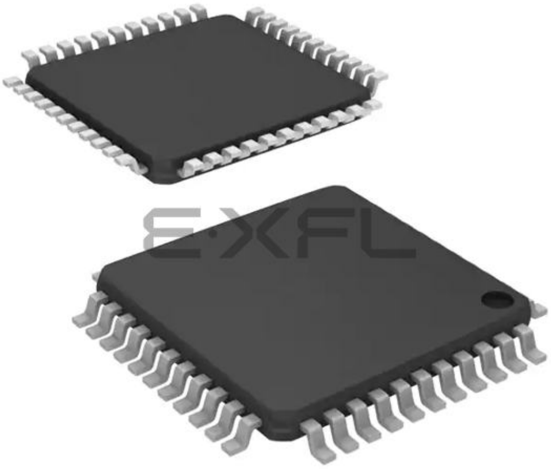
Microchip Technology - PIC18F4550-I/PT Datasheet



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	48MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4550-i-pt

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	PDIP, SOIC			
				PORTA is a bidirectional I/O port.
RA0/AN0	2			
RA0		I/O	TTL	Digital I/O.
AN0		I	Analog	Analog input 0.
RA1/AN1	3			
RA1		I/O	TTL	Digital I/O.
AN1		I	Analog	Analog input 1.
RA2/AN2/VREF-/CVREF	4			
RA2		I/O	TTL	Digital I/O.
AN2		I	Analog	Analog input 2.
VREF-		I	Analog	A/D reference voltage (low) input.
CVREF		O	Analog	Analog comparator reference output.
RA3/AN3/VREF+	5			
RA3		I/O	TTL	Digital I/O.
AN3		I	Analog	Analog input 3.
VREF+		I	Analog	A/D reference voltage (high) input.
RA4/T0CKI/C1OUT/RCV	6			
RA4		I/O	ST	Digital I/O.
T0CKI		I	ST	Timer0 external clock input.
C1OUT		O	—	Comparator 1 output.
RCV		I	TTL	External USB transceiver RCV input.
RA5/AN4/$\overline{\text{SS}}$/ HLVDIN/C2OUT	7			
RA5		I/O	TTL	Digital I/O.
AN4		I	Analog	Analog input 4.
$\overline{\text{SS}}$		I	TTL	SPI slave select input.
HLVDIN		I	Analog	High/Low-Voltage Detect input.
C2OUT		O	—	Comparator 2 output.
RA6	—	—	—	See the OSC2/CLKO/RA6 pin.

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

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

5.1.2.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration Register 4L. When STVREN is set, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit but not cause a device Reset. The STKFUL or STKUNF bits are cleared by user software or a Power-on Reset.

5.1.3 FAST REGISTER STACK

A Fast Register Stack is provided for the STATUS, WREG and BSR registers to provide a "fast return" option for interrupts. Each stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the stack registers. The values in the registers are then loaded back into their associated registers if the `RETFIE, FAST` instruction is used to return from the interrupt.

If both low and high-priority interrupts are enabled, the stack registers cannot be used reliably to return from low-priority interrupts. If a high-priority interrupt occurs while servicing a low-priority interrupt, the stack register values stored by the low-priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low-priority interrupt.

If interrupt priority is not used, all interrupts may use the Fast Register Stack for returns from interrupt. If no interrupts are used, the Fast Register Stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the Fast Register Stack for a subroutine call, a `CALL label, FAST` instruction must be executed to save the STATUS, WREG and BSR registers to the Fast Register Stack. A `RETURN, FAST` instruction is then executed to restore these registers from the Fast Register Stack.

Example 5-1 shows a source code example that uses the Fast Register Stack during a subroutine call and return.

EXAMPLE 5-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL  SUB1, FAST     ;STATUS, WREG, BSR
                     ;SAVED IN FAST REGISTER
                     ;STACK
      •
      •

SUB1  •
      •
      RETURN, FAST   ;RESTORE VALUES SAVED
                     ;IN FAST REGISTER STACK
```

5.1.4 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures, or look-up tables, in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed `GOTO`
- Table Reads

5.1.4.1 Computed `GOTO`

A computed `GOTO` is accomplished by adding an offset to the program counter. An example is shown in Example 5-2.

A look-up table can be formed with an `ADDWF PCL` instruction and a group of `RETLW nn` instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the `ADDWF PCL` instruction. The next instruction executed will be one of the `RETLW nn` instructions that returns the value '`nn`' to the calling function.

The offset value (in WREG) specifies the number of bytes that the program counter should advance and should be multiples of 2 (LSb = `0`).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

EXAMPLE 5-2: COMPUTED GOTO USING AN OFFSET VALUE

```
        MOVF    OFFSET, W
        CALL    TABLE
ORG     nn00h
TABLE   ADDWF   PCL
        RETLW   nnh
        RETLW   nnh
        RETLW   nnh
        .
        .
        .
```

5.1.4.2 Table Reads and Table Writes

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location.

Look-up table data may be stored two bytes per program word by using table reads and writes. The Table Pointer (TBLPTR) register specifies the byte address and the Table Latch (TABLAT) register contains the data that is read from or written to program memory. Data is transferred to or from program memory one byte at a time.

Table read and table write operations are discussed further in **Section 6.1 "Table Reads and Table Writes"**.

5.2 PIC18 Instruction Cycle

5.2.1 CLOCKING SCHEME

The microcontroller clock input, whether from an internal or external source, is internally divided by four to generate four non-overlapping quadrature clocks (Q1, Q2, Q3 and Q4). Internally, the program counter is incremented on every Q1; the instruction is fetched from the program memory and latched into the Instruction Register (IR) during Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 5-3.

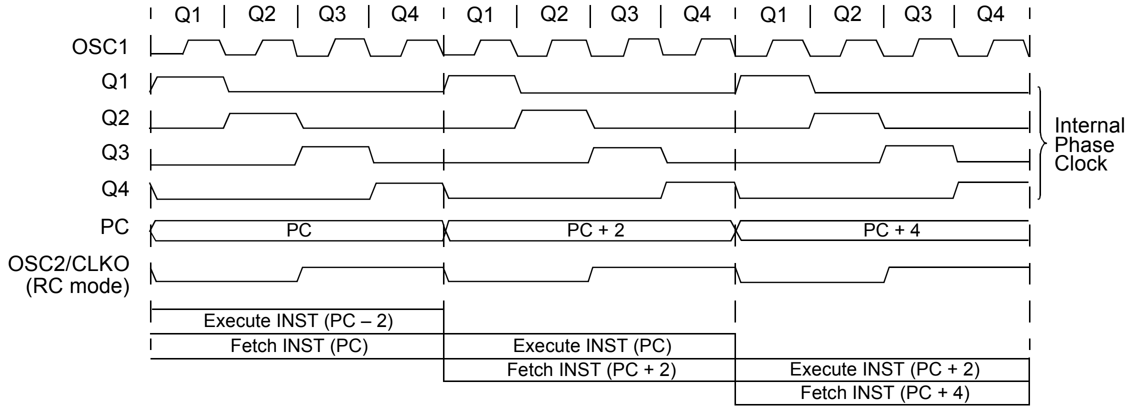
5.2.2 INSTRUCTION FLOW/PIPELINING

An “Instruction Cycle” consists of four Q cycles: Q1 through Q4. The instruction fetch and execute are pipelined in such a manner that a fetch takes one instruction cycle, while the decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., `GOTO`), then two cycles are required to complete the instruction (Example 5-3).

A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 5-3: CLOCK/INSTRUCTION CYCLE



EXAMPLE 5-3: INSTRUCTION PIPELINE FLOW

	TCY0	TCY1	TCY2	TCY3	TCY4	TCY5
`1. MOVLW 55h`	Fetch 1	Execute 1				
`2. MOVWF PORTB`		Fetch 2	Execute 2			
`3. BRA SUB_1`			Fetch 3	Execute 3		
`4. BSF PORTA, BIT3 (Forced NOP)`				Fetch 4	Flush (`NOP`)	
`5. Instruction @ address SUB_1`					Fetch SUB_1	Execute SUB_1

Note: All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

TABLE 10-1: PORTA I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RA0/AN0	RA0	0	OUT	DIG	LATA<0> data output; not affected by analog input.
		1	IN	TTL	PORTA<0> data input; disabled when analog input enabled.
	AN0	1	IN	ANA	A/D Input Channel 0 and Comparator C1- input. Default configuration on POR; does not affect digital output.
RA1/AN1	RA1	0	OUT	DIG	LATA<1> data output; not affected by analog input.
		1	IN	TTL	PORTA<1> data input; reads '0' on POR.
	AN1	1	IN	ANA	A/D Input Channel 1 and Comparator C2- input. Default configuration on POR; does not affect digital output.
RA2/AN2/ VREF-/CVREF	RA2	0	OUT	DIG	LATA<2> data output; not affected by analog input. Disabled when CVREF output enabled.
		1	IN	TTL	PORTA<2> data input. Disabled when analog functions enabled; disabled when CVREF output enabled.
	AN2	1	IN	ANA	A/D Input Channel 2 and Comparator C2+ input. Default configuration on POR; not affected by analog output.
	VREF-	1	IN	ANA	A/D and comparator voltage reference low input.
	CVREF	x	OUT	ANA	Comparator voltage reference output. Enabling this feature disables digital I/O.
RA3/AN3/ VREF+	RA3	0	OUT	DIG	LATA<3> data output; not affected by analog input.
		1	IN	TTL	PORTA<3> data input; disabled when analog input enabled.
	AN3	1	IN	ANA	A/D Input Channel 3 and Comparator C1+ input. Default configuration on POR.
	VREF+	1	IN	ANA	A/D and comparator voltage reference high input.
RA4/T0CKI/ C1OUT/RCV	RA4	0	OUT	DIG	LATA<4> data output; not affected by analog input.
		1	IN	ST	PORTA<4> data input; disabled when analog input enabled.
	T0CKI	1	IN	ST	Timer0 clock input.
	C1OUT	0	OUT	DIG	Comparator 1 output; takes priority over port data.
	RCV	x	IN	TTL	External USB transceiver RCV input.
RA5/AN4/$\overline{\text{SS}}$/ HLVDIN/C2OUT	RA5	0	OUT	DIG	LATA<5> data output; not affected by analog input.
		1	IN	TTL	PORTA<5> data input; disabled when analog input enabled.
	AN4	1	IN	ANA	A/D Input Channel 4. Default configuration on POR.
	$\overline{\text{SS}}$	1	IN	TTL	Slave select input for MSSP module.
	HLVDIN	1	IN	ANA	High/Low-Voltage Detect external trip point input.
	C2OUT	0	OUT	DIG	Comparator 2 output; takes priority over port data.
OSC2/CLKO/ RA6	OSC2	x	OUT	ANA	Main oscillator feedback output connection (all XT and HS modes).
	CLKO	x	OUT	DIG	System cycle clock output (FOSC/4); available in EC, ECPLL and INTCKO modes.
	RA6	0	OUT	DIG	LATA<6> data output. Available only in ECIO, ECPIO and INTIO modes; otherwise, reads as '0'.
		1	IN	TTL	PORTA<6> data input. Available only in ECIO, ECPIO and INTIO modes; otherwise, reads as '0'.

Legend: OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

TABLE 10-3: PORTB I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RB6/KBI2/ PGC	RB6	0	OUT	DIG	LATB<6> data output.
		1	IN	TTL	PORTB<6> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI2	1	IN	TTL	Interrupt-on-pin change.
	PGC	x	IN	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation.(3)
RB7/KBI3/ PGD	RB7	0	OUT	DIG	LATB<7> data output.
		1	IN	TTL	PORTB<7> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI3	1	IN	TTL	Interrupt-on-pin change.
	PGD	x	OUT	DIG	Serial execution data output for ICSP and ICD operation.(3)
		x	IN	ST	Serial execution data input for ICSP and ICD operation.(3)

Legend: OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, I^{2}C/SMB = I^{2}C/SMBus input buffer, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: Configuration on POR is determined by PBADEN Configuration bit. Pins are configured as analog inputs when PBADEN is set and digital inputs when PBADEN is cleared.

2: Alternate pin assignment for CCP2 when CCP2MX = 0. Default assignment is RC1.

3: All other pin functions are disabled when ICSP™ or ICD operation is enabled.

4: 40/44-pin devices only.

TABLE 10-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	56
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	56
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	56
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	53
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	53
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	53
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	54
SPPCON(1)	—	—	—	—	—	—	SPPOWN	SPPEN	57
SPPCFG(1)	CLKCFG1	CLKCFG0	CSEN	CLK1EN	WS3	WS2	WS1	WS0	57
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	57

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTB.

Note 1: These registers are unimplemented on 28-pin devices.

13.0 TIMER2 MODULE

The Timer2 module timer incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP module

The module is controlled through the T2CON register (Register 13-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 13-1.

13.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (FOSC/4). A 2-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by-16 prescale options. These are selected by the prescaler control bits, T2CKPS1:T2CKPS0 (T2CON<1:0>). The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 13.2 “Timer2 Interrupt”**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, while the PR2 register initializes at FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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 7 **Unimplemented:** Read as ‘0’

bit 6-3 **T2OUTPS3:T2OUTPS0**: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

16.0 ENHANCED CAPTURE/COMPARE/PWM (ECCP) MODULE

Note:	The ECCP module is implemented only in 40/44-pin devices.

In 28-pin devices, CCP1 is implemented as a standard CCP module with Enhanced PWM capabilities. These include the provision for 2 or 4 output channels, user-selectable polarity, dead-band control and automatic shutdown and restart. The Enhanced features are discussed in detail in **Section 16.4 “Enhanced PWM Mode”**. Capture, Compare and single output PWM functions of the ECCP module are the same as described for the standard CCP module.

The control register for the Enhanced CCP module is shown in Register 16-1. It differs from the CCPxCON registers in 28-pin devices in that the two Most Significant bits are implemented to control PWM functionality.

REGISTER 16-1: CCP1CON: ECCP CONTROL REGISTER (40/44-PIN DEVICES)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
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 7-6 **P1M1:P1M0:** Enhanced PWM Output Configuration bits

If CCP1M3:CCP1M2 = 00, 01, 10:
xx = P1A assigned as Capture/Compare input/output; P1B, P1C, P1D assigned as port pins

If CCP1M3:CCP1M2 = 11:
00 = Single output: P1A modulated; P1B, P1C, P1D assigned as port pins
01 = Full-bridge output forward: P1D modulated; P1A active; P1B, P1C inactive
10 = Half-bridge output: P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins
11 = Full-bridge output reverse: P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B1:DC1B0**: PWM Duty Cycle Bit 1 and Bit 0

Capture mode:
Unused.

Compare mode:
Unused.

PWM mode:
These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR1L.

bit 3-0 **CCP1M3:CCP1M0**: Enhanced CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCP module)
0001 = Reserved
0010 = Compare mode, toggle output on match
0011 = Capture mode
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, initialize CCP1 pin low, set output on compare match (set CCP1IF)
1001 = Compare mode, initialize CCP1 pin high, clear output on compare match (set CCP1IF)
1010 = Compare mode, generate software interrupt only, CCP1 pin reverts to I/O state
1011 = Compare mode, trigger special event (CCP1 resets TMR1 or TMR3, sets CCP1IF bit)
1100 = PWM mode: P1A, P1C active-high; P1B, P1D active-high
1101 = PWM mode: P1A, P1C active-high; P1B, P1D active-low
1110 = PWM mode: P1A, P1C active-low; P1B, P1D active-high
1111 = PWM mode: P1A, P1C active-low; P1B, P1D active-low

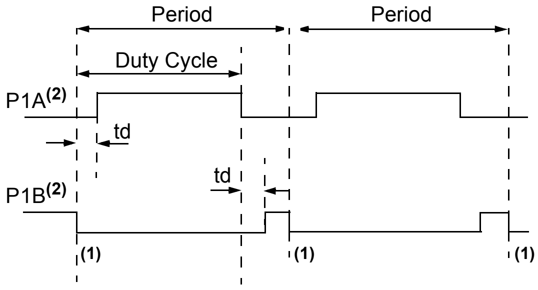
16.4.4 HALF-BRIDGE MODE

In the Half-Bridge Output mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the P1A pin, while the complementary PWM output signal is output on the P1B pin (Figure 16-4). This mode can be used for half-bridge applications, as shown in Figure 16-5, or for full-bridge applications where four power switches are being modulated with two PWM signals.

In Half-Bridge Output mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of bits PDC6:PDC0 sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 16.4.6 "Programmable Dead-Band Delay"** for more details of the dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORTC<2> and PORTD<5> data latches, the TRISC<2> and TRISD<5> bits must be cleared to configure P1A and P1B as outputs.

FIGURE 16-4: HALF-BRIDGE PWM OUTPUT



td = Dead-Band Delay

Note 1: At this time, the TMR2 register is equal to the PR2 register.

2: Output signals are shown as active-high.

FIGURE 16-5: EXAMPLES OF HALF-BRIDGE OUTPUT MODE APPLICATIONS

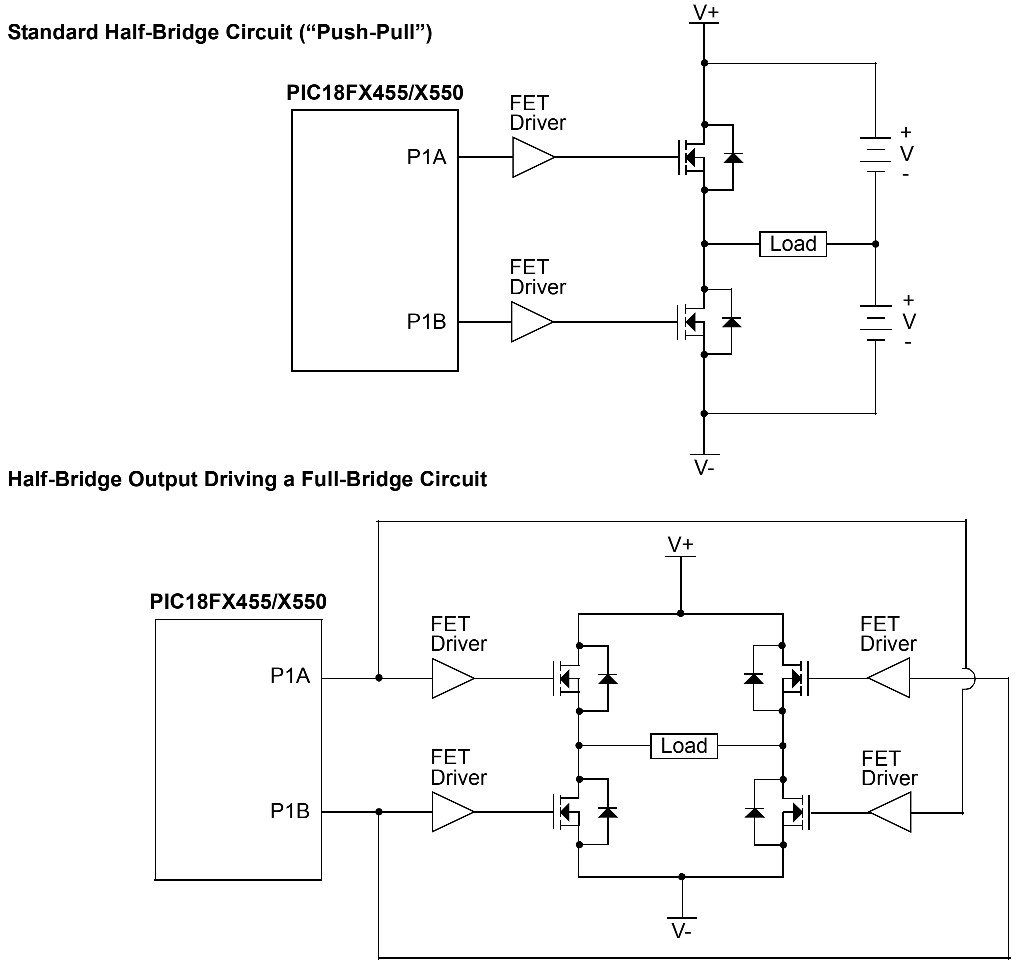


TABLE 17-1: DIFFERENTIAL OUTPUTS TO TRANSCEIVER

VPO	VMO	Bus State
0	0	Single-Ended Zero
0	1	Differential ‘0’
1	0	Differential ‘1’
1	1	Illegal Condition

TABLE 17-2: SINGLE-ENDED INPUTS FROM TRANSCEIVER

VP	VM	Bus State
0	0	Single-Ended Zero
0	1	Low Speed
1	0	High Speed
1	1	Error

The $\overline{\text{UOE}}$ signal toggles the state of the external transceiver. This line is pulled low by the device to enable the transmission of data from the SIE to an external device.

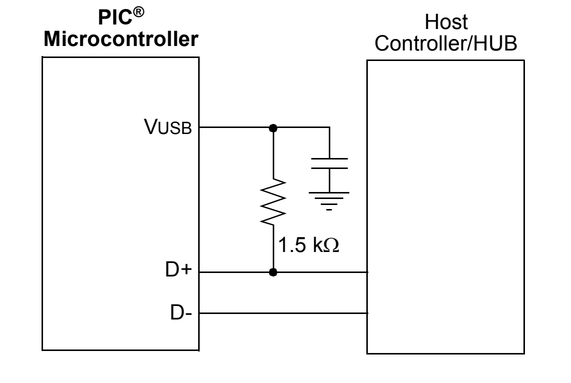
17.2.2.3 Internal Pull-up Resistors

The PIC18FX455/X550 devices have built-in pull-up resistors designed to meet the requirements for low-speed and full-speed USB. The UPUEN bit (UCFG<4>) enables the internal pull-ups. Figure 17-1 shows the pull-ups and their control.

17.2.2.4 External Pull-up Resistors

External pull-up may also be used if the internal resistors are not used. The VUSB pin may be used to pull up D+ or D-. The pull-up resistor must be 1.5 kΩ (±5%) as required by the USB specifications. Figure 17-3 shows an example.

FIGURE 17-3: EXTERNAL CIRCUITRY



Note: The above setting shows a typical connection for a full-speed configuration using an on-chip regulator and an external pull-up resistor.

17.2.2.5 Ping-Pong Buffer Configuration

The usage of ping-pong buffers is configured using the PPB1:PPB0 bits. Refer to **Section 17.4.4 “Ping-Pong Buffering”** for a complete explanation of the ping-pong buffers.

17.2.2.6 USB Output Enable Monitor

The USB $\overline{\text{OE}}$ monitor provides indication as to whether the SIE is listening to the bus or actively driving the bus. This is enabled by default when using an external transceiver or when UCFG<6> = 1.

The USB $\overline{\text{OE}}$ monitoring is useful for initial system debugging, as well as scope triggering during eye pattern generation tests.

17.2.2.7 Eye Pattern Test Enable

An automatic eye pattern test can be generated by the module when the UCFG<7> bit is set. The eye pattern output will be observable based on module settings, meaning that the user is first responsible for configuring the SIE clock settings, pull-up resistor and Transceiver mode. In addition, the module has to be enabled.

Once UTEYE is set, the module emulates a switch from a receive to transmit state and will start transmitting a J-K-J-K bit sequence (K-J-K-J for full speed). The sequence will be repeated indefinitely while the Eye Pattern Test mode is enabled.

Note that this bit should never be set while the module is connected to an actual USB system. This test mode is intended for board verification to aid with USB certification tests. It is intended to show a system developer the noise integrity of the USB signals which can be affected by board traces, impedance mismatches and proximity to other system components. It does not properly test the transition from a receive to a transmit state. Although the eye pattern is not meant to replace the more complex USB certification test, it should aid during first order system debugging.

REGISTER 19-4: SSPCON1: MSSP CONTROL REGISTER 1 (I²C™ MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
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 7 **WCOL:** Write Collision Detect bit

<u>In Master Transmit mode:</u>
1 = A write to the SSPBUF register was attempted while the I²C conditions were not valid for a transmission to be started (must be cleared in software)
0 = No collision

<u>In Slave Transmit mode:</u>
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

<u>In Receive mode (Master or Slave modes):</u>
This is a “don’t care” bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

<u>In Receive mode:</u>
1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)
0 = No overflow

<u>In Transmit mode:</u>
This is a “don’t care” bit in Transmit mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit

1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins**(1)**
0 = Disables serial port and configures these pins as I/O port pins**(1)**

bit 4 **CKP:** SCK Release Control bit

<u>In Slave mode:</u>
1 = Release clock
0 = Holds clock low (clock stretch), used to ensure data setup time

<u>In Master mode:</u>
Unused in this mode.

bit 3-0 **SSPM3:SSPM0:** Master Synchronous Serial Port Mode Select bits

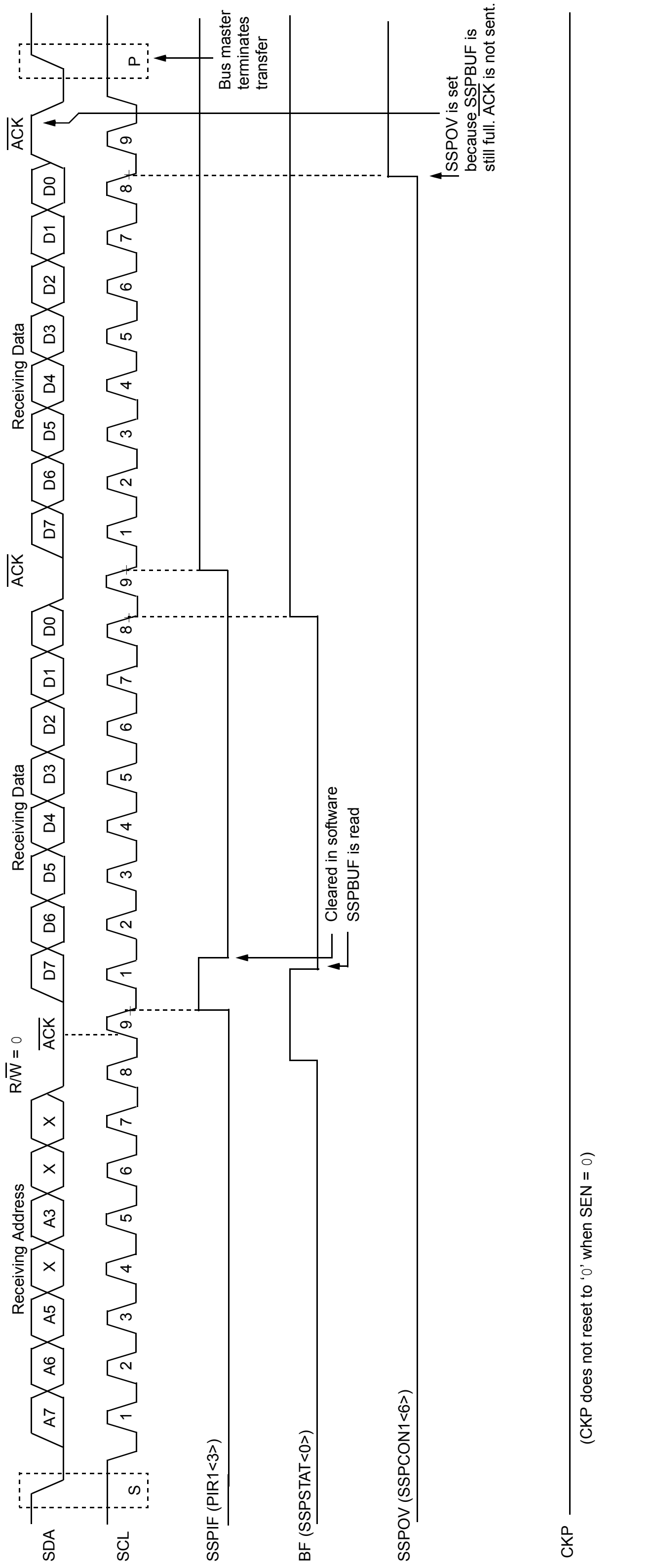
1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled**(2)**
1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled**(2)**
1011 = I²C Firmware Controlled Master mode (slave Idle)**(2)**
1000 = I²C Master mode, clock = FOSC/(4 * (SSPADD + 1))**(2,3)**
0111 = I²C Slave mode, 10-bit address**(2)**
0110 = I²C Slave mode, 7-bit address**(2)**

Note 1: When enabled, the SDA and SCL pins must be properly configured as input or output.

2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

3: Guideline only; exact baud rate slightly dependent upon circuit conditions, but the highest clock rate should not exceed this formula. SSPADD values of ‘0’ and ‘1’ are not supported.

FIGURE 19-9: I^{2}C™ SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01011 (RECEPTION, 7-BIT ADDRESS)



Note 1: x = Don't care (i.e., address bit can be either a '1' or a '0').

2: In this example, an address equal to A7.A6.A5.X.A3.X.X will be Acknowledged and cause an interrupt.

20.3.2 EUSART SYNCHRONOUS MASTER RECEPTION

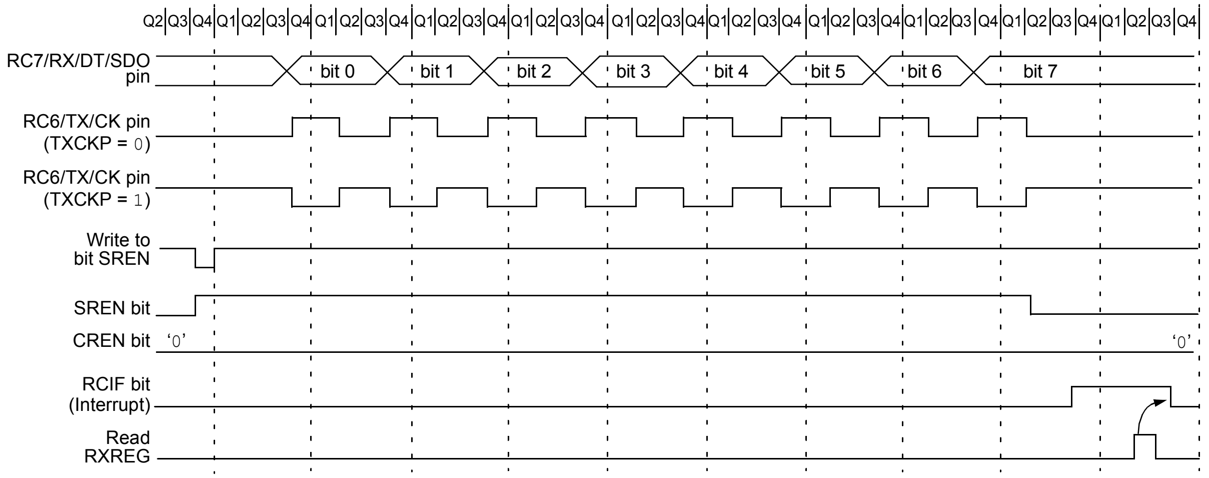
Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTA<5>), or the Continuous Receive Enable bit, CREN (RCSTA<4>). Data is sampled on the RX pin on the falling edge of the clock.

If enable bit, SREN, is set, only a single word is received. If enable bit, CREN, is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
3. Ensure bits, CREN and SREN, are clear.
4. If the signal from the CK pin is to be inverted, set the TXCKP bit. If the signal from the DT pin is to be inverted, set the RXDTP bit.
5. If interrupts are desired, set enable bit, RCIE.
6. If 9-bit reception is desired, set bit, RX9.
7. If a single reception is required, set bit, SREN. For continuous reception, set bit, CREN.
8. Interrupt flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if the enable bit, RCIE, was set.
9. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREG register.
11. If any error occurred, clear the error by clearing bit, CREN.
12. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)


Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
RC7/RX/DT/SDO pin
bit 0
bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
bit 7
RC6/TX/CK pin (TXCKP = 0)
RC6/TX/CK pin (TXCKP = 1)
Write to bit SREN
SREN bit
CREN bit
'0'
'0'
RCIF bit (Interrupt)
Read RXREG

Note: Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0.

TABLE 20-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	53
PIR1	SPPIF[1]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	56
PIE1	SPPIE[1]	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	56
IPR1	SPPIP[1]	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	56
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	55
RCREG	EUSART Receive Register								55
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	55
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	55
SPBRGH	EUSART Baud Rate Generator Register High Byte								55
SPBRG	EUSART Baud Rate Generator Register Low Byte								55

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.
Note 1: Reserved in 28-pin devices; always maintain these bits clear.

REGISTER 25-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)

U-0	U-0	R/P-0	R/P-0	R/P-0	R/P-0	R/P-0	R/P-0
—	—	USBDIV	CPUDIV1	CPUDIV0	PLLDIV2	PLLDIV1	PLLDIV0
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’
-n = Value when device is unprogrammed		u = Unchanged from programmed state

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **USBDIV:** USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1)

1 = USB clock source comes from the 96 MHz PLL divided by 2
0 = USB clock source comes directly from the primary oscillator block with no postscale

bit 4-3 **CPUDIV1:CPUDIV0:** System Clock Postscaler Selection bits

For XT, HS, EC and ECIO Oscillator modes:
11 = Primary oscillator divided by 4 to derive system clock
10 = Primary oscillator divided by 3 to derive system clock
01 = Primary oscillator divided by 2 to derive system clock
00 = Primary oscillator used directly for system clock (no postscaler)

For XTPLL, HSPLL, ECPLL and ECPIO Oscillator modes:
11 = 96 MHz PLL divided by 6 to derive system clock
10 = 96 MHz PLL divided by 4 to derive system clock
01 = 96 MHz PLL divided by 3 to derive system clock
00 = 96 MHz PLL divided by 2 to derive system clock

bit 2-0 **PLLDIV2:PLLDIV0:** PLL Prescaler Selection bits

111 = Divide by 12 (48 MHz oscillator input)
110 = Divide by 10 (40 MHz oscillator input)
101 = Divide by 6 (24 MHz oscillator input)
100 = Divide by 5 (20 MHz oscillator input)
011 = Divide by 4 (16 MHz oscillator input)
010 = Divide by 3 (12 MHz oscillator input)
001 = Divide by 2 (8 MHz oscillator input)
000 = No prescale (4 MHz oscillator input drives PLL directly)

CPFSGT — Compare f with W, Skip if f > W

Syntax: CPFSGT f {,a}

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: (f) – (W),
skip if (f) > (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	010a	ffff	ffff

Description: Compares the contents of data memory location ‘f’ to the contents of the W by performing an unsigned subtraction.
If the contents of ‘f’ are greater than the contents of WREG, then the fetched instruction is discarded and a `NOP` is executed instead, making this a two-cycle instruction.
If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank (default).
If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 26.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE       CPFSGT REG, 0
NGREATER   :
GREATER    :
```

```
Before Instruction
    PC        =    Address (HERE)
    W         =    ?
After Instruction
    If REG    >    W;
        PC    =    Address (GREATER)
    If REG    ≤    W;
        PC    =    Address (NGREATER)
```

CPFSLT — Compare f with W, Skip if f < W

Syntax: CPFSLT f {,a}

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	000a	ffff	ffff

Description: Compares the contents of data memory location ‘f’ to the contents of W by performing an unsigned subtraction.
If the contents of ‘f’ are less than the contents of W, then the fetched instruction is discarded and a `NOP` is executed instead, making this a two-cycle instruction.
If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE     CPFSLT REG, 1
NLESS    :
LESS     :
```

```
Before Instruction
    PC        =    Address (HERE)
    W         =    ?
After Instruction
    If REG    <    W;
    PC        =    Address (LESS)
    If REG    ≥    W;
    PC        =    Address (NLESS)
```

MOVLW Move Literal to W

Syntax: MOVLW k

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow W$

Status Affected: None

Encoding:

0000	1110	kkkk	kkkk

Description: The eight-bit literal ‘k’ is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example: `MOVLW 5Ah`

After Instruction
W = 5Ah

MOVWF Move W to f

Syntax: MOVWF f {,a}

Operands: $0 \leq f \leq 255$
$a \in [0,1]$

Operation: $(W) \rightarrow f$

Status Affected: None

Encoding:

0110	111a	ffff	ffff

Description: Move data from W to register ‘f’. Location ‘f’ can be anywhere in the 256-byte bank.
If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank (default).
If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 26.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write register ‘f’

Example: `MOVWF REG, 0`

Before Instruction
W = 4Fh
REG = FFh
After Instruction
W = 4Fh
REG = 4Fh

NEGF Negate f

Syntax: NEGF f {,a}

Operands: $0 \leq f \leq 255$
$a \in [0,1]$

Operation: $(\bar{f}) + 1 \rightarrow f$

Status Affected: N, OV, C, DC, Z

Encoding:

`0110`	`110a`	`ffff`	`ffff`

Description: Location ‘f’ is negated using two’s complement. The result is placed in the data memory location ‘f’.
If ‘a’ is ‘`0`’, the Access Bank is selected. If ‘a’ is ‘`1`’, the BSR is used to select the GPR bank (default).
If ‘a’ is ‘`0`’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 26.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write register ‘f’

Example: `NEGF    REG, 1`

Before Instruction
REG = `0011 1010` [3Ah]
After Instruction
REG = `1100 0110` [C6h]

NOP No Operation

Syntax: NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

`0000`
`1111`
	`0000`
	`xxxx`
		`0000`
		`xxxx`
			`0000`
			`xxxx`

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

TBLWT Table Write

Syntax: TBLWT (*; *+; *-; +*)

Operands: None

Operation: if TBLWT*,
(TABLAT) → Holding Register,
TBLPTR – No Change;
if TBLWT*+,
(TABLAT) → Holding Register,
(TBLPTR) + 1 → TBLPTR;
if TBLWT*-,
(TABLAT) → Holding Register,
(TBLPTR) – 1 → TBLPTR;
if TBLWT+*,
(TBLPTR) + 1 → TBLPTR;
(TABLAT) → Holding Register

Status Affected: None

Encoding:

0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*

Description: This instruction uses the 3 LSBs of TBLPTR to determine which of the 8 holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 6.0 “Flash Program Memory”** for additional details on programming Flash memory.)
The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range. The LSb of the TBLPTR selects which byte of the program memory location to access.

TBLPTR[0] = `0`:	Least Significant Byte of Program Memory Word
TBLPTR[0] = `1`:	Most Significant Byte of Program Memory Word

The `TBLWT` instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read TABLAT)	No operation	No operation (Write to Holding Register)

TBLWT Table Write (Continued)

Example 1: `TBLWT *+;`

Before Instruction

TABLAT	=	55h
TBLPTR	=	00A356h
HOLDING REGISTER (00A356h)	=	FFh

After Instructions (table write completion)

TABLAT	=	55h
TBLPTR	=	00A357h
HOLDING REGISTER (00A356h)	=	55h

Example 2: `TBLWT +*;`

Before Instruction

TABLAT	=	34h
TBLPTR	=	01389Ah
HOLDING REGISTER (01389Ah)	=	FFh
HOLDING REGISTER (01389Bh)	=	FFh

After Instruction (table write completion)

TABLAT	=	34h
TBLPTR	=	01389Bh
HOLDING REGISTER (01389Ah)	=	FFh
HOLDING REGISTER (01389Bh)	=	34h

ADDWF — ADD W to Indexed (Indexed Literal Offset mode)

Syntax:	ADDWF [k] {,d}
Operands:	0 ≤ k ≤ 95 d ∈ [0,1]
Operation:	(W) + ((FSR2) + k) → dest
Status Affected:	N, OV, C, DC, Z
Encoding:	0010 01d0 kkkk kkkk
Description:	The contents of W are added to the contents of the register indicated by FSR2, offset by the value ‘k’. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’ (default).
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read ‘k’	Process Data	Write to destination

Example: `ADDWF [OFST] ,0`

Before Instruction		
W	=	17h
OFST	=	2Ch
FSR2	=	0A00h
Contents of 0A2Ch	=	20h
After Instruction		
W	=	37h
Contents of 0A2Ch	=	20h

BSF — Bit Set Indexed (Indexed Literal Offset mode)

Syntax:	BSF [k], b
Operands:	0 ≤ f ≤ 95 0 ≤ b ≤ 7
Operation:	1 → ((FSR2) + k)<b>
Status Affected:	None
Encoding:	1000 bbb0 kkkk kkkk
Description:	Bit ‘b’ of the register indicated by FSR2, offset by the value ‘k’, is set.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination

Example: `BSF [FLAG_OFST], 7`

Before Instruction		
FLAG_OFST	=	0Ah
FSR2	=	0A00h
Contents of 0A0Ah	=	55h
After Instruction		
Contents of 0A0Ah	=	D5h

SETF — Set Indexed (Indexed Literal Offset mode)

Syntax:	SETF [k]
Operands:	0 ≤ k ≤ 95
Operation:	FFh → ((FSR2) + k)
Status Affected:	None
Encoding:	0110 1000 kkkk kkkk
Description:	The contents of the register indicated by FSR2, offset by ‘k’, are set to FFh.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read ‘k’	Process Data	Write register

Example: `SETF [OFST]`

Before Instruction		
OFST	=	2Ch
FSR2	=	0A00h
Contents of 0A2Ch	=	00h
After Instruction		
Contents of 0A2Ch	=	FFh

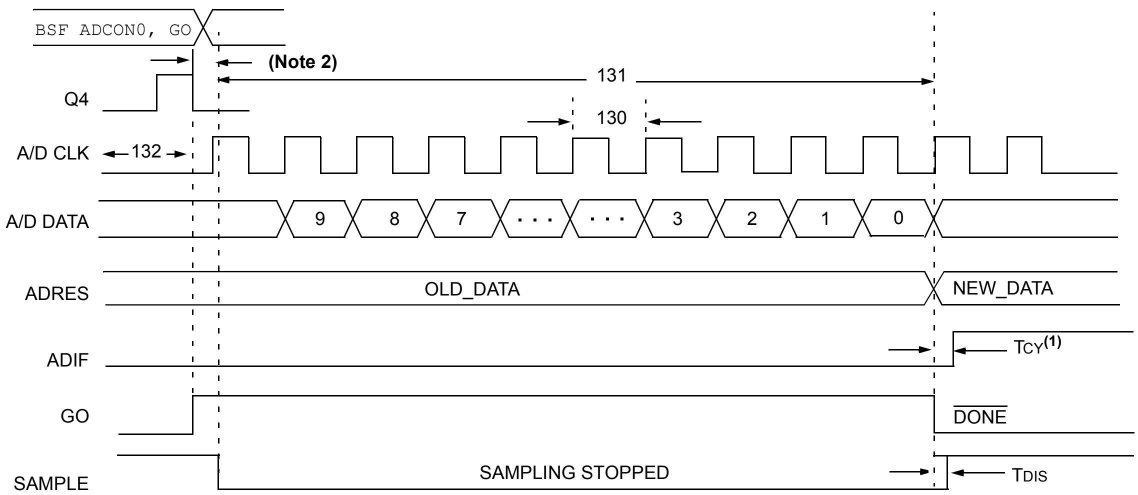
TABLE 28-28: A/D CONVERTER CHARACTERISTICS: PIC18F2455/2550/4455/4550 (INDUSTRIAL) PIC18LF2455/2550/4455/4550 (INDUSTRIAL)

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
A01	N_R	Resolution	—	—	10	bit	$\Delta V_{REF} \geq 3.0V$
A03	E_{IL}	Integral Linearity Error	—	—	<±1	LSB	$\Delta V_{REF} \geq 3.0V$
A04	E_{DL}	Differential Linearity Error	—	—	<±1	LSB	$\Delta V_{REF} \geq 3.0V$
A06	E_{OFF}	Offset Error	—	—	<±2.0	LSB	$\Delta V_{REF} \geq 3.0V$
A07	E_{GN}	Gain Error	—	—	<±1	LSB	$\Delta V_{REF} \geq 3.0V$
A10	—	Monotonicity	Guaranteed[1]			—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	ΔV_{REF}	Reference Voltage Range ($V_{REFH} - V_{REFL}$)	1.8 3.0	— —	$V_{DD} - V_{SS}$ $V_{DD} - V_{SS}$	V V	$V_{DD} < 3.0V$ $V_{DD} \geq 3.0V$
A21	V_{REFH}	Reference Voltage High	$V_{ss} + \Delta V_{REF}$	—	V_{DD}	V	
A22	V_{REFL}	Reference Voltage Low	V_{SS}	—	$V_{DD} - \Delta V_{REF}$	V	
A25	V_{AIN}	Analog Input Voltage	V_{REFL}	—	V_{REFH}	V	
A30	Z_{AIN}	Recommended Impedance of Analog Voltage Source	—	—	2.5	kΩ	
A50	I_{REF}	V_{REF} Input Current[2]	— —	— —	5 150	μA μA	During V_{AIN} acquisition. During A/D conversion cycle.

Note 1: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

2: V_{REFH} current is from RA3/AN3/V_{REF+} pin or V_{DD}, whichever is selected as the V_{REFH} source. V_{REFL} current is from RA2/AN2/V_{REF-}/CV_{REF} pin or V_{SS}, whichever is selected as the V_{REFL} source.

FIGURE 28-23: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of T_{CY} is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

2: This is a minimal RC delay (typically 100 ns), which also disconnects the holding capacitor from the analog input.