



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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, I ² C, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	60
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 2.75V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12e128vfue

0x0140 – 0x016F TIM1 (Timer 16 Bit 4 Channels) (Sheet 3 of 4)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0157	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0158	TC4 (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x0159	TC4 (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x015A	TC5 (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x015B	TC5 (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x015C	TC6 (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x015D	TC6 (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x015E	TC7 (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x015F	TC7 (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0160	PACTL	R	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
		W								
0x0161	PAFLG	R	0	0	0	0	0	0	PAOVF	PAIF
		W								
0x0162	PACNT (hi)	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x0163	PACNT (lo)	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0164	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0165	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0166	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0167	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0168	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0169	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0240 – 0x027F PIM (Port Interface Module) (Sheet 2 of 4)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x024A	DDRS	R	DDRS7	DDRS6	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
		W								
0x024B	RDRS	R	RDRS7	RDRS6	RDRS5	RDRS4	RDRS3	RDRS2	RDRS1	RDRS0
		W								
0x024C	PERS	R	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0
		W								
0x024D	PPSS	R	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0
		W								
0x024E	WOMS	R	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0
		W								
0x024F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0250	PTM	R	PTM7	PTM6	PTM5	PTM4	PTM3	0	PTM1	PTM0
		W								
0x0251	PTIM	R	PTIM7	PTIM6	PTIM5	PTIM4	PTIM3	0	PTIM1	PTIM0
		W								
0x0252	DDRM	R	DDRM7	DDRM6	DDRM5	DDRM4	DDRM3	0	DDRM1	DDRM0
		W								
0x0253	RDRM	R	RDRM7	RDRM6	RDRM5	RDRM4	RDRM3	0	RDRM1	RDRM0
		W								
0x0254	PERM	R	PERM7	PERM6	PERM5	PERM4	PERM3	0	PERM1	PERM0
		W								
0x0255	PPSM	R	PPSM7	PPSM6	PPSM5	PPSM4	PPSM3	0	PPSM1	PPSM0
		W								
0x0256	WOMM	R	WOMM7	WOMM6	WOMM5	WOMM4	0	0	0	0
		W								
0x0257	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0258	PTP	R	0	0	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
		W								
0x0259	PTIP	R	0	0	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
		W								
0x025A	DDRP	R	0	0	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
		W								
0x025B	RDRP	R	0	0	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
		W								
0x025C	PERP	R	0	0	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
		W								

3.3 Memory Map and Register Definition

This section provides a detailed description of all registers. Table 3-2 is a standard memory map of port integration module.

Table 3-2. PIM9HZ256 Memory Map

Address Offset	Use	Access
0x0000	Port T I/O Register (PTT)	R/W
0x0001	Port T Input Register (PTIT)	R
0x0002	Port T Data Direction Register (DDRT)	R/W
0x0003	Port T Reduced Drive Register (RDRT)	R/W
0x0004	Port T Pull Device Enable Register (PERT)	R/W
0x0005	Port T Polarity Select Register (PPST)	R/W
0x0006 - 0x0007	Reserved	—
0x0008	Port S I/O Register (PTS)	R/W
0x0009	Port S Input Register (PTIS)	R
0x000A	Port S Data Direction Register (DDRS)	R/W
0x000B	Port S Reduced Drive Register (RDRS)	R/W
0x000C	Port S Pull Device Enable Register (PERS)	R/W
0x000D	Port S Polarity Select Register (PPSS)	R/W
0x000E	Port S Wired-OR Mode Register (WOMS)	R/W
0x000F	Reserved	—
0x0010	Port M I/O Register (PTM)	R/W
0x0011	Port M Input Register (PTIM)	R
0x0012	Port M Data Direction Register (DDRM)	R/W
0x0013	Port M Reduced Drive Register (RDRM)	R/W
0x0014	Port M Pull Device Enable Register (PERM)	R/W
0x0015	Port M Polarity Select Register (PPSM)	R/W
0x0016	Port M Wired-OR Mode Register (WOMM)	R/W
0x0017	Reserved	—
0x0018	Port P I/O Register (PTP)	R/W
0x0019	Port P Input Register (PTIP)	R
0x001A	Port P Data Direction Register (DDRP)	R/W
0x001B	Port P Reduced Drive Register (RDRP)	R/W
0x001C	Port P Pull Device Enable Register (PERP)	R/W
0x001D	Port P Polarity Select Register (PPSP)	R/W
0x001E - 0x001F	Reserved	—

3.3.1.8 Port AD Interrupt Flag Register (PIFAD)

	7	6	5	4	3	2	1	0
R	PIFAD15	PIFAD14	PIFAD13	PIFAD12	PIFAD11	PIFAD10	PIFAD9	PIFAD8
W								
Reset	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
R	PIFAD7	PIFAD6	PIFAD5	PIFAD4	PIFAD3	PIFAD2	PIFAD1	PIFAD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 3-9. Port AD Interrupt Flag Register (PIFAD)

Read: Anytime. Write: Anytime.

Each flag is set by an active edge on the associated input pin. The active edge could be rising or falling based on the state of the corresponding PPSADx bit. To clear each flag, write “1” to the corresponding PIFADx bit. Writing a “0” has no effect.

NOTE

If the ATDDIEN0(1) bit of the associated pin is set to 0 (digital input buffer is disabled), active edges can not be detected.

Table 3-8. PIFAD Field Descriptions

Field	Description
15:0 PIFAD[15:0]	Interrupt Flags Port AD 0 No active edge pending. Writing a “0” has no effect. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set). Writing a “1” clears the associated flag.

4.3.2.6 CRG Clock Select Register (CLKSEL)

This register controls CRG clock selection. Refer to [Figure 4-17](#) for details on the effect of each bit.

	7	6	5	4	3	2	1	0
R	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI
W								
Reset	0	0	0	0	0	0	0	0

Figure 4-9. CRG Clock Select Register (CLKSEL)

Read: anytime

Write: refer to each bit for individual write conditions

Table 4-4. CLKSEL Field Descriptions

Field	Description
7 PLLSEL	<p>PLL Select Bit — Write anytime. Writing a 1 when LOCK = 0 and AUTO = 1, or TRACK = 0 and AUTO = 0 has no effect. This prevents the selection of an unstable PLLCLK as SYSCCLK. PLLSEL bit is cleared when the MCU enters self-clock mode, stop mode or wait mode with PLLWAI bit set.</p> <p>0 System clocks are derived from OSCCLK (Bus Clock = OSCCLK / 2).</p> <p>1 System clocks are derived from PLLCLK (Bus Clock = PLLCLK / 2).</p>
6 PSTP	<p>Pseudo-Stop Bit — Write: anytime — This bit controls the functionality of the oscillator during stop mode.</p> <p>0 Oscillator is disabled in stop mode.</p> <p>1 Oscillator continues to run in stop mode (pseudo-stop). The oscillator amplitude is reduced. Refer to oscillator block description for availability of a reduced oscillator amplitude.</p> <p>Note: Pseudo-stop allows for faster stop recovery and reduces the mechanical stress and aging of the resonator in case of frequent stop conditions at the expense of a slightly increased power consumption.</p> <p>Note: Lower oscillator amplitude exhibits lower power consumption but could have adverse effects during any electro-magnetic susceptibility (EMS) tests.</p>
5 SYSWAI	<p>System Clocks Stop in Wait Mode Bit — Write: anytime</p> <p>0 In wait mode, the system clocks continue to run.</p> <p>1 In wait mode, the system clocks stop.</p> <p>Note: RTI and COP are not affected by SYSWAI bit.</p>
4 ROAWAI	<p>Reduced Oscillator Amplitude in Wait Mode Bit — Write: anytime — Refer to oscillator block description chapter for availability of a reduced oscillator amplitude. If no such feature exists in the oscillator block then setting this bit to 1 will not have any effect on power consumption.</p> <p>0 Normal oscillator amplitude in wait mode.</p> <p>1 Reduced oscillator amplitude in wait mode.</p> <p>Note: Lower oscillator amplitude exhibits lower power consumption but could have adverse effects during any electro-magnetic susceptibility (EMS) tests.</p>
3 PLLWAI	<p>PLL Stops in Wait Mode Bit — Write: anytime — If PLLWAI is set, the CRGV4 will clear the PLLSEL bit before entering wait mode. The PLLON bit remains set during wait mode but the PLL is powered down. Upon exiting wait mode, the PLLSEL bit has to be set manually if PLL clock is required.</p> <p>While the PLLWAI bit is set the AUTO bit is set to 1 in order to allow the PLL to automatically lock on the selected target frequency after exiting wait mode.</p> <p>0 PLL keeps running in wait mode.</p> <p>1 PLL stops in wait mode.</p>
2 CWAI	<p>Core Stops in Wait Mode Bit — Write: anytime</p> <p>0 Core clock keeps running in wait mode.</p> <p>1 Core clock stops in wait mode.</p>

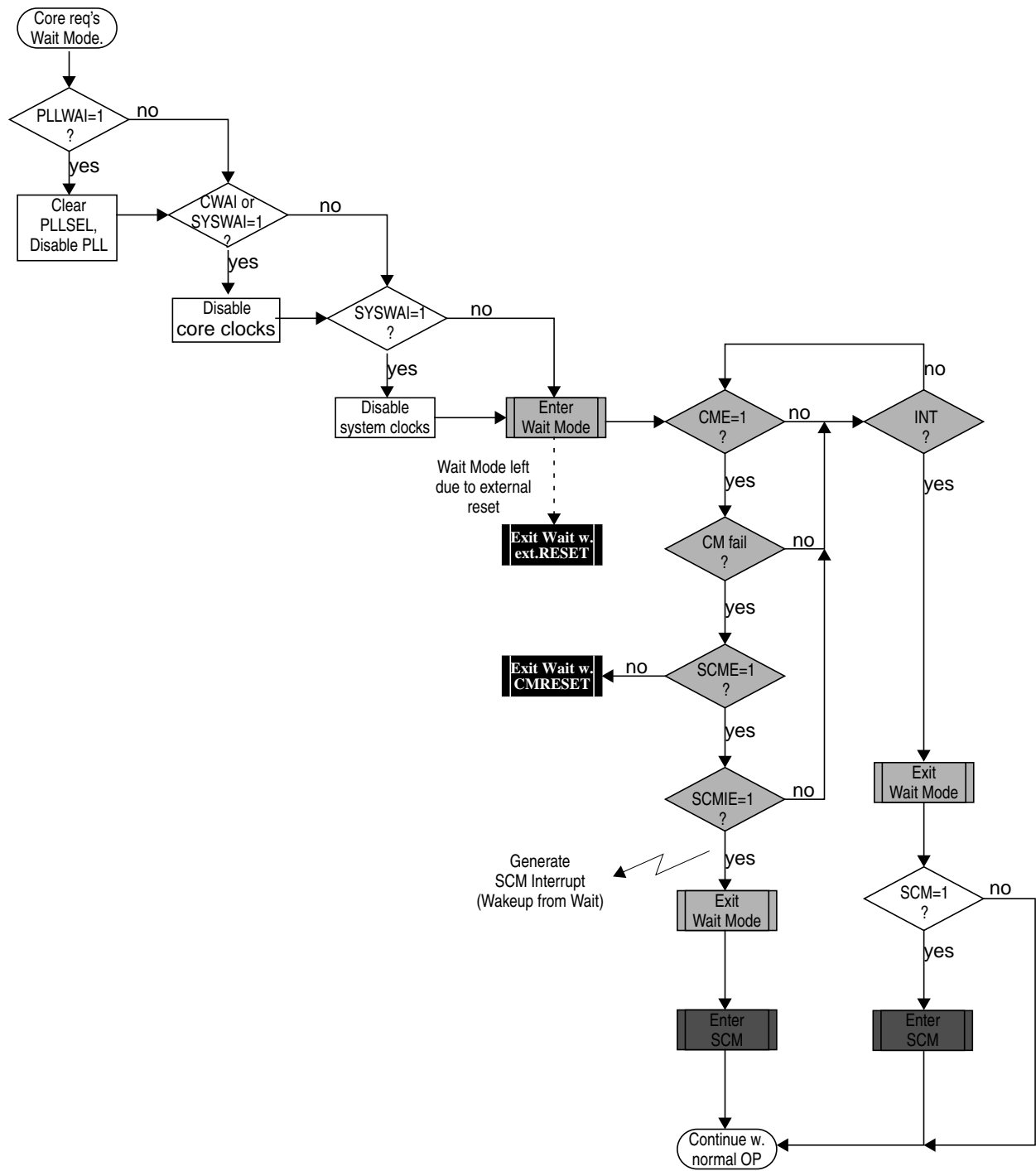


Figure 4-23. Wait Mode Entry/Exit Sequence

8.4.1 Infrared Interface Submodule

This module provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the SCI. The IrDA physical layer specification defines a half-duplex infrared communication link for exchange data. The full standard includes data rates up to 16 Mbits/s. This design covers only data rates between 2.4 kbits/s and 115.2 kbits/s.

The infrared submodule consists of two major blocks: the transmit encoder and the receive decoder. The SCI transmits serial bits of data which are encoded by the infrared submodule to transmit a narrow pulse for every 0 bit. No pulse is transmitted for every 1 bit. When receiving data, the IR pulses should be detected using an IR photo diode and transformed to CMOS levels by the IR receive decoder (external from the MCU). The narrow pulses are then stretched by the infrared submodule to get back to a serial bit stream to be received by the SCI. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that uses active low pulses.

The infrared submodule receives its clock sources from the SCI. One of these two clocks are selected in the infrared submodule in order to generate either 3/16, 1/16, or 1/32 narrow pulses during transmission. The infrared block receives two clock sources from the SCI, R16XCLK, and R32XCLK, which are configured to generate the narrow pulse width during transmission. The R16XCLK and R32XCLK are internal clocks with frequencies 16 and 32 times the baud rate respectively. Both R16XCLK and R32XCLK clocks are used for transmitting data. The receive decoder uses only the R16XCLK clock.

8.4.1.1 Infrared Transmit Encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the TXD pin. A narrow pulse is transmitted for a 0 bit and no pulse for a 1 bit. The narrow pulse is sent in the middle of the bit with a duration of 1/32, 1/16, or 3/16 of a bit time.

8.4.1.2 Infrared Receive Decoder

The infrared receive block converts data from the RXD pin to the receive shift register. A narrow pulse is expected for each 0 received and no pulse is expected for each 1 received. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

8.4.2 Data Format

The SCI uses the standard NRZ mark/space data format. When Infrared is enabled, the SCI uses RZI data format where 0s are represented by light pulses and 1s remain low. See [Figure 8-12](#).

10.2 External Signal Description

The IICV2 module has two external pins.

10.2.1 IIC_SCL — Serial Clock Line Pin

This is the bidirectional serial clock line (SCL) of the module, compatible to the IIC bus specification.

10.2.2 IIC_SDA — Serial Data Line Pin

This is the bidirectional serial data line (SDA) of the module, compatible to the IIC bus specification.

10.3 Memory Map and Register Definition

This section provides a detailed description of all memory and registers for the IIC module.

10.3.1 Module Memory Map

The memory map for the IIC module is given below in [Table 10-1](#). The address listed for each register is the address offset. The total address for each register is the sum of the base address for the IIC module and the address offset for each register.

11.3.2.20 PMF Enable Control A Register (PMFENCA)

Module Base + 0x0020

	7	6	5	4	3	2	1	0
R	PWMENA	0	0	0	0	0	LDOKA	PWMRIEA
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 11-26. PMF Enable Control A Register (PMFENCA)

Read and write anytime.

Table 11-25. PMFENCA Field Descriptions

Field	Description
7 PWMENA	<p>PWM Generator A Enable — When MTG is clear, this bit when set enables the PWM generators A, B and C and the PWM0–5 pins. When PWMENA is clear, PWM generators A, B and C are disabled, and the PWM0–5 pins are in their inactive states unless the corresponding OUTCTLx bits are set.</p> <p>When MTG is set, this bit when set enables the PWM generator A and the PWM0 and PWM1 pins. When PWMENA is clear, the PWM generator A is disabled and PWM0 and PWM1 pins are in their inactive states unless the OUTCTL0 and OUTCTL1 bits are set.</p> <p>0 PWM generator A and PWM0–1 (2–5 if MTG=0) pins disabled unless the respective OUTCTL bit is set. 1 PWM generator A and PWM0–1 (2–5 if MTG=0) pins enabled.</p>
1 LDOKA	<p>Load Okay A — When MTG is clear, this bit allows loads of the PRSCA bits, the PMFMODA register and the PWMVAL0–5 registers into a set of buffers. The buffered prescaler A divisor, PWM counter modulus A value, and all PWM pulse widths take effect at the next PWM reload.</p> <p>When MTG is set, this bit allows loads of the PRSCA bits, the PMFMODA register and the PWMVAL0–1 registers into a set of buffers. The buffered prescaler divisor A, PWM counter modulus A value, PWM0–1 pulse widths take effect at the next PWM reload.</p> <p>Set LDOKA by reading it when it is logic zero and then writing a logic one to it. LDOKA is automatically cleared after the new values are loaded, or can be manually cleared before a reload by writing a logic zero to it. Reset clears LDOKA.</p> <p>0 Do not load new modulus A, prescaler A, and PWM0–1 (2–5 if MTG=0) values 1 Load prescaler A, modulus A, and PWM0–1 (2–5 if MTG=0) values</p> <p>Note: Do not set PWMENA bit before setting the LDOKA bit and do not clear the LDOKA bit at the same time as setting the PWMENA bit.</p>
0 PWMRIEA	<p>PWM Reload Interrupt Enable A — This bit enables the PWMRFA flag to generate CPU interrupt requests.</p> <p>0 PWMRFA CPU interrupt requests disabled 1 PWMRFA CPU interrupt requests enabled</p>

11.3.2.30 PMF Enable Control C Register (PMFENCC)

Module Base + 0x0030

	7	6	5	4	3	2	1	0
R	PWMENC	0	0	0	0	0	LDOKC	PWMRIEC
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 11-36. PMF Enable Control C Register (PMFENCC)

Read anytime and write only if MTG is set.

Table 11-39. PMFENCC Field Descriptions

Field	Description
7 PWMENC	<p>PWM Generator C Enable — If MTG is clear, this bit reads zero and cannot be written.</p> <p>If MTG is set, this bit when set enables the PWM generator C and the PWM4 and PWM5 pins. When PWMENC is clear, PWM generator C is disabled, and the PWM4 and PWM5 pins are in their inactive states unless the OUTCTL4 and OUTCTL5 bits are set.</p> <p>0 PWM generator C and PWM4–5 pins disabled unless the respective OUTCTL bit is set. 1 PWM generator C and PWM4–5 pins enabled.</p>
1 LDOKC	<p>Load Okay C — If MTG is clear, this bit reads zero and can not be written.</p> <p>If MTG is set, this bit loads the PRSCC bits, the PMFMODEC register and the PWMVAL4–5 registers into a set of buffers. The buffered prescaler divisor C, PWM counter modulus C value, PWM4–5 pulse widths take effect at the next PWM reload.</p> <p>Set LDOKC by reading it when it is logic zero and then writing a logic one to it. LDOKC is automatically cleared after the new values are loaded, or can be manually cleared before a reload by writing a logic zero to it. Reset clears LDOKC.</p> <p>0 Do not load new modulus C, prescaler C, and PWM4–5 values. 1 Load prescaler C, modulus C, and PWM4–5 values.</p> <p>Note: Do not set PWMENC bit before setting the LDOKC bit and do not clear the LDOKC bit at the same time as setting the PWMENC bit.</p>
0 PWMRIEC	<p>PWM Reload Interrupt Enable C — If MTG is clear, this bit reads zero and cannot be written.</p> <p>If MTG is set, this bit enables the PWMRFC flag to generate CPU interrupt requests.</p> <p>0 PWMRFC CPU interrupt requests disabled 1 PWMRFC CPU interrupt requests enabled</p>

Table 11-42. PWM Prescaler C

PRSCC	PWM Clock Frequency
00	f_{bus}
01	$f_{bus}/2$
10	$f_{bus}/4$
11	$f_{bus}/8$

11.3.2.32 PMF Counter C Register (PMFCNTC)

Module Base + 0x0032

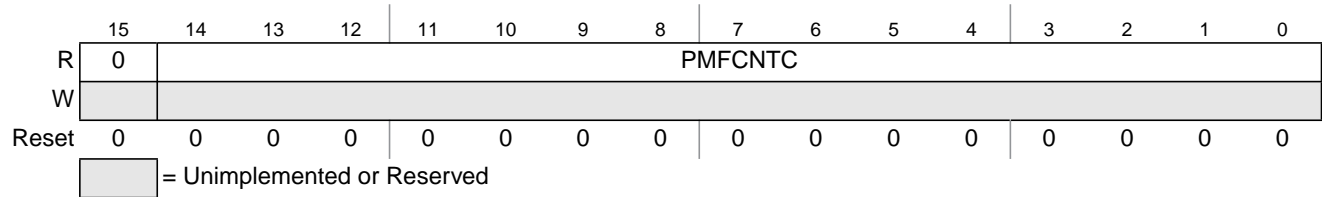


Figure 11-38. PMF Counter C Register (PMFCNTC)

Read anytime and writes have no effect.

Table 11-43. PMFCNTC Field Descriptions

Field	Description
14–0 PMFCNTC'	PMF Counter C — This register displays the state of the 15-bit PWM C counter.

11.3.2.33 PMF Counter Modulo C Register (PMFMODC)

Module Base + 0x0034

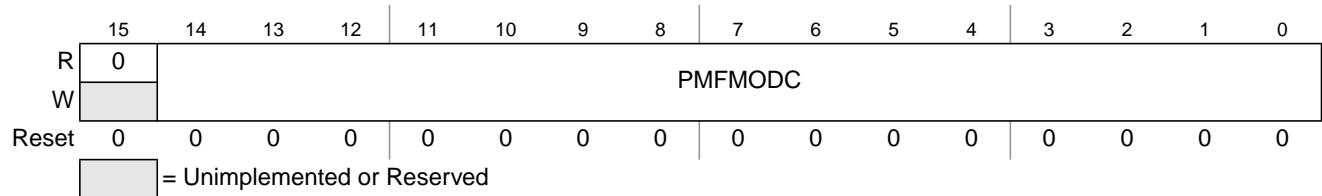


Figure 11-39. PMF Counter Modulo C Register (PMFMODC)

Read anytime and write only if MTG is set.

Table 11-44. PMFMODC Field Descriptions

Field	Description
14–0 PMFMODC	PMF Couter Modulo C — The 15-bit unsigned value written to this register is the PWM period in PWM clock periods. Do not write a modulus value of zero. Note: The PWM counter modulo register is buffered. The value written does not take effect until the LDOKC bit is set and the next PWM load cycle begins. Reading PMFMODC reads the value in the buffer. It is not necessarily the value the PWM generator A is currently using.

12.3.2.1 PWM Enable Register (PWME)

Each PWM channel has an enable bit (PWME_x) to start its waveform output. When any of the PWME_x bits are set (PWME_x = 1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWME_x and the clock source.

NOTE

The first PWM cycle after enabling the channel can be irregular.

An exception to this is when channels are concatenated. After concatenated mode is enabled (CON_{xx} bits set in PWMCTL register), enabling/disabling the corresponding 16-bit PWM channel is controlled by the low-order PWME_x bit. In this case, the high-order bytes PWME_x bits have no effect and their corresponding PWM output lines are disabled.

While in run mode, if all six PWM channels are disabled (PWME5–PWME0 = 0), the prescaler counter shuts off for power savings.

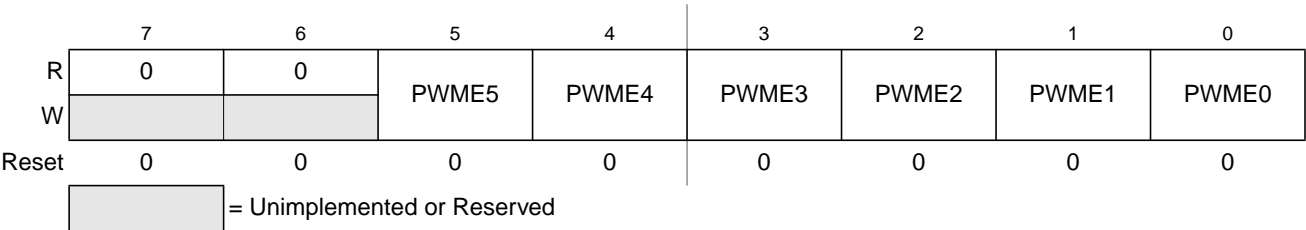


Figure 12-3. PWM Enable Register (PWME)

Read: anytime

Write: anytime

Table 12-2. PWME Field Descriptions

Field	Description
5 PWME5	Pulse Width Channel 5 Enable 0 Pulse width channel 5 is disabled. 1 Pulse width channel 5 is enabled. The pulse modulated signal becomes available at PWM,output bit 5 when its clock source begins its next cycle.
4 PWME4	Pulse Width Channel 4 Enable 0 Pulse width channel 4 is disabled. 1 Pulse width channel 4 is enabled. The pulse modulated signal becomes available at PWM, output bit 4 when its clock source begins its next cycle. If CON45 = 1, then bit has no effect and PWM output line 4 is disabled.
3 PWME3	Pulse Width Channel 3 Enable 0 Pulse width channel 3 is disabled. 1 Pulse width channel 3 is enabled. The pulse modulated signal becomes available at PWM, output bit 3 when its clock source begins its next cycle.
2 PWME2	Pulse Width Channel 2 Enable 0 Pulse width channel 2 is disabled. 1 Pulse width channel 2 is enabled. The pulse modulated signal becomes available at PWM, output bit 2 when its clock source begins its next cycle. If CON23 = 1, then bit has no effect and PWM output line 2 is disabled.

Table 13-14. Timer Clock Selection

PR2	PR1	PR0	Timer Clock
0	0	0	Bus Clock / 1
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

NOTE

The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

13.3.2.12 Main Timer Interrupt Flag 1 (TFLG1)

	7	6	5	4	3	2	1	0
R	C7F	C6F	C5F	C4F	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Figure 13-18. Main Timer Interrupt Flag 1 (TFLG1)

Read: Anytime

Write: Used in the clearing mechanism (set bits cause corresponding bits to be cleared). Writing a zero will not affect current status of the bit.

Table 13-15. TRLG1 Field Descriptions

Field	Description
7:4 C[7:4]F	Input Capture/Output Compare Channel “x” Flag — These flags are set when an input capture or output compare event occurs. Clear a channel flag by writing one to it. When TFFCA bit in TSCR register is set, a read from an input capture or a write into an output compare channel (0x0010–0x001F) will cause the corresponding channel flag CxF to be cleared.

Table 15-2. BDMSTS Field Descriptions (continued)

Field	Description
5 ENTAG	Tagging Enable — This bit indicates whether instruction tagging is enabled or disabled. It is set when the TAGGO command is executed and cleared when BDM is entered. The serial system is disabled and the tag function enabled 16 cycles after this bit is written. BDM cannot process serial commands while tagging is active. 0 Tagging not enabled or BDM active 1 Tagging enabled
4 SDV	Shift Data Valid — This bit is set and cleared by the BDM hardware. It is set after data has been transmitted as part of a firmware read command or after data has been received as part of a firmware write command. It is cleared when the next BDM command has been received or BDM is exited. SDV is used by the standard BDM firmware to control program flow execution. 0 Data phase of command not complete 1 Data phase of command is complete
3 TRACE	TRACE1 BDM Firmware Command is Being Executed — This bit gets set when a BDM TRACE1 firmware command is first recognized. It will stay set as long as continuous back-to-back TRACE1 commands are executed. This bit will get cleared when the next command that is not a TRACE1 command is recognized. 0 TRACE1 command is not being executed 1 TRACE1 command is being executed
2 CLKSW	Clock Switch — The CLKSW bit controls which clock the BDM operates with. It is only writable from a hardware BDM command. A 150 cycle delay at the clock speed that is active during the data portion of the command will occur before the new clock source is guaranteed to be active. The start of the next BDM command uses the new clock for timing subsequent BDM communications. Table 15-3 shows the resulting BDM clock source based on the CLKSW and the PLLSEL (PLL select from the clock and reset generator) bits. Note: The BDM alternate clock source can only be selected when CLKSW = 0 and PLLSEL = 1. The BDM serial interface is now fully synchronized to the alternate clock source, when enabled. This eliminates frequency restriction on the alternate clock which was required on previous versions. Refer to the device overview section to determine which clock connects to the alternate clock source input. Note: If the acknowledge function is turned on, changing the CLKSW bit will cause the ACK to be at the new rate for the write command which changes it.
1 UNSEC	Unsecure — This bit is only writable in special single-chip mode from the BDM secure firmware and always gets reset to zero. It is in a zero state as secure mode is entered so that the secure BDM firmware lookup table is enabled and put into the memory map along with the standard BDM firmware lookup table. The secure BDM firmware lookup table verifies that the on-chip EEPROM and FLASH EEPROM are erased. This being the case, the UNSEC bit is set and the BDM program jumps to the start of the standard BDM firmware lookup table and the secure BDM firmware lookup table is turned off. If the erase test fails, the UNSEC bit will not be asserted. 0 System is in a secured mode 1 System is in a unsecured mode Note: When UNSEC is set, security is off and the user can change the state of the secure bits in the on-chip FLASH EEPROM. Note that if the user does not change the state of the bits to “unsecured” mode, the system will be secured again when it is next taken out of reset.

Table 15-3. BDM Clock Sources

PLLSEL	CLKSW	BDMCLK
0	0	Bus clock
0	1	Bus clock

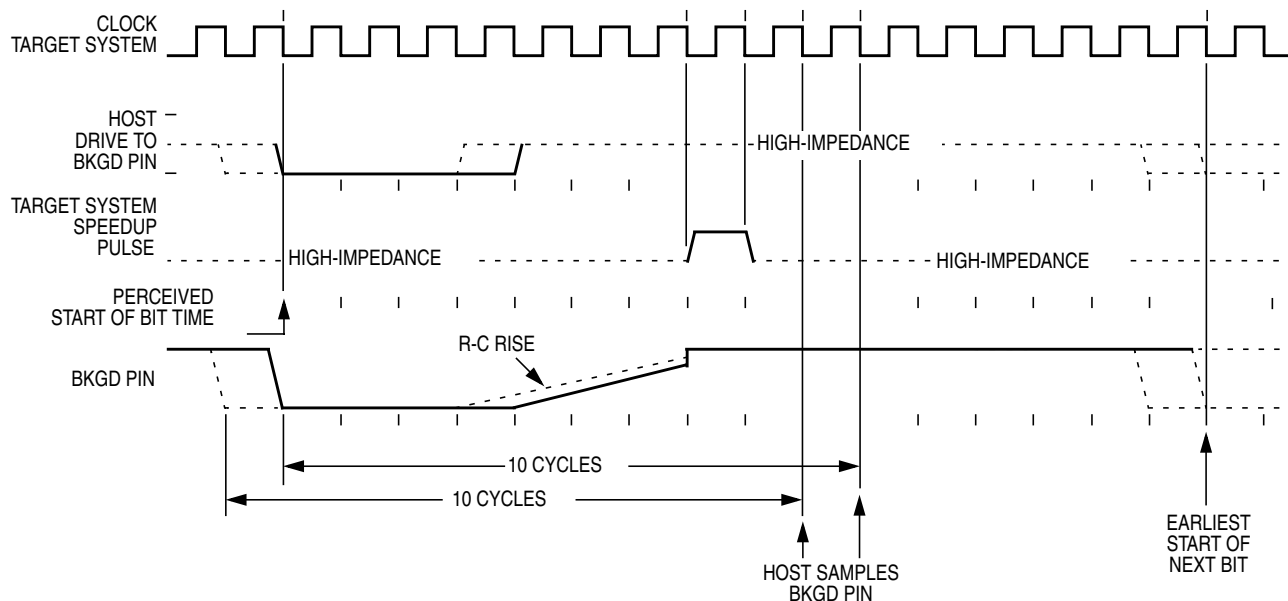


Figure 15-8. BDM Target-to-Host Serial Bit Timing (Logic 1)

Figure 15-9 shows the host receiving a logic 0 from the target. Because the host is asynchronous to the target, there is up to a one clock-cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target. The host initiates the bit time but the target finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

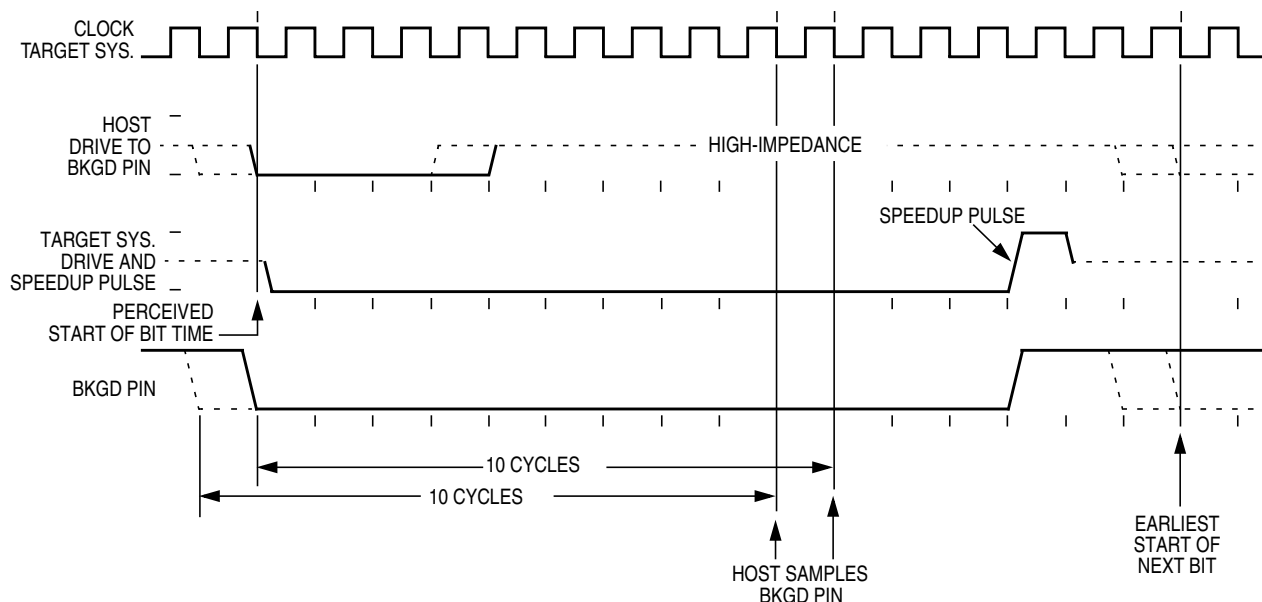


Figure 15-9. BDM Target-to-Host Serial Bit Timing (Logic 0)

least six addresses higher than address A (or B is lower than A) and there are not changes of flow to put these in the queue at the same time, then this operation should trigger properly.

16.4.2.5.4 Event-Only B (Store Data)

In the event-only B trigger mode, if the match condition for B is met, the B flag in DBGSC is set and a trigger occurs. The event-only B trigger mode is considered a begin-trigger type and the BEGIN bit in DBGSC1 is ignored. Event-only B is incompatible with instruction tagging (TRGSEL = 1), and thus the value of TRGSEL is ignored. Please refer to [Section 16.4.2.7, “Storage Memory,”](#) for more information.

This trigger mode is incompatible with the detail capture mode so the detail capture mode will have priority. TRGSEL and BEGIN will not be ignored and this trigger mode will behave as if it were “B only”.

16.4.2.5.5 A then Event-Only B (Store Data)

In the A then event-only B trigger mode, the match condition for A must be met before the match condition for B is compared, after the A match has occurred, a trigger occurs each time B matches. When the match condition for A or B is met, the corresponding flag in DBGSC is set. The A then event-only B trigger mode is considered a begin-trigger type and BEGIN in DBGSC1 is ignored. TRGSEL in DBGSC1 applies only to the match condition for A. Please refer to [Section 16.4.2.7, “Storage Memory,”](#) for more information.

This trigger mode is incompatible with the detail capture mode so the detail capture mode will have priority. TRGSEL and BEGIN will not be ignored and this trigger mode will be the same as A then B.

16.4.2.5.6 A and B (Full Mode)

In the A and B trigger mode, comparator A compares to the address bus and comparator B compares to the data bus. In the A and B trigger mode, if the match condition for A and B happen on the same bus cycle, both the A and B flags in the DBGSC register are set and a trigger occurs.

If TRGSEL = 1, only matches from comparator A are used to determine if the trigger condition is met and comparator B matches are ignored. If TRGSEL = 0, full-word data matches on an odd address boundary (misaligned access) do not work unless the access is to a RAM that manages misaligned accesses in a single clock cycle (which is typical of RAM modules used in HCS12 MCUs).

16.4.2.5.7 A and Not B (Full Mode)

In the A and not B trigger mode, comparator A compares to the address bus and comparator B compares to the data bus. In the A and not B trigger mode, if the match condition for A and not B happen on the same bus cycle, both the A and B flags in DBGSC are set and a trigger occurs.

If TRGSEL = 1, only matches from comparator A are used to determine if the trigger condition is met and comparator B matches are ignored. As described in [Section 16.4.2.5.6, “A and B \(Full Mode\),”](#) full-word data compares on misaligned accesses will not match expected data (and thus will cause a trigger in this mode) unless the access is to a RAM that manages misaligned accesses in a single clock cycle.

18.4.3 Modes of Operation

The operating mode out of reset is determined by the states of the MODC, MODB, and MODA pins during reset (Table 18-16). The MODC, MODB, and MODA bits in the MODE register show the current operating mode and provide limited mode switching during operation. The states of the MODC, MODB, and MODA pins are latched into these bits on the rising edge of the reset signal.

Table 18-16. Mode Selection

MODC	MODB	MODA	Mode Description
0	0	0	Special Single Chip, BDM allowed and ACTIVE. BDM is allowed in all other modes but a serial command is required to make BDM active.
0	0	1	Emulation Expanded Narrow, BDM allowed
0	1	0	Special Test (Expanded Wide), BDM allowed
0	1	1	Emulation Expanded Wide, BDM allowed
1	0	0	Normal Single Chip, BDM allowed
1	0	1	Normal Expanded Narrow, BDM allowed
1	1	0	Peripheral; BDM allowed but bus operations would cause bus conflicts (must not be used)
1	1	1	Normal Expanded Wide, BDM allowed

There are two basic types of operating modes:

1. Normal modes: Some registers and bits are protected against accidental changes.
2. Special modes: Allow greater access to protected control registers and bits for special purposes such as testing.

A system development and debug feature, background debug mode (BDM), is available in all modes. In special single-chip mode, BDM is active immediately after reset.

Some aspects of Port E are not mode dependent. Bit 1 of Port E is a general purpose input or the $\overline{\text{IRQ}}$ interrupt input. $\overline{\text{IRQ}}$ can be enabled by bits in the CPU's condition codes register but it is inhibited at reset so this pin is initially configured as a simple input with a pull-up. Bit 0 of Port E is a general purpose input or the $\overline{\text{XIRQ}}$ interrupt input. $\overline{\text{XIRQ}}$ can be enabled by bits in the CPU's condition codes register but it is inhibited at reset so this pin is initially configured as a simple input with a pull-up. The ESTR bit in the EBICTL register is set to one by reset in any user mode. This assures that the reset vector can be fetched even if it is located in an external slow memory device. The PE6/MODB/IPIPE1 and PE5/MODA/IPIPE0 pins act as high-impedance mode select inputs during reset.

The following paragraphs discuss the default bus setup and describe which aspects of the bus can be changed after reset on a per mode basis.

- Calculates the address of the next instruction after the CALL instruction (the return address), and pushes this 16-bit value onto the stack.
- Pushes the old PPAGE value onto the stack.
- Calculates the effective address of the subroutine, refills the queue, and begins execution at the new address on the selected page of the expansion window.

This sequence is uninterruptable; there is no need to inhibit interrupts during CALL execution. A CALL can be performed from any address in memory to any other address.

The PPAGE value supplied by the instruction is part of the effective address. For all addressing mode variations except indexed-indirect modes, the new page value is provided by an immediate operand in the instruction. In indexed-indirect variations of CALL, a pointer specifies memory locations where the new page value and the address of the called subroutine are stored. Using indirect addressing for both the new page value and the address within the page allows values calculated at run time rather than immediate values that must be known at the time of assembly.

The RTC instruction terminates subroutines invoked by a CALL instruction. RTC unstacks the PPAGE value and the return address and refills the queue. Execution resumes with the next instruction after the CALL.

During the execution of an RTC instruction, the CPU:

- Pulls the old PPAGE value from the stack
- Pulls the 16-bit return address from the stack and loads it into the PC
- Writes the old PPAGE value into the PPAGE register
- Refills the queue and resumes execution at the return address

This sequence is uninterruptable; an RTC can be executed from anywhere in memory, even from a different page of extended memory in the expansion window.

The CALL and RTC instructions behave like JSR and RTS, except they use more execution cycles. Therefore, routinely substituting CALL/RTC for JSR/RTS is not recommended. JSR and RTS can be used to access subroutines that are on the same page in expanded memory. However, a subroutine in expanded memory that can be called from other pages must be terminated with an RTC. And the RTC unstacks a PPAGE value. So any access to the subroutine, even from the same page, must use a CALL instruction so that the correct PPAGE value is in the stack.

19.4.3.2 Extended Address (XAB19:14) and $\overline{\text{ECS}}$ Signal Functionality

If the EMK bit in the MODE register is set (see MEBI block description chapter) the PIX5:0 values will be output on XAB19:14 respectively (port K bits 5:0) when the system is addressing within the physical program page window address space (0x8000–0xBFFF) and is in an expanded mode. When addressing anywhere else within the physical address space (outside of the paging space), the XAB19:14 signals will be assigned a constant value based upon the physical address space selected. In addition, the active-low emulation chip select signal, $\overline{\text{ECS}}$, will likewise function based upon the assigned memory allocation. In the cases of 48K byte and 64K byte allocated physical FLASH/ROM space, the operation of the $\overline{\text{ECS}}$ signal will additionally depend upon the state of the ROMHM bit (see [Section 19.3.2.4, “Miscellaneous System Control Register \(MISC\)”](#)) in the MISC register. [Table 19-18](#), [Table 19-19](#), [Table 19-20](#), and

A.3.1.4 External Reset

When external reset is asserted for a time greater than PW_{RSTL} the CRG module generates an internal reset, and the CPU starts fetching the reset vector without doing a clock quality check, if there was an oscillation before reset.

A.3.1.5 Stop Recovery

Out of STOP the controller can be woken up by an external interrupt. A clock quality check as after POR is performed before releasing the clocks to the system.

A.3.1.6 Pseudo Stop and Wait Recovery

The recovery from Pseudo STOP and Wait are essentially the same since the oscillator was not stopped in both modes. The controller can be woken up by internal or external interrupts. After t_{WRS} the CPU starts fetching the interrupt vector.

The loop bandwidth f_C should be chosen to fulfill the Gardner's stability criteria by at least a factor of 10, typical values are 50. $\zeta = 0.9$ ensures a good transient response.

$$f_C < \frac{2 \cdot \zeta \cdot f_{\text{ref}}}{\pi \cdot \left(\zeta + \sqrt{1 + \zeta^2} \right)} \cdot \frac{1}{10} \rightarrow f_C < \frac{f_{\text{ref}}}{4 \cdot 10}; (\zeta = 0.9)$$

$$f_C < 25\text{kHz}$$

And finally the frequency relationship is defined as

$$n = \frac{f_{\text{VCO}}}{f_{\text{ref}}} = 2 \cdot (\text{synr} + 1) = 50$$

With the above values the resistance can be calculated. The example is shown for a loop bandwidth $f_C=10\text{kHz}$:

$$R = \frac{2 \cdot \pi \cdot n \cdot f_C}{K_{\Phi}} = 2 \cdot \pi \cdot 50 \cdot 10\text{kHz} / (316.7\text{Hz}/\Omega) = 9.9\text{k}\Omega \approx 10\text{k}\Omega$$

The capacitance C_s can now be calculated as:

$$C_s = \frac{2 \cdot \zeta^2}{\pi \cdot f_C \cdot R} \approx \frac{0.516}{f_C \cdot R}; (\zeta = 0.9) = 5.19\text{nF} \approx 4.7\text{nF}$$

The capacitance C_p should be chosen in the range of:

$$C_s/20 \leq C_p \leq C_s/10 \quad C_p = 470\text{pF}$$

A.3.3.2 Jitter Information

The basic functionality of the PLL is shown in [Figure A-2](#). With each transition of the clock f_{cmp} , the deviation from the reference clock f_{ref} is measured and input voltage to the VCO is adjusted accordingly. The adjustment is done continuously with no abrupt changes in the clock output frequency. Noise, voltage, temperature and other factors cause slight variations in the control loop resulting in a clock jitter. This jitter affects the real minimum and maximum clock periods as illustrated in [Figure A-3](#).