NXP USA Inc. - MC9S08DV60CLH Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dv60clh

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

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x1883	CANBTR1	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
0x1884	CANRFLG	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
0x1885	CANRIER	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
0x1886	CANTFLG	0	0	0	0	0	TXE2	TXE1	TXE0
0x1887	CANTIER	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
0x1888	CANTARQ	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
0x1889	CANTAAK	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
0x188A	CANTBSEL	0	0	0	0	0	TX2	TX1	TX0
0x188B	CANIDAC	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
0x188C	Reserved	0	0	0	0	0	0	0	0
0x188D	CANMISC	0	0	0	0	0	0	0	BOHOLD
0x188E	CANRXERR	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
0x188F	CANTXERR	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
0x1890 – 0x1893	CANIDAR0 – CANIDAR3	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
0x1894 – 0x1897	CANIDMR0 – CANIDMR3	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
0x1898 – 0x189B	CANIDAR4 – CANIDAR7	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
0x189C– 0x189F	CANIDMR4 – CANIDMR7	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
0x18BE	CANTTSRH	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
0x18BF	CANTTSRL	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
0x18C0– 0x18FF	Reserved	— —	— —	— —	— —	— —	— —	— —	— —

[1] This bit is reserved. User must write a 1 to this bit. Failing to do so may result in unexpected behavior.

Figure 4-4 shows the structure of receive and transmit buffers for extended identifier mapping. These registers vary depending on whether standard or extended mapping is selected. See Chapter 12, “Freescale Controller Area Network (S08MSCANV1),” for details on extended and standard identifier mapping.

Table 4-4. MSCAN Foreground Receive and Transmit Buffer Layouts — Extended Mapping Shown

0x18A0	CANRIDR0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
0x18A1	CANRIDR1	ID20	ID19	ID18	SRR[1]	IDE[1]	ID17	ID16	ID15
0x18A2	CANRIDR2	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
0x18A3	CANRIDR3	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR[2]
0x18A4 – 0x18AB	CANRDSR0 – CANRDSR7	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0x18AC	CANRDLR	—	—	—	—	DLC3	DLC2	DLC1	DLC0
0x18AD	Reserved	—	—	—	—	—	—	—	—
0x18AE	CANRTSRH	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8

4.5.2 Program and Erase Times

Before any program or erase command can be accepted, the Flash clock divider register (FCDIV) must be written to set the internal clock for the Flash module to a frequency (f_{FCLK}) between 150 kHz and 200 kHz (see Section 4.5.10.1, “Flash Clock Divider Register (FCDIV)”). This register can be written only once, so normally this write is performed during reset initialization. The user must ensure that FACCERR is not set before writing to the FCDIV register. One period of the resulting clock ($1/f_{FCLK}$) is used by the command processor to time program and erase pulses. An integer number of these timing pulses is used by the command processor to complete a program or erase command.

Table 4-6 shows program and erase times. The bus clock frequency and FCDIV determine the frequency of FCLK (f_{FCLK}). The time for one cycle of FCLK is $t_{FCLK} = 1/f_{FCLK}$. The times are shown as a number of cycles of FCLK and as an absolute time for the case where $t_{FCLK} = 5\ \mu s$. Program and erase times shown include overhead for the command state machine and enabling and disabling of program and erase voltages.

Table 4-6. Program and Erase Times

Parameter	Cycles of FCLK	Time if FCLK = 200 kHz
Byte program	9	45 μs
Burst program	4	20 μs[1]
Sector erase	4000	20 ms
Mass erase	20,000	100 ms
Sector erase abort	4	20 μs[1]

[1] Excluding start/end overhead

4.5.3 Program and Erase Command Execution

The FCDIV register must be initialized after any reset and any error flag is 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 sector erase commands, the address can be any address in the sector of Flash to be erased. For mass erase and blank check commands, the address can be any address in the Flash .

NOTE

> Before programming a particular byte in the Flash , the sector in which that particular byte resides must be erased by a mass or sector erase operation. Reprogramming bits in an already programmed byte without first performing an erase operation may disturb data stored in the Flash memory.

2. Write the command code for the desired command to FCMD. The six valid commands are blank check (0x05), byte program (0x20), burst program (0x25), sector erase (0x40), mass erase[1] (0x41), and sector erase abort (0x47). The command code is latched into the command buffer.

[1] A mass erase is possible only when the Flash block is fully unprotected.

5.8.1 Interrupt Pin Request Status and Control Register (IRQSC)

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

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

▨ = Unimplemented or Reserved

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

Table 5-2. IRQSC Register Field Descriptions

Field	Description
6 IRQPDD	**Interrupt Request (IRQ) Pull Device Disable**— This read/write control bit is used to disable the internal pull-up/pull-down device when the IRQ pin is enabled (IRQPE = 1) allowing for an external device to be used. 0 IRQ pull device enabled if IRQPE = 1. 1 IRQ pull device disabled if IRQPE = 1.
5 IRQEDG	**Interrupt Request (IRQ) Edge Select** — This read/write control bit is used to select the polarity of edges or levels on the IRQ pin that cause IRQF to be set. The IRQMOD control bit determines whether the IRQ pin is sensitive to both edges and levels or only edges. When the IRQ pin is enabled as the IRQ input and is configured to detect rising edges, it has a pull-down. When the IRQ pin is enabled as the IRQ input and is configured to detect falling edges, it has a pull-up. 0 IRQ is falling edge or falling edge/low-level sensitive. 1 IRQ is rising edge or rising edge/high-level sensitive.
4 IRQPE	**IRQ Pin Enable** — This read/write control bit enables the IRQ pin function. When this bit is set the IRQ pin can be used as an interrupt request. 0 IRQ pin function is disabled. 1 IRQ pin function is enabled.
3 IRQF	**IRQ Flag** — This read-only status bit indicates when an interrupt request event has occurred. 0 No IRQ request. 1 IRQ event detected.
2 IRQACK	**IRQ Acknowledge** — This write-only bit is used to acknowledge interrupt request events (write 1 to clear IRQF). Writing 0 has no meaning or effect. Reads always return 0. If edge-and-level detection is selected (IRQMOD = 1), IRQF cannot be cleared while the IRQ pin remains at its asserted level.
1 IRQIE	**IRQ Interrupt Enable** — This read/write control bit determines whether IRQ events generate an interrupt request. 0 Interrupt request when IRQF set is disabled (use polling). 1 Interrupt requested whenever IRQF = 1.
0 IRQMOD	**IRQ Detection Mode** — This read/write control bit selects either edge-only detection or edge-and-level detection. The IRQEDG control bit determines the polarity of edges and levels that are detected as interrupt request events. See Section 5.5.2.2, “Edge and Level Sensitivity” for more details. 0 IRQ event on falling edges or rising edges only. 1 IRQ event on falling edges and low levels or on rising edges and high levels.

5.8.5 System Options Register 2 (SOPT2)

This high page register contains bits to configure MCU specific features on the MC9S08DV60 Series devices.

	7	6	5	4	3	2	1	0
R	COPCLKS[1]	COPW[1]	0	ADHTS	0	MCSEL		
W			▨		▨			
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 5-6. System Options Register 2 (SOPT2)

[1] This bit can be written only one time after reset. Additional writes are ignored.

Table 5-7. SOPT2 Register Field Descriptions

Field	Description
7 COPCLKS	**COP Watchdog Clock Select** — This write-once bit selects the clock source of the COP watchdog. See Table 5-6 for details. 0 Internal 1-kHz clock is source to COP. 1 Bus clock is source to COP.
6 COPW	**COP Window** — This write-once bit selects the COP operation mode. When set, the 0x55-0xAA write sequence to the SRS register must occur in the last 25% of the selected period. Any write to the SRS register during the first 75% of the selected period will reset the MCU. 0 Normal COP operation. 1 Window COP operation.
4 ADHTS	**ADC Hardware Trigger Select** — This bit selects which hardware trigger initiates conversion for the analog to digital converter when the ADC hardware trigger is enabled (ADCTRG is set in ADCSC2 register). 0 Real Time Counter (RTC) overflow. 1 External Interrupt Request (IRQ) pin.
2:0 MCSEL	**MCLK Divide Select**— These bits enable the MCLK output on PTA0 pin and select the divide ratio for the MCLK output according to the formula below when the MCSEL bits are not equal to all zeroes. In case that the MCSEL bits are all zeroes, the MCLK output is disabled. MCLK frequency = Bus Clock frequency ÷ (2 * MCSEL)

5.8.8 System Power Management Status and Control 2 Register (SPMSC2)

This register is used to report the status of the low-voltage warning function, and to configure the stop mode behavior of the MCU. This register should be written during the user's reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

	7	6	5	4	3	2	1	0
R	0	0	LVDV[1]	LVWV	PPDF	0	0	PPDC[2]
W	▨	▨			▨	PPDACK	▨	
Power-on Reset:	0	0	0	0	0	0	0	0
LVD Reset:	0	0	u	u	0	0	0	0
Any other Reset:	0	0	u	u	0	0	0	0

▨ = Unimplemented or Reserved

u = Unaffected by reset

[1] This bit can be written only one time after power-on reset. Additional writes are ignored.

[2] This bit can be written only one time after reset. Additional writes are ignored.

Figure 5-10. System Power Management Status and Control 2 Register (SPMSC2)

Table 5-11. SPMSC2 Register Field Descriptions

Field	Description
5 LVDV	**Low-Voltage Detect Voltage Select** — This write-once bit selects the low-voltage detect (LVD) trip point setting. It also selects the warning voltage range. See Table 5-12.
4 LVWV	**Low-Voltage Warning Voltage Select** — This bit selects the low-voltage warning (LVW) trip point voltage. See Table 5-12.
3 PPDF	**Partial Power Down Flag** — This read-only status bit indicates that the MCU has recovered from stop2 mode. 0 MCU has not recovered from stop2 mode. 1 MCU recovered from stop2 mode.
2 PPDACK	**Partial Power Down Acknowledge** — Writing a 1 to PPDACK clears the PPDF bit.
0 PPDC	**Partial Power Down Control** — This write-once bit controls whether stop2 or stop3 mode is selected. 0 Stop3 mode enabled. 1 Stop2, partial power down, mode enabled.

Table 5-12. LVD and LVW Trip Point Typical Values[1]

LVDV:LVWV	LVW Trip Point	LVD Trip Point
0:0	V_{LVW0} = 2.74 V	V_{LVD0} = 2.56 V
0:1	V_{LVW1} = 2.92 V	
1:0	V_{LVW2} = 4.3 V	V_{LVD1} = 4.0 V
1:1	V_{LVW3} = 4.6 V	

[1] See Electrical Characteristics appendix for minimum and maximum values.

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.

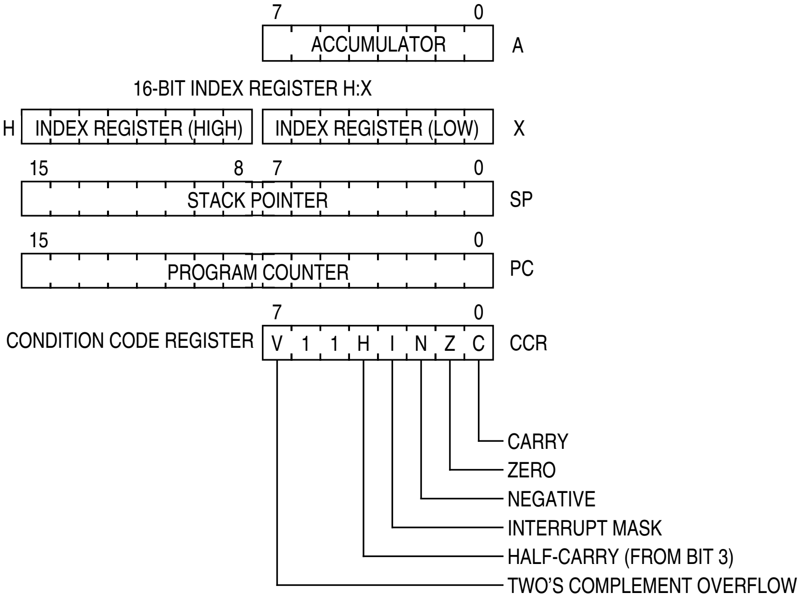

Figure 7-1. CPU Registers

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.

7.2.3 Stack Pointer (SP)

This 16-bit address pointer register points at the next available location on the automatic last-in-first-out (LIFO) stack. The stack may be located anywhere in the 64-Kbyte address space that has RAM and can be any size up to the amount of available RAM. The stack is used to automatically save the return address for subroutine calls, the return address and CPU registers during interrupts, and for local variables. The AIS (add immediate to stack pointer) instruction adds an 8-bit signed immediate value to SP. This is most often used to allocate or deallocate space for local variables on the stack.

SP is forced to 0x00FF at reset for compatibility with the earlier M68HC05 Family. HCS08 programs normally change the value in SP to the address of the last location (highest address) in on-chip RAM during reset initialization to free up direct page RAM (from the end of the on-chip registers to 0x00FF).

The RSP (reset stack pointer) instruction was included for compatibility with the M68HC05 Family and is seldom used in new HCS08 programs because it only affects the low-order half of the stack pointer.

7.2.4 Program Counter (PC)

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched.

During normal program execution, the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, interrupt, and return operations load the program counter with an address other than that of the next sequential location. This is called a change-of-flow.

During reset, the program counter is loaded with the reset vector that is located at 0xFFFE and 0xFFFF. The vector stored there is the address of the first instruction that will be executed after exiting the reset state.

7.2.5 Condition Code Register (CCR)

The 8-bit condition code register contains the interrupt mask (I) and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to 1. The following paragraphs describe the functions of the condition code bits in general terms. For a more detailed explanation of how each instruction sets the CCR bits, refer to the *HCS08 Family Reference Manual, volume 1,* Freescale Semiconductor document order number HCS08RMv1.

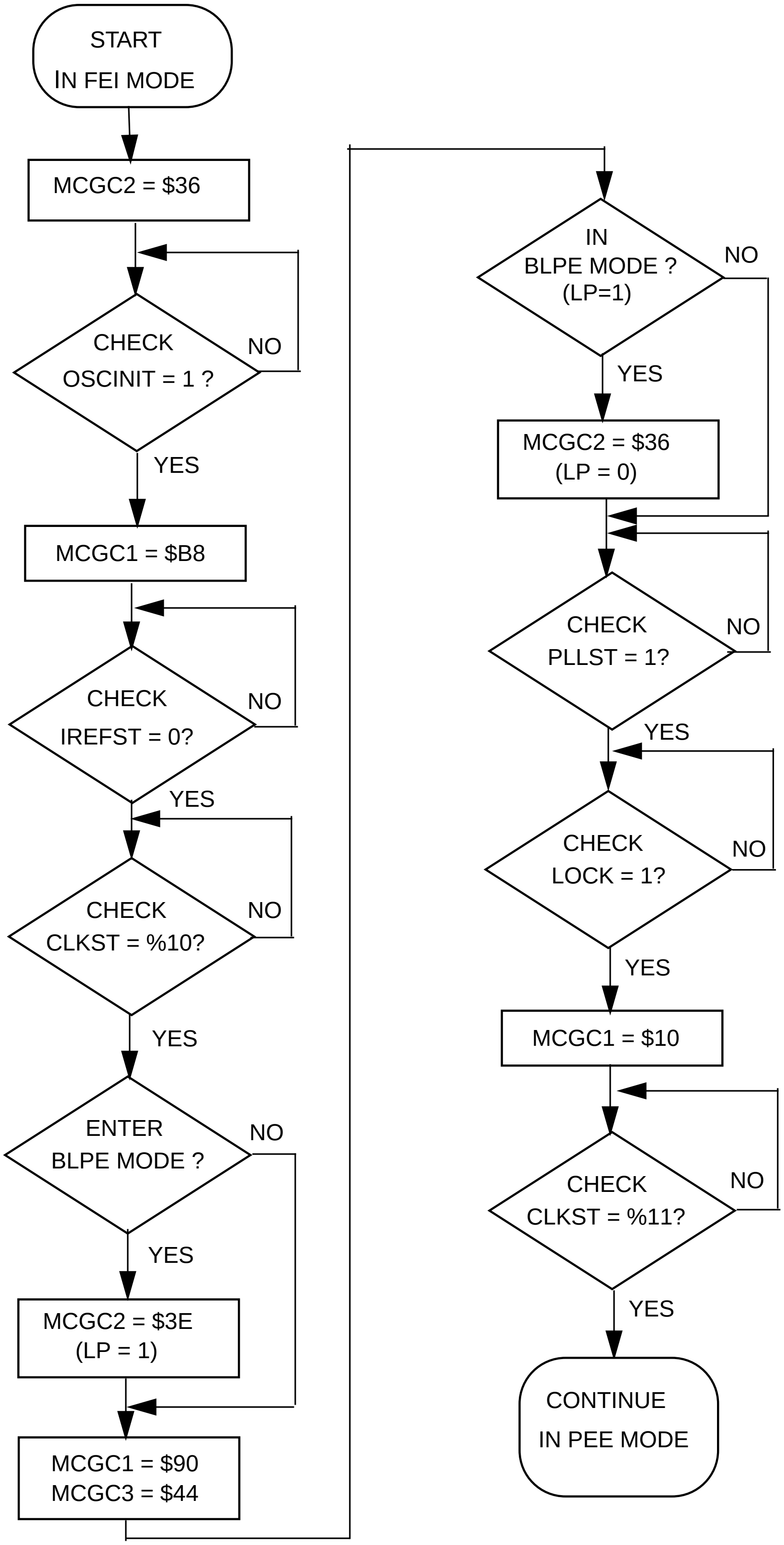

Figure 8-9. Flowchart of FEI to PEE Mode Transition using a 4 MHz crystal

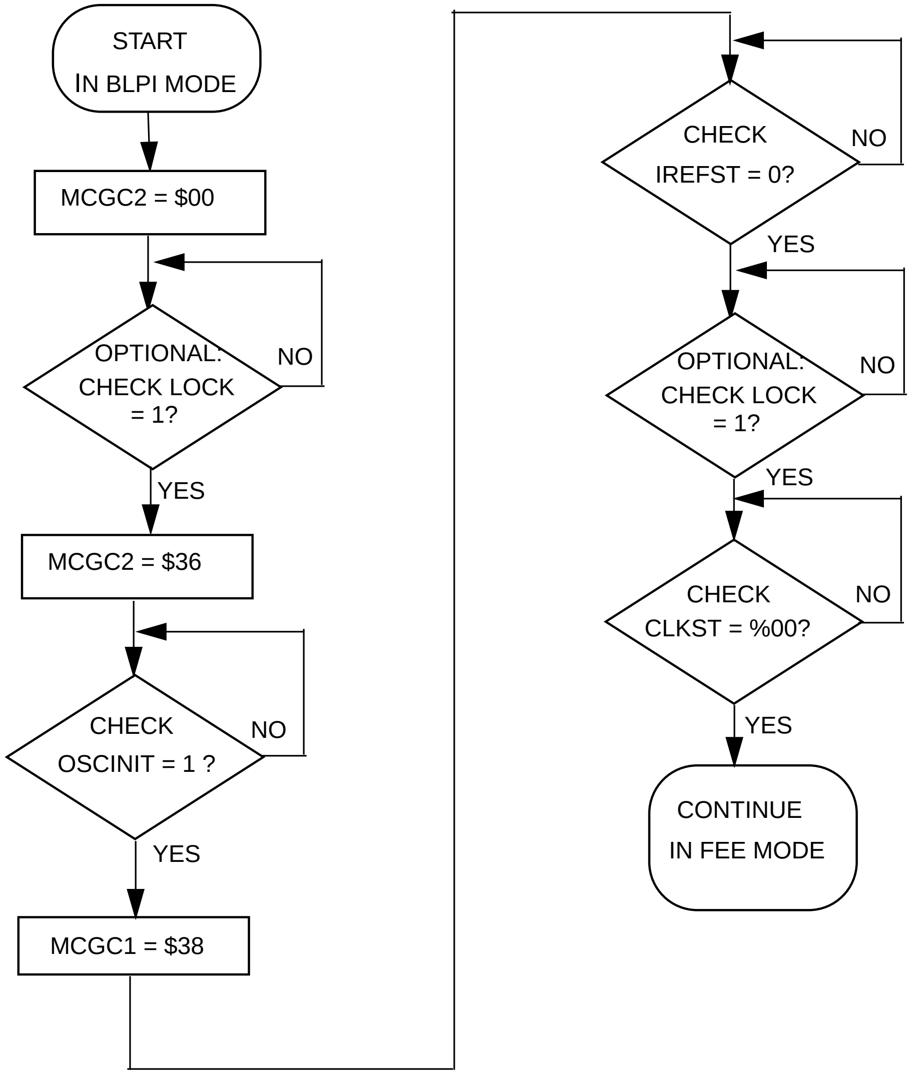

Figure 8-11. Flowchart of BLPI to FEE Mode Transition using a 4 MHz crystal

8.5.2.4 Example # 4: Moving from FEI to PEE Mode: External Crystal = 8 MHz, Bus Frequency = 8 MHz

In this example, the MCG will move through the proper operational modes from FEI to PEE mode until the 8 MHz crystal reference frequency is set to achieve a bus frequency of 8 MHz.

This example is similar to example number one except that in this case the frequency of the external crystal is 8 MHz instead of 4 MHz. Special consideration must be taken with this case since there is a period of time along the way from FEI mode to PEE mode where the FLL operates based on a reference clock with a frequency that is greater than the maximum allowed for the FLL. This occurs because with an 8 MHz

10.1.5 Temperature Sensor

To use the on-chip temperature sensor, the user must perform the following:

- Configure ADC for long sample with a maximum of 1 MHz clock
- Convert the bandgap voltage reference channel (AD27)
  - By converting the digital value of the bandgap voltage reference channel using the value of V_{BG} the user can determine V_{DD}. For value of bandgap voltage, see Section A.6, “DC Characteristics”.
- Convert the temperature sensor channel (AD26)
  - By using the calculated value of V_{DD}, convert the digital value of AD26 into a voltage, V_{TEMP}

Equation 10-1 provides an approximate transfer function of the temperature sensor.

$$\text{Temp} = 25 - ((V_{TEMP} - V_{TEMP25}) \div m) \qquad \textit{Eqn. 10-1}$$

where:

- V_{TEMP} is the voltage of the temperature sensor channel at the ambient temperature.
- V_{TEMP25} is the voltage of the temperature sensor channel at 25°C.
- m is the hot or cold voltage versus temperature slope in V/°C.

For temperature calculations, use the V_{TEMP25} and m values from the ADC Electricals table.

In application code, the user reads the temperature sensor channel, calculates V_{TEMP}, and compares to V_{TEMP25}. If V_{TEMP} is greater than V_{TEMP25} the cold slope value is applied in Equation 10-1. If V_{TEMP} is less than V_{TEMP25} the hot slope value is applied in Equation 10-1. To improve accuracy the user should calibrate the bandgap voltage reference and temperature sensor.

Calibrating at 25°C will improve accuracy to ± 4.5°C.

Calibration at three points, -40°C, 25°C, and 125°C will improve accuracy to ± 2.5°C. Once calibration has been completed, the user will need to calculate the slope for both hot and cold. In application code, the user would then calculate the temperature using Equation 10-1 as detailed above and then determine if the temperature is above or below 25°C. Once determined if the temperature is above or below 25°C, the user can recalculate the temperature using the hot or cold slope value obtained during calibration.

Table 12-22. CANIDMR0–CANIDMR3 Register Field Descriptions

Field	Description
7:0 AM[7:0]	**Acceptance Mask Bits** — If a particular bit in this register is cleared, this indicates that the corresponding bit in the identifier acceptance register must be the same as its identifier bit before a match is detected. The message is accepted if all such bits match. If a bit is set, it indicates that the state of the corresponding bit in the identifier acceptance register does not affect whether or not the message is accepted. 0 Match corresponding acceptance code register and identifier bits 1 Ignore corresponding acceptance code register bit (don't care)

	7	6	5	4	3	2	1	0
R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset	0	0	0	0	0	0	0	0

Figure 12-22. MSCAN Identifier Mask Registers (Second Bank) — CANIDMR4–CANIDMR7

Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 12-23. CANIDMR4–CANIDMR7 Register Field Descriptions

Field	Description
7:0 AM[7:0]	**Acceptance Mask Bits** — If a particular bit in this register is cleared, this indicates that the corresponding bit in the identifier acceptance register must be the same as its identifier bit before a match is detected. The message is accepted if all such bits match. If a bit is set, it indicates that the state of the corresponding bit in the identifier acceptance register does not affect whether or not the message is accepted. 0 Match corresponding acceptance code register and identifier bits 1 Ignore corresponding acceptance code register bit (don't care)

12.4 Programmer's Model of Message Storage

The following section details the organization of the receive and transmit message buffers and the associated control registers.

To simplify the programmer interface, the receive and transmit message buffers have the same outline. Each message buffer allocates 16 bytes in the memory map containing a 13 byte data structure.

An additional transmit buffer priority register (TBPR) is defined for the transmit buffers. Within the last two bytes of this memory map, the MSCAN stores a special 16-bit time stamp, which is sampled from an internal timer after successful transmission or reception of a message. This feature is only available for transmit and receiver buffers if the TIME bit is set (see Section 12.3.1, "MSCAN Control Register 0 (CANCTL0)").

The time stamp register is written by the MSCAN. The CPU can only read these registers.

13.5.2 SPI Interrupts

There are three flag bits, two interrupt mask bits, and one interrupt vector associated with the SPI system. The SPI interrupt enable mask (SPIE) enables interrupts from the SPI receiver full flag (SPRF) and mode fault flag (MODF). The SPI transmit interrupt enable mask (SPTIE) enables interrupts from the SPI transmit buffer empty flag (SPTEF). When one of the flag bits is set, and the associated interrupt mask bit is set, a hardware interrupt request is sent to the CPU. If the interrupt mask bits are cleared, software can poll the associated flag bits instead of using interrupts. The SPI interrupt service routine (ISR) should check the flag bits to determine what event caused the interrupt. The service routine should also clear the flag bit(s) before returning from the ISR (usually near the beginning of the ISR).

13.5.3 Mode Fault Detection

A mode fault occurs and the mode fault flag (MODF) becomes set when a master SPI device detects an error on the $\overline{\text{SS}}$ pin (provided the $\overline{\text{SS}}$ pin is configured as the mode fault input signal). The $\overline{\text{SS}}$ pin is configured to be the mode fault input signal when MSTR = 1, mode fault enable is set (MODFEN = 1), and slave select output enable is clear (SSOE = 0).

The mode fault detection feature can be used in a system where more than one SPI device might become a master at the same time. The error is detected when a master’s $\overline{\text{SS}}$ pin is low, indicating that some other SPI device is trying to address this master as if it were a slave. This could indicate a harmful output driver conflict, so the mode fault logic is designed to disable all SPI output drivers when such an error is detected.

When a mode fault is detected, MODF is set and MSTR is cleared to change the SPI configuration back to slave mode. The output drivers on the SPSCK, MOSI, and MISO (if not bidirectional mode) are disabled.

MODF is cleared by reading it while it is set, then writing to the SPI control register 1 (SPIC1). User software should verify the error condition has been corrected before changing the SPI back to master mode.

The TPM channels are programmable independently as input capture, output compare, or edge-aligned PWM channels. Alternately, the TPM can be configured to produce CPWM outputs on all channels. When the TPM is configured for CPWMs, the counter operates as an up/down counter; input capture, output compare, and EPWM functions are not practical.

If a channel is configured as input capture, an internal pullup device may be enabled for that channel. The details of how a module interacts with pin controls depends upon the chip implementation because the I/O pins and associated general purpose I/O controls are not part of the module. Refer to the discussion of the I/O port logic in a full-chip specification.

Because center-aligned PWMs are usually used to drive 3-phase AC-induction motors and brushless DC motors, they are typically used in sets of three or six channels.

16.2 Signal Description

Table 16-1 shows the user-accessible signals for the TPM. The number of channels may be varied from one to eight. When an external clock is included, it can be shared with the same pin as any TPM channel; however, it could be connected to a separate input pin. Refer to the I/O pin descriptions in full-chip specification for the specific chip implementation.

Table 16-1. Signal Properties

Name	Function
EXTCLK[1]	External clock source which may be selected to drive the TPM counter.
TPMxCHn[2]	I/O pin associated with TPM channel n

[1] When preset, this signal can share any channel pin; however depending upon full-chip implementation, this signal could be connected to a separate external pin.

[2] n=channel number (1 to 8)

Refer to documentation for the full-chip for details about reset states, port connections, and whether there is any pullup device on these pins.

TPM channel pins can be associated with general purpose I/O pins and have passive pullup devices which can be enabled with a control bit when the TPM or general purpose I/O controls have configured the associated pin as an input. When no TPM function is enabled to use a corresponding pin, the pin reverts to being controlled by general purpose I/O controls, including the port-data and data-direction registers. Immediately after reset, no TPM functions are enabled, so all associated pins revert to general purpose I/O control.

16.2.1 Detailed Signal Descriptions

This section describes each user-accessible pin signal in detail. Although Table 16-1 grouped all channel pins together, any TPM pin can be shared with the external clock source signal. Since I/O pin logic is not part of the TPM, refer to full-chip documentation for a specific derivative for more details about the interaction of TPM pin functions and general purpose I/O controls including port data, data direction, and pullup controls.

	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 16-13. TPM 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 16-14. TPM 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 half is read. This latching mechanism also resets (becomes unlatched) when the TPMxCnSC register is written (whether BDM mode is active or not). Any write to the channel registers will be ignored during the input capture mode.

When BDM is active, the coherency mechanism is frozen (unless reset by writing to TPMxCnSC register) such that the buffer latches remain in the state they were in when the BDM became active, even if one or both halves of the channel register are read while BDM is active. This assures that if the user was in the middle of reading a 16-bit register when BDM became active, it will read the appropriate value from the other half of the 16-bit value after returning to normal execution. The value read from the TPMxCnVH and TPMxCnVL registers in BDM mode is the value of these registers and not the value of their read buffer.

In output compare or PWM modes, writing to either byte (TPMxCnVH or TPMxCnVL) latches the value into a buffer. After both bytes are written, they are transferred as a coherent 16-bit value into the timer-channel registers according to the value of CLKSB:CLKSA bits and the selected mode, so:

- If (CLKSB:CLKSA = 0:0), then the registers are updated when the second byte is written.
- If (CLKSB:CLKSA not = 0:0 and in output compare mode) then the registers are updated after the second byte is written and on the next change of the TPM counter (end of the prescaler counting).
- If (CLKSB:CLKSA not = 0:0 and in EPWM or CPWM modes), then the registers are updated after the both bytes were written, and the TPM counter changes from (TPMxMODH:TPMxMODL - 1) to (TPMxMODH:TPMxMODL). If the TPM counter is a free-running counter then the update is made when the TPM counter changes from 0xFFFE to 0xFFFF.

The latching mechanism may be manually reset by writing to the TPMxCnSC register (whether BDM mode is active or not). This latching mechanism allows coherent 16-bit writes in either big-endian or little-endian order which is friendly to various compiler implementations.

When BDM is active, the coherency mechanism is frozen such that the buffer latches remain in the state they were in when the BDM became active even if one or both halves of the channel register are written while BDM is active. Any write to the channel registers bypasses the buffer latches and directly write to the channel register while BDM is active. The values written to the channel register while BDM is active

17.1.2 Features

Features of the BDC module include:

- Single pin for mode selection and background communications
- BDC registers are not located in the memory map
- SYNC command to determine target communications rate
- Non-intrusive commands for memory access
- Active background mode commands for CPU register access
- GO and TRACE1 commands
- BACKGROUND command can wake CPU from stop or wait modes
- One hardware address breakpoint built into BDC
- Oscillator runs in stop mode, if BDC enabled
- COP watchdog disabled while in active background mode

Features of the ICE system include:

- Two trigger comparators: Two address + read/write (R/W) or one full address + data + R/W
- Flexible 8-word by 16-bit FIFO (first-in, first-out) buffer for capture information:
  - Change-of-flow addresses or
  - Event-only data
- Two types of breakpoints:
  - Tag breakpoints for instruction opcodes
  - Force breakpoints for any address access
- Nine trigger modes:
  - Basic: A-only, A OR B
  - Sequence: A then B
  - Full: A AND B data, A AND NOT B data
  - Event (store data): Event-only B, A then event-only B
  - Range: Inside range (A ≤ address ≤ B), outside range (address < A or address > B)

17.2 Background Debug Controller (BDC)

All MCUs in the HCS08 Family contain a single-wire background debug interface that supports in-circuit programming of on-chip nonvolatile memory and sophisticated non-intrusive debug capabilities. Unlike debug interfaces on earlier 8-bit MCUs, this system does not interfere with normal application resources. It does not use any user memory or locations in the memory map and does not share any on-chip peripherals.

BDC commands are divided into two groups:

- Active background mode commands require that the target MCU is in active background mode (the user program is not running). Active background mode commands allow the CPU registers to be read or written, and allow the user to trace one user instruction at a time, or GO to the user program from active background mode.

Figure 17-3 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.

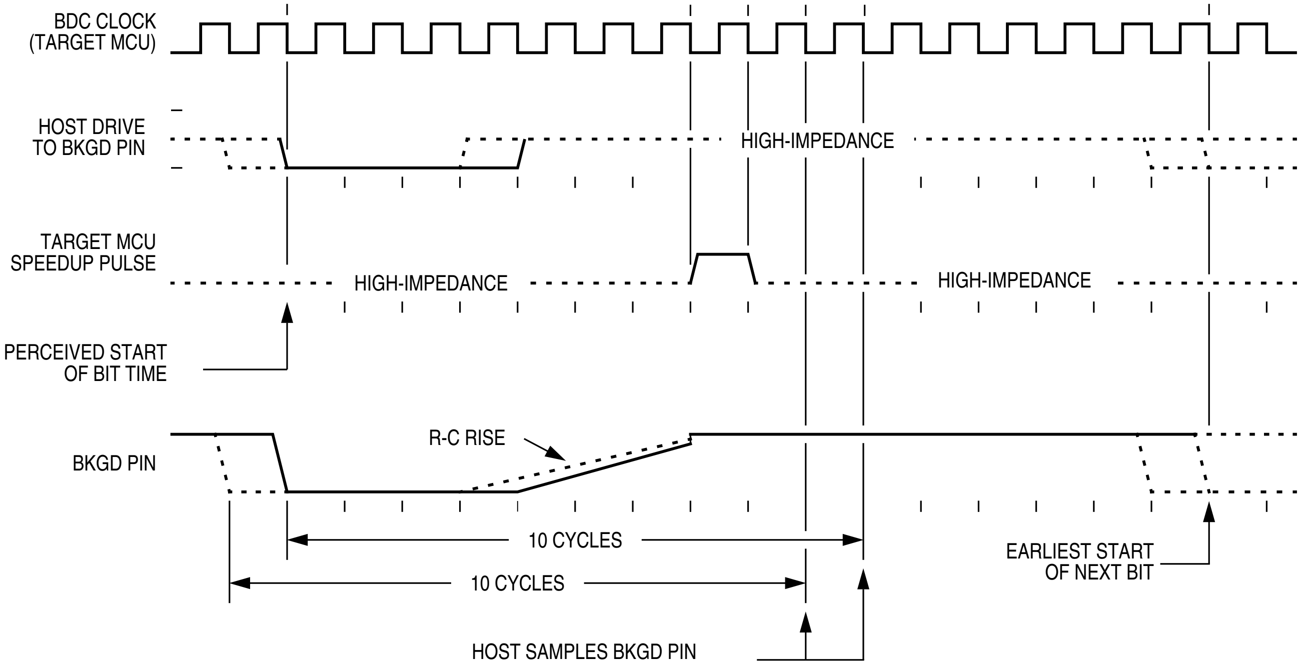

Figure 17-3. BDC Target-to-Host Serial Bit Timing (Logic 1)

where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations 1 and 2 iteratively for any value of T_A.

A.5 ESD Protection and Latch-Up Immunity

Although damage from electrostatic discharge (ESD) is much less common on these devices than on early CMOS circuits, normal handling precautions should be used to avoid exposure to static discharge. Qualification tests are performed to ensure that these devices can withstand exposure to reasonable levels of static without suffering any permanent damage.

All ESD testing is in conformity with AEC-Q100 Stress Test Qualification for Automotive Grade Integrated Circuits. During the device qualification ESD stresses were performed for the Human Body Model (HBM) and the Charge Device Model (CDM).

A device is defined as a failure if after exposure to ESD pulses the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature followed by hot temperature, unless specified otherwise in the device specification.

Table A-4. ESD and Latch-up Test Conditions

Model	Description	Symbol	Value	Unit
Human Body	Series Resistance	R1	1500	Ω
	Storage Capacitance	C	100	pF
	Number of Pulse per pin	–	3	
Latch-up	Minimum input voltage limit		–2.5	V
	Maximum input voltage limit		7.5	V

Table A-5. ESD and Latch-Up Protection Characteristics

Num	Rating	Symbol	Min	Max	Unit
1	Human Body Model (HBM)	V_{HBM}	+/- 2000	–	V
2	Charge Device Model (CDM)	V_{CDM}	+/- 500	–	V
3	Latch-up Current at T_A = 125°C	I_{LAT}	+/- 100	–	mA

[7] Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive and negative clamp voltages, then use the larger of the two values.

[8] PTE1 does not have a clamp diode to V_{DD}. Do not drive PTE1 above V_{DD}.

A.7 Supply Current Characteristics

Table A-7. Supply Current Characteristics

Num	C	Parameter	Symbol	V_{DD} (V)	Typical[1]	Max[2]	Unit
1	C	Run supply current[3] measured at (CPU clock = 2 MHz, f_{Bus} = 1 MHz)	RI_{DD}	5	3	7.5	mA
	C			3	2.8	7.4	
2	P	Run supply current[3] measured at (CPU clock = 16 MHz, f_{Bus} = 8 MHz)	RI_{DD}	5	7.7	11.4	mA
	C			3	7.4	11.2	
3	P	Run supply current[3] measured at (CPU clock = 40 MHz, f_{Bus} = 20 MHz)	RI_{DD}	5	15	24	mA
	C			3	14	23	
4	P[4]	Stop3 mode supply current –40 °C (C, V, & M suffix)	$S3I_{DD}$	5	0.9	—	µA
	P[4]	25 °C (All parts)			1.0	—	
	P	105 °C (V suffix only)			26	39	
	P	125 °C (M suffix only)			62	90	
	C	–40 °C (C, V, & M suffix)		3	0.8	—	
	C	25 °C (All parts)			0.9	—	
	C	105 °C (V suffix only)			21	32	
	C	125 °C (M suffix only)			52	80	
5	P[4]	Stop2 mode supply current –40 °C (C, V, & M suffix)	$S2I_{DD}$	5	0.8	—	µA
	P[4]	25 °C (All parts)			0.9	—	
	P	105 °C (V suffix only)			25	37	
	P	125 °C (M suffix only)			46	70	
	C	–40 °C (C, V, & M suffix)		3	0.7	—	
	C	25 °C (All parts)			0.8	—	
	C	105 °C (V suffix only)			20	30	
	C	125 °C (M suffix only)			40	60	

the value in the TPMxCNTH:TPMxCNTL counter is 0x0000. (The new duty cycle does not take effect until the next full period.)

B.3.3 Center-Aligned PWM Mode

This type of PWM output uses the up-/down-counting mode of the timer counter (CPWMS = 1). The output compare value in TPMxCnVH:TPMxCnVL determines the pulse width (duty cycle) of the PWM signal and the period is determined by the value in TPMxMODH:TPMxMODL. TPMxMODH:TPMxMODL should be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results. ELSnA will determine the polarity of the CPWM output.

pulse width = 2 x (TPMxCnVH:TPMxCnVL) *Eqn. 17-1*

period = 2 x (TPMxMODH:TPMxMODL);
for TPMxMODH:TPMxMODL = 0x0001–0x7FFF *Eqn. 17-2*

If the channel value register TPMxCnVH:TPMxCnVL is zero or negative (bit 15 set), the duty cycle will be 0%. If TPMxCnVH:TPMxCnVL is a positive value (bit 15 clear) and is greater than the (nonzero) modulus setting, the duty cycle will be 100% because the duty cycle compare will never occur. This implies the usable range of periods set by the modulus register is 0x0001 through 0x7FFE (0x7FFF if generation of 100% duty cycle is not necessary). This is not a significant limitation because the resulting period is much longer than required for normal applications.

TPMxMODH:TPMxMODL = 0x0000 is a special case that should not be used with center-aligned PWM mode. When CPWMS = 0, this case corresponds to the counter running free from 0x0000 through 0xFFFF, but when CPWMS = 1 the counter needs a valid match to the modulus register somewhere other than at 0x0000 in order to change directions from up-counting to down-counting.

Figure B-11 shows the output compare value in the TPM channel registers (multiplied by 2), which determines the pulse width (duty cycle) of the CPWM signal. If ELSnA = 0, the compare match while counting up forces the CPWM output signal low and a compare match while counting down forces the output high. The counter counts up until it reaches the modulo setting in TPMxMODH:TPMxMODL, then counts down until it reaches zero. This sets the period equal to two times TPMxMODH:TPMxMODL.

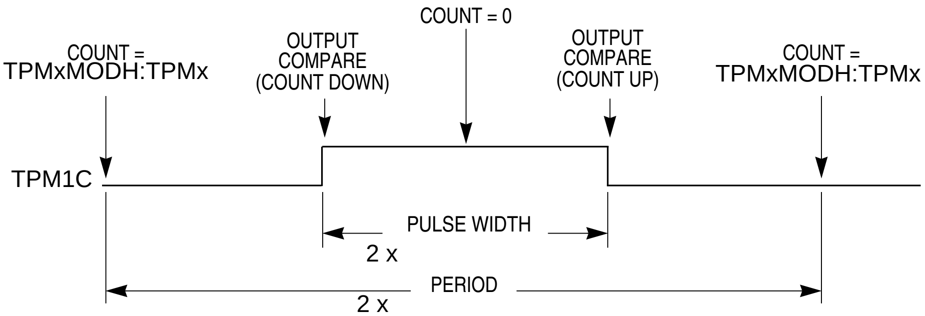

Figure B-11. CPWM Period and Pulse Width (ELSnA = 0)

Center-aligned PWM outputs typically produce less noise than edge-aligned PWMs because fewer I/O pin transitions are lined up at the same system clock edge. This type of PWM is also required for some types of motor drives.

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.)

B.4.3 Channel Event Interrupt Description

The meaning of channel interrupts depends on the current mode of the channel (input capture, output compare, edge-aligned PWM, or center-aligned PWM).

When a channel is configured as an input capture channel, the ELSnB:ELSnA control bits select rising edges, falling edges, any edge, or no edge (off) as the edge that triggers an input capture event. When the selected edge is detected, the interrupt flag is set. The flag is cleared by the 2-step sequence described in Section B.4.1, "Clearing Timer Interrupt Flags."

When a channel is configured as an output compare channel, the interrupt flag is set each time the main timer counter matches the 16-bit value in the channel value register. The flag is cleared by the 2-step sequence described in Section B.4.1, "Clearing Timer Interrupt Flags."

B.4.4 PWM End-of-Duty-Cycle Events

For channels that are configured for PWM operation, there are two possibilities:

- When the channel is configured for edge-aligned PWM, the channel flag is set when the timer counter matches the channel value register that marks the end of the active duty cycle period.
- When the channel is configured for center-aligned PWM, the timer count matches the channel value register twice during each PWM cycle. In this CPWM case, the channel flag is set at the start and at the end of the active duty cycle, which are the times when the timer counter matches the channel value register.

The flag is cleared by the 2-step sequence described in Section B.4.1, "Clearing Timer Interrupt Flags."