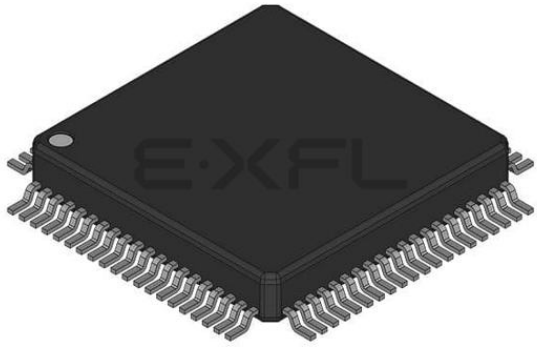

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, I²C, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	60
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K 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 ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12e32cfue

0x0200 – 0x023F PMF (Pulse width Modulator with Fault protection) (Sheet 4 of 4)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0239	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x023A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x023B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x023C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x023D	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x023E	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x023F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0240 – 0x027F PIM (Port Interface Module) (Sheet 1 of 4)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0240	PTT	R	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0
		W								
0x0241	PTIT	R	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
		W								
0x0242	DDRT	R	DDRT7	DDRT7	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
		W								
0x0243	RDRT	R	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0
		W								
0x0244	PERT	R	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0
		W								
0x0245	PPST	R	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
		W								
0x0246	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0247	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0248	PTS	R	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
		W								
0x0249	PTIS	R	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
		W								

impedance input pins. Consult the Port Integration Module (PIM) PIM_9E128 block description chapter, TIM_16B4C block description chapter, and the PWM_8B6C block description chapter for information about pin configurations.

1.4.43 VDDX,VSSX — Power & Ground Pins for I/O Drivers

External power and ground for I/O drivers. Bypass requirements depend on how heavily the MCU pins are loaded.

1.4.44 VDDR, VSSR — Power Supply Pins for I/O Drivers & for Internal Voltage Regulator

External power and ground for I/O drivers and input to the internal voltage regulator. Bypass requirements depend on how heavily the MCU pins are loaded.

1.4.45 VDD1, VDD2, VSS1, VSS2 — Power Supply Pins for Internal Logic

Power is supplied to the MCU through VDD and VSS. This 2.5V supply is derived from the internal voltage regulator. There is no static load on those pins allowed. The internal voltage regulator is turned off, if VDDR is tied to ground.

1.4.46 VDDA, VSSA — Power Supply Pins for ATD and VREG

VDDA, VSSA are the power supply and ground input pins for the voltage regulator and the analog to digital converter.

1.4.47 VRH, VRL — ATD Reference Voltage Input Pins

VRH and VRL are the reference voltage input pins for the analog to digital converter.

1.4.48 VDDPLL, VSSPLL — Power Supply Pins for PLL

Provides operating voltage and ground for the Oscillator and the Phased-Locked Loop. This allows the supply voltage to the Oscillator and PLL to be bypassed independently. This 2.5V voltage is generated by the internal voltage regulator.

3.3.2.7 Port M Wired-OR Mode Register (WOMM)

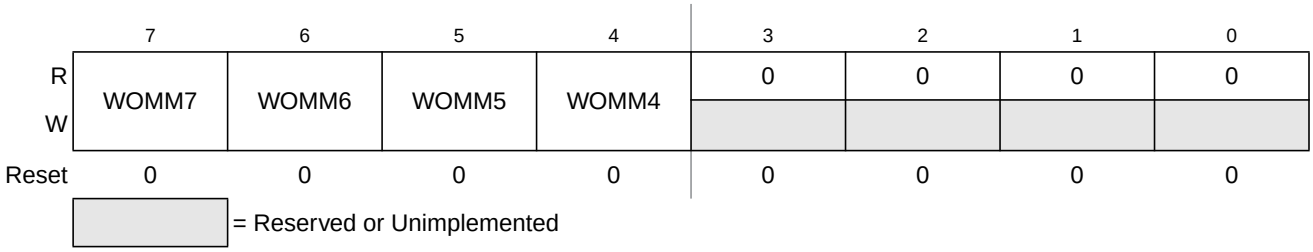

Figure 3-16. Port M Wired-OR Mode Register (WOMM)

Read: Anytime. Write: Anytime.

This register selects whether a port M output is configured as push-pull or wired-or. When a Wired-OR Mode Register bit is set to 1, the corresponding output pin is driven active low only (open drain) and a high level is not driven. A Wired-OR Mode Register bit has no effect if the corresponding pin is configured as an input.

These bits apply also to the SCI2 outputs and allow a multipoint connection of several serial modules.

If the IIC is enabled, the associated pins are always set to wired-or mode, and the state of the WOMM[7:6] bits have no effect. The WOMM[7:6] bits will not change to reflect their wired-or mode configuration when the IIC is enabled.

Table 3-13. WOMM Field Descriptions

Field	Description
7:4 WOMM[7:4]	**Wired-OR Mode Port M** 0 Output buffers operate as push-pull outputs. 1 Output buffers operate as open-drain outputs.

4.3.2.2 CRG Reference Divider Register (REFDV)

The REFDV register provides a finer granularity for the PLL multiplier steps. The count in the reference divider divides OSCCLK frequency by REFDV + 1.

	7	6	5	4	3	2	1	0
R	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 4-5. CRG Reference Divider Register (REFDV)

Read: anytime

Write: anytime except when PLLSEL = 1

NOTE

Write to this register initializes the lock detector bit and the track detector bit.

4.3.2.3 Reserved Register (CTFLG)

This register is reserved for factory testing of the CRGV4 module and is not available in normal modes.

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

= Unimplemented or Reserved

Figure 4-6. CRG Reserved Register (CTFLG)

Read: always reads 0x0000 in normal modes

Write: unimplemented in normal modes

NOTE

Writing to this register when in special mode can alter the CRGV4 functionality.

6.3.2.14 Port Data Register 0 (PORTAD0)

The data port associated with the ATD is input-only. The port pins are shared with the analog A/D inputs AN[15:8].

	7	6	5	4	3	2	1	0
R	PTAD15	PTAD14	PTAD13	PTAD12	PTAD11	PTAD10	PTAD9	PTAD8
W								
Reset	1	1	1	1	1	1	1	1
Pin Function	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8

= Unimplemented or Reserved

Figure 6-16. Port Data Register 0 (PORTAD0)

Read: Anytime

Write: Anytime, no effect

The A/D input channels may be used for general-purpose digital input.

Table 6-21. PORTAD0 Field Descriptions

Field	Description
7:0 PTAD[15:8]	**A/D Channel x (ANx) Digital Input Bits**— If the digital input buffer on the ANx pin is enabled (IENx = 1) or channel x is enabled as external trigger (ETRIGE = 1, ETRIGCH[3-0] = x, ETRIGSEL = 0) read returns the logic level on ANx pin (signal potentials not meeting V_{IL} or V_{IH} specifications will have an indeterminate value)). If the digital input buffers are disabled (IENx = 0) and channel x is not enabled as external trigger, read returns a “1”. Reset sets all PORTAD0 bits to “1”.

7.3.2 Register Descriptions

This section consists of register descriptions arranged in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in descending bit order.

7.3.2.1 DAC Control Register 0 (DACC0)

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	DACE	DACTE	0	0	DJM	DSGN	DACWAI	DACOE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-3. DAC Control Register 0 (DACC0)

Read: anytime (reserved locations read zero)

Write: anytime except DACTE is available only in special modes

Table 7-2. DACC0 Field Descriptions

Field	Description
7 DACE	**DAC Enable** — This bit enables digital-to-analog converter functionality. When enabled, an analog voltage based on the digital value in the DAC data register will be output. When disabled, DAO pin is high-impedance. 0 DAC is disabled and powered down 1 DAC is enabled for conversion
6 DACTE	**DAC Test Enable** — This reserved bit is designed for factory test purposes only and is not intended for general user access. Writing to this bit when in special test modes can alter DAC functionality.
3 DJM	**Data Register Data Justification** — This bit controls the justification of the data in the DAC data register (DACD). If DJM is clear (left-justified), the data to be converted must be written to left justified DACD and the right justified DACD register will read zeroes. If DJM is set (right-justified), the data to be converted is written to right justified DACD register and left justified DACD register reads zeroes. Data is preserved if DJM bit is changed after data is written. 0 Left justified data in DAC data register 1 Right justified data in DAC data register
2 DSGN	**Data Register Signed** — This bit selects between signed and unsigned conversion data representation in the DAC data register. Signed data is represented as 2's complement. 0 Unsigned data representation in DAC data register 1 Signed data representation in DAC data register
1 DACWAI	**DAC Stop in WAIT Mode** — DACWAI disables the DAC8B1C module (no new conversion is done) during wait mode. 0 DAC is enabled during wait mode 1 DAC is disabled and powered down during wait mode
0 DACOE	**DAC Output Enable** — This bit enables the output on the DAO pin. To output the DAC voltage, the DACOE bit and the DACE bit must be set. When disabled, DAO pin is high-impedance. 0 Output is not available for external use 1 Output on DAO pin enabled.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 8-15 summarizes the results of the data bit samples.

Table 8-15. Data Bit Recovery

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

NOTE

The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are logic 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit (logic 0).

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 8-16 summarizes the results of the stop bit samples.

Table 8-16. Stop Bit Recovery

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

9.3.2.3 SPI Baud Rate Register (SPIBR)

	7	6	5	4	3	2	1	0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-5. SPI Baud Rate Register (SPIBR)

Read: anytime

Write: anytime; writes to the reserved bits have no effect

Table 9-6. SPIBR Field Descriptions

Field	Description
6:4 SPPR[2:0]	**SPI Baud Rate Preselection Bits** — These bits specify the SPI baud rates as shown in Table 9-7. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.
2:0 SPR[2:0}	**SPI Baud Rate Selection Bits** — These bits specify the SPI baud rates as shown in Table 9-7. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.

The baud rate divisor equation is as follows:

$$BaudRateDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}$$

The baud rate can be calculated with the following equation:

$$Baud\ Rate = BusClock / BaudRateDivisor$$

10.4 Functional Description

This section provides a complete functional description of the IICV2.

10.4.1 I-Bus Protocol

The IIC bus system uses a serial data line (SDA) and a serial clock line (SCL) for data transfer. All devices connected to it must have open drain or open collector outputs. Logic AND function is exercised on both lines with external pull-up resistors. The value of these resistors is system dependent.

Normally, a standard communication is composed of four parts: START signal, slave address transmission, data transfer and STOP signal. They are described briefly in the following sections and illustrated in Figure 10-8.

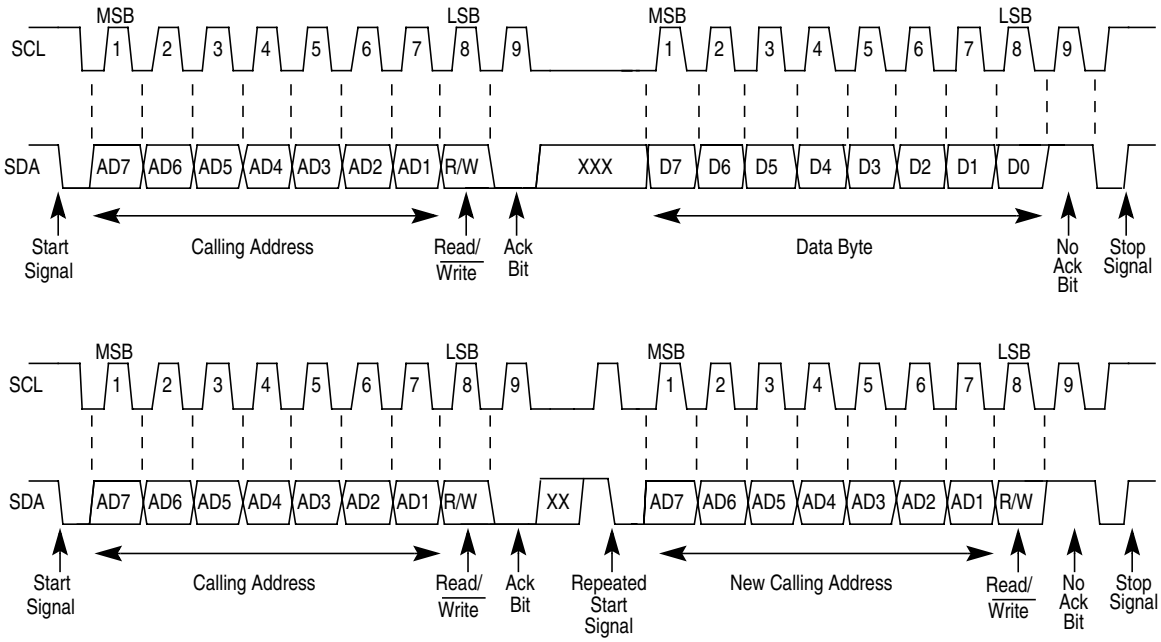

Figure 10-8. IIC-Bus Transmission Signals

10.4.1.1 START Signal

When the bus is free, i.e. no master device is engaging the bus (both SCL and SDA lines are at logical high), a master may initiate communication by sending a START signal.As shown in Figure 10-8, a START signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and brings all slaves out of their idle states.

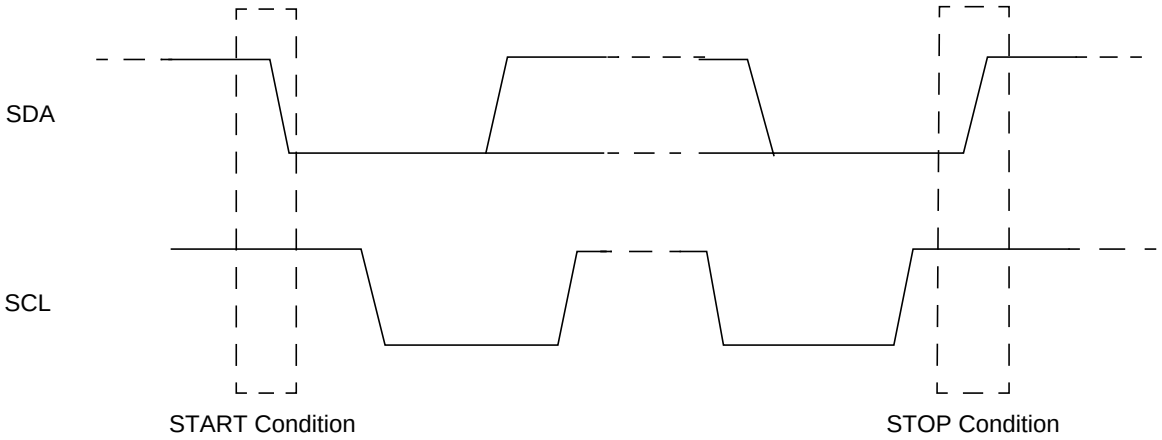

Figure 10-9. Start and Stop Conditions

10.4.1.2 Slave Address Transmission

The first byte of data transfer immediately after the START signal is the slave address transmitted by the master. This is a seven-bit calling address followed by a R/W bit. The R/W bit tells the slave the desired direction of data transfer.

1 = Read transfer, the slave transmits data to the master.
0 = Write transfer, the master transmits data to the slave.

Only the slave with a calling address that matches the one transmitted by the master will respond by sending back an acknowledge bit. This is done by pulling the SDA low at the 9th clock (see Figure 10-8).

No two slaves in the system may have the same address. If the IIC bus is master, it must not transmit an address that is equal to its own slave address. The IIC bus cannot be master and slave at the same time.However, if arbitration is lost during an address cycle the IIC bus will revert to slave mode and operate correctly even if it is being addressed by another master.

10.4.1.3 Data Transfer

As soon as successful slave addressing is achieved, the data transfer can proceed byte-by-byte in a direction specified by the R/W bit sent by the calling master

All transfers that come after an address cycle are referred to as data transfers, even if they carry sub-address information for the slave device.

Each data byte is 8 bits long. Data may be changed only while SCL is low and must be held stable while SCL is high as shown in Figure 10-8. There is one clock pulse on SCL for each data bit, the MSB being transferred first. Each data byte has to be followed by an acknowledge bit, which is signalled from the receiving device by pulling the SDA low at the ninth clock. So one complete data byte transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master, the SDA line must be left high by the slave. The master can then generate a stop signal to abort the data transfer or a start signal (repeated start) to commence a new calling.

Chapter 11
Pulse Width Modulator with Fault Protection (PMF15B6CV2)

11.1 Introduction

The Pulse width Modulator with Fault protection (PMF) module can be configured for one, two, or three complementary pairs. For example:

- One complementary pair and four independent PWM outputs
- Two complementary pair and two independent PWM outputs
- Three complementary pair and zero independent PWM outputs
- Zero complementary pair and six independent PWM outputs

All PWM outputs can be generated from the same counter, or each pair can have its own counter for three independent PWM frequencies. Complementary operation permits programmable dead-time insertion, distortion correction through current sensing by software, and separate top and bottom output polarity control. Each counter value is programmable to support a continuously variable PWM frequency. Both edge- and center-aligned synchronous pulse width-control and full range modulation from 0 percent to 100 percent, are supported. The PMF is capable of controlling most motor types: AC induction motors (ACIM), both brushless (BLDC) and brush DC motors (BDC), switched (SRM), and variable reluctance motors (VRM), and stepper motors.

11.1.1 Features

- Three complementary PWM signal pairs, or six independent PWM signals
- Three 15-bit counters
- Features of complementary channel operation
  - Deadtime insertion
  - Separate top and bottom pulse width correction via current status inputs or software
  - Separate top and bottom polarity control
- Edge-aligned or center-aligned PWM signals
- Half-cycle reload capability
- Integral reload rates from 1 to 16
- Individual software-controlled PWM output
- Programmable fault protection
- Polarity control

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0028	PMFENCB	R	PWMENB	0	0	0	0	0	LDOKB	PWMRIEB
		W								
0x0029	PMFFQCB	R	LDFQB				HALFB	PRSCB		PWMRFB
		W								
0x002A	PMFCNTB	R	0	PMFCNTB						
		W								
0x002B	PMFCNTB	R	PMFCNTB							
		W								
0x002C	PMFMODB	R	0	PMFMODB						
		W								
0x002D	PMFMODB	R	PMFMODB							
		W								
0x002E	PMFDTMB	R	0	0	0	0	PMFDTMB			
		W								
0x002F	PMFDTMB	R	PMFDTMB							
		W								
0x0030	PMFENCC	R	PWMENC	0	0	0	0	0	LDOKC	PWMRIEC
		W								
0x0031	PMFFQCC	R	LDFQC				HALFC	PRSCC		PWMRFC
		W								
0x0032	PMFCNTC	R	0	PMFCNTC						
		W								
0x0033	PMFCNTC	R	PMFCNTC							
		W								
0x0034	PMFMODC	R	0	PMFMODC						
		W								
0x0035	PMFMODC	R	PMFMODC							
		W								
0x0036	PMFDTMC	R	0	0	0	0	PMFDTMC			
		W								
0x0037	PMFDTMC	R	PMFDTMC							
		W								
0x0038 ↓ 0X003F	Reserved	R								
		W								

= Unimplemented or Reserved

Figure 11-3. PMF15B6C Register Summary (Sheet 3 of 3)

11.3.2.24 PMF Deadtime A Register (PMFDTMA)

Module Base + 0x0026

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

= Unimplemented or Reserved

Figure 11-30. PMF Deadtime A Register (PMFDTMA)

Read anytime. This register cannot be modified after the WP bit is set.

Table 11-31. PMFDTMA Field Descriptions

Field	Description
11–0 PMFDTMA	**PMF Deadtime A Bits** — The 12-bit value written to this register is the number of PWM clock cycles in complementary channel operation. A reset sets the PWM deadtime register to a default value of 0x0FFF, selecting a deadtime of 256-PWM clock cycles minus one bus clock cycle. **Note:** Deadtime is affected by changes to the prescaler value. The deadtime duration is determined as follows: DT = P × PMFDTMA – 1, where DT is deadtime, P is the prescaler value, PMFDTMA is the programmed value of dead time. For example: if the prescaler is programmed for a divide-by-two and the PMFDTMA is set to five, then P = 2 and the deadtime value is equal to DT = 2 × 5 – 1 = 9 IPbus clock cycles. A special case exists when the P = 1, then DT = PMFDTMA.

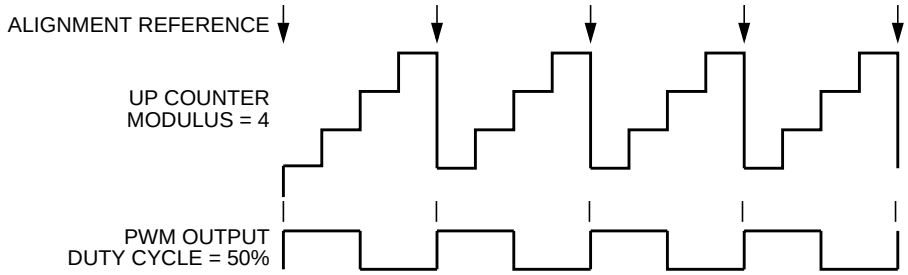

Figure 11-42. Edge-Aligned PWM Output

NOTE

Because of the equals-comparator architecture of this PMF, the modulus equals zero case is considered illegal. Therefore, the modulus register does not return to zero, and a modulus value of zero will result in waveforms inconsistent with the other modulus waveforms. If a modulus of zero is loaded, the counter will continually count down from $7FFF. This operation will not be tested or guaranteed. Consider it illegal. However, the dead-time constraints and fault conditions will still be guaranteed.

11.4.3.2 Period

A PWM period is determined by the value written to the PWM counter modulo register.

The PWM counter is an up/down counter in a center-aligned operation. In this mode the PWM highest output resolution is two bus clock cycles.

$$\text{PWM period} = (\text{PWM modulus}) \times (\text{PWM clock period}) \times 2$$

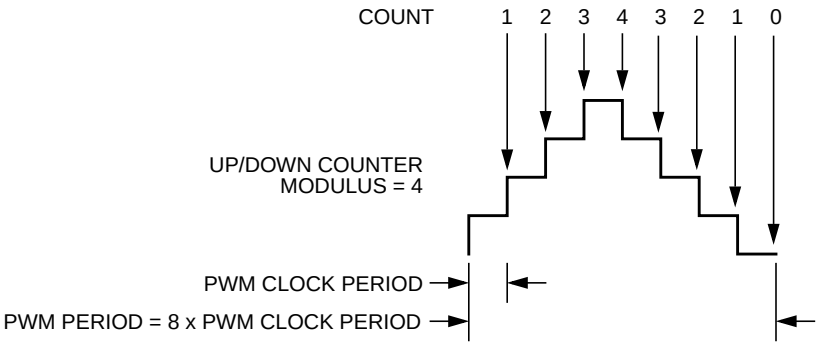

Figure 11-43. Center-Aligned PWM Period

Setting the OUTCTLx bits do not disable the PWM generators and current status sensing circuitry. They continue to run, but no longer control the output pins. When the OUTCTLx bits are cleared, the outputs of the PWM generator become the inputs to the deadtime generators at the beginning of the next PWM cycle. Software can drive the PWM outputs even when PWM enable bit (PWMEN) is set to zero.

NOTE

Avoid an unexpected deadtime insertion by clearing the OUTx bits before setting and after clearing the OUTCTLx bits.

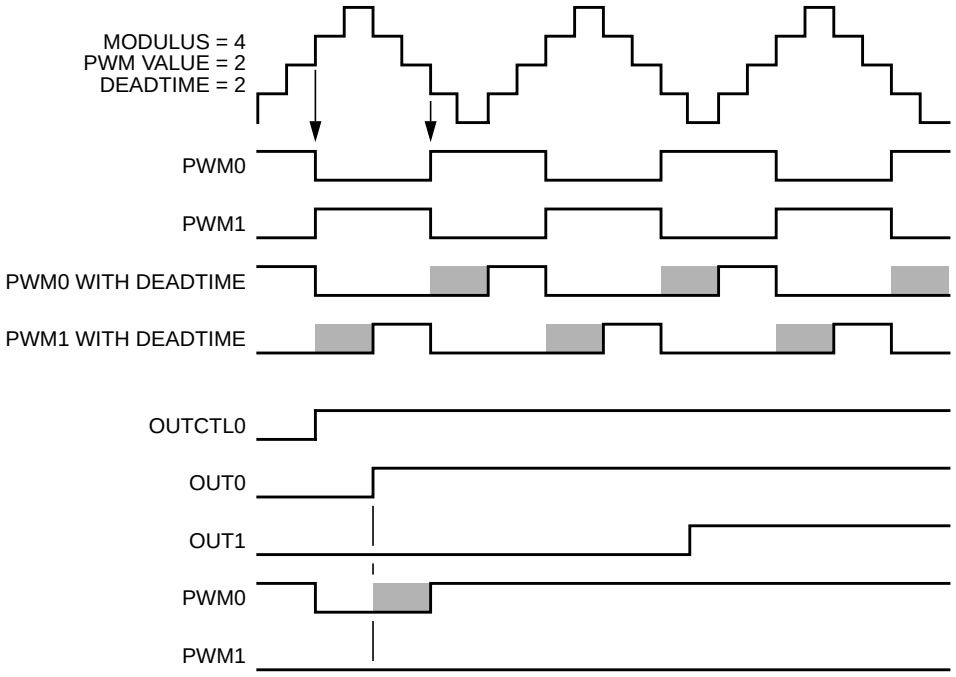

Figure 11-62. Setting OUT0 with OUTCTL Set in Complementary Mode

Table 12-2. PWME Field Descriptions (continued)

Field	Description
1 PWME1	**Pulse Width Channel 1 Enable** 0 Pulse width channel 1 is disabled. 1 Pulse width channel 1 is enabled. The pulse modulated signal becomes available at PWM, output bit 1 when its clock source begins its next cycle.
0 PWME0	**Pulse Width Channel 0 Enable** 0 Pulse width channel 0 is disabled. 1 Pulse width channel 0 is enabled. The pulse modulated signal becomes available at PWM, output bit 0 when its clock source begins its next cycle. If CON01 = 1, then bit has no effect and PWM output line 0 is disabled.

12.3.2.2 PWM Polarity Register (PWMPOL)

The starting polarity of each PWM channel waveform is determined by the associated PPOLx bit in the PWMPOL register. If the polarity bit is 1, the PWM channel output is high at the beginning of the cycle and then goes low when the duty count is reached. Conversely, if the polarity bit is 0 the output starts low and then goes high when the duty count is reached.

	7	6	5	4	3	2	1	0
R	0	0	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 12-4. PWM Polarity Register (PWMPOL)

Read: anytime

Write: anytime

NOTE

PPOLx register bits can be written anytime. If the polarity is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition

Table 12-3. PWMPOL Field Descriptions

Field	Description
5 PPOL5	**Pulse Width Channel 5 Polarity** 0 PWM channel 5 output is low at the beginning of the period, then goes high when the duty count is reached. 1 PWM channel 5 output is high at the beginning of the period, then goes low when the duty count is reached.
4 PPOL4	**Pulse Width Channel 4 Polarity** 0 PWM channel 4 output is low at the beginning of the period, then goes high when the duty count is reached. 1 PWM channel 4 output is high at the beginning of the period, then goes low when the duty count is reached.
3 PPOL3	**Pulse Width Channel 3 Polarity** 0 PWM channel 3 output is low at the beginning of the period, then goes high when the duty count is reached. 1 PWM channel 3 output is high at the beginning of the period, then goes low when the duty count is reached.

13.3.2.1 Timer Input Capture/Output Compare Select (TIOS)

	7	6	5	4	3	2	1	0
R	IOS7	IOS6	IOS5	IOS4	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Figure 13-6. Timer Input Capture/Output Compare Select (TIOS)

Read: Anytime

Write: Anytime

Table 13-2. TIOS Field Descriptions

Field	Description
7:4 IOS[7:4]	**Input Capture or Output Compare Channel Configuration** 0 The corresponding channel acts as an input capture. 1 The corresponding channel acts as an output compare.

13.3.2.2 Timer Compare Force Register (CFORC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	FOC7	FOC6	FOC5	FOC4				
Reset	0	0	0	0	0	0	0	0

Figure 13-7. Timer Compare Force Register (CFORC)

Read: Anytime but will always return 0x0000 (1 state is transient)

Write: Anytime

Table 13-3. CFORC Field Descriptions

Field	Description
7:4 FOC[7:4]	**Force Output Compare Action for Channel 7:4** — A write to this register with the corresponding data bit(s) set causes the action which is programmed for output compare “x” to occur immediately. The action taken is the same as if a successful comparison had just taken place with the TCx register except the interrupt flag does not get set. **Note:** A successful channel 7 output compare overrides any channel 6:4 compares. If forced output compare on any channel occurs at the same time as the successful output compare then forced output compare action will take precedence and interrupt flag won’t get set.

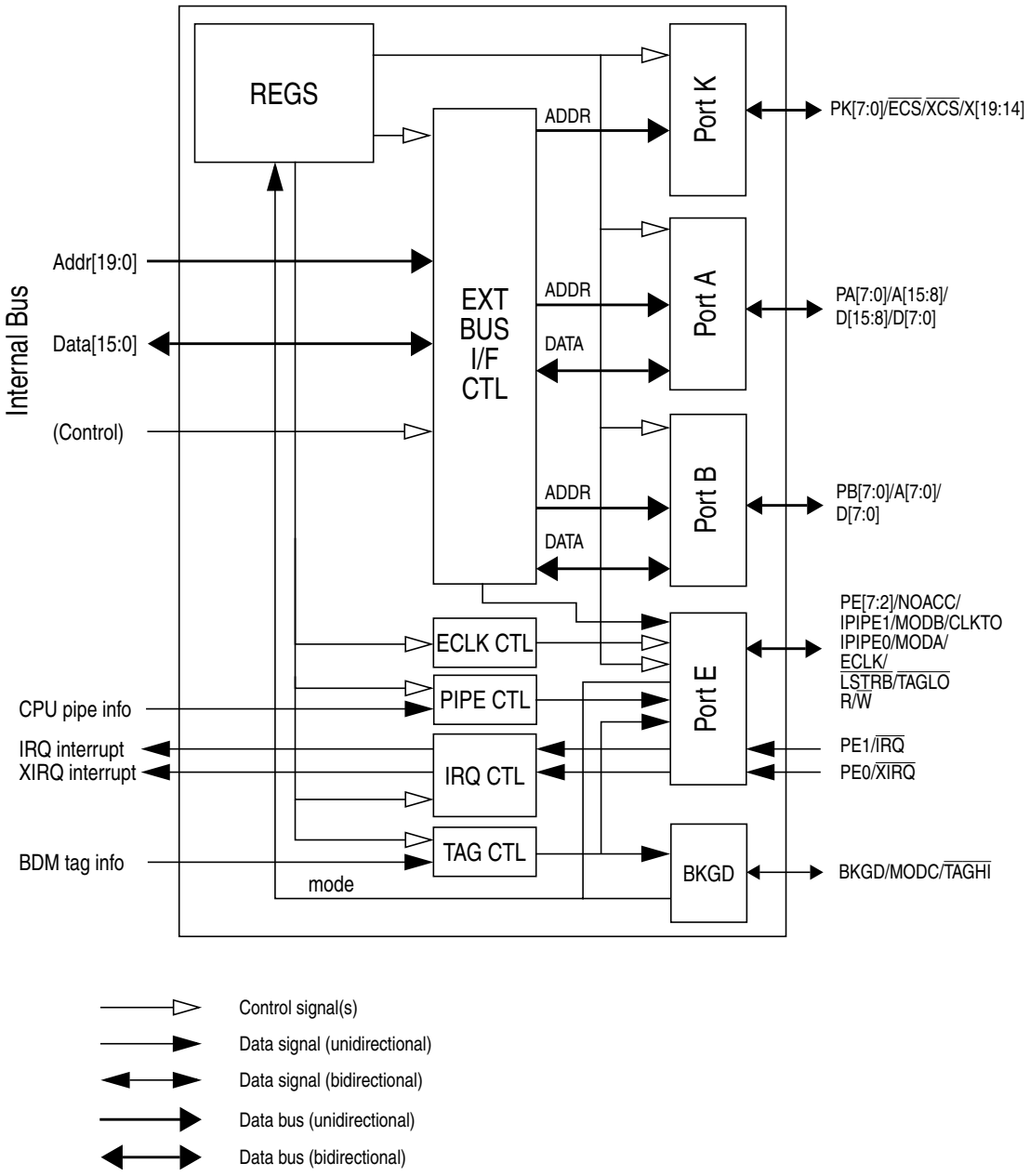

Figure 18-1. MEBI Block Diagram

18.3.2.9 Mode Register (MODE)

	7	6	5	4	3	2	1	0
R	MODC	MODB	MODA	0	IVIS	0	EMK	EME
W								
Reset								
Special Single Chip	0	0	0	0	0	0	0	0
Emulation Expanded Narrow	0	0	1	0	1	0	1	1
Special Test	0	1	0	0	1	0	0	0
Emulation Expanded Wide	0	1	1	0	1	0	1	1
Normal Single Chip	1	0	0	0	0	0	0	0
Normal Expanded Narrow	1	0	1	0	0	0	0	0
Peripheral	1	1	0	0	0	0	0	0
Normal Expanded Wide	1	1	1	0	0	0	0	0

= Unimplemented or Reserved

Figure 18-13. Mode Register (MODE)

Read: Anytime (provided this register is in the map).

Write: Each bit has specific write conditions. Please refer to the descriptions of each bit on the following pages.

The MODE register is used to establish the operating mode and other miscellaneous functions (i.e., internal visibility and emulation of port E and K).

In special peripheral mode, this register is not accessible but it is reset as shown to system configuration features. Changes to bits in the MODE register are delayed one cycle after the write.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

18.4.3.1.5 Emulation Expanded Narrow Mode

Expanded narrow modes are intended to allow connection of single 8-bit external memory devices for lower cost systems that do not need the performance of a full 16-bit external data bus. Accesses to internal resources that have been mapped external (i.e. PORTA, PORTB, DDRA, DDRB, PORTE, DDRE, PEAR, PUCR, RDRIV) will be accessed with a 16-bit data bus on Ports A and B. Accesses of 16-bit external words to addresses which are normally mapped external will be broken into two separate 8-bit accesses using Port A as an 8-bit data bus. Internal operations continue to use full 16-bit data paths. They are only visible externally as 16-bit information if IVIS=1.

Ports A and B are configured as multiplexed address and data output ports. During external accesses, address A15, data D15 and D7 are associated with PA7, address A0 is associated with PB0 and data D8 and D0 are associated with PA0. During internal visible accesses and accesses to internal resources that have been mapped external, address A15 and data D15 is associated with PA7 and address A0 and data D0 is associated with PB0.

The bus control related pins in Port E (PE7/NOACC, PE6/MODB/IPIPE1, PE5/MODA/IPIPE0, PE4/ECLK, PE3/$\overline{\text{LSTRB}}$/$\overline{\text{TAGLO}}$, and PE2/R/$\overline{\text{W}}$) are all configured to serve their bus control output functions rather than general purpose I/O. Notice that writes to the bus control enable bits in the PEAR register in emulation mode are restricted.

The main difference between special modes and normal modes is that some of the bus control and system control signals cannot be written in emulation modes.

18.4.3.2 Special Operating Modes

There are two special operating modes that correspond to normal operating modes. These operating modes are commonly used in factory testing and system development.

18.4.3.2.1 Special Single-Chip Mode

When the MCU is reset in this mode, the background debug mode is enabled and active. The MCU does not fetch the reset vector and execute application code as it would in other modes. Instead the active background mode is in control of CPU execution and BDM firmware is waiting for additional serial commands through the BKGD pin. When a serial command instructs the MCU to return to normal execution, the system will be configured as described below unless the reset states of internal control registers have been changed through background commands after the MCU was reset.

There is no external expansion bus after reset in this mode. Ports A and B are initially simple bidirectional I/O pins that are configured as high-impedance inputs with internal pull resistors disabled; however, writing to the mode select bits in the MODE register (which is allowed in special modes) can change this after reset. All of the Port E pins (except PE4/ECLK) are initially configured as general purpose high-impedance inputs with internal pull resistors enabled. PE4/ECLK is configured as the E clock output in this mode.

The pins associated with Port E bits 6, 5, 3, and 2 cannot be configured for their alternate functions IPIPE1, IPIPE0, $\overline{\text{LSTRB}}$, and R/$\overline{\text{W}}$ while the MCU is in single chip modes. In single chip modes, the associated