



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 17x10b; D/A 1x5b, 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1716-e-ss

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Banl	« 4										
20Ch	WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	1111 1111	11 1111
20Dh	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	1111
20Eh	WPUC	WPUC7	WPUC6	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	1111 1111	1111 1111
20Fh	_	Unimplement	ted							_	_
210h	WPUE	_	_	_	—	WPUE3	—	—	—	1	1
211h	SSP1BUF	Synchronous	Serial Port Re	eceive Buffer/1	ransmit Regis	ster		•		XXXX XXXX	uuuu uuuu
212h	SSP1ADD		ADD<7:0>								0000 0000
213h	SSP1MSK				MSK	<7:0>				XXXX XXXX	1111 1111
214h	SSP1STAT	SMP	CKE	D/Ā	Р	S	R/W	UA	BF	0000 0000	0000 0000
215h	SSP1CON1	WCOL	SSPOV	SSPEN	CKP		SSPI	M<3:0>		0000 0000	0000 0000
216h	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
217h	SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000
218h 21Fh	_	Unimplemented							-	_	
Banl	< 5										
28Ch	ODCONA	ODA7	ODA6	ODA5	ODA4	ODA3	ODA2	ODA1	ODA0	0000 0000	00 -000
28Dh	ODCONB	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000 000-	0000
28Eh	ODCONC	ODC7	ODC6	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	0000 0000	0000 0000
28Fh	_	Unimplemented								_	_
290h	_	Unimplemented							_	_	
291h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)							XXXX XXXX	uuuu uuuu	
292h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								XXXX XXXX	uuuu uuuu
293h	CCP1CON	—	_	DC1B	<1:0>		CCP1	M<3:0>		00 0000	00 0000
294h 297h	_	Unimplemented							_	_	
298h	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								XXXX XXXX	uuuu uuuu
299h	CCPR2H	Capture/Compare/PWM Register 2 (MSB)							XXXX XXXX	uuuu uuuu	
29Ah	CCP2CON	— — DC2B<1:0> ССР2М<3:0>						00 0000	00 0000		
29Bh 29Dh	_	Unimplemented						-	_		
29Eh	CCPTMRS	P4TSEL<1:0> P3TSEL<1:0> C2TSEL<1:0> C1TSEL<1:0>						0000 0000	0000 0000		
29Fh	—	Unimplemented						—	—		
Banl	x 6										
30Ch	SLRCONA	SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0	1111 1111	00 -000
30Dh	SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0	1111 1111	0000
30Eh	SLRCONC	SLRC7	SLRC6	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	1111 1111	0000 0000
30Fh 31Fh	hUnimplemented							_	_		

TABLE 3-11: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Legend:x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.Shaded locations are unimplemented, read as '0'.

Note 1: Unimplemented, read as '1'.

2: Unimplemented on PIC16(L)F1713/6.

6.2 Clock Source Types

Clock sources can be classified as external or internal.

External clock sources rely on external circuitry for the clock source to function. Examples are: oscillator modules (ECH, ECM, ECL mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes) and Resistor-Capacitor (EXTRC) mode circuits.

Internal clock sources are contained within the oscillator module. The internal oscillator block has two internal oscillators and a dedicated Phase-Lock Loop (HFPLL) that are used to generate three internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC), 500 kHz (MFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bits in the OSCCON register. See **Section 6.3 "Clock Switching"** for additional information.

6.2.1 EXTERNAL CLOCK SOURCES

An external clock source can be used as the device system clock by performing one of the following actions:

- Program the FOSC<2:0> bits in the Configuration Words to select an external clock source that will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to:
 - Secondary oscillator during run-time, or
 - An external clock source determined by the value of the FOSC bits.

See **Section 6.3 "Clock Switching**" for more information.

6.2.1.1 EC Mode

The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. Figure 6-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through Configuration Words:

- ECH High power, 4-32 MHz
- ECM Medium power, 0.5-4 MHz
- ECL Low power, 0-0.5 MHz

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC[®] MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.



EXTERNAL CLOCK (EC) MODE OPERATION



6.2.1.2 LP, XT, HS Modes

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 6-3). The three modes select a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

XT Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

HS Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting.

Figure 6-3 and Figure 6-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- · GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 or PIE2 registers)

The INTCON, PIR1 and PIR2 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- · GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See "Section 7.5 "Automatic Context Saving")
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The RETFIE instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual	inte	rrupt	flag	bits	s are	set,
	regardless	of	the	state	of	any	other
	enable bits						

2: All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

TABLE 9-2: WDT CLEARING CONDITIONS

Conditions	WDT	
WDTE<1:0> = 00		
WDTE<1:0> = 01 and SWDTEN = 0		
WDTE<1:0> = 10 and enter Sleep	Cleared	
CLRWDT Command	Cleared	
Oscillator Fail Detected		
Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK		
Exit Sleep + System Clock = XT, HS, LP	Cleared until the end of OST	
Change INTOSC divider (IRCF bits)	Unaffected	

13.6 Register Definitions: Interrupt-on-Change Control

REGISTER 13-1: IOCAP: INTERRUPT-ON-CHANGE PORTA POSITIVE EDGE REGISTER

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| IOCAP7 | IOCAP6 | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 |
| bit 7 | | | | | | | bit 0 |

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0

IOCAP<7:0>: Interrupt-on-Change PORTA Positive Edge Enable bits

1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 13-2: IOCAN: INTERRUPT-ON-CHANGE PORTA NEGATIVE EDGE REGISTER

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| IOCAN7 | IOCAN6 | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 |
| bit 7 | | | | | | | bit 0 |

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **IOCAN<7:0>:** Interrupt-on-Change PORTA Negative Edge Enable bits

- 1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
- 0 = Interrupt-on-Change disabled for the associated pin.

17.1.9 SETUP FOR PWM OPERATION USING PWMx PINS

The following steps should be taken when configuring the module for PWM operation using the PWMx pins:

- 1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
- 2. Clear the PWMxCON register.
- 3. Load the PR2 register with the PWM period value.
- 4. Load the PWMxDCH register and bits <7:6> of the PWMxDCL register with the PWM duty cycle value.
- 5. Configure and start Timer2:
 - Clear the TMR2IF interrupt flag bit of the PIR1 register. See Note below.
 - Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
 - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
- Enable PWM output pin and wait until Timer2 overflows, TMR2IF bit of the PIR1 register is set. See Note below.
- 7. Enable the PWMx pin output driver(s) by clearing the associated TRIS bit(s) and setting the desired pin PPS control bits.
- 8. Configure the PWM module by loading the PWMxCON register with the appropriate values.
 - Note 1: In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then move Step 8 to replace Step 4.
 - **2:** For operation with other peripherals only, disable PWMx pin outputs.

17.1.10 SETUP FOR PWM OPERATION TO OTHER DEVICE PERIPHERALS

The following steps should be taken when configuring the module for PWM operation to be used by other device peripherals:

- 1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
- 2. Clear the PWMxCON register.
- 3. Load the PR2 register with the PWM period value.
- 4. Load the PWMxDCH register and bits <7:6> of the PWMxDCL register with the PWM duty cycle value.
- 5. Configure and start Timer2:
 - Clear the TMR2IF interrupt flag bit of the PIR1 register. See Note below.
 - Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
 - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
- 6. Enable PWM output pin:
- Wait until Timer2 overflows, TMR2IF bit of the PIR1 register is set. See Note below.
- 7. Configure the PWM module by loading the PWMxCON register with the appropriate values.
- Note: In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

18.5.4 RISING EVENT DEAD-BAND

Rising event dead band delays the turn-on of the primary outputs from when complementary outputs are turned off. The rising event dead-band time starts when the rising_ event output goes true.

See Section 18.5.1, Asynchronous Delay Chain Dead-band Delay and Section 18.5.2, Synchronous Counter Dead-band Delay for more information on setting the rising edge dead-band time.

18.5.5 FALLING EVENT DEAD-BAND

Falling event dead band delays the turn-on of complementary outputs from when the primary outputs are turned off. The falling event dead-band time starts when the falling event output goes true.

See Section 18.5.1, Asynchronous Delay Chain Dead-band Delay and Section 18.5.2, Synchronous Counter Dead-band Delay for more information on setting the rising edge dead-band time.

18.5.6 DEAD-BAND OVERLAP

There are two cases of dead-band overlap:

- Rising-to-falling
- Falling-to-rising

18.5.6.1 Rising-to-Falling Overlap

In this case, the falling event occurs while the rising event dead-band counter is still counting. When this happens, the primary drives are suppressed and the dead-band extends by the falling event dead-band time. At the termination of the extended dead-band time, the complementary drive goes true.

18.5.6.2 Falling-to-Rising Overlap

In this case, the rising event occurs while the falling event dead-band counter is still counting. When this happens, the complementary drive is suppressed and the dead-band extends by the rising event dead-band time. At the termination of the extended dead-band time, the primary drive goes true.

18.6 Blanking Control

Input blanking is a function, whereby, the event inputs can be masked or blanked for a short period of time. This is to prevent electrical transients caused by the turn-on/off of power components from generating a false input event.

The COG contains two blanking counters: one triggered by the rising event and the other triggered by the falling event. The counters are cross coupled with the events they are blanking. The falling event blanking counter is used to blank rising input events and the rising event blanking counter is used to blank falling input events. Once started, blanking extends for the time specified by the corresponding blanking counter. Blanking is timed by counting COG_clock periods from zero up to the value in the blanking count register. Use Equation 18-1 to calculate blanking times.

18.6.1 FALLING EVENT BLANKING OF RISING EVENT INPUTS

The falling event blanking counter inhibits rising event inputs from triggering a rising event. The falling event blanking time starts when the rising event output drive goes false.

The falling event blanking time is set by the value contained in the COGxBLKF register (Register 18-13). Blanking times are calculated using the formula shown in Equation 18-1.

When the COGxBLKF value is zero, falling event blanking is disabled and the blanking counter output is true, thereby, allowing the event signal to pass straight through to the event trigger circuit.

18.6.2 RISING EVENT BLANKING OF FALLING EVENT INPUTS

The rising event blanking counter inhibits falling event inputs from triggering a falling event. The rising event blanking time starts when the falling event output drive goes false.

The rising event blanking time is set by the value contained in the COGxBLKR register (Register 18-12).

When the COGxBLKR value is zero, rising event blanking is disabled and the blanking counter output is true, thereby, allowing the event signal to pass straight through to the event trigger circuit.

18.6.3 BLANKING TIME UNCERTAINTY

When the rising and falling sources that trigger the blanking counters are asynchronous to the COG_clock, it creates uncertainty in the blanking time. The maximum uncertainty is equal to one COG_clock period. Refer to Equation 18-1 and Example 18-1 for more detail.

18.7 Phase Delay

It is possible to delay the assertion of either or both the rising event and falling events. This is accomplished by placing a non-zero value in COGxPHR or COGxPHF phase-delay count register, respectively (Register 18-14 and Register 18-15). Refer to Figure 18-10 for COG operation with CCP1 and phase delay. The delay from the input rising event signal switching to the actual assertion of the events is calculated the same as the dead-band and blanking delays. Refer to Equation 18-1.

When the phase-delay count value is zero, phase delay is disabled and the phase-delay counter output is true, thereby, allowing the event signal to pass straight through to the complementary output driver flop.

^{© 2013-2016} Microchip Technology Inc.

REGISTER 18-4: COGxRSIM: COG RISING EVENT SOURCE INPUT MODE REGISTER

bit 0

GxRSIM0: COGx Rising Event Input Source 0 Mode bit

<u>GxRIS0 = 1:</u>

- 1 = Pin selected with COGxPPS control low-to-high transition will cause a rising event after rising event phase delay
- 0 = Pin selected with COGxPPS control high level will cause an immediate rising event

<u>GxRIS0 = 0:</u>

Pin selected with COGxPPS control has no effect on rising event

REGISTER 20-3: NCOxACCL: NCOx ACCUMULATOR REGISTER – LOW BYTE

bit 7	R/W-0/0							
	bit 7			NCOAN	00~1.02			bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 NCOxACC<7:0>: NCOx Accumulator, Low Byte

REGISTER 20-4: NCOxACCH: NCOx ACCUMULATOR REGISTER – HIGH BYTE

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| | | | NCOxAC | C<15:8> | | | |
| bit 7 | | | | | | | bit 0 |
| | | | | | | | |
| Legend: | | | | | | | |

Legenu.		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 NCOxACC<15:8>: NCOx Accumulator, High Byte

REGISTER 20-5: NCOxACCU: NCOx ACCUMULATOR REGISTER – UPPER BYTE

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0		
—	—	—	—	NCOxACC<19:16>					
bit 7							bit 0		

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 Unimplemented: Read as '0'

bit 3-0 NCOxACC<19:16>: NCOx Accumulator, Upper Byte

REGISTER 21-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
			ADRE	S<9:2>			
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit		it	U = Unimpler	nented bit, read	d as '0'		
u = Bit is unchanged x = Bit is unknow		own	-n/n = Value a	at POR and BC	R/Value at all o	other Resets	
'1' = Bit is set '0' = Bit is cleared			red				

bit 7-0 **ADRES<9:2>**: ADC Result Register bits Upper eight bits of 10-bit conversion result

REGISTER 21-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| ADRES | S<1:0> | — | — | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 ADRES<1:0>: ADC Result Register bits Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

24.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC2) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- · Comparator positive input
- · ADC input channel
- DAC2OUT1/DAC2OUT2 pin
- Comparators
- Op Amps

EQUATION 24-1: DAC OUTPUT VOLTAGE

<u>IF DACEN = 1</u>

 $VOUT = \left((VSOURCE + -VSOURCE -) \times \frac{DACR[4:0]}{2^5} \right) + VSOURCE -$

IF DACEN = 0 and DACLPS = 1 and DACR[4:0] = 11111

VOUT = VSOURCE +

IF DACEN = 0 and DACLPS = 0 and DACR[4:0] = 00000

VOUT = VSOURCE -

VSOURCE+ = VDD, VREF, or FVR BUFFER 2

VSOURCE- = VSS

24.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in Table 34-20.

The Digital-to-Analog Converter (DAC) can be enabled by setting the DACEN bit of the DACCON0 register.

24.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the DACR<4:0> bits of the DACCON1 register.

The DAC output voltage is determined by the following equations:

24.3 DAC Voltage Reference Output

The DAC can be output to the DACOUT pin by setting the DACOE bit of the DACCON0 register to '1'. Selecting the DAC reference voltage for output on the DACOUT pin automatically overrides the digital output buffer and digital input threshold detector functions of that pin. Reading the DACOUT pin when it has been configured for DAC reference voltage output will always return a '0'.

Due to the limited current drive capability, a buffer must be used on the DAC voltage reference output for external connections to DACOUT. Figure 24-2 shows an example buffering technique.





FIGURE 24-2:

VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page	
CCP1CON	—	-	DC1B	3<1:0> CCP1M<3:0>						
CCPR1L	Capture/Corr	pare/PWM Re	egister 1 (LSB	er 1 (LSB)						
CCPTMRS	P4TSE	L<1:0>	P3TSE	L<1:0>	1:0> C2TSEL<1:0>			C1TSEL<1:0>		
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	83	
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	84	
PIE2	OSFIE	C2IE	C1IE	_	BCL1IE	TMR6IE	TMR4IE	CCP2IE	85	
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	87	
PIR2	OSFIF	C2IF	C1IF	_	BCL1IF	TMR6IF	TMR4IF	CCP2IF	88	
PR2	Timer2 Perio	d Register								
ANSELB	—		ANSB5	ANSB4 ANSB3 ANSB2 ANSB1 ANSB0					126	
ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ISC4 ANSC3 ANSC2 — —				131	
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125	
TRISC	TRISC7	TRISC6	TRISC5	TRISC4 TRISC3 TRISC2 TRISC1 TRISC0						
RxyPPS	—		_	RxyPPS<4:0>						
CCP1PPS	—	-	—	CCP1PPS<4:0>						
CCP2PPS	—	_	—	CCP2PPS<4:0>						
T2CON	—		T2OUT	PS<3:0>		TMR2ON	T2CKP	'S<1:0>	270	
TMR2	Timer2 Modu	ule Register							268	

TABLE 29-3. SUMIMART OF REGISTERS ASSOCIATED WITH CO	TABLE 29-3:	SUMMARY OF REGISTERS ASSOCIATED WITH CCP
--	-------------	--

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the CCP. * Page provides register information.



© 2013-2016 Microchip Technology Inc.

30.5.8 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I²C protocol, defined as address 0x00. When the GCEN bit of the SSPCON<u>2</u> register is set, the slave module will automatically ACK the reception of this address regardless of the value stored in SSPADD. After the slave clocks in an address of all zeros with the R/W bit clear, an interrupt is generated and slave

software can read SSPBUF and respond. Figure 30-24 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 30-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



30.5.9 SSP MASK REGISTER

An SSP Mask (SSPMSK) register (Register 30-5) is available in I²C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

30.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPSTAT register is set when the CPU writes to SSPBUF and is cleared when all eight bits are shifted out.

30.6.6.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

WCOL must be cleared by software before the next transmission.

30.6.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit of the SSPCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK} = 0$) and is set when the slave does not Acknowledge ($\overline{ACK} = 1$). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

30.6.6.4 Typical transmit sequence:

- 1. The user generates a Start condition by setting the SEN bit of the SSPCON2 register.
- 2. SSPIF is set by hardware on completion of the Start.
- 3. SSPIF is cleared by software.
- 4. The MSSP module will wait the required start time before any other operation takes place.
- 5. The user loads the SSPBUF with the slave address to transmit.
- 6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPBUF is written to.
- 7. The MSSP module shifts in the ACK bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
- 8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- 9. The user loads the SSPBUF with eight bits of data.
- 10. Data is shifted out the SDA pin until all eight bits are transmitted.
- 11. The MSSP module shifts in the ACK bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
- 12. Steps 8-11 are repeated for all transmitted data bytes.
- 13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPCON2 register. Interrupt is generated once the Stop/Restart condition is complete.

FIGURE 31-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION

CRECT SC BOOM	la jacjosjak AUNUNUN Bitsetepi		atatata Nununun			ujoke ju Urugun				katesiokia NUNUNUN Aksio	: \$22\$23\$294 _ALALAL_ šewieś
80003 Line 8039 8039							: : : ::::::::::::::::::::::::::::::::		111 - 12 20 - 12 - 12 - 12 11 - 11 - 12 - 12 - 12 -		- - - - - - - - - - - - - - - - - - -
FIGURE	31-8:	AUTO-\	WAKE-UI	P BIT (V	VUE) TI	MINGS [DURING S	SLEEP			



(i) The SUSSARY remains in His while the Vicie hit is ad

TABLE 34-24: SPI MODE REQUIREMENTS

Standar	Standard Operating Conditions (unless otherwise stated)									
Param No.	Symbol	Characteristic	Min.	Тур†	Max.	Units	Conditions			
SP70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK \downarrow or SCK \uparrow input	2.25 TCY	_	—	ns				
SP71*	TscH	SCK input high time (Slave mode)	Tcy + 20	—	_	ns				
SP72*	TscL	SCK input low time (Slave mode)	TCY + 20	—	—	ns				
SP73*	TDIV2scH, TDIV2scL	Setup time of SDI data input to SCK edge	100	_	—	ns				
SP74*	TscH2DIL, TscL2DIL	Hold time of SDI data input to SCK edge	100	_	_	ns				
SP75*	TDOR	SDO data output rise time	—	10	25	ns	$3.0V \le V\text{DD} \le 5.5V$			
			—	25	50	ns	$1.8V \leq V\text{DD} \leq 5.5V$			
SP76*	TDOF	SDO data output fall time	—	10	25	ns				
SP77*	TssH2doZ	\overline{SS} to SDO output high-impedance	10	—	50	ns				
SP78*	TscR	SCK output rise time	—	10	25	ns	$3.0V \le V\text{DD} \le 5.5V$			
		(Master mode)	—	25	50	ns	$1.8V \leq V\text{DD} \leq 5.5V$			
SP79*	TscF	SCK output fall time (Master mode)	—	10	25	ns				
SP80*	TscH2doV,	SDO data output valid after SCK	—	_	50	ns	$3.0V \le V\text{DD} \le 5.5V$			
	TscL2DoV	edge	—	_	145	ns	$1.8V \leq V\text{DD} \leq 5.5V$			
SP81*	TDOV2scH, TDOV2scL	SDO data output setup to SCK edge	1 Tcy	—	_	ns				
SP82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge	—	—	50	ns				
SP83*	TscH2ssH, TscL2ssH	SS ↑ after SCK edge	1.5 Tcy + 40	_		ns				

These parameters are characterized but not tested. *

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note: Unless otherwise noted, VIN = 5V, Fosc = 300 kHz, CIN = 0.1 μ F, TA = 25°C.



FIGURE 35-25: IDD Typical, HFINTOSC Mode, PIC16F1713/6 Only.



FIGURE 35-26: IDD Maximum, HFINTOSC Mode, PIC16F1713/6 Only.



FIGURE 35-27: IDD Typical, HS Oscillator, 25°C, PIC16LF1713/6 Only.



FIGURE 35-28: IDD Maximum, HS Oscillator, PIC16LF1713/6 Only.



FIGURE 35-29: IDD Typical, HS Oscillator, 25°C, PIC16F1713/6 Only.



FIGURE 35-30: IDD Maximum, HS Oscillator, PIC16F1713/6 Only.

Note: Unless otherwise noted, VIN = 5V, Fosc = 300 kHz, CIN = 0.1 μ F, TA = 25°C.



FIGURE 35-79: ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1μ S, 25° C.



FIGURE 35-80: ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 4μ S, 25° C.



FIGURE 35-81: ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, VREF = 3.0V.







Single-Ended DNL, VDD = 3.0V, $TAD = 1 \ \mu$ S.



Single-Ended INL, VDD = 3.0V, $TaD = 1 \ \mu S$.