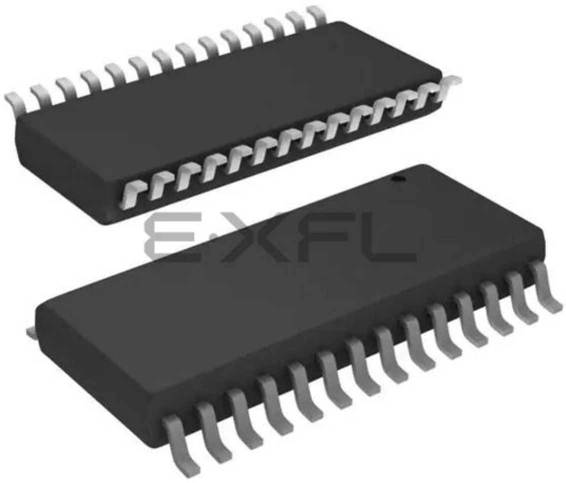
Microchip Technology - DSPIC30F2020-20E/SO Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	20 MIPS
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	21
Program Memory Size	12KB (4K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f2020-20e-so

TABLE 1-1: PINOUT I/O DESCRIPTIONS FOR dsPIC30F1010 (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
RE0-RE7	I/O	ST	PORTE is a bidirectional I/O port.
RF6, RF7, RF8	I/O	ST	PORTF is a bidirectional I/O port.
SCK1 SDI1 SDO1	I/O I O	ST ST —	Synchronous serial clock input/output for SPI #1. SPI #1 Data In. SPI #1 Data Out.
SCL SDA	I/O I/O	ST ST	Synchronous serial clock input/output for I^2C™. Synchronous serial data input/output for I^2C.
T1CK T2CK	I I	ST ST	Timer1 external clock input. Timer2 external clock input.
U1RX U1TX U1ARX U1ATX	I O I O	ST — ST —	UART1 Receive. UART1 Transmit. Alternate UART1 Receive. Alternate UART1 Transmit.
CMP1A CMP1B CMP1C CMP1D CMP2A CMP2B CMP2C CMP2D	I I I I I I I I	Analog Analog Analog Analog Analog Analog Analog Analog	Comparator 1 Channel A Comparator 1 Channel B Comparator 1 Channel C Comparator 1 Channel D Comparator 2 Channel A Comparator 2 Channel B Comparator 2 Channel C Comparator 2 Channel D
CN0-CN7	I	ST	Input Change notification inputs Can be software programmed for internal weak pull-ups on all inputs.
V_{DD}	P	—	Positive supply for logic and I/O pins.
V_{SS}	P	—	Ground reference for logic and I/O pins.
EXTREF	I	Analog	External reference to Comparator DAC

Legend: CMOS = CMOS compatible input or output; Analog = Analog input
ST = Schmitt Trigger input with CMOS levels; O = Output
I = Input; P = Power

2.4.1 MULTIPLIER

The 17x17-bit multiplier is capable of signed or unsigned operation and can multiplex its output using a scaler to support either 1.31 fractional (Q31) or 32-bit integer results. Unsigned operands are zero-extended into the 17th bit of the multiplier input value. Signed operands are sign-extended into the 17th bit of the multiplier input value. The output of the 17x17-bit multiplier/scaler is a 33-bit value, which is sign-extended to 40 bits. Integer data is inherently represented as a signed two's complement value, where the MSB is defined as a sign bit. Generally speaking, the range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1} - 1$. For a 16-bit integer, the data range is -32768 (0x8000) to 32767 (0x7FFF), including 0. For a 32-bit integer, the data range is -2,147,483,648 (0x8000 0000) to 2,147,483,645 (0x7FFF FFFF).

When the multiplier is configured for fractional multiplication, the data is represented as a two's complement fraction, where the MSB is defined as a sign bit and the radix point is implied to lie just after the sign bit (QX format). The range of an N-bit two's complement fraction with this implied radix point is -1.0 to $(1-2^{1-N})$. For a 16-bit fraction, the Q15 data range is -1.0 (0x8000) to 0.999969482 (0x7FFF), including 0, and has a precision of 3.01518×10^{-5}. In Fractional mode, a 16x16 multiply operation generates a 1.31 product, which has a precision of 4.65661×10^{-10}.

The same multiplier is used to support the MCU multiply instructions, which include integer 16-bit signed, unsigned and mixed sign multiplies.

The `MUL` instruction may be directed to use byte or word sized operands. Byte operands will direct a 16-bit result, and word operands will direct a 32-bit result to the specified register(s) in the W array.

2.4.2 DATA ACCUMULATORS AND ADDER/SUBTRACTER

The data accumulator consists of a 40-bit adder/subtracter with automatic sign extension logic. It can select one of two accumulators (A or B) as its pre-accumulation source and post-accumulation destination. For the `ADD` and `LAC` instructions, the data to be accumulated or loaded can be optionally scaled via the barrel shifter, prior to accumulation.

2.4.2.1 Adder/Subtracter, Overflow and Saturation

The adder/subtracter is a 40-bit adder with an optional zero input into one side and either true or complement data into the other input. In the case of addition, the carry/$\overline{\text{borrow}}$ input is active high and the other input is true data (not complemented), whereas in the case of subtraction, the carry/$\overline{\text{borrow}}$ input is active low and the other input is complemented. The adder/subtracter generates overflow Status bits SA/SB and OA/OB, which are latched and reflected in the STATUS register.

- Overflow from bit 39: this is a catastrophic overflow in which the sign of the accumulator is destroyed.
- Overflow into guard bits 32 through 39: this is a recoverable overflow. This bit is set whenever all the guard bits are not identical to each other.

The adder has an additional saturation block which controls accumulator data saturation, if selected. It uses the result of the adder, the overflow Status bits described above, and the SATA/B (CORCON<7:6>) and ACCSAT (CORCON<4>) mode control bits to determine when and to what value to saturate.

Six STATUS register bits have been provided to support saturation and overflow; they are:

1. OA:
   ACCA overflowed into guard bits
2. OB:
   ACCB overflowed into guard bits
3. SA:
   ACCA saturated (bit 31 overflow and saturation)
   or
   ACCA overflowed into guard bits and saturated (bit 39 overflow and saturation)
4. SB:
   ACCB saturated (bit 31 overflow and saturation)
   or
   ACCB overflowed into guard bits and saturated (bit 39 overflow and saturation)
5. OAB:
   Logical OR of OA and OB
6. SAB:
   Logical OR of SA and SB

The OA and OB bits are modified each time data passes through the adder/subtracter. When set, they indicate that the most recent operation has overflowed into the accumulator guard bits (bits 32 through 39). The OA and OB bits can also optionally generate an arithmetic warning trap when set and the corresponding overflow trap flag enable bit (OVATE, OVBTE) in the INTCON1 register (refer to **Section 5.0 "Interrupts"**) is set. This allows the user to take immediate action, for example, to correct system gain.

NOTES:

5.2 Reset Sequence

A Reset is not a true exception, because the interrupt controller is not involved in the Reset process. The processor initializes its registers in response to a Reset, which forces the PC to zero. The processor then begins program execution at location 0x000000. A `GOTO` instruction is stored in the first program memory location, immediately followed by the address target for the `GOTO` instruction. The processor executes the `GOTO` to the specified address and then begins operation at the specified target (start) address.

5.2.1 RESET SOURCES

In addition to External Reset and Power-on Reset (POR), there are 6 sources of error conditions which 'trap' to the Reset vector.

- Watchdog Time-out:
  The watchdog has timed out, indicating that the processor is no longer executing the correct flow of code.
- Uninitialized W Register Trap:
  An attempt to use an uninitialized W register as an Address Pointer will cause a Reset.
- Illegal Instruction Trap:
  Attempted execution of any unused opcodes will result in an illegal instruction trap. Note that a fetch of an illegal instruction does not result in an illegal instruction trap if that instruction is flushed prior to execution due to a flow change.
- Trap Lockout:
  Occurrence of multiple Trap conditions simultaneously will cause a Reset.

5.3 Traps

Traps can be considered as non-maskable interrupts indicating a software or hardware error, which adhere to a predefined priority as shown in Figure 5-1. They are intended to provide the user a means to correct erroneous operation during debug and when operating within the application.

Note:	If the user does not intend to take corrective action in the event of a Trap Error condition, these vectors must be loaded with the address of a default handler that simply contains the `RESET` instruction. If, on the other hand, one of the vectors containing an invalid address is called, an address error trap is generated.

Note that many of these trap conditions can only be detected when they occur. Consequently, the questionable instruction is allowed to complete prior to trap exception processing. If the user chooses to recover from the error, the result of the erroneous action that caused the trap may have to be corrected.

There are 8 fixed priority levels for traps: Level 8 through Level 15, which implies that the IPL3 is always set during processing of a trap.

If the user is not currently executing a trap, and he sets the IPL<3:0> bits to a value of '`0111`' (Level 7), then all interrupts are disabled, but traps can still be processed.

5.3.1 TRAP SOURCES

The following traps are provided with increasing priority. However, since all traps can be nested, priority has little effect.

Math Error Trap:

The Math Error trap executes under the following four circumstances:

1. Should an attempt be made to divide by zero, the divide operation will be aborted on a cycle boundary and the trap taken.
2. If enabled, a Math Error trap will be taken when an arithmetic operation on either accumulator A or B causes an overflow from bit 31 and the accumulator guard bits are not utilized.
3. If enabled, a Math Error trap will be taken when an arithmetic operation on either accumulator A or B causes a catastrophic overflow from bit 39 and all saturation is disabled.
4. If the shift amount specified in a shift instruction is greater than the maximum allowed shift amount, a trap will occur.

REGISTER 5-9: IPC0: INTERRUPT PRIORITY CONTROL REGISTER 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP<2:0>			—	OC1IP<2:0>		
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP<2:0>			—	INT0IP<2:0>		
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 **T1IP<2:0>:** Timer1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **OC1IP<2:0>:** Output Compare Channel 1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **IC1IP<2:0>:** Input Capture Channel 1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **INT0IP<2:0>:** External Interrupt 0 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 5-10: IPC1: INTERRUPT PRIORITY CONTROL REGISTER 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T3IP<2:0>			—	T2IP<2:0>		
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC2IP<2:0>			—	—	—	—
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 **T3IP<2:0>:** Timer3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **T2IP<2:0>:** Timer2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **OC2IP<2:0>:** Output Compare Channel 2 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’

TABLE 6-1: dsPIC30F1010/2020 PORT REGISTER MAP

SFR Name	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	Reset State
TRISA	02C0	—	—	—	—	—	—	TRISA9	—	—	—	—	—	—	—	—	—	0000 0010 0000 0000
PORTA	02C2	—	—	—	—	—	—	RA9	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATA	02C4	—	—	—	—	—	—	LAT9	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISB	02C6	—	—	—	—	—	—	—	—	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 0000 0011 1111
PORTB	02C8	—	—	—	—	—	—	—	—	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CA	—	—	—	—	—	—	—	—	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRISD0	0000 0000 0000 0001
PORTD	02D4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LATD0	0000 0000 0000 0000
TRISE	02D8	—	—	—	—	—	—	—	—	TRSE7	TRSE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	0000 0000 1111 1111
PORTE	02DA	—	—	—	—	—	—	—	—	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	0000 0000 0000 0000
LATE	02DC	—	—	—	—	—	—	—	—	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	0000 0000 0000 0000
TRISF	02DE	—	—	—	—	—	—	—	TRISF8	TRISF7	TRISF6	—	—	—	—	—	—	0000 0001 1100 0000
PORTF	02E0	—	—	—	—	—	—	—	RF8	RF7	RF6	—	—	—	—	—	—	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	LATF8	LATF7	LATF6	—	—	—	—	—	—	0000 0000 0000 0000

Note: Refer to the "*dsPIC30F Family Reference Manual*" (DS70046) for descriptions of register bit fields.

TABLE 6-2: dsPIC30F2023 PORT REGISTER MAP

SFR Name	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	Reset State
TRISA	02C0	—	—	—	—	TRISA11	TRISA10	TRIS9	TRISA8	—	—	—	—	—	—	—	—	0000 1111 0000 0000
PORTA	02C2	—	—	—	—	RA11	RA10	RA9	RA8	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATA	02C4	—	—	—	—	LATA11	LATA10	LATA9	LATA8	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISB	02C6	—	—	—	—	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRIS6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 1111 1111 1111
PORTB	02C8	—	—	—	—	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CA	—	—	—	—	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRISD1	TRISD0	0000 0000 0000 0011
PORTD	02D4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RD1	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LATD1	LATD0	0000 0000 0000 0000
TRISE	02D8	—	—	—	—	—	—	—	—	TRSE7	TRSE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	0000 0000 1111 1111
PORTE	02DA	—	—	—	—	—	—	—	—	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	0000 0000 0000 0000
LATE	02DC	—	—	—	—	—	—	—	—	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	0000 0000 0000 0000
TRISF	02DE	TRISF15	TRISF14	—	—	—	—	—	TRISF8	TRISF7	TRISF6	—	—	TRISF3	TRISF2	—	—	1100 0001 1100 1100
PORTF	02E0	RF15	RF14	—	—	—	—	—	RF8	RF7	RF6	—	—	RF3	RF2	—	—	0000 0000 0000 0000
LATF	02E2	LATF15	LATF14	—	—	—	—	—	LATF8	LATF7	LATF6	—	—	LATF3	LATF2	—	—	0000 0000 0000 0000
TRISG	02E4	—	—	—	—	—	—	—	—	—	—	—	—	TRISG3	TRISG2	—	—	0000 0000 0000 1100
PORTG	02E6	—	—	—	—	—	—	—	—	—	—	—	—	RG3	RG2	—	—	0000 0000 0000 0000
LATG	02E8	—	—	—	—	—	—	—	—	—	—	—	—	LATG3	LATG2	—	—	0000 0000 0000 0000

Note: Refer to the "*dsPIC30F Family Reference Manual*" (DS70046) for descriptions of register bit fields.

TABLE 6-3: dsPIC30F1010/202X INPUT CHANGE NOTIFICATION REGISTER MAP

SFR Name	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	Reset State
CNEN1	0060	—	—	—	—	—	—	—	—	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNPU1	0064	—	—	—	—	—	—	—	—	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000

Note: Refer to the "*dsPIC30F Family Reference Manual*" (DS70046) for descriptions of register bit fields.

NOTES:

10.2 Input Capture Operation During Sleep and Idle Modes

An input capture event will generate a device wake-up or interrupt, if enabled, if the device is in CPU Idle or Sleep mode.

Independent of the timer being enabled, the input capture module will wake-up from the CPU Sleep or Idle mode when a capture event occurs, if ICM<2:0> = `111` and the interrupt enable bit is asserted. The same wake-up can generate an interrupt, if the conditions for processing the interrupt have been satisfied. The wake-up feature is useful as a method of adding extra external pin interrupts.

10.2.1 INPUT CAPTURE IN CPU SLEEP MODE

CPU Sleep mode allows input capture module operation with reduced functionality. In the CPU Sleep mode, the ICI<1:0> bits are not applicable, and the input capture module can only function as an external interrupt source.

The capture module must be configured for interrupt only on the rising edge (ICM<2:0> = `111`), in order for the input capture module to be used while the device is in Sleep mode. The prescale settings of 4:1 or 16:1 are not applicable in this mode.

10.2.2 INPUT CAPTURE IN CPU IDLE MODE

CPU Idle mode allows input capture module operation with full functionality. In the CPU Idle mode, the Interrupt mode selected by the ICI<1:0> bits are applicable, as well as the 4:1 and 16:1 capture prescale settings, which are defined by control bits ICM<2:0>. This mode requires the selected timer to be enabled. Moreover, the ICSIDL bit must be asserted to a logic '`0`'.

If the input capture module is defined as ICM<2:0> = `111` in CPU Idle mode, the input capture pin will serve only as an external interrupt pin.

10.3 Input Capture Interrupts

The input capture channels have the ability to generate an interrupt, based upon the selected number of capture events. The selection number is set by control bits ICI<1:0> (ICxCON<6:5>).

Each channel provides an interrupt flag (ICxIF) bit. The respective capture channel interrupt flag is located in the corresponding IFSx STATUS register.

Enabling an interrupt is accomplished via the respective capture channel interrupt enable (ICxIE) bit. The capture interrupt enable bit is located in the corresponding IEC Control register.

REGISTER 13-3: 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 enabled ($\overline{SSx}$ pin used as frame sync pulse input/output)
0 = Framed SPIx support 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.

TABLE 15-1: UART1 REGISTER MAP

SFR 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
U1MODE	0220	UARTEN	—	USIDL	IREN	—	ALTIO	—	—	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL<1:0>		STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	UART Transmit Register									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	UART Receive Register									0000
U1BRG	0228	Baud Rate Generator Prescaler																0000

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

REGISTER 16-5: A/D CONVERT PAIR CONTROL REGISTER 0 (ADCPC0)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRQEN1	PEND1	SWTRG1	TRGSRC1<4:0>				
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
IRQEN0	PEND0	SWTRG0	TRGSRC0<4:0>				
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 **IRQEN1:** Interrupt Request Enable 1 bit
1 = Enable IRQ generation when requested conversion of channels AN3 and AN2 is completed
0 = IRQ is not generated

bit 14 **PEND1:** Pending Conversion Status 1 bit
1 = Conversion of channels AN3 and AN2 is pending. Set when selected trigger is asserted
0 = Conversion is complete

bit 13 **SWTRG1:** Software Trigger 1 bit
1 = Start conversion of AN3 and AN2 (if selected in TRGSRC bits). If other conversions are in progress, then conversion will be performed when the conversion resources are available. This bit will be reset when the PEND bit is set.

bit 12-8 **TRGSRC1<4:0>:** Trigger 1 Source Selection bits
Selects trigger source for conversion of analog channels AN3 and AN2.
00000 = No conversion enabled
00001 = Individual software trigger selected
00010 = Global software trigger selected
00011 = PWM Special Event Trigger selected
00100 = PWM generator #1 trigger selected
00101 = PWM generator #2 trigger selected
00110 = PWM generator #3 trigger selected
00111 = PWM generator #4 trigger selected
01100 = Timer #1 period match
01101 = Timer #2 period match
01110 = PWM GEN #1 current-limit ADC trigger
01111 = PWM GEN #2 current-limit ADC trigger
10000 = PWM GEN #3 current-limit ADC trigger
10001 = PWM GEN #4 current-limit ADC trigger
10110 = PWM GEN #1 fault ADC trigger
10111 = PWM GEN #2 fault ADC trigger
11000 = PWM GEN #3 fault ADC trigger
11001 = PWM GEN #4 fault ADC trigger

bit 7 **IRQEN0:** Interrupt Request Enable 0 bit
1 = Enable IRQ generation when requested conversion of channels AN1 and AN0 is completed
0 = IRQ is not generated

bit 6 **PEND0:** Pending Conversion Status 0 bit
1 = Conversion of channels AN1 and AN0 is pending. Set when selected trigger is asserted.
0 = Conversion is complete

bit 5 **SWTRG0:** Software Trigger 0 bit
1 = Start conversion of AN1 and AN0 (if selected by TRGSRC bits). If other conversions are in progress, then conversion will be performed when the conversion resources are available. This bit will be reset when the PEND bit is set

Table 18-3 shows the Reset conditions for the RCON register. Since the control bits within the RCON register are R/W, the information in the table implies that all the bits are negated prior to the action specified in the condition column.

TABLE 18-3: INITIALIZATION CONDITION FOR RCON REGISTER CASE 1

Condition	Program Counter	TRAPR	IOPUWR	EXTR	SWR	WDTO	IDLE	SLEEP	POR
Power-on Reset	0x000000	0	0	0	0	0	0	0	1
$\overline{MCLR}$ Reset during normal operation	0x000000	0	0	1	0	0	0	0	0
Software Reset during normal operation	0x000000	0	0	0	1	0	0	0	0
$\overline{MCLR}$ Reset during Sleep	0x000000	0	0	1	0	0	0	1	0
$\overline{MCLR}$ Reset during Idle	0x000000	0	0	1	0	0	1	0	0
WDT Time-out Reset	0x000000	0	0	0	0	1	0	0	0
WDT Wake-up	PC + 2	0	0	0	0	1	0	1	0
Interrupt Wake-up from Sleep	PC + 2(1)	0	0	0	0	0	0	1	0
Clock Failure Trap	0x000004	0	0	0	0	0	0	0	0
Trap Reset	0x000000	1	0	0	0	0	0	0	0
Illegal Operation Trap	0x000000	0	1	0	0	0	0	0	0

Note 1: When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

Table 18-4 shows a second example of the bit conditions for the RCON register. In this case, it is not assumed the user has set/cleared specific bits prior to action specified in the condition column.

TABLE 18-4: INITIALIZATION CONDITION FOR RCON REGISTER CASE 2

Condition	Program Counter	TRAPR	IOPUWR	EXTR	SWR	WDTO	IDLE	SLEEP	POR
Power-on Reset	0x000000	0	0	0	0	0	0	0	1
$\overline{MCLR}$ Reset during normal operation	0x000000	u	u	1	0	0	0	0	u
Software Reset during normal operation	0x000000	u	u	0	1	0	0	0	u
$\overline{MCLR}$ Reset during Sleep	0x000000	u	u	1	u	0	0	1	u
$\overline{MCLR}$ Reset during Idle	0x000000	u	u	1	u	0	1	0	u
WDT Time-out Reset	0x000000	u	u	0	0	1	0	0	u
WDT Wake-up	PC + 2	u	u	u	u	1	u	1	u
Interrupt Wake-up from Sleep	PC + 2(1)	u	u	u	u	u	u	1	u
Clock Failure Trap	0x000004	u	u	u	u	u	u	u	u
Trap Reset	0x000000	1	u	u	u	u	u	u	u
Illegal Operation Reset	0x000000	u	1	u	u	u	u	u	u

Legend: u = unchanged

Note 1: When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

18.9.2 IDLE MODE

In Idle mode, the clock to the CPU is shutdown while peripherals keep running. Unlike Sleep mode, the clock source remains active.

Several peripherals have a control bit in each module that allows them to operate during Idle.

LPRC fail-safe clock remains active if clock failure detect is enabled.

The processor wakes up from Idle if at least one of the following conditions is true:

- on any interrupt that is individually enabled (IE bit is '1') and meets the required priority level
- on any Reset (POR, $\overline{\text{MCLR}}$)
- on WDT time-out

Upon wake-up from Idle mode, the clock is reapplied to the CPU and instruction execution begins immediately, starting with the instruction following the PWRSAV instruction.

Any interrupt that is individually enabled (using IE bit) and meets the prevailing priority level will be able to wake-up the processor. The processor will process the interrupt and branch to the ISR. The Idle status bit in RCON register is set upon wake-up.

Any Reset, other than POR, will set the Idle status bit. On a POR, the Idle bit is cleared.

If Watchdog Timer is enabled, then the processor will wake-up from Idle mode upon WDT time-out. The Idle and WDTO status bits are both set.

Unlike wake-up from Sleep, there are no time delays involved in wake-up from Idle.

18.10 Device Configuration Registers

The Configuration bits in each device Configuration register specify some of the device modes and are programmed by a device programmer, or by using the In-Circuit Serial Programming (ICSP) feature of the device. Each device Configuration register is a 24-bit register, but only the lower 16 bits of each register are used to hold configuration data. There are six Configuration registers available to the user:

1. FBS (0xF80000): Boot Code Segment Configuration Register
2. FGS (0xF80004): General Code Segment Configuration Register
3. FOSCEL (0xF80006): Oscillator Selection Configuration Register
4. FOSC (0xF80008): Oscillator Configuration Register
5. FWDT (0xF8000A): Watchdog Timer Configuration Register
6. FPOR (0xF8000C): Power-On Reset Configuration Register

The placement of the Configuration bits is automatically handled when you select the device in your device programmer. The desired state of the Configuration bits may be specified in the source code (dependent on the language tool used), or through the programming interface. After the device has been programmed, the application software may read the Configuration bit values through the table read instructions. For additional information, please refer to the programming specifications of the device.

Note:	If the code protection configuration fuse bits (GSS<1:0> and GWRP in the FGS register) have been programmed, an erase of the entire code-protected device is only possible at voltages $V_{DD} \geq 4.5V$.

Table 18-5 shows the bit descriptions of the FGS and FBS registers for the dsPIC30F1010. Table 18-6 shows the bit descriptions of the FGS and FBS registers for dsPIC30F202x devices. Table 18-7 shows the bit descriptions of FWDT and the FPOR registers for dsPIC30F1010/202X devices.

TABLE 19-1: SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)

Field	Description
Wb	Base W register ∈ {W0..W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor working register pair (direct addressing)
Wm*Wm	Multiplicand and Multiplier working register pair for Square instructions ∈ {W4 * W4,W5 * W5,W6 * W6,W7 * W7}
Wm*Wn	Multiplicand and Multiplier working register pair for DSP instructions ∈ {W4 * W5,W4 * W6,W4 * W7,W5 * W6,W5 * W7,W6 * W7}
Wn	One of 16 working registers ∈ {W0..W15}
Wnd	One of 16 destination working registers ∈ {W0..W15}
Wns	One of 16 source working registers ∈ {W0..W15}
WREG	W0 (working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }
Wx	X data space prefetch address register for DSP instructions ∈ {[W8] + = 6, [W8] + = 4, [W8] + = 2, [W8], [W8] – = 6, [W8] – = 4, [W8] – = 2, [W9] + = 6, [W9] + = 4, [W9] + = 2, [W9], [W9] – = 6, [W9] – = 4, [W9] – = 2, [W9 + W12],none}
Wxd	X data space prefetch destination register for DSP instructions ∈ {W4..W7}
Wy	Y data space prefetch address register for DSP instructions ∈ {[W10] + = 6, [W10] + = 4, [W10] + = 2, [W10], [W10] - = 6, [W10] - = 4, [W10] - = 2, [W11] + = 6, [W11] + = 4, [W11] + = 2, [W11], [W11] – = 6, [W11] – = 4, [W11] – = 2, [W11 + W12], none}
Wyd	Y data space prefetch destination register for DSP instructions ∈ {W4..W7}

TABLE 19-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
1	ADD	ADD	Acc	Add Accumulators	1	1	OA,OB,SA,SB
		ADD	f	f = f + WREG	1	1	C,DC,N,OV,Z
		ADD	f,WREG	WREG = f + WREG	1	1	C,DC,N,OV,Z
		ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
		ADD	Wso,#Slit4,Acc	16-bit Signed Add to Accumulator	1	1	OA,OB,SA,SB
2	ADDC	ADDC	f	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	AND	AND	f	f = f .AND. WREG	1	1	N,Z
		AND	f,WREG	WREG = f .AND. WREG	1	1	N,Z
		AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N,Z
		AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N,Z
		AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N,Z
4	ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C,N,OV,Z
		ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N,Z
		ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N,Z
5	BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
		BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
6	BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
		BRA	GE,Expr	Branch if greater than or equal	1	1 (2)	None
		BRA	GEU,Expr	Branch if unsigned greater than or equal	1	1 (2)	None
		BRA	GT,Expr	Branch if greater than	1	1 (2)	None
		BRA	GTU,Expr	Branch if unsigned greater than	1	1 (2)	None
		BRA	LE,Expr	Branch if less than or equal	1	1 (2)	None
		BRA	LEU,Expr	Branch if unsigned less than or equal	1	1 (2)	None
		BRA	LT,Expr	Branch if less than	1	1 (2)	None
		BRA	LTU,Expr	Branch if unsigned less than	1	1 (2)	None
		BRA	N,Expr	Branch if Negative	1	1 (2)	None
		BRA	NC,Expr	Branch if Not Carry	1	1 (2)	None
		BRA	NN,Expr	Branch if Not Negative	1	1 (2)	None
		BRA	NOV,Expr	Branch if Not Overflow	1	1 (2)	None
		BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
		BRA	OA,Expr	Branch if accumulator A overflow	1	1 (2)	None
		BRA	OB,Expr	Branch if accumulator B overflow	1	1 (2)	None
		BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
		BRA	SA,Expr	Branch if accumulator A saturated	1	1 (2)	None
		BRA	SB,Expr	Branch if accumulator B saturated	1	1 (2)	None
		BRA	Expr	Branch Unconditionally	1	2	None
		BRA	Z,Expr	Branch if Zero	1	1 (2)	None
		BRA	Wn	Computed Branch	1	2	None
7	BSET	BSET	f,#bit4	Bit Set f	1	1	None
		BSET	Ws,#bit4	Bit Set Ws	1	1	None
8	BSW	BSW.C	Ws,Wb	Write C bit to Ws<Wb>	1	1	None
		BSW.Z	Ws,Wb	Write Z bit to Ws<Wb>	1	1	None
9	BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
		BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
10	BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None

20.2 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

20.3 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

20.4 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

20.5 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 21-5: DC CHARACTERISTICS: OPERATING CURRENT (IDD)

DC CHARACTERISTICS				Standard Operating Conditions: 3.3V and 5.0V (±10%) (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	Typical(1)	Max	Units	Conditions		
Operating Current (IDD)(2)						
DC20a	13	16	mA	+25°C	3.3V	FRC 3.2 MIPS, PLL disabled
DC20b	14	16	mA	+85°C		
DC20c	14	17	mA	+125°C		
DC20d	22	26	mA	+25°C	5V	
DC20e	22	26	mA	+85°C		
DC20f	22	27	mA	+125°C		
DC22a	19	22	mA	+25°C	3.3V	FRC, 4.9 MIPS, PLL disabled
DC22b	19	23	mA	+85°C		
DC22c	19	23	mA	+125°C		
DC22d	30	36	mA	+25°C	5V	
DC22e	30	37	mA	+85°C		
DC22f	31	37	mA	+125°C		
DC23a	27	33	mA	+25°C	3.3V	FRC, 7.3 MIPS, PLL disabled
DC23b	28	33	mA	+85°C		
DC23c	28	34	mA	+125°C		
DC23d	44	53	mA	+25°C	5V	
DC23e	45	53	mA	+85°C		
DC23f	45	54	mA	+125°C		
DC24a	66	79	mA	+25°C	3.3V	FRC 13 MIPS, PLL enabled
DC24b	67	80	mA	+85°C		
DC24c	68	81	mA	+125°C		
DC24d	108	129	mA	+25°C	5V	
DC24e	109	130	mA	+85°C		
DC24f	110	131	mA	+125°C		
DC26a	98	118	mA	+25°C	3.3V	FRC 20 MIPS, PLL enabled
DC26b	99	118	mA	+85°C		
DC26d	159	191	mA	+25°C	5V	
DC26e	160	192	mA	+85°C		
DC26f	161	193	mA	+125°C		
DC27d	222	267	mA	+25°C	5V	FRC, 30 MIPS, PLL enabled
DC27e	223	267	mA	+85°C		

Note 1: Data in “Typical” column is at 5V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature also have an impact on the current consumption. The test conditions for all IDD measurements are as follows:
- All I/O pins are configured as Outputs and pulled to VSS.
- MCLR = VDD, WDT and FSCM are disabled.
- CPU, SRAM, Program Memory and Data Memory are operational.
- No peripheral modules are operating.

FIGURE 21-18: I²C™ BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)

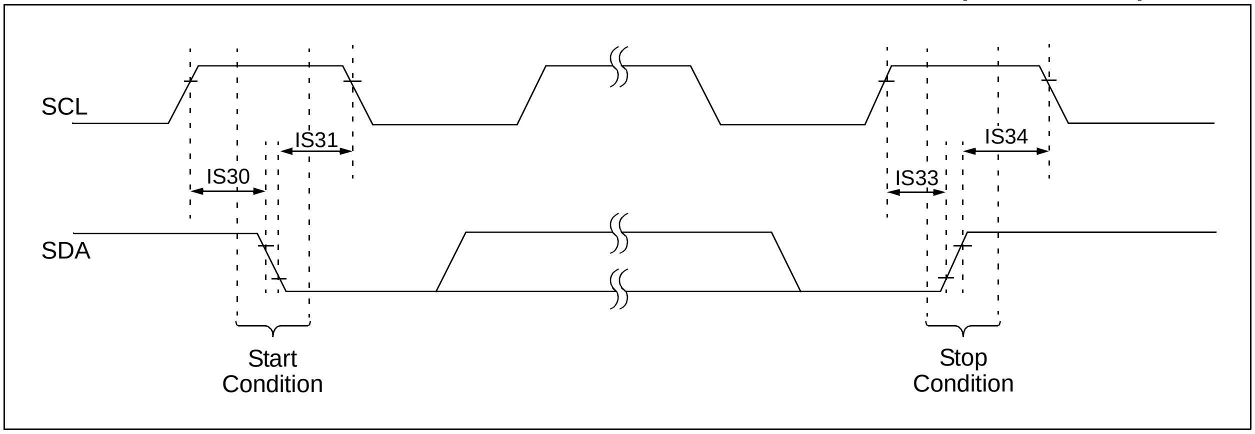


FIGURE 21-19: I²C™ BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)