT8A are divided into these three groups:

- Direct-page registers are located in the first 128 locations in the memory map, so they are accessible with efficient direct addressing mode instructions.
- High-page registers are used much less often, so they are located above 0x1800 in the memory map. This leaves more room in the direct page for more frequently used registers and variables.
- The nonvolatile register area consists of a block of 16 locations in FLASH memory at 0xFFB0–0xFFBF.

  Nonvolatile register locations include:

  — Three values which are loaded into working registers at reset

  — An 8-byte backdoor comparison key which optionally allows a user to gain controlled access to secure memory

  Because the nonvolatile register locations are FLASH memory, they must be erased and programmed like other FLASH memory locations.

Direct-page registers can be accessed with efficient direct addressing mode instructions. Bit manipulation instructions can be used to access any bit in any direct-page register. Table 4-2 is a summary of all user-accessible direct-page registers and control bits.

The direct page registers in Table 4-2 can use the more efficient direct addressing mode which only requires the lower byte of the address. Because of this, the lower byte of the address in column one is shown in bold text. In Table 4-3 and Table 4-4 the whole address in column one is shown in bold. In Table 4-2, Table 4-3, and Table 4-4, the register names in column two are shown in bold to set them apart from the bit names to the right. Cells that are not associated with named bits are shaded. A shaded cell with a 0 indicates this unused bit always reads as a 0. Shaded cells with dashes indicate unused or reserved bit locations that could read as 1s or 0s.

Table 4-2. Direct-Page Register Summary (Sheet 2 of 3)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x0028	**SPIC1**	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
0x0029	**SPIC2**	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
0x002A	**SPIBR**	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
0x002B	**SPIS**	SPRF	0	SPTEF	MODF	0	0	0	0
0x002C	Reserved	0	0	0	0	0	0	0	0
0x002D	**SPID**	Bit 7	6	5	4	3	2	1	Bit 0
0x002E	Reserved	0	0	0	0	0	0	0	0
0x002F	Reserved	0	0	0	0	0	0	0	0
0x0030	**TPM1SC**	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0031	**TPM1CNTH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0032	**TPM1CNTL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0033	**TPM1MODH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0034	**TPM1MODL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0035	**TPM1C0SC**	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0036	**TPM1C0VH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0037	**TPM1C0VL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0038	**TPM1C1SC**	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x0039	**TPM1C1VH**	Bit 15	14	13	12	11	10	9	Bit 8
0x003A	**TPM1C1VL**	Bit 7	6	5	4	3	2	1	Bit 0
0x003B	**TPM1C2SC**	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	0	0
0x003C	**TPM1C2VH**	Bit 15	14	13	12	11	10	9	Bit 8
0x003D	**TPM1C2VL**	Bit 7	6	5	4	3	2	1	Bit 0
0x003E– 0x0043	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x0044	**PTGD**	0	0	0	0	PTGD3	PTGD2	PTGD1	PTGD0
0x0045	**PTGPE**	0	0	0	0	PTGPE3	PTGPE2	PTGPE1	PTGPE0
0x0046	**PTGSE**	0	0	0	0	PTGSE3	PTGSE2	PTGSE1	PTGSE0
0x0047	**PTGDD**	0	0	0	0	PTGDD3	PTGDD2	PTGDD1	PTGDD0
0x0048	**ICGC1**	HGO	RANGE	REFS	CLKS		OSCSTEN	LOCD	0
0x0049	**ICGC2**	LOLRE	MFD			LOCRE	RFD		
0x004A	**ICGS1**	CLKST		REFST	LOLS	LOCK	LOCS	ERCS	ICGIF
0x004B	**ICGS2**	0	0	0	0	0	0	0	DCOS
0x004C	**ICGFLTU**	0	0	0	0	FLT			
0x004D	**ICGFLTL**	FLT							
0x004E	**ICGTRM**	TRIM							
0x004F	Reserved	0	0	0	0	0	0	0	0
0x0050	**ATDC**	ATDPU	DJM	RES8	SGN	PRS			
0x0051	**ATDSC**	CCF	ATDIE	ATDCO	ATDCH				
0x0052	**ATDRH**	Bit 7	6	5	4	3	2	1	Bit 0
0x0053	**ATDRL**	Bit 7	6	5	4	3	2	1	Bit 0

makes the MCU secure. During development, whenever the FLASH is erased, it is good practice to immediately program the SEC00 bit to 0 in NVOPT so SEC01:SEC00 = 1:0. This would allow the MCU to remain unsecured after a subsequent reset.

The on-chip debug module cannot be enabled while the MCU is secure. The separate background debug controller can still be used for background memory access commands, but the MCU cannot enter active background mode except by holding BKGD/MS low at the rising edge of reset.

A user can choose to allow or disallow a security unlocking mechanism through an 8-byte backdoor security key. If the nonvolatile KEYEN bit in NVOPT/FOPT is 0, the backdoor key is disabled and there is no way to disengage security without completely erasing all FLASH locations. If KEYEN is 1, a secure user program can temporarily disengage security by:

1. Writing 1 to KEYACC in the FCNFG register. This makes the FLASH module interpret writes to the backdoor comparison key locations (NVBACKKEY through NVBACKKEY+7) as values to be compared against the key rather than as the first step in a FLASH program or erase command.
2. Writing the user-entered key values to the NVBACKKEY through NVBACKKEY+7 locations. These writes must be done in order, starting with the value for NVBACKKEY and ending with NVBACKKEY+7. STHX should not be used for these writes because these writes cannot be done on adjacent bus cycles. User software normally would get the key codes from outside the MCU system through a communication interface such as a serial I/O.
3. Writing 0 to KEYACC in the FCNFG register. If the 8-byte key that was just written matches the key stored in the FLASH locations, SEC01:SEC00 are automatically changed to 1:0 and security will be disengaged until the next reset.

The security key can be written only from RAM, so it cannot be entered through background commands without the cooperation of a secure user program.

The backdoor comparison key (NVBACKKEY through NVBACKKEY+7) is located in FLASH memory locations in the nonvolatile register space so users can program these locations exactly as they would program any other FLASH memory location. The nonvolatile registers are in the same 512-byte block of FLASH as the reset and interrupt vectors, so block protecting that space also block protects the backdoor comparison key. Block protects cannot be changed from user application programs, so if the vector space is block protected, the backdoor security key mechanism cannot permanently change the block protect, security settings, or the backdoor key.

Security can always be disengaged through the background debug interface by performing these steps:

1. Disable any block protections by writing FPROT. FPROT can be written only with background debug commands, not from application software.
2. Mass erase FLASH, if necessary.
3. Blank check FLASH. Provided FLASH is completely erased, security is disengaged until the next reset.

   To avoid returning to secure mode after the next reset, program NVOPT so SEC01:SEC00 = 1:0.

5.4.2.2 Edge and Level Sensitivity

The IRQMOD control bit re-configures the detection logic so it detects edge events and pin levels. In this edge detection mode, the IRQF status flag becomes set when an edge is detected (when the IRQ pin changes from the deasserted to the asserted level), but the flag is continuously set (and cannot be cleared) as long as the IRQ pin remains at the asserted level.

5.4.3 Interrupt Vectors, Sources, and Local Masks

Table 5-1 provides a summary of all interrupt sources. Higher-priority sources are located toward the bottom of the table. The high-order byte of the address for the interrupt service routine is located at the first address in the vector address column, and the low-order byte of the address for the interrupt service routine is located at the next higher address.

When an interrupt condition occurs, an associated flag bit becomes set. If the associated local interrupt enable is 1, an interrupt request is sent to the CPU. Within the CPU, if the global interrupt mask (I bit in the CCR) is 0, the CPU will finish the current instruction, stack the PCL, PCH, X, A, and CCR CPU registers, set the I bit, and then fetch the interrupt vector for the highest priority pending interrupt. Processing then continues in the interrupt service routine.

5.7.5 System Device Identification Register (SDIDH, SDIDL)

This read-only register is included so host development systems can identify the HCS08 derivative and revision number. This allows the development software to recognize where specific memory blocks, registers, and control bits are located in a target MCU.

	7	6	5	4	3	2	1	0
R					ID11	ID10	ID9	ID8
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-6. System Device Identification Register High (SDIDH)

Table 5-6. SDIDH Field Descriptions

Field	Description
3:0 ID[11:8]	**Part Identification Number** — Each derivative in the HCS08 Family has a unique identification number. The MC9S08GT16A/GT8A is hard coded to the value 0x00D. See also ID bits in Table 5-7.

	7	6	5	4	3	2	1	0
R	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
W								
Reset	0	0	0	0	1	1	0	1

= Unimplemented or Reserved

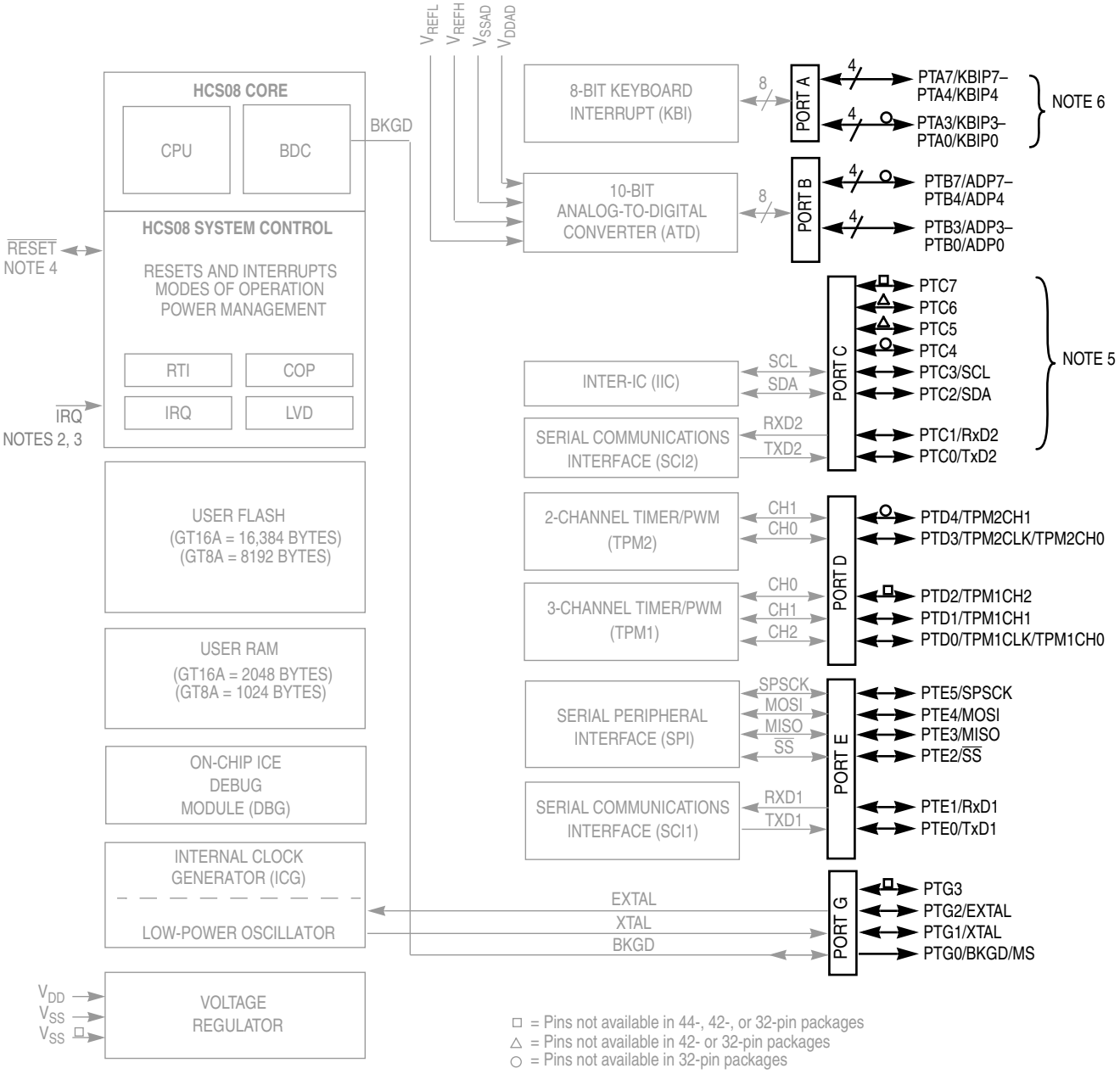
Figure 5-7. System Device Identification Register Low (SDIDL)

Table 5-7. SDIDL Field Descriptions

Field	Description
7:0 ID[7:0]	**Part Identification Number** — Each derivative in the HCS08 Family has a unique identification number. The MC9S08GT16A/GT8A is hard coded to the value 0x00D. See also ID bits in Table 5-6.

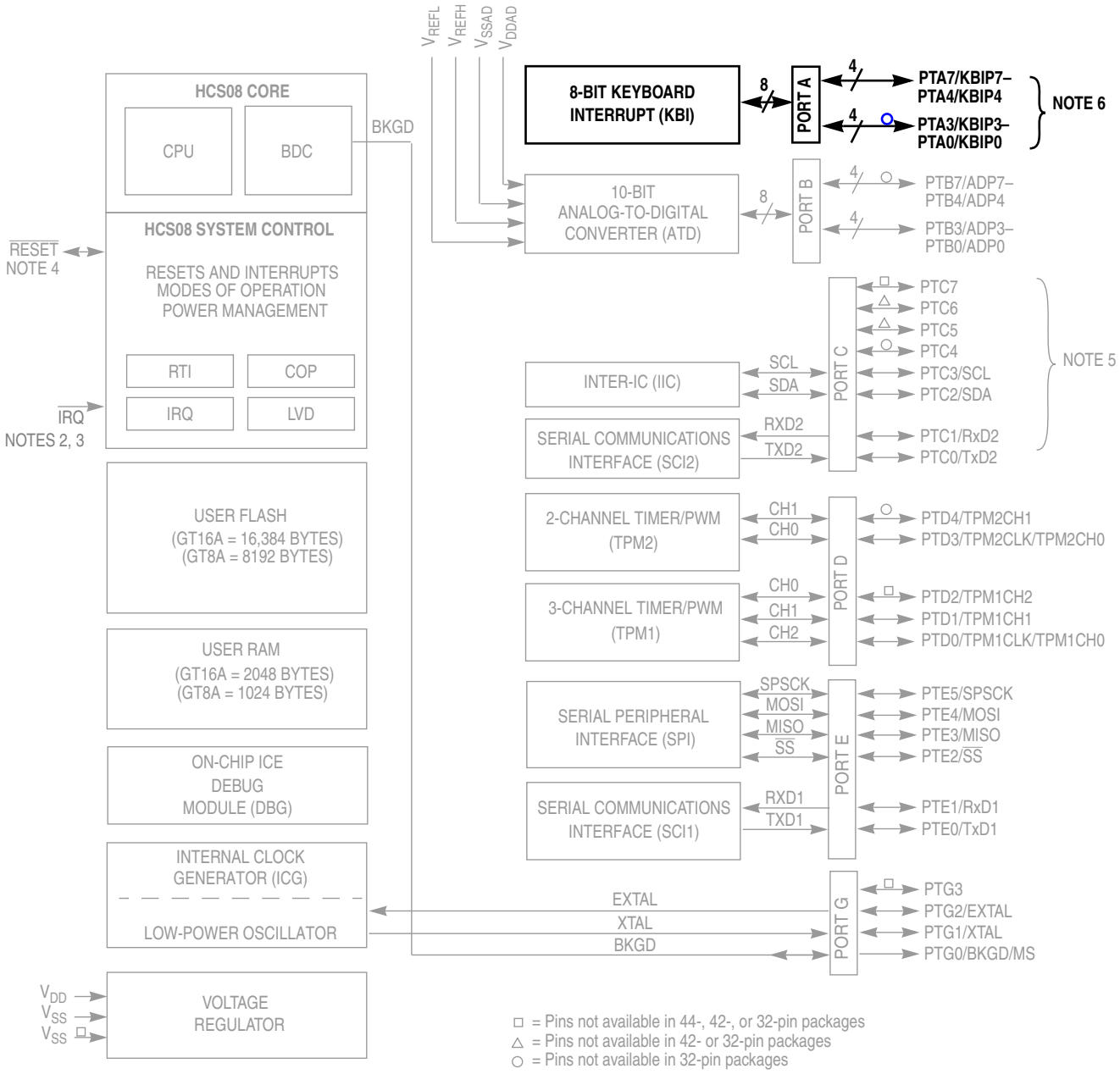
- Eight port B pins shared with ATD
- Eight high-current port C pins shared with SCI2 and IIC
- Five port D pins shared with TPM1 and TPM2
- Six port E pins shared with SCI1 and SPI
- Four port G pins shared with EXTAL, XTAL, and BKGD/MS

6.1.2 Block Diagram



NOTES:
1. Port pins are software configurable with pullup device if input port.
2. Pin contains pullup/pulldown device if IRQ enabled (IRQPE = 1).
3. IRQ does not have a clamp diode to VDD. IRQ should not be driven above VDD.
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 6-1. Block Diagram Highlighting Parallel Input/Output Pins



NOTES:

1. Port pins are software configurable with pullup device if input port.
2. Pin contains pullup/pulldown device if IRQ enabled (IRQPE = 1).
3. IRQ does not have a clamp diode to VDD. IRQ should not be driven above VDD.
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 7-2. Block Diagram Highlighting the KBI Module

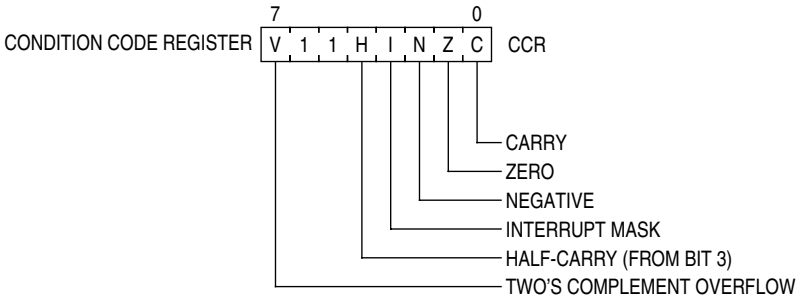


Figure 8-2. Condition Code Register

Table 8-1. 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

Table 10-5. Mode, Edge, and Level Selection

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	XX	00	Pin not used for TPM channel; use as an external clock for the TPM or revert to general-purpose I/O	
0	00	01	Input capture	Capture on rising edge only
		10		Capture on falling edge only
		11		Capture on rising or falling edge
	01	00	Output compare	Software compare only
		01		Toggle output on compare
		10		Clear output on compare
		11		Set output on compare
	1X	10	Edge-aligned PWM	High-true pulses (clear output on compare)
		X1		Low-true pulses (set output on compare)
1	XX	10	Center-aligned PWM	High-true pulses (clear output on compare-up)
		X1		Low-true pulses (set output on compare-up)

If the associated port pin is not stable for at least two bus clock cycles before changing to input capture mode, it is possible to get an unexpected indication of an edge trigger. Typically, a program would clear status flags after changing channel configuration bits and before enabling channel interrupts or using the status flags to avoid any unexpected behavior.

10.3.5 Timer x Channel Value Registers (TPMxCnVH:TPMxCnVL)

These read/write registers contain the captured TPM counter value of the input capture function or the output compare value for the output compare or PWM functions. The channel value registers are cleared by reset.

	7	6	5	4	3	2	1	0
R W	Bit 15	14	13	12	11	10	9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 10-9. Timer x Channel Value Register High (TPMxCnVH)

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0

Figure 10-10. Timer 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.

transferred to the corresponding timer channel registers only after both 8-bit bytes of a 16-bit register have been written and the timer counter overflows (reverses direction from up-counting to down-counting at the end of the terminal count in the modulus register). This TPMxCNT overflow requirement only applies to PWM channels, not output compares.

Optionally, when TPMxCNTH:TPMxCNTL = TPMxMODH:TPMxMODL, the TPM can generate a TOF interrupt at the end of this count. The user can choose to reload any number of the PWM buffers, and they will all update simultaneously at the start of a new period.

Writing to TPMxSC cancels any values written to TPMxMODH and/or TPMxMODL and resets the coherency mechanism for the modulo registers. Writing to TPMxCnSC cancels any values written to the channel value registers and resets the coherency mechanism for TPMxCnVH:TPMxCnVL.

10.5 TPM Interrupts

The TPM generates an optional interrupt for the main counter overflow and an interrupt for each channel. The meaning of channel interrupts depends on the mode of operation for each channel. If the channel is configured for input capture, the interrupt flag is set each time the selected input capture edge is recognized. If the channel is configured for output compare or PWM modes, the interrupt flag is set each time the main timer counter matches the value in the 16-bit channel value register. See the Resets, Interrupts, and System Configuration chapter for absolute interrupt vector addresses, priority, and local interrupt mask control bits.

For each interrupt source in the TPM, a flag bit is set on recognition of the interrupt condition such as timer overflow, channel input capture, or output compare events. This flag may be read (polled) by software to verify that the action has occurred, or an associated enable bit (TOIE or CHnIE) can be set to enable hardware interrupt generation. While the interrupt enable bit is set, a static interrupt will be generated whenever the associated interrupt flag equals 1. It is the responsibility of user software to perform a sequence of steps to clear the interrupt flag before returning from the interrupt service routine.

10.5.1 Clearing Timer Interrupt Flags

TPM interrupt flags are cleared by a 2-step process that includes a read of the flag bit while it is set (1) followed by a write of 0 to the bit. If a new event is detected between these two steps, the sequence is reset and the interrupt flag remains set after the second step to avoid the possibility of missing the new event.

10.5.2 Timer Overflow Interrupt Description

The conditions that cause TOF to become set depend on the counting mode (up or up/down). In up-counting mode, the 16-bit timer counter counts from 0x0000 through 0xFFFF and overflows to 0x0000 on the next counting clock. TOF becomes set at the transition from 0xFFFF to 0x0000. When a modulus limit is set, TOF becomes set at the transition from the value set in the modulus register to 0x0000. When the counter is operating in up-/down-counting mode, the TOF flag gets set as the counter changes direction at the transition from the value set in the modulus register and the next lower count value. This corresponds to the end of a PWM period. (The 0x0000 count value corresponds to the center of a period.)

12.6.1.2 Pseudo—Code Example

In this example, the SPI module will be set up for master mode with only transmit interrupts enabled to run at a maximum baud rate of bus clock divided by 2. Clock phase and polarity will be set for an active-high SPI clock where the first edge on SPSCK occurs at the start of the first cycle of a data transfer.

SPIC1 = 0x74(%01110100)

Bit 7	SPIE	= 0	Disables receive and mode fault interrupts
Bit 6	SPE	= 1	Enables the SPI system
Bit 5	SPTIE	= 1	Enables SPI transmit interrupts
Bit 4	MSTR	= 1	Sets the SPI module as a master SPI device
Bit 3	CPOL	= 0	Configures SPI clock as active-high
Bit 2	CPHA	= 1	First edge on SPSCK at start of first data transfer cycle
Bit 1	SSOE	= 0	Determines $\overline{SS}$ pin function when mode fault enabled
Bit 0	LSBFE	= 0	SPI serial data transfers start with most significant bit

SPIC2 = 0x00(%00000000)

Bit 7:5		= 000	Unimplemented
Bit 4	MODFEN	= 0	Disables mode fault function
Bit 3	BIDIROE	= 0	SPI data I/O pin acts as input
Bit 2		= 0	Unimplemented
Bit 1	SPISWAI	= 0	SPI clocks operate in wait mode
Bit 0	SPC0	= 0	SPI uses separate pins for data input and output

SPIBR = 0x00(%00000000)

Bit 7	= 0	Unimplemented
Bit 6:4	= 000	Sets prescale divisor to 1
Bit 3	= 0	Unimplemented
Bit 2:0	= 000	Sets baud rate divisor to 2

SPIS = 0x00(%00000000)

Bit 7	SPRF	= 0	Flag is set when receive data buffer is full
Bit 6		= 0	Unimplemented
Bit 5	SPTEF	= 0	Flag is set when transmit data buffer is empty
Bit 4	MODF	= 0	Mode fault flag for master mode
Bit 3:0		= 0	Unimplemented

SPID = 0xxx

Holds data to be transmitted by transmit buffer and data received by receive buffer.

of how straight the line is (how far it deviates from a straight line). The adjusted ideal transition voltage is:

Eqn. 14-6

$$\text{Adjusted Ideal Trans. V} = \frac{(\text{Current Code} - 1/2)}{2^N} * ((V_{REFH} + E_{FS}) - (V_{REFL} + E_{ZS}))$$

- Zero scale error (E_{ZS}) — This is the difference between the transition voltage to the first valid code and the ideal transition to that code. Normally, it is defined as the difference between the actual and ideal transition to code $001, but in some cases the first transition may be to a higher code. The ideal transition to any code is:

Eqn. 14-7

$$\text{Ideal Transition V} = \frac{(\text{Current Code} - 1/2)}{2^N} *(V_{REFH} - V_{REFL})$$

- Full scale error (E_{FS}) — This is the difference between the transition voltage to the last valid code and the ideal transition to that code. Normally, it is defined as the difference between the actual and ideal transition to code $3FF, but in some cases the last transition may be to a lower code. The ideal transition to any code is:

Eqn. 14-8

$$\text{Ideal Transition V} = \frac{(\text{Current Code} - 1/2)}{2^N} *(V_{REFH} - V_{REFL})$$

- Total unadjusted error (E_{TU}) — This is the difference between the transition voltage to a given code and the ideal straight-line transfer function. An alternate definition (with the same result) is the difference between the actual transfer function and the ideal straight-line transfer function. This measure of error includes inherent quantization error and all forms of circuit error (INL, DNL, zero-scale, and full-scale) except input leakage error, which is not due to the ATD.
- Input leakage error (E_{IL}) — This is the error between the transition voltage to the current code and the ideal transition to that code that is the result of input leakage across the real portion of the impedance of the network that drives the analog input. This error is a system-observable error which is not inherent to the ATD, so it is not added to total error. This error is:

$$\mathbf{E_{IL}\ (in\ V) = input\ leakage * R_{AS}}$$ *Eqn. 14-9*

There are two other forms of error which are not specified which can also affect ATD accuracy. These are:

- Sampling error (E_S) — The error due to inadequate time to charge the ATD circuitry
- Noise error (E_N) — The error due to noise on V_{AIN}, V_{REFH}, or V_{REFL} due to either direct coupling (noise source capacitively coupled directly on the signal) or power supply (V_{DDAD}, V_{SSAD}, V_{DD}, and V_{SS}) noise interfering with the ATD's ability to resolve the input accurately. The error due to internal sources can be reduced (and specified operation achieved) by operating the ATD conversion in wait mode and ceasing all IO activity. Reducing the error due to external sources is dependent on system activity and board layout.

14.5 Resets

The ATD module is reset on system reset. If the system reset signal is activated, the ATD registers are initialized back to their reset state and the ATD module is powered down. This occurs as a function of the register file initialization; the reset definition of the ATDPU bit (power down bit) is zero or disabled.

The MCU places the module back into an initialized state. If the module is performing a conversion, the current conversion is terminated, the conversion complete flag is cleared, and the SAR register bits are cleared. Any pending interrupts are also cancelled. Note that the control, test, and status registers are initialized on reset; the initialized register state is defined in the register description section of this specification.

Enabling the module (using the ATDPU bit) does not cause the module to reset since the register file is not initialized. Finally, writing to control register ATDC does not cause the module to reset; the current conversion will be terminated.

14.6 Interrupts

The ATD module originates interrupt requests and the MCU handles or services these requests. Details on how the ATD interrupt requests are handled can be found in the resets and interrupts chapter of this data sheet.

The ATD interrupt function is enabled by setting the ATDIE bit in the ATDSC register. When the ATDIE bit is set, an interrupt is generated at the end of an ATD conversion and the ATD result registers (ATDRH and ATDRL) contain the result data generated by the conversion. If the interrupt function is disabled (ATDIE = 0), then the CCF flag must be polled to determine when a conversion is complete.

The interrupt will remain pending as long as the CCF flag is set. The CCF bit is cleared whenever the ATD status and control (ATDSC) register is written. The CCF bit is also cleared whenever the ATD result registers (ATDRH or ATDRL) are read.

Table 14-8. Interrupt Summary

Interrupt	Local Enable	Description
CCF	ATDIE	Conversion complete

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.

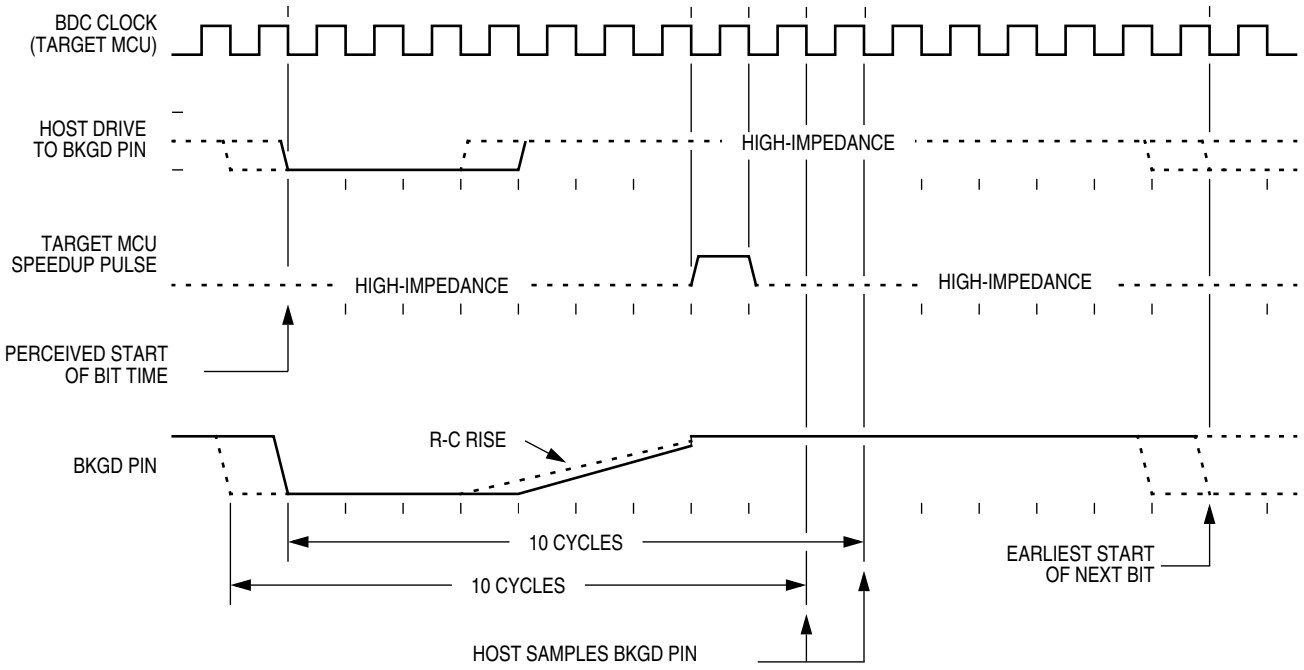


Figure 15-3. BDC Target-to-Host Serial Bit Timing (Logic 1)

15.3.6 Hardware Breakpoints

The BRKEN control bit in the DBGC register may be set to 1 to allow any of the trigger conditions described in Section 15.3.5, “Trigger Modes,” to be used to generate a hardware breakpoint request to the CPU. TAG in DBGC controls whether the breakpoint request will be treated as a tag-type breakpoint or a force-type breakpoint. A tag breakpoint causes the current opcode to be marked as it enters the instruction queue. If a tagged opcode reaches the end of the pipe, the CPU executes a BGND instruction to go to active background mode rather than executing the tagged opcode. A force-type breakpoint causes the CPU to finish the current instruction and then go to active background mode.

If the background mode has not been enabled (ENBDM = 1) by a serial WRITE_CONTROL command through the BKGD pin, the CPU will execute an SWI instruction instead of going to active background mode.

15.4 Register Definition

This section contains the descriptions of the BDC and DBG registers and control bits.

Refer to the high-page register summary in the device overview chapter of this data sheet for the absolute address assignments for all DBG registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

15.4.1 BDC Registers and Control Bits

The BDC has two registers:

- The BDC status and control register (BDCSCR) is an 8-bit register containing control and status bits for the background debug controller.
- The BDC breakpoint match register (BDCBKPT) holds a 16-bit breakpoint match address.

These registers are accessed with dedicated serial BDC commands and are not located in the memory space of the target MCU (so they do not have addresses and cannot be accessed by user programs).

Some of the bits in the BDCSCR have write limitations; otherwise, these registers may be read or written at any time. For example, the ENBDM control bit may not be written while the MCU is in active background mode. (This prevents the ambiguous condition of the control bit forbidding active background mode while the MCU is already in active background mode.) Also, the four status bits (BDMACT, WS, WSF, and DVF) are read-only status indicators and can never be written by the WRITE_CONTROL serial BDC command. The clock switch (CLKSW) control bit may be read or written at any time.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								BDFR[1]
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

[1] BDFR is writable only through serial background mode debug commands, not from user programs.

Figure 15-6. System Background Debug Force Reset Register (SBDFR)

Table 15-3. SBDFR Register Field Description

Field	Description
0 BDFR	**Background Debug Force Reset** — A serial active background mode command such as WRITE_BYTE allows an external debug host to force a target system reset. Writing 1 to this bit forces an MCU reset. This bit cannot be written from a user program.

15.4.3 DBG Registers and Control Bits

The debug module includes nine bytes of register space for three 16-bit registers and three 8-bit control and status registers. These registers are located in the high register space of the normal memory map so they are accessible to normal application programs. These registers are rarely if ever accessed by normal user application programs with the possible exception of a ROM patching mechanism that uses the breakpoint logic.

15.4.3.1 Debug Comparator A High Register (DBGCAH)

This register contains compare value bits for the high-order eight bits of comparator A. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

15.4.3.2 Debug Comparator A Low Register (DBGCAL)

This register contains compare value bits for the low-order eight bits of comparator A. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

15.4.3.3 Debug Comparator B High Register (DBGCBH)

This register contains compare value bits for the high-order eight bits of comparator B. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

15.4.3.4 Debug Comparator B Low Register (DBGCBL)

This register contains compare value bits for the low-order eight bits of comparator B. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

A.8 ATD Characteristics

Table A-8. ATD Electrical Characteristics (Operating)

No.	Characteristic	Condition	Symbol	Min	Typ	Max	Unit
1	ATD supply[1]		V_{DDAD}	1.80	—	3.6	V
2	ATD supply current	Enabled	$I_{DDADrun}$	—	0.7	1.2	mA
		Disabled (ATDPU = 0 or STOP)	$I_{DDADstop}$	—	0.02	0.6	µA
3	Differential supply voltage	V_{DD}–V_{DDAD}	$\lvert V_{DDLT}\rvert$	—	—	100	mV
4	Differential ground voltage	V_{SS}–V_{SSAD}	$\lvert V_{SDLT}\rvert$	—	—	100	mV
5	Reference potential, low		$\lvert V_{REFL}\rvert$	—	—	V_{SSAD}	V
	Reference potential, high	$2.08V \le V_{DDAD} \le 3.6V$	V_{REFH}	2.08	—	V_{DDAD}	V
		$1.80V \le V_{DDAD} < 2.08V$		V_{DDAD}	—	V_{DDAD}	
6	Reference supply current (V_{REFH} to V_{REFL})	Enabled	I_{REF}	—	200	300	µA
		Disabled (ATDPU = 0 or STOP)	I_{REF}	—	<0.01	0.02	
7	Analog input voltage[2]		V_{INDC}	V_{SSAD} – 0.3	—	V_{DDAD} + 0.3	V

1 V_{DDAD} must be at same potential as V_{DD}.
2 Maximum electrical operating range, not valid conversion range.

Table A-9. ATD Timing/Performance Characteristics[1]

No.	Characteristic	Condition	Symbol	Min	Typ	Max	Unit
1	ATD conversion clock frequency	$2.08V \le V_{DDAD} \le 3.6V$	f_{ATDCLK}	0.5	—	2.0	MHz
		$1.80V \le V_{DDAD} < 2.08V$		0.5	—	1.0	
2	Conversion cycles (continuous convert)[2]		CC	28	28	<30	ATDCLK cycles
3	Conversion time (Including sample time)	$2.08V \le V_{DDAD} \le 3.6V$	T_{conv}	14.0	—	60.0	µs
		$1.80V \le V_{DDAD} < 2.08V$		28.0	—	60.0	
4	ATD sample time	t_{ADS}	t_{ADS}	—	14	—	ATDCLK cycles
5	Source impedance at input[3]		R_{AS}	—	—	10	kΩ
6	Analog Input Voltage[4]		V_{AIN}	V_{REFL}		V_{REFH}	V

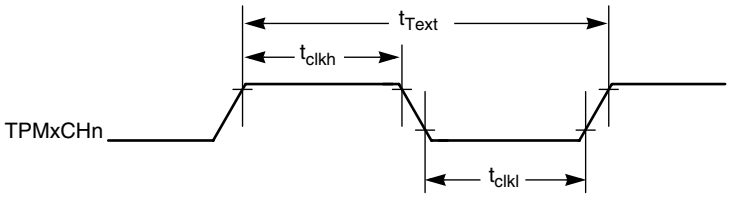


Figure A-15. Timer External Clock

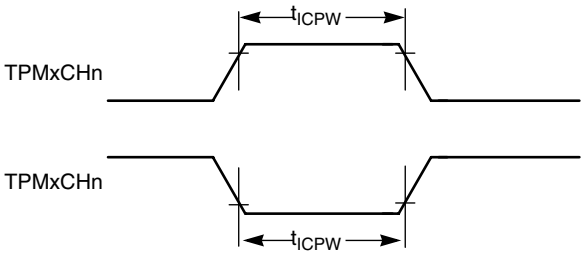


Figure A-16. Timer Input Capture Pulse