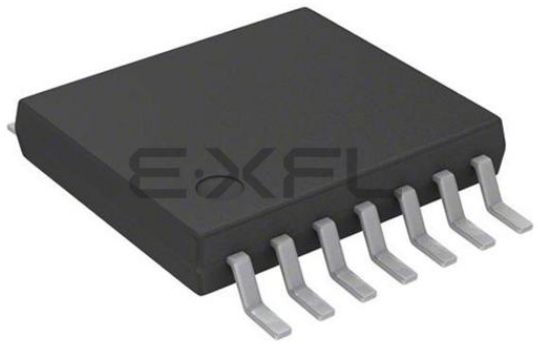


Welcome to E-XFL.COM

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	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 11x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	14-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	14-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18426t-i-st

1.4.2.1 Short Bit Names

Short bit names are an abbreviation for the bit function. For example, some peripherals are enabled with the EN bit. The bit names shown in the registers are the short name variant.

Short bit names are useful when accessing bits in C programs. The general format for accessing bits by the short name is RegisterNamebits.ShortName. For example, the enable bit, EN, in the CM1CON0 register can be set in C programs with the instruction `CM1CON0bits.EN = 1`.

Short names are generally not useful in assembly programs because the same name may be used by different peripherals in different bit positions. When this occurs, during the include file generation, all instances of that short bit name are appended with an underscore plus the name of the register in which the bit resides to avoid naming contentions.

1.4.2.2 Long Bit Names

Long bit names are constructed by adding a peripheral abbreviation prefix to the short name. The prefix is unique to the peripheral thereby making every long bit name unique. The long bit name for the COG1 enable bit is the COG1 prefix, G1, appended with the enable bit short name, EN, resulting in the unique bit name G1EN.

Long bit names are useful in both C and assembly programs. For example, in C the COG1CON0 enable bit can be set with the `G1EN = 1` instruction. In assembly, this bit can be set with the `BSF COG1CON0,G1EN` instruction.

1.4.2.3 Bit Fields

Bit fields are two or more adjacent bits in the same register. Bit fields adhere only to the short bit naming convention. For example, the three Least Significant bits of the COG1CON0 register contain the mode control bits. The short name for this field is MD. There is no long bit name variant. Bit field access is only possible in C programs. The following example demonstrates a C program instruction for setting the COG1 to the Push-Pull mode:

COG1CON0bits.MD = 0x5;

Individual bits in a bit field can also be accessed with long and short bit names. Each bit is the field name appended with the number of the bit position within the field. For example, the Most Significant mode bit has the short bit name MD2 and the long bit name is G1MD2. The following two examples demonstrate assembly program sequences for setting the COG1 to Push-Pull mode:

Example 1:

```
MOVLW  ~(1<<G1MD1)
ANDWF  COG1CON0,F
MOVLW  1<<G1MD2 | 1<<G1MD0
IORWF  COG1CON0,F
```

Example 2:

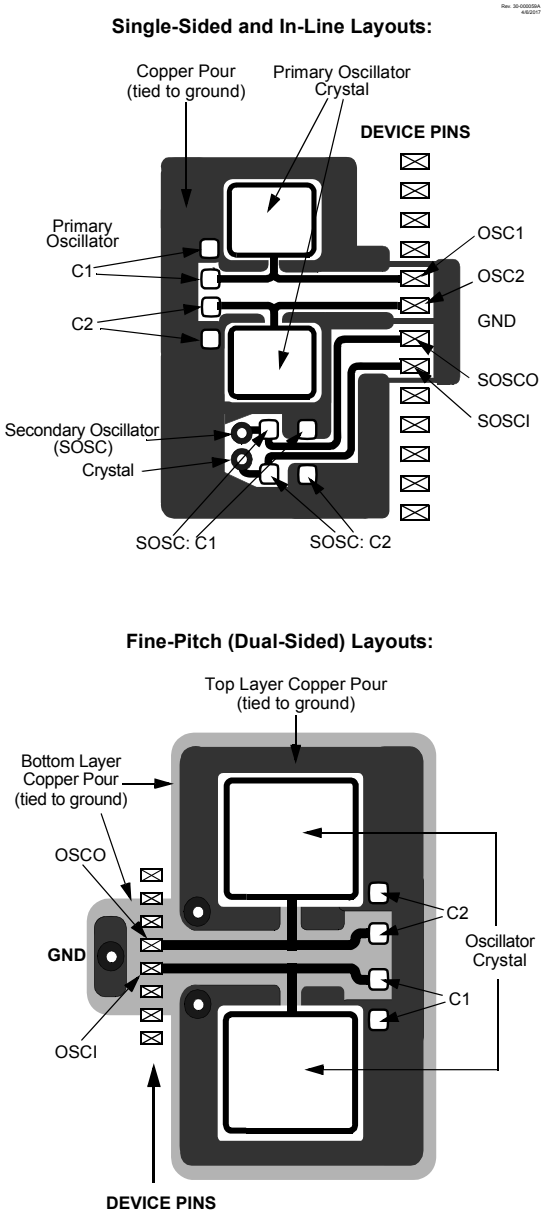
```
BSF    COG1CON0,G1MD2
BCF    COG1CON0,G1MD1
BSF    COG1CON0,G1MD0
```

1.4.3 Register and Bit Naming Exceptions

1.4.3.1 Status, Interrupt, and Mirror Bits

Status, interrupt enables, interrupt flags, and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

Figure 2-3. Suggested Placement of the Oscillator Circuit



In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

For additional information and design guidance on oscillator circuits, refer to these Microchip Application Notes, available at the corporate website (www.microchip.com):

- AN826, *"Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- AN849, *"Basic PICmicro® Oscillator Design"*
- AN943, *"Practical PICmicro® Oscillator Analysis and Design"*
- AN949, *"Making Your Oscillator Work"*

Related Links

Oscillator Module (with Fail-Safe Clock Monitor)

7.8.7 BSR

Name: BSR
Address: 0x08 + n*0x80 [n=0..63]

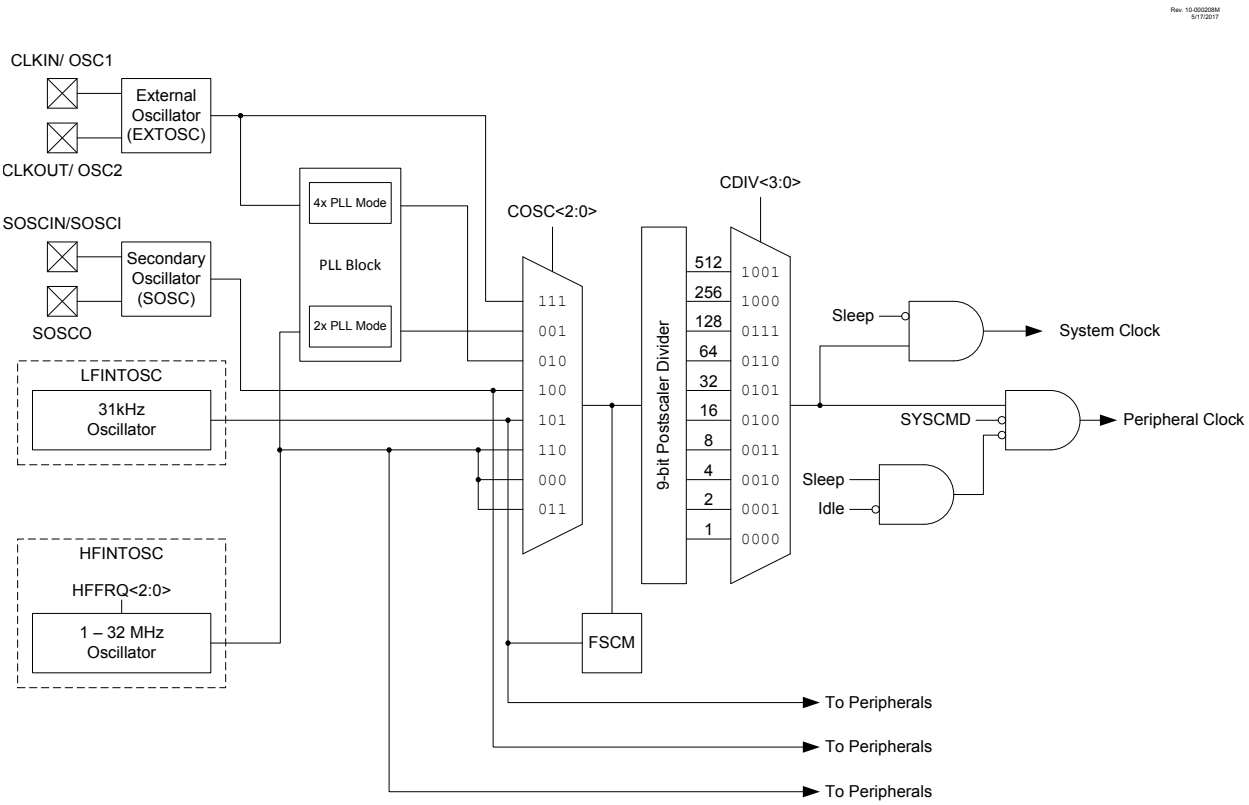
Bank Select Register
The BSR indicates the data memory bank by writing the bank number into the register. All data memory can be accessed directly via instructions, or indirectly via FSRs.

Bit	7	6	5	4	3	2	1	0
			BSR[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – BSR[5:0]
Six most significant bits of the data memory address

Related Links
Core Registers

Figure 9-1. Simplified PIC® MCU Clock Source Block Diagram



Related Links

CONFIG1

9.2 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).

There is also a secondary oscillator block which is optimized for a 32.768 kHz external clock source, which can be used as an alternate clock source.

There are two internal oscillator blocks:

- HFINTOSC
- LFINTOSC

The HFINTOSC can produce clock frequencies from 1-32 MHz, and is responsible for generating the two MFINTOSC frequencies (500 kHz and 32 kHz) that can be used by some peripherals. The LFINTOSC generates a 31 kHz clock frequency.

There is a 4x PLL that can be used by the external oscillator. Additionally, there is a PLL that can be used by the HFINTOSC at certain frequencies.

Related Links

4x PLL

Related Links

OSCSTAT

OSCEN

9.2.2.8 HFOR and MFOR Bits

The HFOR and MFOR bits indicate that the HFINTOSC and MFINTOSC is ready. These clocks are always valid for use at all times, but only accurate after they are ready.

When a new value is loaded into the OSCFRQ register, the HFOR and MFOR bits will clear, and set again when the oscillator is ready. During pending OSCFRQ changes the MFINTOSC clock will stall at a high or a low state, until the HFINTOSC resumes operation.

9.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the New Oscillator Source (NOSC) and New Divider selection request (NDIV) bits. The following clock sources can be selected:

- External Oscillator (EXTOSC)
- High-Frequency Internal Oscillator (HFINTOSC)
- Low-Frequency Internal Oscillator (LFINTOSC)
- Secondary Oscillator (SOSC)
- EXTOSC with 4x PLL
- HFINTOSC with 2x PLL

9.3.1 New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) Bits

The New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) bits select the system clock source and frequency that are used for the CPU and peripherals.

When new values of NOSC and NDIV are written to OSCCON1, the current oscillator selection will continue to operate while waiting for the new clock source to indicate that it is stable and ready. In some cases, the newly requested source may already be in use, and is ready immediately. In the case of a divider-only change, the new and old sources are the same, so the old source will be ready immediately. The device may enter Sleep while waiting for the switch.

When the new oscillator is ready, the New Oscillator Ready (NOSCR) bit is set and also the Clock Switch Interrupt Flag (CSWIF) bit of PIR1 sets. If Clock Switch Interrupts are enabled (CSWIE = 1), an interrupt will be generated at that time. The Oscillator Ready (ORDY) bit of OSCCON3 can also be polled to determine when the oscillator is ready in lieu of an interrupt.

If the Clock Switch Hold (CSWHOLD) bit is clear, the oscillator switch will occur when the New Oscillator is READY bit (NOSCR) is set, and the interrupt (if enabled) will be serviced at the new oscillator setting.

If CSWHOLD is set, the oscillator switch is suspended, while execution continues using the current (old) clock source. When the NOSCR bit is set, software should:

- Set CSWHOLD = 0 so the switch can complete, or
- Copy COSC into NOSC to abandon the switch.

If DOZE is in effect, the switch occurs on the next clock cycle, whether or not the CPU is operating during that cycle.

Changing the clock post-divider without changing the clock source (i.e., changing F_{OSC} from 1 MHz to 2 MHz) is handled in the same manner as a clock source change, as described previously. The clock

CDIV/NDIV	Clock Divider
0001	2
0000	1

Note:

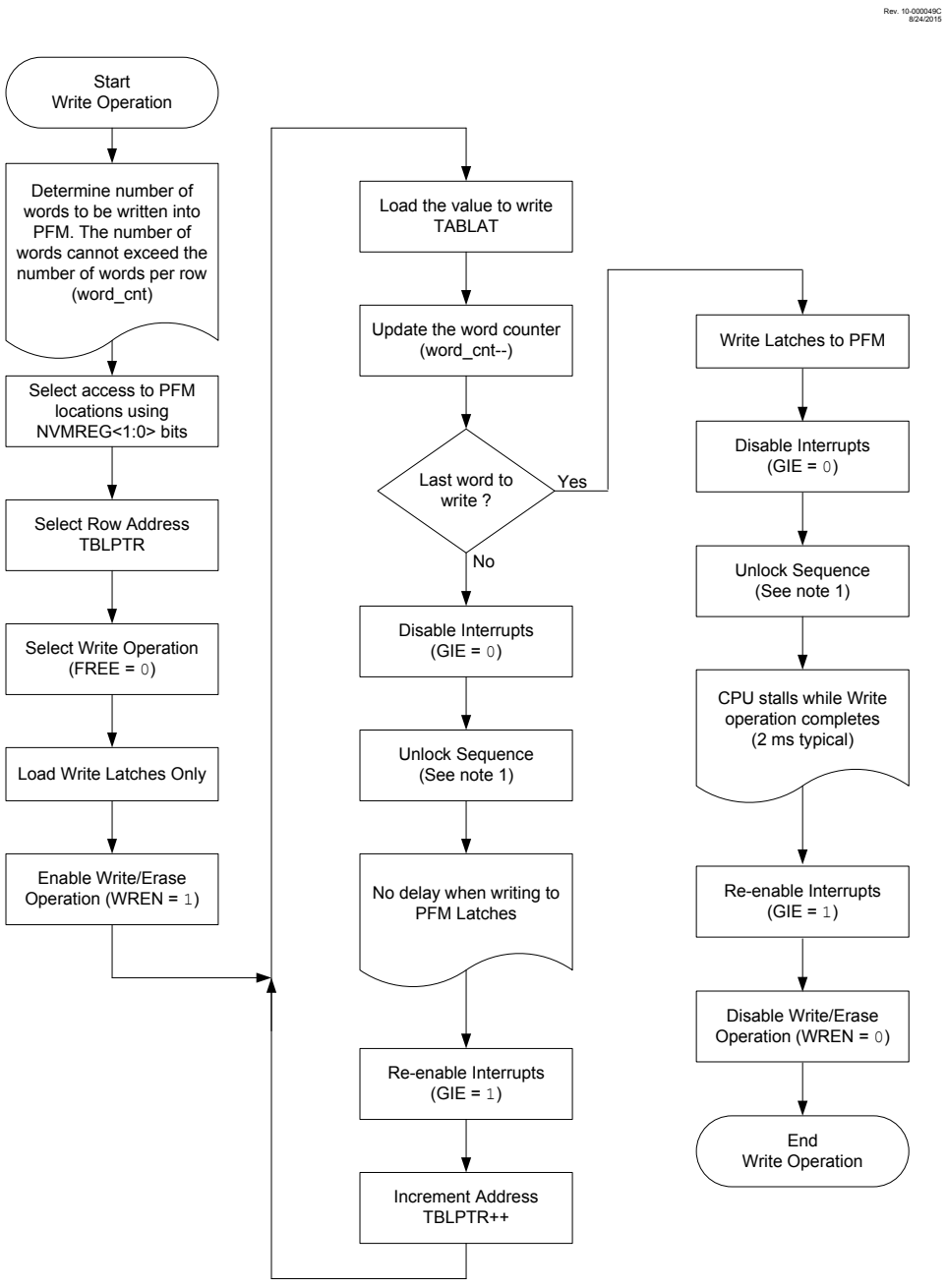
1. The POR value is the value present when user code execution begins.
2. The Reset value (n) is the same as the OSCCON1[NOSC/NDIV] bits.
3. EXTOSC configured by the CONFIG1[FEXTOSC] bits.
4. HFINTOSC frequency is configured with the FRQ bits of the OSCFRQ register

Related Links

CONFIG1

PLL Specifications

Figure 13-5. Program Flash Memory Flowchart



Note:

1. See NVM Unlock Sequence Flowchart

Writing to Program Flash Memory

```
   ; This write routine assumes the following:
   ; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR
   ; 2. Each word of data to be written is made up of two adjacent bytes in
DATA_ADDR,
   ; stored in little endian format
   ; 3. A valid starting address (the least significant bits = 00000) is
```

14.6.14 WPUB

Name: WPUB
Address: 0x1F44

Weak Pull-up Register

Bit	7	6	5	4	3	2	1	0
	WPUB7	WPUB6	WPUB5	WPUB4				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bits 4, 5, 6, 7 – WPUBn Weak Pull-up PORTB Control bits

Value	Description
1	Weak Pull-up enabled
0	Weak Pull-up disabled

17.6.6 IOCBF

Name: IOCBF
Address: 0x1F4A

PORTB Interrupt-on-Change Flag Register

Bit	7	6	5	4	3	2	1	0
	IOCBF7	IOCBF6	IOCBF5	IOCBF4				
Access	R/W/HS	R/W/HS	R/W/HS	R/W/HS				
Reset	0	0	0	0				

Bits 4, 5, 6, 7 – IOCBFn Interrupt-on-Change Flag bits

Value	Condition	Description
1	IOCBP[n]=1	A positive edge was detected on the RB[n] pin
1	IOCBN[n]=1	A negative edge was detected on the RB[n] pin
0	IOCBP[n]=x and IOCBN[n]=x	No change was detected, or the user cleared the detected change

Note: PORTB associated registers are available on 20-pin or higher pin-count devices only.

If SOI = 1, a threshold interrupt condition will clear GO and the conversions will stop.

20.6.9 Double Sample Conversion

Double sampling is enabled by setting the DSEN bit. When this bit is set, two conversions are required before the module will calculate threshold error (each conversion must still be triggered separately). The first conversion will set the MATH bit and update ADACC, but will not calculate ADERR or trigger ADTIF. When the second conversion completes, the first value is transferred to ADPREV (depending on the setting of PSIS) and the value of the second conversion is placed into ADRES. Only upon the completion of the second conversion is ADERR calculated and ADTIF triggered (depending on the value of CALC).

20.8.15 ADRES

Name: ADRES
Address: 0x09D

ADC Result Register

Bit	15	14	13	12	11	10	9	8
	RESH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	RESL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 15:8 – RESH[7:0] ADC Result Register bits. High bits
Reset States: POR/BOR = xxxxxxxx
All Other Resets = uuuuuuuu

Value	Condition	Description
0x00 to 0x0F	FRM = 1	Upper 4 bits of result
0x00 to 0xFF	FRM = 0	Upper 8 bits of result

Bits 7:0 – RESL[7:0] ADC Result Register bits. Lower bits
Reset States: POR/BOR = xxxxxxxx
All Other Resets = uuuuuuuu

Value	Condition	Description
0x00 to 0xFF	FRM = 1	Lower 8 bits of result
0x00,0x10 to 0xF0	FRM = 0	Lower 4 bits of result

- TMRxON bit of the TxCON register
- TMRxIE bits of the PIEx register
- PEIE/GIEL bit of the INTCON register
- GIE/GIEH bit of the INTCON register

The interrupt is cleared by clearing the TMRxIF bit in the Interrupt Service Routine.

For more information on selecting high or low priority status for the Timer1 Overflow Interrupt, see the Interrupts chapter.

Important: The TMRxH:TMRxL register pair and the TMRxIF bit should be cleared before enabling interrupts.

26.9 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when set up in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMRxON bit of the TxCON register must be set
- TMRxIE bit of the PIEx register must be set
- PEIE/GIEL bit of the INTCON register must be set
- $\overline{\text{TxSYNC}}$ bit of the TxCON register must be set
- Configure the TMRxCLK register for using secondary oscillator as the clock source
- Enable the SOSCEN bit of the OSCEN register

The device will wake-up on an overflow and execute the next instruction. If the GIE/GIEH bit of the INTCON register is set, the device will call the Interrupt Service Routine.

The secondary oscillator will continue to operate in Sleep regardless of the $\overline{\text{TxSYNC}}$ bit setting.

26.10 CCP Capture/Compare Time Base

The CCP modules use the TMRxH:TMRxL register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMRxH:TMRxL register pair is copied into the CCPRxH:CCPRxL register pair on a configured event.

In Compare mode, an event is triggered when the value in the CCPRxH:CCPRxL register pair matches the value in the TMRxH:TMRxL register pair. This event can be a Special Event Trigger.

For more information, see Capture/Compare/PWM Module(CCP) chapter.

Related Links

Capture/Compare/PWM Module

- T2TMR is cleared
- The PWM output is active. (Exception: When the PWM duty cycle = 0%, the PWM output will remain inactive.)
- The PWMxDCH and PWMxDCL register values are latched into the buffers.

Important: The Timer2 postscaler has no effect on the PWM operation.

30.4 PWM Duty Cycle

The PWM duty cycle is specified by writing a 10-bit value to the PWMxDCH and PWMxDCL register pair. The PWMxDCH register contains the eight MSbs and the PWMxDCL<7:6>, the two LSbs. The PWMxDCH and PWMxDCL registers can be written to at any time.

The formulas below are used to calculate the PWM pulse width and the PWM duty cycle ratio.

Equation 30-2. Pulse Width

$$PulseWidth = (PWMxDCH{:}PWMxDCL<7{:}6>) \bullet Tosc \bullet (TMR2\mathrm{P}rescaleValue)$$

Note: $T_{OSC} = 1/F_{OSC}$

Equation 30-3. Duty Cycle Ratio

$$DutyCycleRatio = \frac{(PWMxDCH{:}PWMxDCL<7{:}6>)}{4(T2PR+1)}$$

The 8-bit timer T2TMR register is concatenated with the two Least Significant bits of $1/F_{OSC}$, adjusted by the Timer2 prescaler to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

30.5 PWM Resolution

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when T2PR is 255. The resolution is a function of the T2PR register value as shown below.

Equation 30-4. PWM Resolution

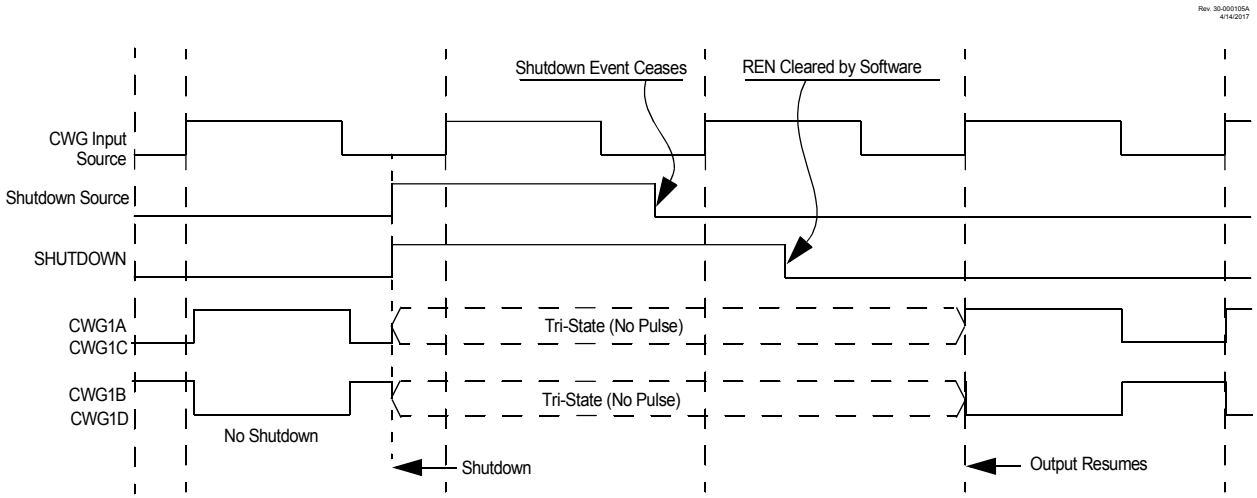
$$\mathrm{R}esolution = \frac{\log[4(T2PR+1)]}{\log(2)} bits$$

Important: If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

Once all auto-shutdown sources are removed, the software must clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.

Important: The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

Figure 31-17. SHUTDOWN FUNCTIONALITY, AUTO-RESTART DISABLED (REN = 0, LSAC = 01, LSBD = 01)



31.11.2.2 Auto-Restart

When the REN bit is set (REN = 1), the CWG module will restart from the shutdown state automatically.

Once all auto-shutdown conditions are removed, the hardware will automatically clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.

Important: The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

32.12.3 MDxCARH

Name: MDxCARH
Address: 0x089B

Modulation High Carrier Control Register

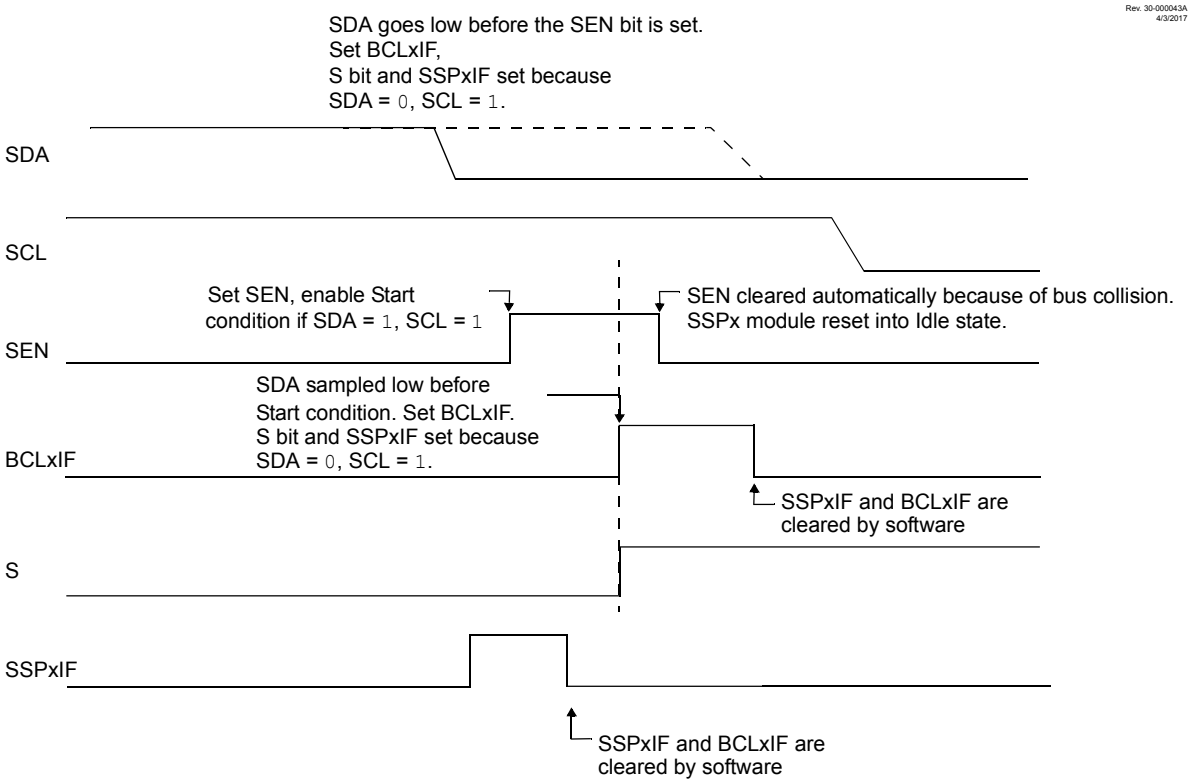
Bit	7	6	5	4	3	2	1	0
					CHS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – CHS[3:0] Modulator Carrier High Selection bits

Table 32-5. MDCARH Source Selections

MDCARH	
CHS<3:0>	**Connection**
1111	Reserved
1110	CLC4 OUT
1101	CLC3 OUT
1100	CLC2 OUT
1011	CLC1 OUT
1010	NCO1 OUT
1001	PWM7 OUT
1000	PWM6 OUT
0111	CCP4 OUT
0110	CCP3 OUT
0101	CCP2 OUT
0100	CCP1 OUT
0011	CLKREF output
0010	HFINTOSC
0001	F_{OSC} (system clock)
0000	Pin selected by MDCARHPPS

Figure 35-33. Bus Collision During Start Condition (SDA Only)



The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

Figure 35-34. Bus Collision During Start Condition (SCL = 0)

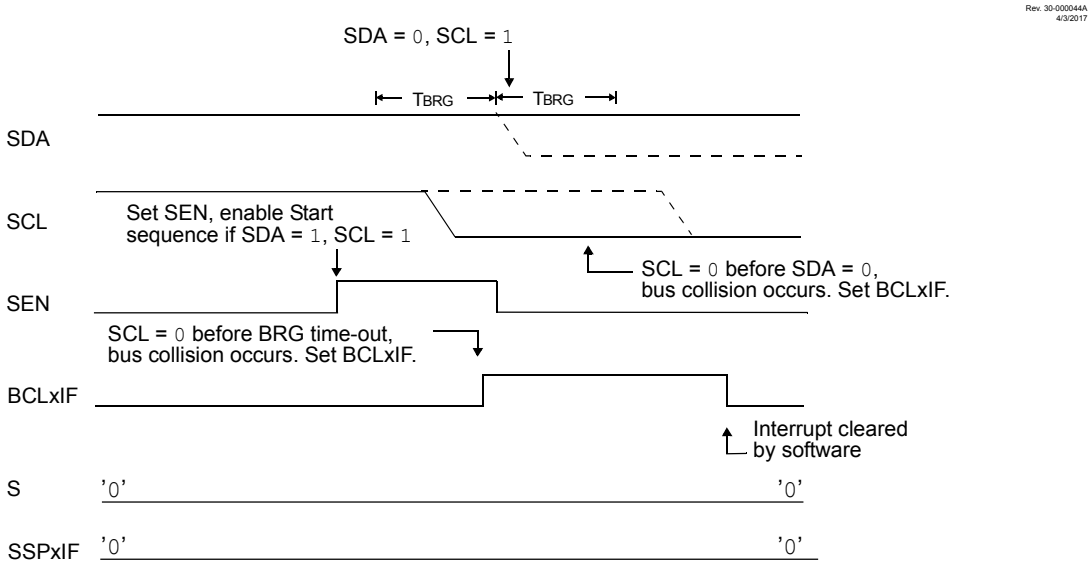
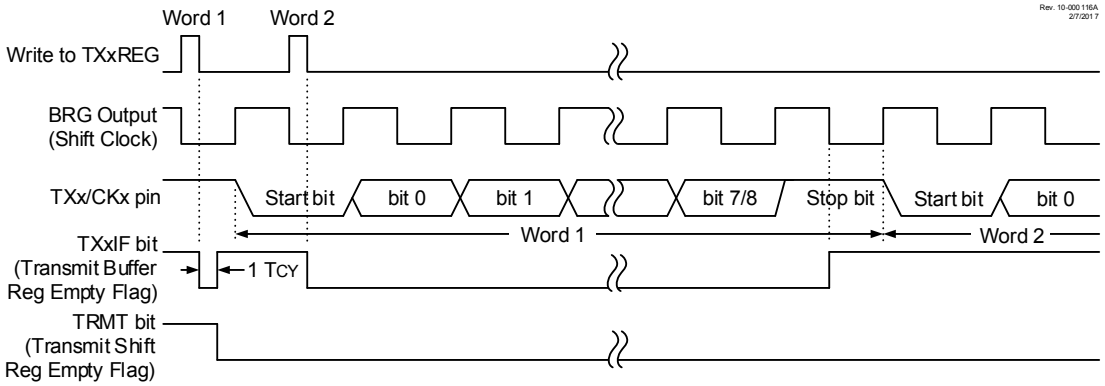


Figure 36-4. Asynchronous Transmission (Back-to-Back)



36.1.2 EUSART Asynchronous Receiver

The Asynchronous mode is typically used in RS-232 systems. A simplified representation of the receiver is shown in the Figure 36-2. The data is received on the RXx/DTx pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCxREG register.

36.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1 (enables the receiver circuitry of the EUSART)
- SYNC = 0 (configures the EUSART for asynchronous operation)
- SPEN = 1 (enables the EUSART)

All other EUSART control bits are assumed to be in their default state.

The user must set the RXxPPS register to select the RXx/DTx I/O pin and set the corresponding TRIS bit to configure the pin as an input.

Important: If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

36.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data

Offset	Name	Bit Pos.								
0x0F0C ... 0x0F7F	Reserved									
0x0F80	INDF0	7:0	INDF0[7:0]							
0x0F81	INDF1	7:0	INDF1[7:0]							
0x0F82	PCL	7:0	PCL[7:0]							
0x0F83	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x0F84	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0F86	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0F88	BSR	7:0			BSR[5:0]					
0x0F89	WREG	7:0	WREG[7:0]							
0x0F8A	PCLATH	7:0		PCLATH[6:0]						
0x0F8B	INTCON	7:0	GIE	PEIE						INTEDG
0x0F8C ... 0x0FEC	Reserved									
0x0FED	STKPTR	7:0				STKPTR[4:0]				
0x0FEE	TOS	7:0	TOSL[7:0]							
		15:8	TOSH[7:0]							
0x0FF0 ... 0x0FFF	Reserved									
0x1000	INDF0	7:0	INDF0[7:0]							
0x1001	INDF1	7:0	INDF1[7:0]							
0x1002	PCL	7:0	PCL[7:0]							
0x1003	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x1004	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1006	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1008	BSR	7:0			BSR[5:0]					
0x1009	WREG	7:0	WREG[7:0]							
0x100A	PCLATH	7:0		PCLATH[6:0]						
0x100B	INTCON	7:0	GIE	PEIE						INTEDG
0x100C ... 0x107F	Reserved									
0x1080	INDF0	7:0	INDF0[7:0]							
0x1081	INDF1	7:0	INDF1[7:0]							
0x1082	PCL	7:0	PCL[7:0]							
0x1083	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x1084	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1086	FSR1	7:0	FSRL[7:0]							

RETLW	Return literal to W			
Description:	The W register is loaded with the 8-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.			
Words:	1			
Cycles:	2			

Example:

```
      CALL TABLE            ; W contains table
 ; offset value
 ; W now has
 ; table value
    :
 TABLE
 ADDWF PC             ; W = offset
 RETLW k1             ; Begin table
 RETLW k2             ;
 :
 :
 RETLW kn             ; End of table
```

Before Instruction
W = 07h

After Instruction

W = value of k8

RETURN	Return from Subroutine			
Syntax:	[*label*] RETURN			
Operands:	None			
Operation:	(TOS) → PC,			
Status Affected:	None			
Encoding:	0000	0000	0001	001s
Description:	Return from subroutine. The stack is POPped and the top of the stack (TOS) is loaded into the Program Counter. This is a 2-cycle instruction.			

RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f, d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<n>) → dest<n + 1>, (f<7>) → C,

ISBN: 978-1-5224-2486-4

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.