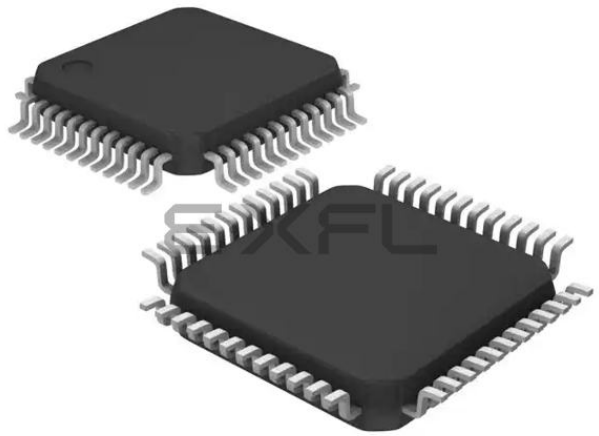

Welcome to E-XFL.COM

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	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	28
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12vr64f2clf

Chapter 1
Device Overview MC9S12VR-Family

Table 1-1. Revision History

Version Number	Revision Date	Description of Changes
1.0	26-November-2010	• Added Block Diagram • Minor Corrections from Shared Review
2.0	11-April-2011	• New Revision for Maskset N05E PartID=$3201 • Added 6 PWM Channels • Pinout changes for PWM channels

1.1 Introduction

The MC9S12VR-Family is an optimized automotive 16-bit microcontroller product line focused on low-cost, high-performance, and low pin-count. This family integrates an S12 microcontroller with a LIN Physical interface, a 5V regulator system to supply the microcontroller, and analog blocks to control other elements of the system which operate at vehicle battery level (e.g. relay drivers, high-side driver outputs, wake up inputs). The MC9S12VR-Family is targeted at generic automotive applications requiring single node LIN communications. Typical examples of these applications include window lift modules, seat modules and sun-roof modules to name a few.

The MC9S12VR-Family uses many of the same features found on the MC9S12G family, including error correction code (ECC) on flash memory, EEPROM for diagnostic or data storage, a fast analog-to-digital converter (ADC) and a frequency modulated phase locked loop (IPLL) that improves the EMC performance. The MC9S12VR-Family delivers an optimized solution with the integration of several key system components into a single device, optimizing system architecture and achieving significant space savings. The MC9S12VR-Family delivers all the advantages and efficiencies of a 16-bit MCU while retaining the low cost, power consumption, EMC, and code-size efficiency advantages currently enjoyed by users of Freescale's existing 8-bit and 16-bit MCU families. Like the MC9S12XS family, the MC9S12VR-Family will run 16-bit wide accesses without wait states for all peripherals and memories. Misaligned single cycle 16 bit RAM access is not supported. The MC9S12VR-Family will be available in 32-pin and 48-pin LQFP. In addition to the I/O ports available in each module, further I/O ports are available with interrupt capability allowing wake-up from stop or wait modes.

The MC9S12VR-Family is a general-purpose family of devices created with relay based motor control in mind and is suitable for a range of applications, including:

- Window lift modules
- Door modules
- Seat controllers
- Smart actuators

Table 1-12. Initial WCOP Configuration

NV[3] in FOPT Register	WCOP in COPCTL Register
1	0
0	1

1.14 ADC External Trigger Input Connection

The ADC module includes external trigger inputs ETRIG0, ETRIG1, ETRIG2, and ETRIG3. The external trigger allows the user to synchronize ADC conversion to external trigger events. ETRIG0 is connected to PP0 / PWM0 and ETRIG1 is connected to PP1 / PWM1. ETRIG2 and ETRIG3 are not used .ETRIG0 can be routed to PS2 and ETRIG1 can be routed to PS3.

1.15 ADC Special Conversion Channels

Whenever the ADC's Special Channel Conversion Bit (SC) in 8.3.2.6 ATD Control Register 5 (ATDCTL5) is set, it is capable of running conversion on a number of internal channels. Table 1-13 lists the internal sources which are connected to these special conversion channels.

Table 1-13. Usage of ADC Special Conversion Channels

ATDCTL5 Register Bits						Usage
SC	CD	CC	CB	CA	ADC Channel	
1	0	0	0	1	Internal_7	Bandgap Voltage V_{BG} or Chip temperature sensor V_{HT} see 4.3.2.13 High Temperature Control Register (CPMUHTCTL)
1	0	0	1	0	Internal_0	Flash Supply Voltage VDDF
1	0	0	1	1	Internal_1	LINPHY temperature sensor
1	1	0	1	0	Internal_4	V_{SENSE} or V_{SUP} selectable in BATS module see 16.1.1 Features
1	1	0	1	1	Internal_5	High voltage inputs Port L see 2.3.34 Port L Analog Access Register (PTAL)

1.16 ADC Result Reference

MCUs of the MC9S12VR-Fanmily are able to measure the internal bandgap reference voltage $V_{BG}$with the analog digital converter. (see Table 1-13.) V_{BG} is a constant voltage with a narrow distribution over temperature and external voltage supply. The ADC conversion result of V_{BG} is provided at address 0x0_405A/0x0_405B in the NVM IFR for reference. By measuring the voltage V_{BG} and comparing the result to the reference value in the IFR it is possible to determine the refrence voltage of the ADC V_{RH} in the application environment.

2.3.44 Port AD Interrupt Enable Register (PIE1AD)

Address 0x027D

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	PIE1AD5	PIE1AD4	PIE1AD3	PIE1AD2	PIE1AD1	PIE1AD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-43. Port AD Interrupt Enable Register (PIE1AD)

[1] Read: Anytime
Write: Anytime

Table 2-45. PIE1AD Register Field Descriptions

Field	Description
5-0 PIE1AD	**Pin Interrupt Enable register 1 port AD** — This bit enables or disables the edge sensitive pin interrupt on the associated pin. An interrupt can be generated if the pin is operating in input or output mode when in use with the general-purpose or related peripheral function. For wakeup from stop mode this bit must be set to allow activating the RC oscillator. 1 Interrupt is enabled 0 Interrupt is disabled (interrupt flag masked)

2.3.45 Port AD Interrupt Flag Register (PIF1AD)

Address 0x027F

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	PIF1AD5	PIF1AD4	PIF1AD3	PIF1AD2	PIF1AD1	PIF1AD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-44. Port AD Interrupt Flag Register (PIF1AD)

[1] Read: Anytime
Write: Anytime, write 1 to clear

Table 2-46. PIF1AD Register Field Descriptions

Field	Description
5-0 PIF1AD	**Pin Interrupt Flag register 1 port AD** — This flag asserts after a valid active edge was detected on the related pin (Section 2.4.4, “Interrupts”). This can be a rising or a falling edge based on the state of the polarity select register. An interrupt will occur if the associated interrupt enable bit is set. 1 Active edge on the associated bit has occurred 0 No active edge occurred

voltage divider can be bypassed (PTAL[PTADIRL]=1). Additionally in latter case the impedance converter in the ADC signal path can be configured to be used or bypassed in direct input mode (PTAL[PTABYPL]).

In run mode the digital input buffer of the selected pin is disabled to avoid shoot-through current. Thus pin interrupts cannot be generated.

In stop mode the digital input buffer is enabled only if DIENL[x]=1 to support wakeup functionality.

Table 2-48 shows the HVI input configuration depending on register bits and operation mode.

Table 2-48. HVI Input Configurations

Mode	DIENL	PTAENL	Digital Input	Analog Input	Resulting Function
Run	0	0	off	off	Input disabled (Reset)
	0	1	off[1]	enabled	Analog input, interrupt not supported
	1	0	enabled	off	Digital input, interrupt supported
	1	1	off[1]	enabled	Analog input, interrupt not supported
Stop	0	0	off	off	Input disabled, wakeup from stop not supported
	0	1	off	off	
	1	0	enabled	off	Digital input, wakeup from stop supported
	1	1	enabled	off	

1 Enabled if (PTAL[PTTEL]=1 & PTAL[PTADIRL]=0)

NOTE

An external resistor R_{EXT_HVI} must always be connected to the high-voltage inputs to protect the device pins from fast transients and to achieve the specified pin input divider ratios when using the HVI in analog mode.

2.4.3.7 Port AD

This port is associated with the ADC.

Port AD pins can be used for either general-purpose I/O, or with the ADC subsystem.

2.4.4 Interrupts

This section describes the interrupts generated by the PIM and their individual sources. Vector addresses and interrupt priorities are defined at MCU level.

Table 2-49. PIM Interrupt Sources

Module Interrupt Sources	Local Enable
$\overline{\text{XIRQ}}$	None
$\overline{\text{IRQ}}$	IRQCR[IRQEN]
Port P pin interrupt	PIEP[PIEP5-PIEP0]

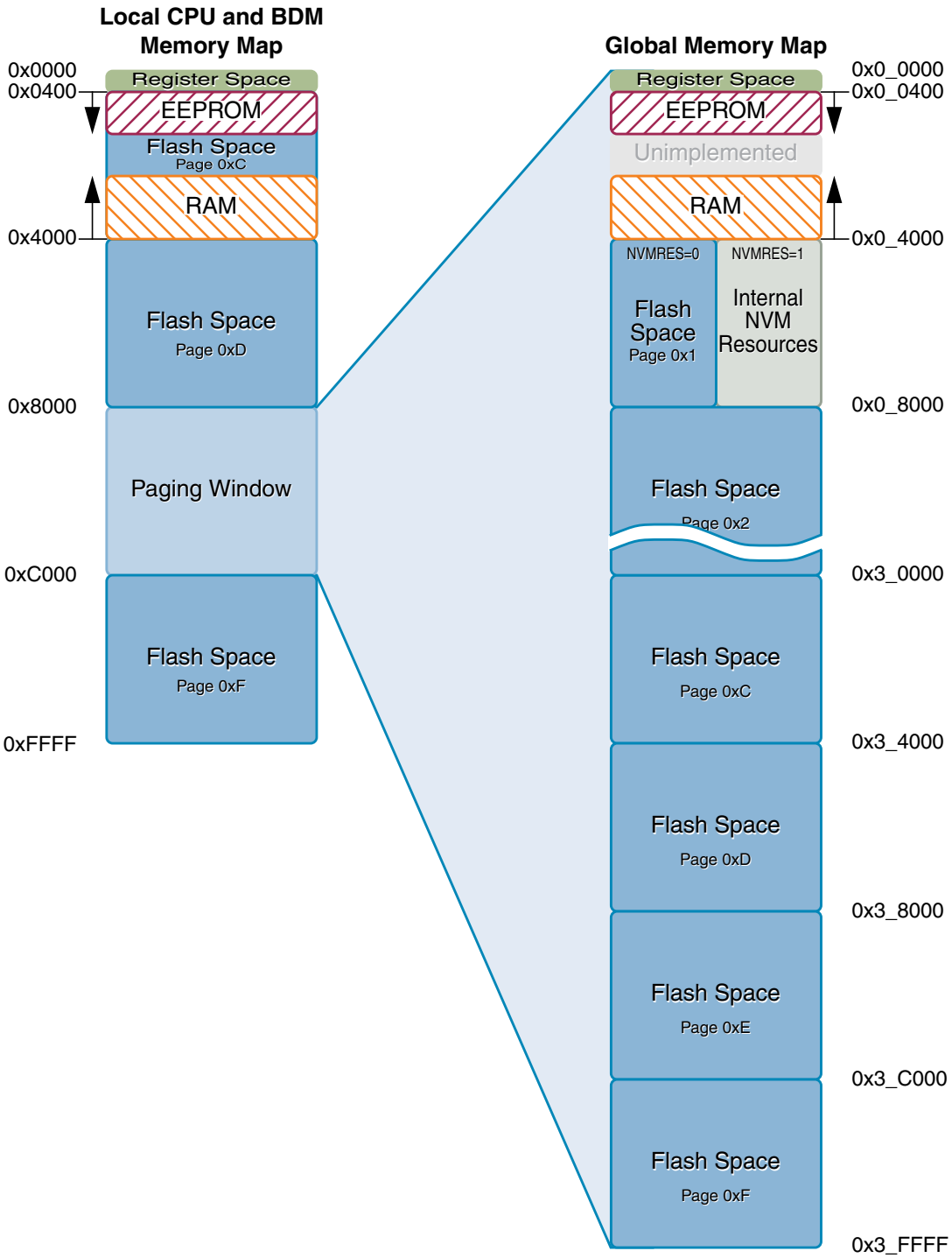

Figure 3-11. Local to Global Address Mapping

Table 4-5. CPMUCLKS Descriptions

Field	Description
7 PLLSEL	**PLL Select Bit** This bit selects the PLLCLK as source of the System Clocks (Core Clock and Bus Clock). PLLSEL can only be set to 0, if UPOSC=1. UPOSC= 0 sets the PLLSEL bit. Entering Full Stop Mode sets the PLLSEL bit. 0 System clocks are derived from OSCCLK if oscillator is up (UPOSC=1, $f_{bus} = f_{osc} / 2$). 1 System clocks are derived from PLLCLK, $f_{bus} = f_{PLL} / 2$.
6 PSTP	**Pseudo Stop Bit** This bit controls the functionality of the oscillator during Stop Mode. 0 Oscillator is disabled in Stop Mode (Full Stop Mode). 1 Oscillator continues to run in Stop Mode (Pseudo Stop Mode), option to run RTI and COP. **Note:** Pseudo Stop Mode 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. **Note:** When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop Mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the startup-time of the external oscillator t_{UPOSC} before entering Pseudo Stop Mode.
4 CSAD	**COP in Stop Mode ACLK Disable** — This bit disables the ACLK for the COP in Stop Mode. Hence the COP is static while in Stop Mode and continues to operate after exit from Stop Mode. Due to clock domain crossing synchronization there is a latency time to enter and exit Stop Mode if COP clock source is ACLK and this clock is stopped in Stop Mode. This maximum latency time is 4 ACLK cycles which must be added to the Stop Mode recovery time tSTP_REC from exit of current Stop Mode to entry of next Stop Mode. This latency time occurs no matter which Stop Mode (Full, Pseudo) is currently exited or entered next. After exit from Stop Mode (Pseudo, Full) for 2 ACLK cycles no Stop Mode request (STOP instruction) should be generated to make sure the COP counter increments at each Stop Mode exit. This bit does not influence the ACLK for the API. 0 COP running in Stop Mode (ACLK for COP enabled in Stop Mode). 1 COP stopped in Stop Mode (ACLK for COP disabled in Stop Mode)
4 COP OSCSEL1	**COP Clock Select 1** — COPOSCSEL0 and COPOSCSEL1 combined determine the clock source to the COP (see also Table 4-6). If COPOSCSEL1 = 1, COPOSCSEL0 has no effect regarding clock select and changing the COPOSCSEL0 bit does not re-start the COP time-out period. COPOSCSEL1 selects the clock source to the COP to be either ACLK (derived from trimmable internal RC-Oscillator) or clock selected via COPOSCSEL0 (IRCCLK or OSCCLK). Changing the COPOSCSEL1 bit re-starts the COP time-out period. COPOSCSEL1 can be set independent from value of UPOSC. UPOSC= 0 does not clear the COPOSCSEL1 bit. 0 COP clock source defined by COPOSCSEL0 1 COP clock source is ACLK derived from a trimmable internal RC-Oscillator
3 PRE	**RTI Enable During Pseudo Stop Bit** — PRE enables the RTI during Pseudo Stop Mode. 0 RTI stops running during Pseudo Stop Mode. 1 RTI continues running during Pseudo Stop Mode if RTIOSCSEL=1. **Note:** If PRE=0 or RTIOSCSEL=0 then the RTI will go static while Stop Mode is active. The RTI counter will not be reset.
2 PCE	**COP Enable During Pseudo Stop Bit** — PCE enables the COP during Pseudo Stop Mode. 0 COP stops running during Pseudo Stop Mode 1 COP continues running during Pseudo Stop Mode if COPOSCSEL=1 **Note:** If PCE=0 or COPOSCSEL=0 then the COP will go static while Stop Mode is active. The COP counter will not be reset.

- 2–58 I bit maskable interrupt vector requests (at addresses vector base + 0x0082–0x00F2).
- I bit maskable interrupts can be nested.
- One X bit maskable interrupt vector request (at address vector base + 0x00F4).
- One non-maskable software interrupt request (SWI) or background debug mode vector request (at address vector base + 0x00F6).
- One non-maskable unimplemented op-code trap (TRAP) vector (at address vector base + 0x00F8).
- Three system reset vectors (at addresses 0xFFFA–0xFFFE).
- Determines the highest priority interrupt vector requests, drives the vector to the bus on CPU request
- Wakes up the system from stop or wait mode when an appropriate interrupt request occurs.

7.1.3 Modes of Operation

- Run mode

  This is the basic mode of operation.
- Wait mode

  In wait mode, the clock to the INT module is disabled. The INT module is however capable of waking-up the CPU from wait mode if an interrupt occurs. Please refer to Section 7.5.3, “Wake Up from Stop or Wait Mode” for details.
- Stop Mode

  In stop mode, the clock to the INT module is disabled. The INT module is however capable of waking-up the CPU from stop mode if an interrupt occurs. Please refer to Section 7.5.3, “Wake Up from Stop or Wait Mode” for details.
- Freeze mode (BDM active)

  In freeze mode (BDM active), the interrupt vector base register is overridden internally. Please refer to Section 7.3.1.1, “Interrupt Vector Base Register (IVBR)” for details.

7.1.4 Block Diagram

Figure 7-1 shows a block diagram of the INT module.

1. The vector base is a 16-bit address which is accumulated from the contents of the interrupt vector base register (IVBR, used as upper byte) and 0x00 (used as lower byte).

The clock source of each PWM channel is determined by PCLKx bits in PWMCLK (see Section 9.3.2.3, “PWM Clock Select Register (PWMCLK)) and PCLKABx bits in PWMCLKAB as shown in Table 9-5 and Table 9-6.

9.3.2.8 PWM Scale A Register (PWMSCLA)

PWMSCLA is the programmable scale value used in scaling clock A to generate clock SA. Clock SA is generated by taking clock A, dividing it by the value in the PWMSCLA register and dividing that by two.

Clock SA = Clock A / (2 * PWMSCLA)

NOTE

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLA).

Module Base + 0x0008

	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 9-10. PWM Scale A Register (PWMSCLA)

Read: Anytime

Write: Anytime (causes the scale counter to load the PWMSCLA value)

9.3.2.9 PWM Scale B Register (PWMSCLB)

PWMSCLB is the programmable scale value used in scaling clock B to generate clock SB. Clock SB is generated by taking clock B, dividing it by the value in the PWMSCLB register and dividing that by two.

Clock SB = Clock B / (2 * PWMSCLB)

NOTE

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLB).

Module Base + 0x0009

	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 9-11. PWM Scale B Register (PWMSCLB)

Read: Anytime

Write: Anytime (causes the scale counter to load the PWMSCLB value).

Clock Source = E, where E = 10 MHz (100 ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 MHz/8 = 1.25 MHz

PWMx Period = 800 ns

PWMx Duty Cycle = 3/4 *100% = 75%

Shown in Figure 9-20 is the output waveform generated.

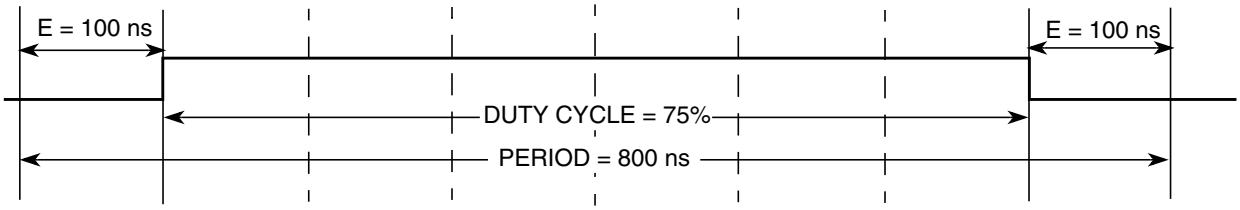

Figure 9-20. PWM Center Aligned Output Example Waveform

9.4.2.7 PWM 16-Bit Functions

The scalable PWM timer also has the option of generating up to 8-channels of 8-bits or 4-channels of 16-bits for greater PWM resolution. This 16-bit channel option is achieved through the concatenation of two 8-bit channels.

The PWMCTL register contains four control bits, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. Channels 6 and 7 are concatenated with the CON67 bit, channels 4 and 5 are concatenated with the CON45 bit, channels 2 and 3 are concatenated with the CON23 bit, and channels 0 and 1 are concatenated with the CON01 bit.

NOTE

Change these bits only when both corresponding channels are disabled.

When channels 6 and 7 are concatenated, channel 6 registers become the high order bytes of the double byte channel, as shown in Figure 9-21. Similarly, when channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

When using the 16-bit concatenated mode, the clock source is determined by the low order 8-bit channel clock select control bits. That is channel 7 when channels 6 and 7 are concatenated, channel 5 when channels 4 and 5 are concatenated, channel 3 when channels 2 and 3 are concatenated, and channel 1 when channels 0 and 1 are concatenated. The resulting PWM is output to the pins of the corresponding low order 8-bit channel as also shown in Figure 9-21. The polarity of the resulting PWM output is controlled by the PPOLx bit of the corresponding low order 8-bit channel as well.

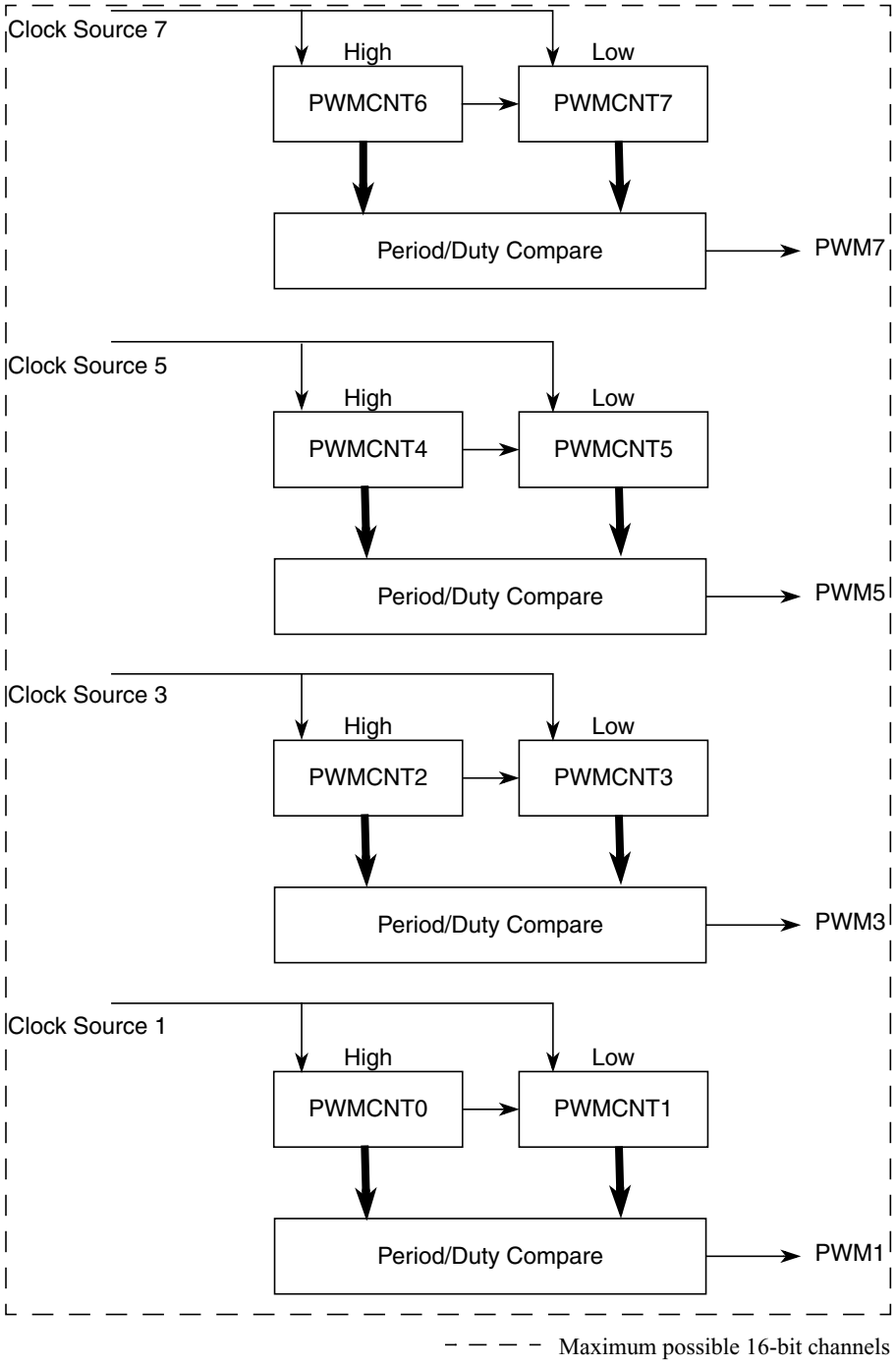

Figure 9-21. PWM 16-Bit Mode

Once concatenated mode is enabled (CONxx bits set in PWMCTL register), enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit. In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output is disabled.

10.3.2.7 SCI Status Register 1 (SCISR1)

The SCISR1 and SCISR2 registers provides inputs to the MCU for generation of SCI interrupts. Also, these registers can be polled by the MCU to check the status of these bits. The flag-clearing procedures require that the status register be read followed by a read or write to the SCI data register.It is permissible to execute other instructions between the two steps as long as it does not compromise the handling of I/O, but the order of operations is important for flag clearing.

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W								
Reset	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-10. SCI Status Register 1 (SCISR1)

Read: Anytime

Write: Has no meaning or effect

Table 10-11. SCISR1 Field Descriptions

Field	Description
7 TDRE	**Transmit Data Register Empty Flag** — TDRE is set when the transmit shift register receives a byte from the SCI data register. When TDRE is 1, the transmit data register (SCIDRH/L) is empty and can receive a new value to transmit.Clear TDRE by reading SCI status register 1 (SCISR1), with TDRE set and then writing to SCI data register low (SCIDRL). 0 No byte transferred to transmit shift register 1 Byte transferred to transmit shift register; transmit data register empty
6 TC	**Transmit Complete Flag** — TC is set low when there is a transmission in progress or when a preamble or break character is loaded. TC is set high when the TDRE flag is set and no data, preamble, or break character is being transmitted.When TC is set, the TXD pin becomes idle (logic 1). Clear TC by reading SCI status register 1 (SCISR1) with TC set and then writing to SCI data register low (SCIDRL). TC is cleared automatically when data, preamble, or break is queued and ready to be sent. TC is cleared in the event of a simultaneous set and clear of the TC flag (transmission not complete). 0 Transmission in progress 1 No transmission in progress
5 RDRF	**Receive Data Register Full Flag** — RDRF is set when the data in the receive shift register transfers to the SCI data register. Clear RDRF by reading SCI status register 1 (SCISR1) with RDRF set and then reading SCI data register low (SCIDRL). 0 Data not available in SCI data register 1 Received data available in SCI data register
4 IDLE	**Idle Line Flag** — IDLE is set when 10 consecutive logic 1s (if M = 0) or 11 consecutive logic 1s (if M =1) appear on the receiver input. Once the IDLE flag is cleared, a valid frame must again set the RDRF flag before an idle condition can set the IDLE flag.Clear IDLE by reading SCI status register 1 (SCISR1) with IDLE set and then reading SCI data register low (SCIDRL). 0 Receiver input is either active now or has never become active since the IDLE flag was last cleared 1 Receiver input has become idle **Note:** When the receiver wakeup bit (RWU) is set, an idle line condition does not set the IDLE flag.

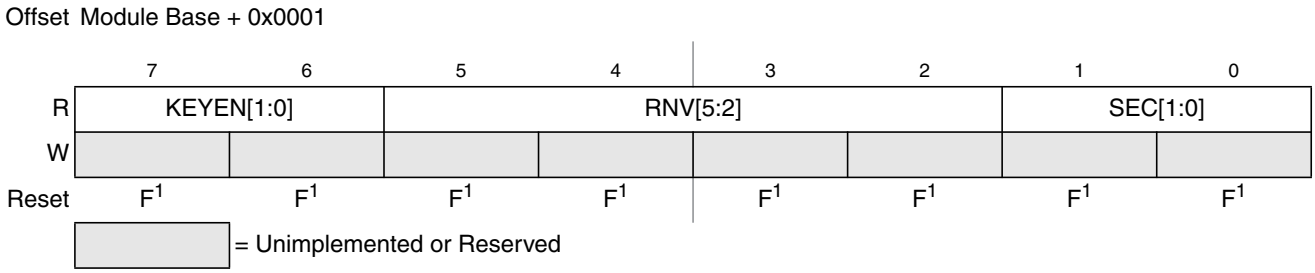

Figure 17-5. Flash Security Register (FSEC)

1 Loaded from IFR Flash configuration field, during reset sequence.

All bits in the FSEC register are readable but not writable.

During the reset sequence, the FSEC register is loaded with the contents of the Flash security byte in the Flash configuration field at global address 0x3_FF0F located in P-Flash memory (see Table 17-3) as indicated by reset condition F in Figure 17-5. If a double bit fault is detected while reading the P-Flash phrase containing the Flash security byte during the reset sequence, all bits in the FSEC register will be set to leave the Flash module in a secured state with backdoor key access disabled.

Table 17-8. FSEC Field Descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 17-9.
5–2 RNV[5:2]	**Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 17-10. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 10.

Table 17-9. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01	DISABLED[1]
10	ENABLED
11	DISABLED

1 Preferred KEYEN state to disable backdoor key access.

Table 17-10. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED[1]
10	UNSECURED
11	SECURED

1 Preferred SEC state to set MCU to secured state.

Table 17-44. Erase Flash Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x09	Global address [17:16] to identify Flash block
001	Global address [15:0] in Flash block to be erased	

Upon clearing CCIF to launch the Erase Flash Block command, the Memory Controller will erase the selected Flash block and verify that it is erased. The CCIF flag will set after the Erase Flash Block operation has completed.

Table 17-45. Erase Flash Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 17-26)
		Set if an invalid global address [17:16] is supplied
		Set if the supplied P-Flash address is not phrase-aligned or if the EEPROM address is not word-aligned
	FPVIOL	Set if an area of the selected Flash block is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

17.4.6.9 Erase P-Flash Sector Command

The Erase P-Flash Sector operation will erase all addresses in a P-Flash sector.

Table 17-46. Erase P-Flash Sector Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0A	Global address [17:16] to identify P-Flash block to be erased
001	Global address [15:0] anywhere within the sector to be erased. Refer to Section 17.1.2.1 for the P-Flash sector size.	

Upon clearing CCIF to launch the Erase P-Flash Sector command, the Memory Controller will erase the selected Flash sector and then verify that it is erased. The CCIF flag will be set after the Erase P-Flash Sector operation has completed.

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller will set the user margin level for the targeted block and then set the CCIF flag.

NOTE

When the EEPROM block is targeted, the EEPROM user margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash user margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply user margin levels to the P-Flash block only.

Valid margin level settings for the Set User Margin Level command are defined in Table 17-53.

Table 17-53. Valid Set User Margin Level Settings

CCOB (CCOBIX=001)	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level[1]
0x0002	User Margin-0 Level[2]

[1] Read margin to the erased state

[2] Read margin to the programmed state

Table 17-54. Set User Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 17-26)
		Set if an invalid global address [17:16] is supplied see)
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

NOTE

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

Table 17-57. Set Field Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 17-26)
		Set if an invalid global address [17:16] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

NOTE

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

17.4.6.14 Erase Verify EEPROM Section Command

The Erase Verify EEPROM Section command will verify that a section of code in the EEPROM is erased. The Erase Verify EEPROM Section command defines the starting point of the data to be verified and the number of words.

Table 17-58. Erase Verify EEPROM Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x10	Global address [17:16] to identify the EEPROM block
001	Global address [15:0] of the first word to be verified	
010	Number of words to be verified	

Upon clearing CCIF to launch the Erase Verify EEPROM Section command, the Memory Controller will verify the selected section of EEPROM memory is erased. The CCIF flag will set after the Erase Verify EEPROM Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

17.4.7 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed or when a Flash command operation has detected an ECC fault.

Table 17-64. Flash Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Command Complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit
ECC Double Bit Fault on Flash Read	DFDIF (FERSTAT register)	DFDIE (FERCNFG register)	I Bit
ECC Single Bit Fault on Flash Read	SFDIF (FERSTAT register)	SFDIE (FERCNFG register)	I Bit

NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

17.4.7.1 Description of Flash Interrupt Operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the DFDIF and SFDIF flags in combination with the DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Section 17.3.2.5, “Flash Configuration Register (FCNFG)”, Section 17.3.2.6, “Flash Error Configuration Register (FERCNFG)”, Section 17.3.2.7, “Flash Status Register (FSTAT)”, and Section 17.3.2.8, “Flash Error Status Register (FERSTAT)”.

The logic used for generating the Flash module interrupts is shown in Figure 17-26.

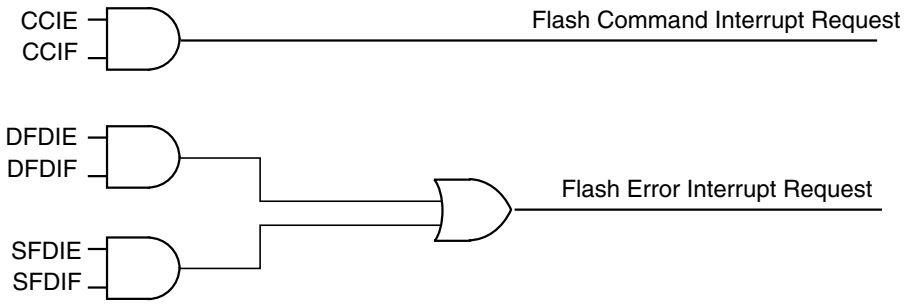

Figure 17-26. Flash Module Interrupts Implementation

17.4.8 Wait Mode

The Flash module is not affected if the MCU enters wait mode. The Flash module can recover the MCU from wait via the CCIF interrupt (see Section 17.4.7, “Interrupts”).

Appendix E
PLL Electrical Specifications

E.1 Reset, Oscillator and PLL

E.1.1 Phase Locked Loop

E.1.1.1 Jitter Information

With each transition of the feedback clock, the deviation from the reference clock is measured and the input voltage to the VCO is adjusted accordingly.The adjustment is done continuously with no abrupt changes in the VCOCLK 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 E-1.**.

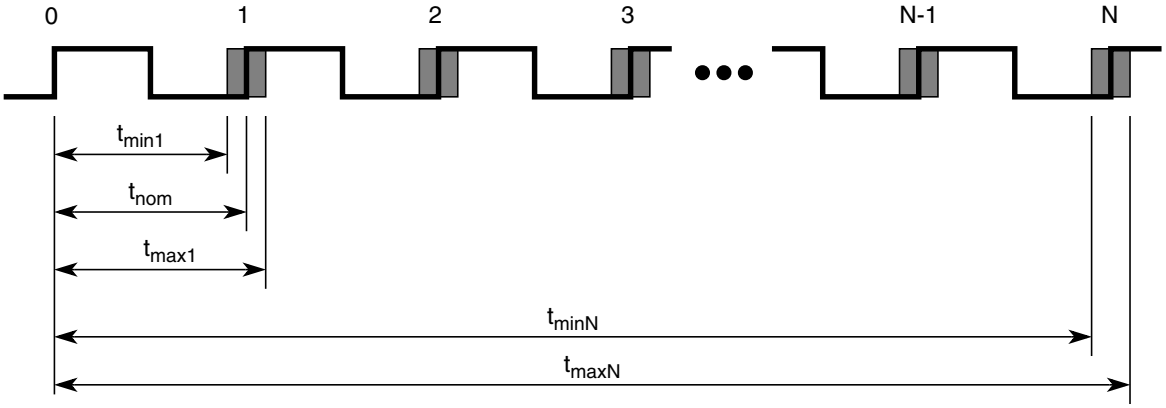

Figure E-1. Jitter Definitions

The relative deviation of t_{nom} is at its maximum for one clock period, and decreases towards zero for larger number of clock periods (N).

Defining the jitter as:

$$J(N) = \max\left(\left|1 - \frac{t_{max}(N)}{N \cdot t_{nom}}\right|, \left|1 - \frac{t_{min}(N)}{N \cdot t_{nom}}\right|\right)$$

The following equation is a good fit for the maximum jitter:

$$J(N) = \frac{j_1}{\sqrt{N}}$$

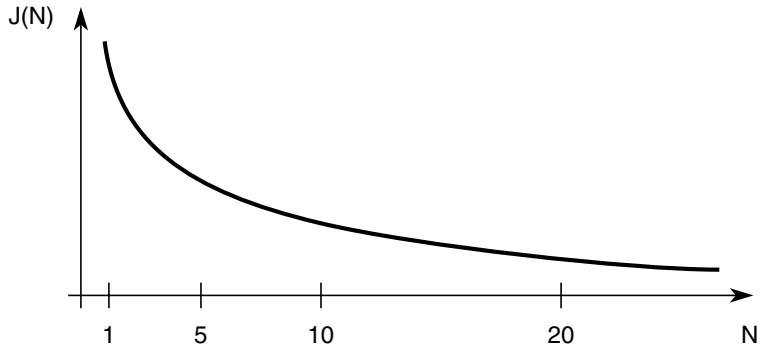

Figure E-2. Maximum Bus Clock Jitter Approximation

NOTE

On timers and serial modules a prescaler will eliminate the effect of the jitter to a large extent.

Table E-1. ipll_1vdd_ll18 Characteristics

Conditions are shown in Figure A-5 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	D	VCO frequency during system reset	f_{VCORST}	8		32	MHz
2	C	VCO locking range	f_{VCO}	32		50	MHz
3	C	Reference Clock	f_{REF}	1			MHz
4	D	Lock Detection	$\|\Delta_{Lock}\|$	0		1.5	%[1]
5	D	Un-Lock Detection	$\|\Delta_{unl}\|$	0.5		2.5	%[1]
6	C	Time to lock	t_{lock}			$150 + 256/f_{REF}$	µs
7	C	Jitter fit parameter 1[2] IRC as reference clock source	j_1			1.4	%
8	C	Jitter fit parameter 1[3] XOSCLCP as reference clock source	j_1			1.0	%

1 Minimum times are based on maximum f_{NVMOP} and maximum f_{NVMBUS}
2 Typical times are based on typical f_{NVMOP} and typical f_{NVMBUS}
3 Maximum times are based on typical f_{NVMOP} and typical f_{NVMBUS} plus aging
4 Worst times are based on minimum f_{NVMOP} and minimum f_{NVMBUS} plus aging
5 Affected by Pflash size
6 Affected by EEPROM size