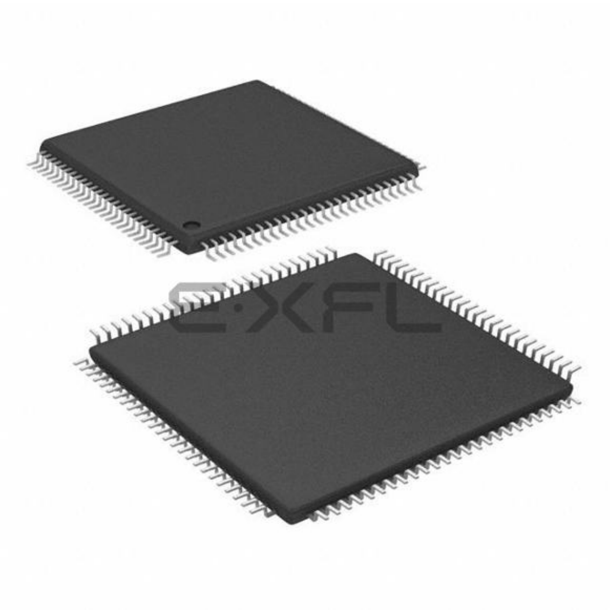


Welcome to E-XFL.COM

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	dsPIC
Core Size	16-Bit
Speed	50 MIPs
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, QEI, POR, PWM, WDT
Number of I/O	85
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 24x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj32gs610t-50i-pf

3.5 Arithmetic Logic Unit (ALU)

The dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610 ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are 2's complement in nature. Depending on the operation, the ALU can affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

Refer to the *"16-bit MCU and DSC Programmer's Reference Manual*" (DS70157) for information on the SR bits affected by each instruction.

The dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610 CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit divisor division.

3.5.1 MULTIPLIER

Using the high-speed, 17-bit x 17-bit multiplier of the DSP engine, the ALU supports unsigned, signed or mixed sign operation in several MCU multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

3.5.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

- 32-bit signed/16-bit signed divide
- 32-bit unsigned/16-bit unsigned divide
- 16-bit signed/16-bit signed divide
- 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. 16-bit signed and unsigned `DIV` instructions can specify any W register for both the 16-bit divisor (Wn) and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

3.6 DSP Engine

The DSP engine consists of a high-speed, 17-bit x 17-bit multiplier, a barrel shifter and a 40-bit adder/subtracter (with two target accumulators, round and saturation logic).

The dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610 is a single-cycle instruction flow architecture; therefore, concurrent operation of the DSP engine with MCU instruction flow is not possible. However, some MCU ALU and DSP engine resources can be used concurrently by the same instruction (for example, ED, EDAC).

The DSP engine can also perform inherent accumulator-to-accumulator operations that require no additional data. These instructions are `ADD`, `SUB` and `NEG`.

The DSP engine has options selected through bits in the CPU Core Control register (CORCON), as listed below:

- Fractional or integer DSP multiply (IF)
- Signed or unsigned DSP multiply (US)
- Conventional or convergent rounding (RND)
- Automatic saturation on/off for ACCA (SATA)
- Automatic saturation on/off for ACCB (SATB)
- Automatic saturation on/off for writes to data memory (SATDW)
- Accumulator Saturation mode selection (ACCSAT)

A block diagram of the DSP engine is shown in Figure 3-3.

TABLE 3-1: DSP INSTRUCTIONS SUMMARY

Instruction	Algebraic Operation	ACC Write-Back
`CLR`	$A = 0$	Yes
`ED`	$A = (x - y)^2$	No
`EDAC`	$A = A + (x - y)^2$	No
`MAC`	$A = A + (x * y)$	Yes
`MAC`	$A = A + x^2$	No
`MOVSAC`	No change in A	Yes
`MPY`	$A = x * y$	No
`MPY`	$A = x^2$	No
`MPY.N`	$A = - x * y$	No
`MSC`	$A = A - x * y$	Yes

TABLE 4-33: HIGH-SPEED 10-BIT ADC REGISTER MAP FOR dsPIC33FJ32GS608 AND dsPIC33FJ64GS608 DEVICES

File Name	SFR Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADCON	0300	ADON	—	ADSIDL	SLOWCLK	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	ASYNCSAMP	—	ADCS2	ADCS1	ADCS0	0003
ADPCFG	0302	PCFG<15:0>																0000
ADPCFG2	0304	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCFG<17:16>		0000
ADSTAT	0306	—	—	—	P12RDY	—	—	—	P8RDY	P7RDY	P6RDY	P5RDY	P4RDY	P3RDY	P2RDY	P1RDY	P0RDY	0000
ADBASE	0308	ADBASE<15:1>															—	0000
ADCPC0	030A	IRQEN1	PEND1	SWTRG1	TRGSRC14	TRGSRC13	TRGSRC12	TRGSRC11	TRGSRC10	IRQEN0	PEND0	SWTRG0	TRGSRC04	TRGSRC03	TRGSRC02	TRGSRC01	TRGSRC00	0000
ADCPC1	030C	IRQEN3	PEND3	SWTRG3	TRGSRC34	TRGSRC33	TRGSRC32	TRGSRC31	TRGSRC30	IRQEN2	PEND2	SWTRG2	TRGSRC24	TRGSRC23	TRGSRC22	TRGSRC21	TRGSRC20	0000
ADCPC2	030E	IRQEN5	PEND5	SWTRG5	TRGSRC54	TRGSRC53	TRGSRC52	TRGSRC51	TRGSRC50	IRQEN4	PEND4	SWTRG4	TRGSRC44	TRGSRC43	TRGSRC42	TRGSRC41	TRGSRC40	0000
ADCPC3	0310	IRQEN7	PEND7	SWTRG7	TRGSRC74	TRGSRC73	TRGSRC72	TRGSRC71	TRGSRC70	IRQEN6	PEND6	SWTRG6	TRGSRC64	TRGSRC63	TRGSRC62	TRGSRC61	TRGSRC640	0000
ADCPC4	0312	—	—	—	—	—	—	—	—	IRQEN8	PEND8	SWTRG8	TRGSRC84	TRGSRC83	TRGSRC82	TRGSRC81	TRGSRC80	0000
ADCPC6	0316	—	—	—	—	—	—	—	—	IRQEN12	PEND12	SWTRG12	TRGSRC124	TRGSRC123	TRGSRC122	TRGSRC121	TRGSRC120	0000
ADCBUF0	0340	ADC Data Buffer 0																xxxx
ADCBUF1	0342	ADC Data Buffer 1																xxxx
ADCBUF2	0344	ADC Data Buffer 2																xxxx
ADCBUF3	0346	ADC Data Buffer 3																xxxx
ADCBUF4	0348	ADC Data Buffer 4																xxxx
ADCBUF5	034A	ADC Data Buffer 5																xxxx
ADCBUF6	034C	ADC Data Buffer 6																xxxx
ADCBUF7	034E	ADC Data Buffer 7																xxxx
ADCBUF8	0350	ADC Data Buffer 8																xxxx
ADCBUF9	0352	ADC Data Buffer 9																xxxx
ADCBUF10	0354	ADC Data Buffer 10																xxxx
ADCBUF11	0356	ADC Data Buffer 11																xxxx
ADCBUF12	0358	ADC Data Buffer 12																xxxx
ADCBUF13	035A	ADC Data Buffer 13																xxxx
ADCBUF14	035C	ADC Data Buffer 14																xxxx
ADCBUF15	035E	ADC Data Buffer 15																xxxx
ADCBUF16	0360	ADC Data Buffer 16																xxxx
ADCBUF17	0362	ADC Data Buffer 17																xxxx
ADCBUF24	0370	ADC Data Buffer 24																xxxx
ADCBUF25	0372	ADC Data Buffer 25																xxxx

Legend: x = unknown value on Reset, — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal.

5.2 RTSP Operation

The dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610 Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user application to erase a page of memory, which consists of eight rows (512 instructions) at a time, and to program one row or one word at a time. Table 27-12 shows typical erase and programming times. The 8-row erase pages and single row write rows are edge-aligned from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

The program memory implements holding buffers that can contain 64 instructions of programming data. Prior to the actual programming operation, the write data must be loaded into the buffers sequentially. The instruction words loaded must always be from a group of 64 boundary.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of `TBLWT` instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register. A total of 64 `TBLWTL` and `TBLWTH` instructions are required to load the instructions.

All of the Table Write operations are single-word writes (two instruction cycles) because only the buffers are written. A programming cycle is required for programming each row.

5.3 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. The processor stalls (waits) until the programming operation is finished.

The programming time depends on the FRC accuracy (see Table 27-20) and the value of the FRC Oscillator Tuning register (see Register 9-4). Use the following formula to calculate the minimum and maximum values for the Row Write Time, Page Erase Time and Word Write Cycle Time parameters (see Table 27-12).

EQUATION 5-1: PROGRAMMING TIME

$$\frac{T}{7.37\ MHz \times (FRC\ Accuracy)\% \times (FRC\ Tuning)\%}$$

For example, if the device is operating at +125°C, the FRC accuracy will be ±2%. If the TUN<5:0> bits (see Register 9-4) are set to `'b000000`, the minimum row write time is equal to Equation 5-2.

EQUATION 5-2: MINIMUM ROW WRITE TIME

$$T_{RW} = \frac{11064\ Cycles}{7.37\ MHz \times (1 + 0.02) \times (1 - 0.000938)} = 1.473\ ms$$

The maximum row write time is equal to Equation 5-3.

EQUATION 5-3: MAXIMUM ROW WRITE TIME

$$T_{RW} = \frac{11064\ Cycles}{7.37\ MHz \times (1 - 0.02) \times (1 - 0.000938)} = 1.533\ ms$$

Setting the WR bit (NVMCON<15>) starts the operation and the WR bit is automatically cleared when the operation is finished.

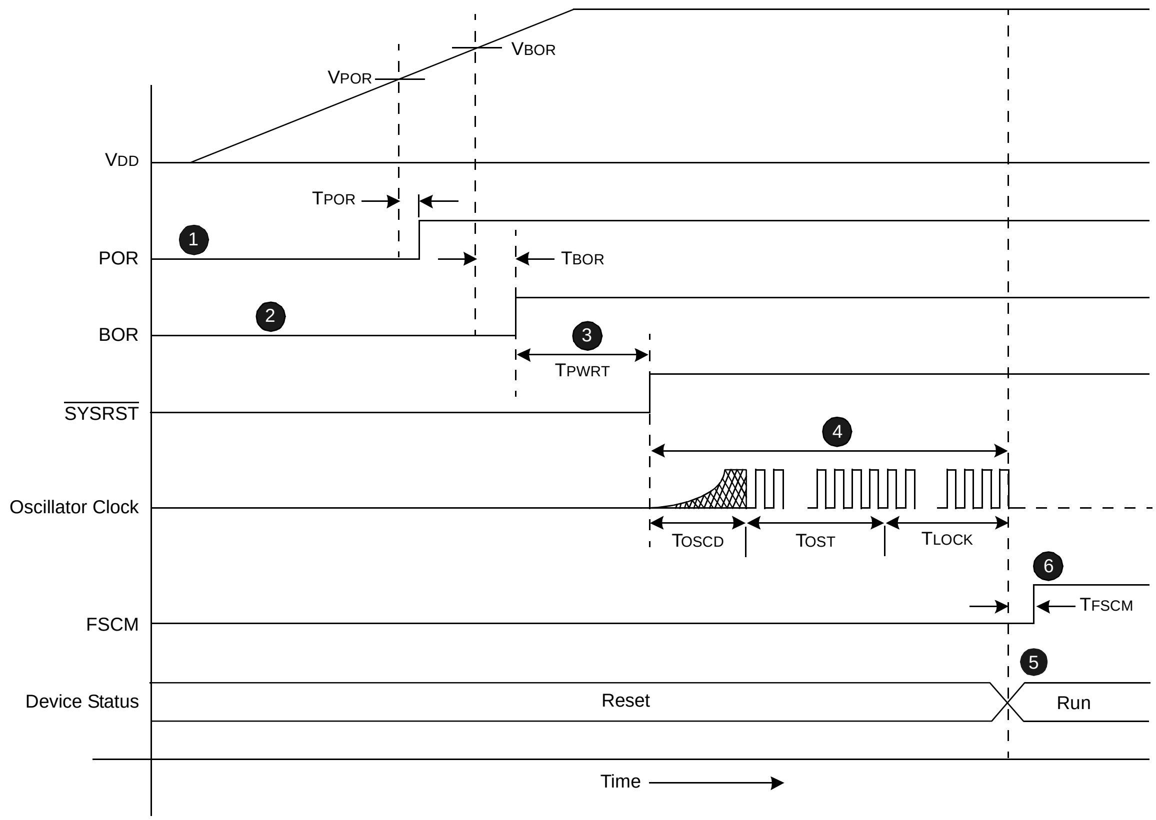
5.4 Control Registers

Two SFRs are used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls which blocks are to be erased, which memory type is to be programmed and the start of the programming cycle.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user application must consecutively write 0x55 and 0xAA to the NVMKEY register. Refer to **Section 5.3 “Programming Operations”** for further details.

FIGURE 6-2: SYSTEM RESET TIMING



Note 1: **POR:** A POR circuit holds the device in Reset when the power supply is turned on. The POR circuit is active until V_{DD} crosses the V_{POR} threshold and the delay, T_{POR}, has elapsed.

2: **BOR:** The on-chip voltage regulator has a BOR circuit that keeps the device in Reset until V_{DD} crosses the V_{BOR} threshold and the delay, T_{BOR}, has elapsed. The delay, T_{BOR}, ensures the voltage regulator output becomes stable.

3: **PWRT Timer:** The programmable Power-up Timer (PWRT) continues to hold the processor in Reset for a specific period of time (T_{PWRT}) after a BOR. The delay, T_{PWRT}, ensures that the system power supplies have stabilized at the appropriate level for full-speed operation. After the delay, T_{PWRT} has elapsed and the $\overline{\text{SYSRST}}$ becomes inactive, which in turn, enables the selected oscillator to start generating clock cycles.

4: **Oscillator Delay:** The total delay for the clock to be ready for various clock source selections is given in Table 6-1. Refer to **Section 9.0 “Oscillator Configuration”** for more information.

5: When the oscillator clock is ready, the processor begins execution from location, 0x000000. The user application programs a `GOTO` instruction at the Reset address, which redirects program execution to the appropriate start-up routine.

6: If the Fail-Safe Clock Monitor (FSCM) is enabled, it begins to monitor the system clock when the system clock is ready and the delay, T_{FSCM}, has elapsed.

TABLE 6-2: OSCILLATOR DELAY

Symbol	Parameter	Value
V_{POR}	POR Threshold	1.8V nominal
T_{POR}	POR Extension Time	30 µs maximum
V_{BOR}	BOR Threshold	2.5V nominal
T_{BOR}	BOR Extension Time	100 µs maximum
T_{PWRT}	Programmable Power-up Time Delay	0-128 ms nominal
T_{FSCM}	Fail-Safe Clock Monitor Delay	900 µs maximum

Note: When the device exits the Reset condition (begins normal operation), the device operating parameters (voltage, frequency, temperature, etc.) must be within their operating ranges; otherwise, the device may not function correctly. The user application must ensure that the delay between the time power is first applied, and the time $\overline{\text{SYSRST}}$ becomes inactive, is long enough to get all operating parameters within specification.

REGISTER 7-15: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	IC4IE	IC3IE	DMA3IE	C1IE(1)	C1RXIE(1)	SPI2IE	SPI2EIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6 **IC4IE:** Input Capture Channel 4 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 5 **IC3IE:** Input Capture Channel 3 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 4 **DMA3IE:** DMA Channel 3 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 3 **C1IE:** ECAN1 Event Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 2 **C1RXIE:** ECAN1 Receive Data Ready Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **SPI2IE:** SPI2 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **SPI2EIE:** SPI2 Error Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

Note 1: Interrupts are disabled on devices without ECAN™ modules.

REGISTER 7-36: IPC18: INTERRUPT PRIORITY CONTROL REGISTER 18

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	QEI2IP2	QEI2IP1	QEI2IP0	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PSESMIP2	PSESMIP1	PSESMIP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **QEI2IP<2:0>:** QEI2 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11-7 **Unimplemented:** Read as '0'

bit 6-4 **PSESMIP<2:0>:** PWM Special Event Secondary Match Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

15.1 Output Compare Modes

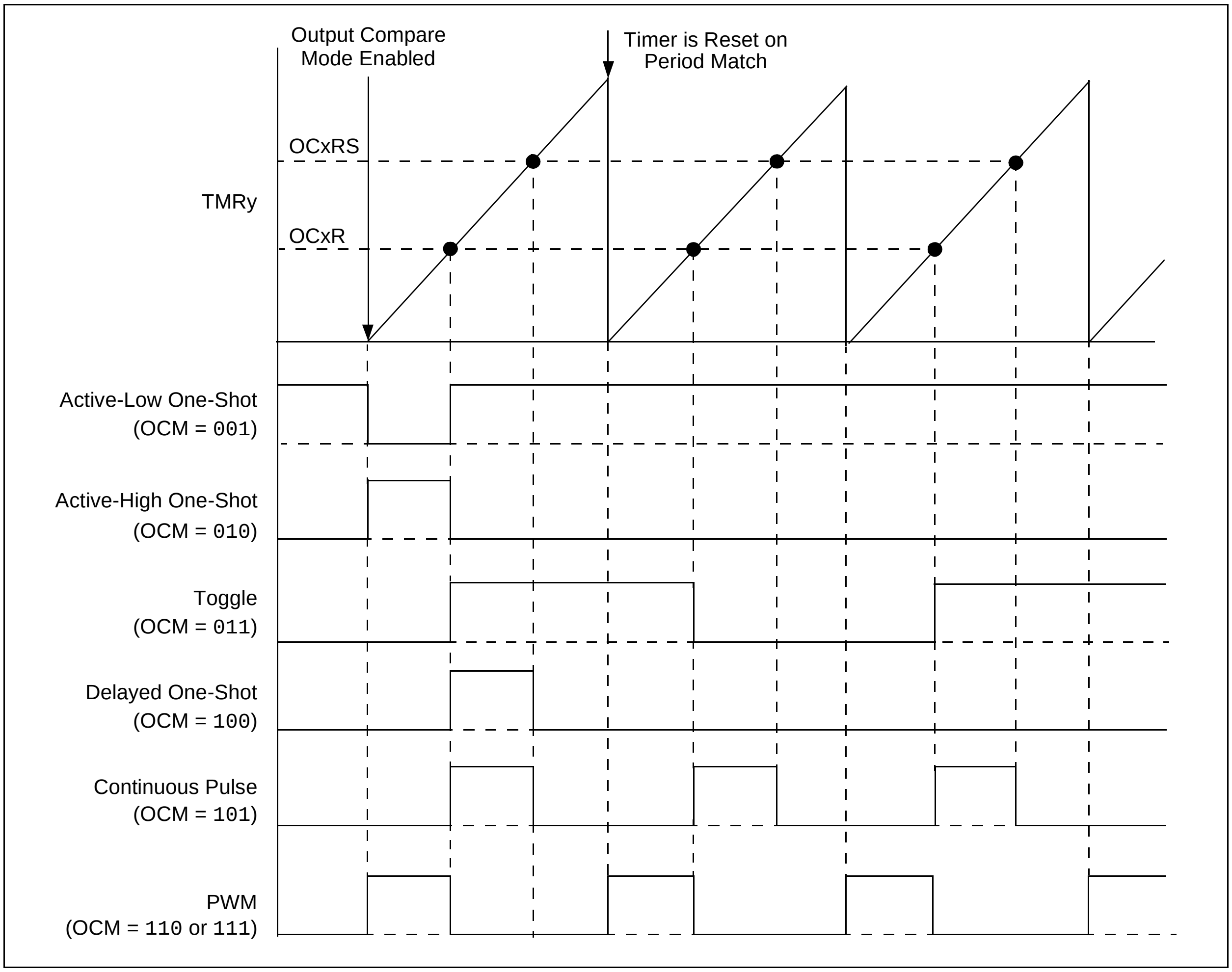
Configure the Output Compare modes by setting the appropriate Output Compare Mode (OCM<2:0>) bits in the Output Compare Control (OCxCON<2:0>) register. Table 15-1 lists the different bit settings for the Output Compare modes. Figure 15-2 illustrates the output compare operation for various modes. The user application must disable the associated timer when writing to the Output Compare Control registers to avoid malfunctions.

Note:	See **"Output Compare"** (DS70005157) in the *"dsPIC33/PIC24 Family Reference Manual"* for OCxR and OCxRS register restrictions.

TABLE 15-1: OUTPUT COMPARE MODES

OCM<2:0>	Mode	OCx Pin Initial State	OCx Interrupt Generation
`000`	Module Disabled	Controlled by GPIO register	—
`001`	Active-Low One-Shot	`0`	OCx rising edge
`010`	Active-High One-Shot	`1`	OCx falling edge
`011`	Toggle	Current output is maintained	OCx rising and falling edge
`100`	Delayed One-Shot	`0`	OCx falling edge
`101`	Continuous Pulse	`0`	OCx falling edge
`110`	PWM without Fault Protection	'`0`' if OCxR is zero, '`1`' if OCxR is non-zero	No interrupt
`111`	PWM with Fault Protection	'`0`' if OCxR is zero, '`1`' if OCxR is non-zero	OCFA falling edge for OC1 to OC4

FIGURE 15-2: OUTPUT COMPARE x OPERATION



NOTES:

16.3 Control Registers

The following registers control the operation of the high-speed PWM module.

- PTCON: PWM Time Base Control Register
- PTCON2: PWM Clock Divider Select Register 2
- PTPER: PWM Primary Master Time Base Period Register(1,2)
- SEVTCMP: PWM Special Event Compare Register(1)
- STCON: PWM Secondary Master Time Base Control Register
- STCON2: PWM Secondary Clock Divider Select Register 2
- STPER: PWM Secondary Master Time Base Period Register
- SSEVTCMP: PWM Secondary Special Event Compare Register
- CHOP: PWM Chop Clock Generator Register(1)
- MDC: PWM Master Duty Cycle Register(1,2)
- PWMCONx: PWM Control x Register
- PDCx: PWM Generator Duty Cycle x Register(1,2,3)
- PHASEx: PWM Primary Phase-Shift x Register(1,2)
- DTRx: PWM Dead-Time x Register
- ALTDTRx: PWM Alternate Dead-Time x Register
- SDCx: PWM Secondary Duty Cycle x Register(1,2,3)
- SPHASEx: PWM Secondary Phase-Shift x Register(1,2)
- TRGCONx: PWM Trigger Control x Register
- IOCONx: PWM I/O Control x Register
- FCLCONx: PWM Fault Current-Limit Control x Register
- TRIGx: PWM Primary Trigger x Compare Value Register
- STRIGx: PWM Secondary Trigger x Compare Value Register(1)
- LEBCONx: Leading-Edge Blanking Control x Register
- LEBDLYx: Leading-Edge Blanking Delay x Register
- AUXCONx: PWM Auxiliary Control x Register
- PWMCAPx: Primary PWM Time Base Capture x Register

REGISTER 18-3: SPIxCON2: SPIx CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	FRMPOL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	FRMDLY	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **FRMEN:** Framed SPIx Support bit
1 = Framed SPIx support is enabled ($\overline{SSx}$ pin is used as Frame Sync pulse input/output)
0 = Framed SPIx support is disabled

bit 14 **SPIFSD:** Frame Sync Pulse Direction Control bit
1 = Frame Sync pulse input (slave)
0 = Frame Sync pulse output (master)

bit 13 **FRMPOL:** Frame Sync Pulse Polarity bit
1 = Frame Sync pulse is active-high
0 = Frame Sync pulse is active-low

bit 12-2 **Unimplemented:** Read as '0'

bit 1 **FRMDLY:** Frame Sync Pulse Edge Select bit
1 = Frame Sync pulse coincides with first bit clock
0 = Frame Sync pulse precedes first bit clock

bit 0 **Unimplemented:** This bit must not be set to '1' by the user application

REGISTER 19-2: I2CxSTAT: I2Cx STATUS REGISTER (CONTINUED)

bit 3	**S:** Start bit 1 = Indicates that a Start (or Repeated Start) bit has been detected last 0 = Start bit was not detected last Hardware is set or clear when Start, Repeated Start or Stop is detected.
bit 2	**R_W:** Read/Write Information bit (when operating as I^2C slave) 1 = Read – indicates data transfer is output from slave 0 = Write – indicates data transfer is input to slave Hardware is set or clear after reception of an I^2C device address byte.
bit 1	**RBF:** Receive Buffer Full Status bit 1 = Receive is complete, I2CxRCV is full 0 = Receive is not complete, I2CxRCV is empty Hardware is set when I2CxRCV is written with a received byte. Hardware is clear when software reads I2CxRCV.
bit 0	**TBF:** Transmit Buffer Full Status bit 1 = Transmit in progress, I2CxTRN is full 0 = Transmit is complete, I2CxTRN is empty Hardware is set when software writes to I2CxTRN. Hardware is clear at completion of the data transmission.

BUFFER 21-5: ECANx MESSAGE BUFFER WORD 4

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 3							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 2							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8	**Byte 3<15:8>:** ECANx Message Byte 3
bit 7-0	**Byte 2<7:0>:** ECANx Message Byte 2

BUFFER 21-6: ECANx MESSAGE BUFFER WORD 5

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 5							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 4							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8	**Byte 5<15:8>:** ECANx Message Byte 5
bit 7-0	**Byte 4<7:0>:** ECANx Message Byte 4

REGISTER 22-6: ADCPC0: ADC CONVERT PAIR CONTROL REGISTER 0 (CONTINUED)

bit 4-0 **TRGSRC0<4:0>:** Trigger 0 Source Selection bits

Selects trigger source for conversion of Analog Channels AN1 and AN0.
`11111` = Timer2 period match
`11110` = PWM Generator 8 current-limit ADC trigger
`11101` = PWM Generator 7 current-limit ADC trigger
`11100` = PWM Generator 6 current-limit ADC trigger
`11011` = PWM Generator 5 current-limit ADC trigger
`11010` = PWM Generator 4 current-limit ADC trigger
`11001` = PWM Generator 3 current-limit ADC trigger
`11000` = PWM Generator 2 current-limit ADC trigger
`10111` = PWM Generator 1 current-limit ADC trigger
`10110` = PWM Generator 9 secondary trigger is selected
`10101` = PWM Generator 8 secondary trigger is selected
`10100` = PWM Generator 7 secondary trigger is selected
`10011` = PWM Generator 6 secondary trigger is selected
`10010` = PWM Generator 5 secondary trigger is selected
`10001` = PWM Generator 4 secondary trigger is selected
`10000` = PWM Generator 3 secondary trigger is selected
`01111` = PWM Generator 2 secondary trigger is selected
`01110` = PWM Generator 1 secondary trigger is selected
`01101` = PWM secondary Special Event Trigger is selected
`01100` = Timer1 period match
`01011` = PWM Generator 8 primary trigger is selected
`01010` = PWM Generator 7 primary trigger is selected
`01001` = PWM Generator 6 primary trigger is selected
`01000` = PWM Generator 5 primary trigger is selected
`00111` = PWM Generator 4 primary trigger is selected
`00110` = PWM Generator 3 primary trigger is selected
`00101` = PWM Generator 2 primary trigger is selected
`00100` = PWM Generator 1 primary trigger is selected
`00011` = PWM Special Event Trigger is selected
`00010` = Global software trigger is selected
`00001` = Individual software trigger is selected
`00000` = No conversion is enabled

Note 1: The trigger source must be set as an individual software trigger prior to setting this bit to '1'. If other conversions are in progress, the conversion is performed when the conversion resources are available.

REGISTER 22-12: ADCPC6: ADC CONVERT PAIR CONTROL REGISTER 6(2)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRQEN12	PEND12	SWTRG12	TRGSRC124	TRGSRC123	TRGSRC122	TRGSRC121	TRGSRC120
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7 **IRQEN12:** Interrupt Request Enable 12 bit
1 = Enables IRQ generation when requested conversion of Channels AN25 and AN24 is completed
0 = IRQ is not generated

bit 6 **PEND12:** Pending Conversion Status 12 bit
1 = Conversion of Channels AN25 and AN24 is pending; set when selected trigger is asserted
0 = Conversion is complete

bit 5 **SWTRG12:** Software Trigger 12 bit
1 = Starts conversion of AN25 (INTREF) and AN24 (EXTREF) if selected by the TRGSRCx<4:0> bits(1)
This bit is automatically cleared by hardware when the PEND12 bit is set.
0 = Conversion has not started

Note 1: The trigger source must be set as an individual software trigger prior to setting this bit to ‘1’. If other conversions are in progress, the conversion is performed when the conversion resources are available.

2: This register is not available on dsPIC33FJ32GS406 and dsPIC33FJ64GS406 devices.

26.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16 and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

26.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

TABLE 27-17: PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 3.0V TO 3.6V)

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
OS50	F_{PLLI}	PLL Voltage Controlled Oscillator (VCO) Input Frequency Range	0.8	—	8	MHz	ECPLL, XTPLL modes
OS51	F_{SYS}	On-Chip VCO System Frequency	100	—	200	MHz	
OS52	T_{LOCK}	PLL Start-up Time (Lock Time)	0.9	1.5	3.1	mS	
OS53	D_{CLK}	CLKO Stability (Jitter)(2)	-3	0.5	3	%	Measured over a 100 ms period

Note 1: Data in “Typ” column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested in manufacturing.

2: These parameters are characterized by similarity, but are not tested in manufacturing. This specification is based on clock cycle by clock cycle measurements. To calculate the effective jitter for individual time bases or communication clocks, use this formula:

$$Peripheral\ Clock\ Jitter = \frac{D_{CLK}}{\sqrt{\left(\frac{F_{OSC}}{Peripheral\ Bit\ Rate\ Clock}\right)}}$$

For example: F_{OSC} = 32 MHz, D_{CLK} = 3%, SPI bit rate clock (i.e., SCK) is 2 MHz.

$$SPI\ SCK\ Jitter = \left[\frac{D_{CLK}}{\sqrt{\left(\frac{32\ MHz}{2\ MHz}\right)}}\right] = \left[\frac{3\%}{\sqrt{16}}\right] = \left[\frac{3\%}{4}\right] = 0.75\%$$

TABLE 27-18: AUXILIARY PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 3.0V TO 3.6V)

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
OS56	F_{HPOUT}	On-Chip, 16x PLL CCO Frequency	112	118	120	MHz	
OS57	F_{HPIN}	On-Chip, 16x PLL Phase Detector Input Frequency	7.0	7.37	7.5	MHz	
OS58	T_{SU}	Frequency Generator Lock Time	—	—	10	µs	

Note 1: Data in “Typ” column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested in manufacturing.

TABLE 27-24: TIMER2/4 EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic(1)		Min	Typ	Max	Units	Conditions
TB10	TtxH	TxCK High Time	Synchronous mode	Greater of: 20 or (TCY + 20)/N	—	—	ns	Must also meet Parameter TB15, N = Prescale value (1, 8, 64, 256)
TB11	TtxL	TxCK Low Time	Synchronous mode	Greater of: 20 or (TCY + 20)/N	—	—	ns	Must also meet Parameter TB15, N = Prescale value (1, 8, 64, 256)
TB15	TtxP	TxCK Input Period	Synchronous mode	Greater of: 40 or (2 TCY + 40)/N	—	—	ns	N = Prescale value (1, 8, 64, 256)
TB20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.75 TCY + 40	—	1.75 TCY + 40	ns	

Note 1: These parameters are characterized, but are not tested in manufacturing.

TABLE 27-25: TIMER3/5 EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic(1)		Min	Typ	Max	Units	Conditions
TC10	TtxH	TxCK High Time	Synchronous	TCY + 20	—	—	ns	Must also meet Parameter TC15
TC11	TtxL	TxCK Low Time	Synchronous	TCY + 20	—	—	ns	Must also meet Parameter TC15
TC15	TtxP	TxCK Input Period	Synchronous, with Prescaler	2 TCY + 40	—	—	ns	
TC20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.75 TCY + 40	—	1.75 TCY + 40	ns	

Note 1: These parameters are characterized, but are not tested in manufacturing.

FIGURE 27-27: ECAN™ MODULE I/O TIMING CHARACTERISTICS

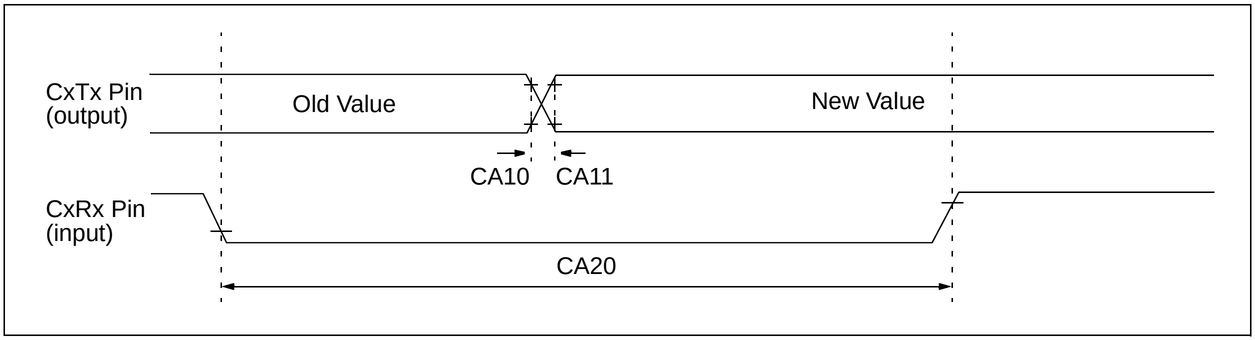


TABLE 27-48: ECAN™ MODULE I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic(1)**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
CA10	TioF	Port Output Fall Time	—	—	—	ns	See Parameter DO32
CA11	TioR	Port Output Rise Time	—	—	—	ns	See Parameter DO31
CA20	Tcwf	Pulse Width to Trigger CAN Wake-up Filter	120	—	—	ns	

Note 1: These parameters are characterized but not tested in manufacturing.

TABLE 27-49: DMA READ/WRITE TIMING REQUIREMENTS

AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended				
Param No.	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
DM1	DMA Read/Write Cycle Time	—	—	1 T_{CY}	ns	

TABLE B-1: MAJOR SECTION UPDATES (CONTINUED)

Section Name	Update Description
Section 9.0 "Oscillator Configuration"	Removed Section 9.2 "FRC Tuning". Removed the PRCDEN, TSEQEN, and LPOSCEN bits from the Oscillator Control Register (see Register 9-1). Updated the Oscillator Tuning Register (see Register 9-4). Removed the Oscillator Tuning Register 2 and the Linear Feedback Shift Register. Updated the default Reset values from R/W-0 to R/W-1 for the SELACLK and APSTSCLR<2:0> bits in the ACLKCON register (see Register 9-5). Renamed the ROSIDL bit to ROSSLP in the REFOCON register (see Register 9-6).
Section 10.0 "Power-Saving Features"	Updated the last paragraph of **Section 10.2.2 "Idle Mode"** to clarify when instruction execution begins. Added Note 1 to the PMD1 register (see Register 10-1).
Section 11.0 "I/O Ports"	Changed the reference to digital-only pins to 5V tolerant pins in the second paragraph of **Section 11.2 "Open-Drain Configuration"**.
Section 16.0 "High-Speed PWM"	Updated the High-Speed PWM Module Register Interconnect Diagram (see Figure 16-2). Updated the SYNCSRC<2:0> = `111`, `101`, and `100` definitions to Reserved in the PTCON and STCON registers (see Register 16-1 and Register 16-5). Updated the PWM time base maximum value from 0xFFFB to 0xFFF8 in the PTPER register (Register 16-3). Updated the smallest pulse width value from 0x0008 to 0x0009 in Note 1 of the shaded note that follows the MDC register (see Register 16-10). Updated the smallest pulse width value from 0x0008 to 0x0009 in Note 2 of the shaded note that follows the PDCx and SDCx registers (see Register 16-12 and Register 16-13). Added Note 2 and updated the FLTDAT<1:0> and CLDAT<1:0> bits, changing the word 'data' to 'state' in the IOCONx register (see Register 16-19).
Section 20.0 "Universal Asynchronous Receiver Transmitter (UART)"	Updated the two baud rate range features to: 10 Mbps to 38 bps at 40 MIPS.
Section 22.0 "High-Speed 10-bit Analog-to-Digital Converter (ADC)"	Updated the TRGSRCx<4:0> = `01101` definition from Reserved to PWM secondary special event trigger selected, and updated Note 1 in the ADCP0-ADCP6 registers (see Register 22-6 through Register 22-12).
Section 24.0 "Special Features"	Updated the second paragraph and removed the fourth paragraph in **Section 24.1 "Configuration Bits"**. Updated the Device Configuration Register Map (see Table 24-1).

NOTES: