

Welcome to [E-XFL.COM](http://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	25MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18f4321-e-p">https://www.e-xfl.com/product-detail/microchip-technology/pic18f4321-e-p</a>

# PIC18F2221/2321/4221/4321 FAMILY

---

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at [docerrors@microchip.com](mailto:docerrors@microchip.com) or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

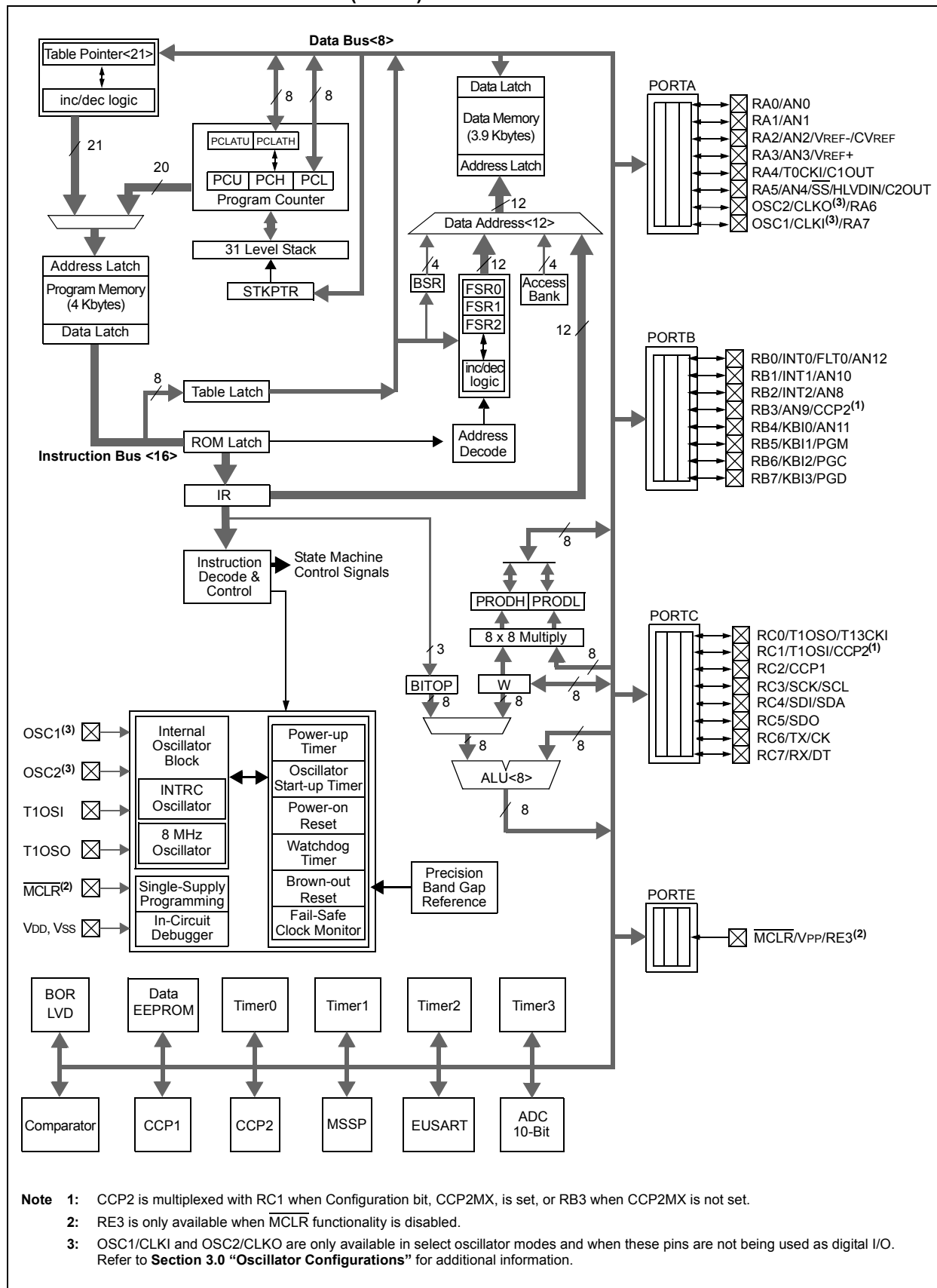
When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

# PIC18F2221/2321/4221/4321 FAMILY

**FIGURE 1-1: PIC18F2221/2321 (28-PIN) BLOCK DIAGRAM**



# PIC18F2221/2321/4221/4321 FAMILY

## 4.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power-managed modes do not invoke the OST at all. There are two cases:

- PRI\_IDLE mode, where the primary clock source is not stopped; and
- the primary clock source is not any of the LP, XT, HS or HSPLL modes.

In these instances, the primary clock source either does not require an oscillator start-up delay since it is already running (PRI\_IDLE), or normally does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes). However, a fixed delay of interval TcSD following the wake event is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

**TABLE 4-2: EXIT DELAY ON WAKE-UP BY RESET FROM SLEEP MODE OR ANY IDLE MODE (BY CLOCK SOURCES)**

Clock Source before Wake-up	Clock Source after Wake-up	Exit Delay	Clock Ready Status Bit (OSCCON)
Primary Device Clock (PRI_IDLE mode)	LP, XT, HS	TcSD <sup>(1)</sup>	OSTS
	HSPLL		
	EC, RC		IOFS
	INTOSC <sup>(2)</sup>		
T1OSC	LP, XT, HS	TOST <sup>(3)</sup>	OSTS
	HSPLL	TOST + t <sub>rc</sub> <sup>(3)</sup>	
	EC, RC	TcSD <sup>(1)</sup>	
	INTOSC <sup>(2)</sup>	TIOBST <sup>(4)</sup>	IOFS
INTOSC <sup>(3)</sup>	LP, XT, HS	TOST <sup>(3)</sup>	OSTS
	HSPLL	TOST + t <sub>rc</sub> <sup>(3)</sup>	
	EC, RC	TcSD <sup>(1)</sup>	
	INTOSC <sup>(2)</sup>	None	IOFS
None (Sleep mode)	LP, XT, HS	TOST <sup>(3)</sup>	OSTS
	HSPLL	TOST + t <sub>rc</sub> <sup>(3)</sup>	
	EC, RC	TcSD <sup>(1)</sup>	
	INTOSC <sup>(2)</sup>	TIOBST <sup>(4)</sup>	IOFS

**Note 1:** TcSD (parameter 38) is a required delay when waking from Sleep and all Idle modes and runs concurrently with any other required delays (see **Section 4.4 “Idle Modes”**). On Reset, INTOSC defaults to 1 MHz.

**2:** Includes both the INTOSC 8 MHz source and postscaler derived frequencies.

**3:** TOST is the Oscillator Start-up Timer (parameter 32). t<sub>rc</sub> is the PLL Lock-out Timer (parameter F12); it is also designated as TPLL.

**4:** Execution continues during TIOBST (parameter 39), the INTOSC stabilization period.

# PIC18F2221/2321/4221/4321 FAMILY

## 5.5 Device Reset Timers

PIC18F2221/2321/4221/4321 family devices incorporate three separate on-chip timers that help regulate the Power-on Reset process. Their main function is to ensure that the device clock is stable before code is executed. These timers are:

- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- PLL Lock Time-out

### 5.5.1 POWER-UP TIMER (PWRT)

The Power-up Timer (PWRT) of the PIC18F2221/2321/4221/4321 family devices is an 11-bit counter which uses the INTRC source as the clock input. This yields an approximate time interval of  $2048 \times 32 \mu\text{s} = 65.6 \text{ ms}$ . While the PWRT is counting, the device is held in Reset.

The power-up time delay depends on the INTRC clock and will vary from chip to chip due to temperature and process variation. See DC parameter 33 for details.

The PWRT is enabled by clearing the  $\overline{\text{PWRTE}}\text{N}$  Configuration bit.

### 5.5.2 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (parameter 33). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP, HS and HSPLL modes and only on Power-on Reset, or on exit from most power-managed modes.

### 5.5.3 PLL LOCK TIME-OUT

With the PLL enabled in HSPLL mode, the time-out sequence following a Power-on Reset is slightly different from other oscillator modes. A separate timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out ( $T_{\text{PLL}}$ ) is typically 2 ms and follows the oscillator start-up time-out.

### 5.5.4 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

1. After the POR pulse has cleared, PWRT time-out is invoked (if enabled).
2. Then, the OST is activated.

The total time-out will vary based on oscillator configuration and the status of the PWRT. Figure 5-3, Figure 5-4, Figure 5-5, Figure 5-6 and Figure 5-7 all depict time-out sequences on power-up, with the Power-up Timer enabled and the device operating in HS Oscillator mode. Figures 5-3 through 5-6 also apply to devices operating in XT or LP modes. For devices in RC mode and with the PWRT disabled, there will be no time-out at all.

Since the time-outs occur from the POR pulse, if  $\overline{\text{MCLR}}$  is kept low long enough, all time-outs will expire. Bringing  $\overline{\text{MCLR}}$  high will begin execution immediately (Figure 5-5). This is useful for testing purposes or to synchronize more than one PIC18FXXXX device operating in parallel.

**TABLE 5-2: TIME-OUT IN VARIOUS SITUATIONS**

Oscillator Configuration	Power-up <sup>(2)</sup> and Brown-out Reset		Exit from Power-Managed Mode
	$\overline{\text{PWRTE}}\text{N} = 0$	$\overline{\text{PWRTE}}\text{N} = 1$	
HSPLL	$66 \text{ ms}^{(1)} + 1024 \text{ TOSC} + 2 \text{ ms}^{(2)}$	$1024 \text{ TOSC} + 2 \text{ ms}^{(2)}$	$1024 \text{ TOSC} + 2 \text{ ms}^{(2)}$
HS, XT, LP	$66 \text{ ms}^{(1)} + 1024 \text{ TOSC}$	$1024 \text{ TOSC}$	$1024 \text{ TOSC}$
EC, ECIO	$66 \text{ ms}^{(1)}$	—	—
RC, RCIO	$66 \text{ ms}^{(1)}$	—	—
INTIO1, INTIO2	$66 \text{ ms}^{(1)}$	—	—

**Note 1:** 66 ms (65.5 ms) is the nominal Power-up Timer (PWRT) delay.

**Note 2:** 2 ms is the nominal time required for the PLL to lock.

# PIC18F2221/2321/4221/4321 FAMILY

## 6.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 Enhanced microcontroller devices:

- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Flash program memory is provided in **Section 7.0 “Flash Program Memory”**. Data EEPROM is discussed separately in **Section 8.0 “Data EEPROM Memory”**.

## 6.1 Program Memory Organization

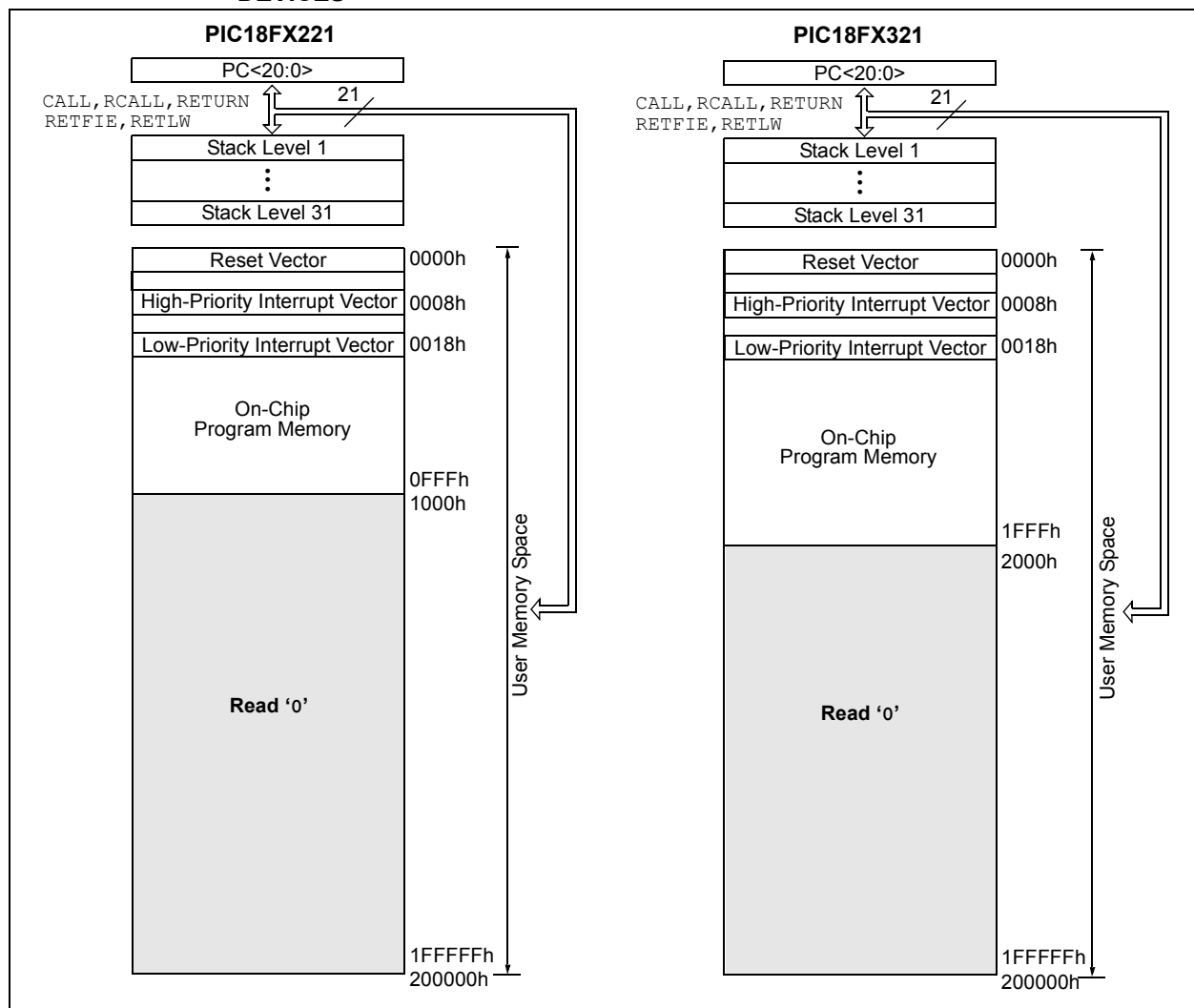
PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all ‘0’s (a NOP instruction).

The PIC18F2221 and PIC18F4221 each have 4 Kbytes of Flash memory and can store up to 2048 single-word instructions. The PIC18F2321 and PIC18F4321 each have 8 Kbytes of Flash memory and can store up to 4096 single-word instructions.

PIC18 devices have two interrupt vectors. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

The program memory maps for PIC18F2221/4221 and PIC18F2321/4321 devices are shown in Figure 6-1.

**FIGURE 6-1: PROGRAM MEMORY MAP AND STACK FOR PIC18F2221/2321/4221/4321 FAMILY DEVICES**



# PIC18F2221/2321/4221/4321 FAMILY

## 6.1.1 PROGRAM COUNTER

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide and is contained in three separate 8-bit registers. The low byte, known as the PCL register, is both readable and writable. The high byte, or PCH register, contains the PC<15:8> bits; it is not directly readable or writable. Updates to the PCH register are performed through the PCLATH register. The upper byte is called PCU. This register contains the PC<20:16> bits; it is also not directly readable or writable. Updates to the PCU register are performed through the PCLATU register.

The contents of PCLATH and PCLATU are transferred to the program counter by any operation that writes PCL. Similarly, the upper two bytes of the program counter are transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 6.1.4.1 “Computed GOTO”**).

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the Least Significant bit of PCL is fixed to a value of ‘0’. The PC increments by 2 to address sequential instructions in the program memory.

The **CALL**, **RCALL**, **GOTO** and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter.

## 6.1.2 RETURN ADDRESS STACK

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC is pushed onto the stack when a **CALL** or **RCALL** instruction is executed or an interrupt is Acknowledged. The PC value is pulled off the stack on a **RETURN**, **RETLW** or a **RETFIE** instruction. PCLATU and PCLATH are not affected by any of the **RETURN** or **CALL** instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, STKPTR. The stack space is not part of either program or data space. The Stack Pointer is readable and writable and the address on the top of the stack is readable and writable through the Top-of-Stack Special Function Registers. Data can also be pushed to, or popped from the stack, using these registers.

A **CALL** type instruction causes a push onto the stack; the Stack Pointer is first incremented and the location pointed to by the Stack Pointer is written with the contents of the PC (already pointing to the instruction following the **CALL**). A **RETURN** type instruction causes a pop from the stack; the contents of the location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

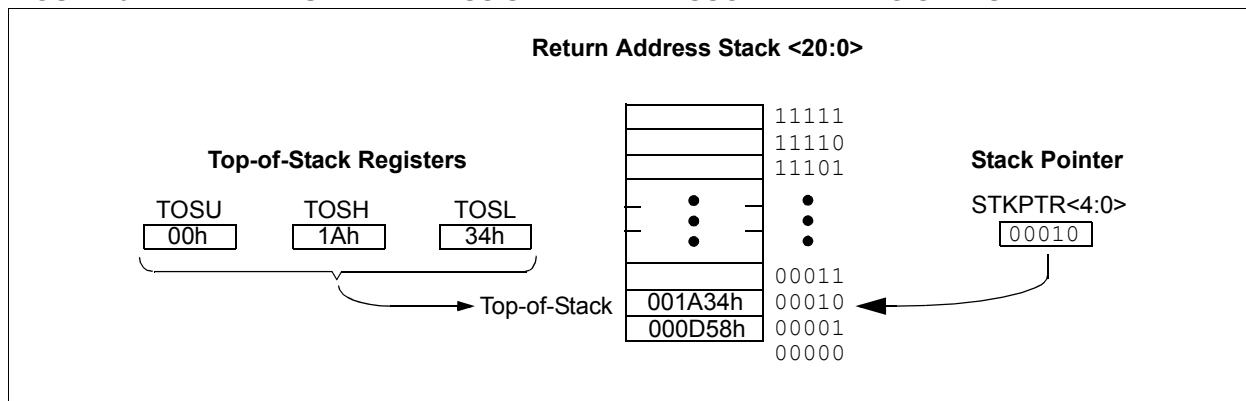
The Stack Pointer is initialized to ‘00000’ after all Resets. There is no RAM associated with the location corresponding to a Stack Pointer value of ‘00000’; this is only a Reset value. Status bits indicate if the stack is full or has overflowed or has underflowed.

### 6.1.2.1 Top-of-Stack Access

Only the top of the return address stack (TOS) is readable and writable. A set of three registers, TOSU:TOSH:TOSL, hold the contents of the stack location pointed to by the STKPTR register (Figure 6-2). This allows users to implement a software stack if necessary. After a **CALL**, **RCALL** or interrupt, the software can read the pushed value by reading the TOSU:TOSH:TOSL registers. These values can be placed on a user-defined software stack. At return time, the software can return these values to TOSU:TOSH:TOSL and do a return.

The user must disable the global interrupt enable bits while accessing the stack to prevent inadvertent stack corruption.

**FIGURE 6-2: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS**



# PIC18F2221/2321/4221/4321 FAMILY

## 8.0 DATA EEPROM MEMORY

The data EEPROM is a nonvolatile memory array, separate from the data RAM and program memory, that is used for long-term storage of program data. It is not directly mapped in either the register file or program memory space but is indirectly addressed through the Special Function Registers (SFRs). The EEPROM is readable and writable during normal operation over the entire VDD range.

Four SFRs are used to read and write to the data EEPROM as well as the program memory. They are:

- EECON1
- EECON2
- EEDATA
- EEADR

The data EEPROM allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and the EEADR register holds the address of the EEPROM location being accessed.

The EEPROM data memory is rated for high erase/write cycle endurance. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer. It will vary with voltage and temperature as well as from chip to chip. Please refer to parameter D122 (Table 27-1 in **Section 27.0 “Electrical Characteristics”**) for exact limits.

### 8.1 EECON1 and EECON2 Registers

Access to the data EEPROM is controlled by two registers: EECON1 and EECON2. These are the same registers which control access to the program memory and are used in a similar manner for the data EEPROM.

The EECON1 register (Register 8-1) is the control register for data and program memory access. Control bit EEPGD determines if the access will be to program or data EEPROM memory. When clear, operations will access the data EEPROM memory. When set, program memory is accessed.

Control bit CFGS determines if the access will be to the Configuration registers or to program memory/data EEPROM memory. When set, subsequent operations access Configuration registers. When CFGS is clear, the EEPGD bit selects either program Flash or data EEPROM memory.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WREN bit is set and cleared when the internal programming timer expires and the write operation is complete.

**Note:** During normal operation, the WRERR bit is read as '1'. This can indicate that a write operation was prematurely terminated by a Reset, or a write operation was attempted improperly.

The WR control bit initiates write operations. The bit cannot be cleared, only set, in software; it is cleared in hardware at the completion of the write operation.

**Note:** The EEIF interrupt flag bit (PIR2<4>) is set when the write is complete. It must be cleared in software.

Control bits, RD and WR, start read and erase/write operations, respectively. These bits are set by firmware and cleared by hardware at the completion of the operation.

The RD bit cannot be set when accessing program memory (EEPGD = 1). Program memory is read using table read instructions. See **Section 7.1 “Table Reads and Table Writes”** regarding table reads.

The EECON2 register is not a physical register. It is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.



# PIC18F2221/2321/4221/4321 FAMILY

## 11.5 PORTE, TRISE and LATE Registers

Depending on the particular PIC18F2221/2321/4221/4321 family device selected, PORTE is implemented in two different ways.

For 40/44-pin devices, PORTE is a 4-bit wide port. Three pins (RE0/RD/AN5, RE1/WR/AN6 and RE2/CS/AN7) are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers. When selected as analog inputs, these pins will read as '0'.

The corresponding Data Direction register is TRISE. Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., put the contents of the output latch on the selected pin).

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

**Note:** On a Power-on Reset, RE<2:0> are configured as analog inputs.

The upper four bits of the TRISE register also control the operation of the Parallel Slave Port. Their operation is explained in Register 11-1.

The Data Latch register (LATE) is also memory mapped. Read-modify-write operations on the LATE register, read and write the latched output value for PORTE.

The fourth pin of PORTE ( $\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$ ) is an input only pin. Its operation is controlled by the MCLRE Configuration bit. When selected as a port pin (MCLRE = 0), it functions as a digital input only pin; as such, it does not have TRIS or LAT bits associated with its operation. Otherwise, it functions as the device's Master Clear input. In either configuration, RE3 also functions as the programming voltage input during programming.

**Note:** On a Power-on Reset, RE3 is enabled as a digital input only if Master Clear functionality is disabled.

### EXAMPLE 11-5: INITIALIZING PORTE

```
CLRF    PORTE    ; Initialize PORTE by
                ; clearing output
                ; data latches
CLRF    LATE      ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0Fh       ; Configure A/D
MOVWF   ADCON1    ; for digital inputs
MOVLW   03h       ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISE     ; Set RE<0> as inputs
                ; RE<1> as outputs
                ; RE<2> as inputs
```

#### 11.5.1 PORTE IN 28-PIN DEVICES

For 28-pin devices, PORTE is only available when Master Clear functionality is disabled (MCLRE = 0). In these cases, PORTE is a single bit, input only port comprised of RE3 only. The pin operates as previously described.

# PIC18F2221/2321/4221/4321 FAMILY

## 17.0 ENHANCED CAPTURE/ COMPARE/PWM (ECCP) MODULE

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

In PIC18F4221/4321 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 17.4 “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 17-1. It differs from the CCPxCON registers in PIC18F2221/2321 devices in that the two Most Significant bits are implemented to control PWM functionality.

### REGISTER 17-1: CCP1CON REGISTER (ECCP1 MODULE, 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			

bit 7-6 **P1M<1:0>**: Enhanced PWM Output Configuration bits

If CCP1M<3:2> = 00, 01, 10:

xx = P1A assigned as Capture/Compare input/output; P1B, P1C, P1D assigned as port pins

If CCP1M<3:2> = 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 **DC1B<1:0>**: 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 **CCP1M<3:0>**: 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 (ECCP resets TMR1 or TMR3, sets CC1IF 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

#### 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

# PIC18F2221/2321/4221/4321 FAMILY

## 18.4.2 OPERATION

The MSSP module functions are enabled by setting MSSP Enable bit, SSPEN (SSPCON1<5>).

The SSPCON1 register allows control of the I<sup>2</sup>C operation. Four mode selection bits (SSPCON1<3:0>) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Master mode clock
- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Slave mode (7-bit address) with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Slave mode (10-bit address) with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Firmware Controlled Master mode, slave is Idle

Selection of any I<sup>2</sup>C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCL and SDA pins.

## 18.4.3 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

The I<sup>2</sup>C Slave mode hardware will always generate an interrupt on an address match. Address masking will allow the hardware to generate an interrupt for more than one address (up to 31 in 7-Bit Addressing mode and up to 63 in 10-Bit Addressing mode). Through the mode select bits, the user can also choose to interrupt on Start and Stop bits

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge (ACK) pulse and load the SSPBUF register with the received value currently in the SSPSR register.

Any combination of the following conditions will cause the MSSP module not to give this ACK pulse:

- The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.
- The overflow bit, SSPOV (SSPCON1<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. The BF bit is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I<sup>2</sup>C specification, as well as the requirement of the MSSP module, are shown in timing parameter 100 and parameter 101.

## 18.4.3.1 Addressing

Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8 bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

1. The SSPSR register value is loaded into the SSPBUF register.
2. The Buffer Full bit, BF, is set.
3. An ACK pulse is generated.
4. MSSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated, if enabled) on the falling edge of the ninth SCL pulse.

In 10-Bit Addressing mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '11110 A9 A8 0', where 'A9' and 'A8' are the two MSBs of the address. The sequence of events for 10-bit address is as follows, with steps 7 through 9 for the slave-transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.
4. Receive second (low) byte of address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.



# PIC18F2221/2321/4221/4321 FAMILY

---

## 18.4.6.1 I<sup>2</sup>C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I<sup>2</sup>C operation. See **Section 18.4.7 "Baud Rate"** for more detail.

A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPCON2<0>).
2. SSPIF is set. The MSSP module will wait the required start time before any other operation takes place.
3. The user loads the SSPBUF with the slave address to transmit.
4. Address is shifted out the SDA pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register.
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
7. The user loads the SSPBUF with eight bits of data.
8. Data is shifted out the SDA pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register.
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

# PIC18F2221/2321/4221/4321 FAMILY

## REGISTER 20-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit

1 = Right justified

0 = Left justified

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT<2:0>:** A/D Acquisition Time Select bits

111 = 20 TAD

110 = 16 TAD

101 = 12 TAD

100 = 8 TAD

011 = 6 TAD

010 = 4 TAD

001 = 2 TAD

000 = 0 TAD<sup>(1)</sup>

bit 2-0 **ADCS<2:0>:** A/D Conversion Clock Select bits

111 = FRC (clock derived from A/D RC oscillator)<sup>(1)</sup>

110 = Fosc/64

101 = Fosc/16

100 = Fosc/4

011 = FRC (clock derived from A/D RC oscillator)<sup>(1)</sup>

010 = Fosc/32

001 = Fosc/8

000 = Fosc/2

**Note 1:** If the A/D FRC clock source is selected, a delay of one T<sub>cy</sub> (instruction cycle) is added before the A/D clock starts. This allows the **SLEEP** instruction to be executed before starting a conversion.

### 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

# PIC18F2221/2321/4221/4321 FAMILY

---

NOTES:

# PIC18F2221/2321/4221/4321 FAMILY

## GOTO Unconditional Branch

Syntax:	GOTO k
Operands:	$0 \leq k \leq 1048575$
Operation:	$k \rightarrow PC<20:1>$
Status Affected:	None
Encoding:	
1st word (k<7:0>)	1110
2nd word (k<19:8>)	1111
	k <sub>19</sub> kkk
	kkkk
	kkkk <sub>0</sub>
	kkkk <sub>8</sub>
Description:	GOTO allows an unconditional branch anywhere within entire 2-Mbyte memory range. The 20-bit value 'k' is loaded into PC<20:1>. GOTO is always a two-cycle instruction.
Words:	2
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	No operation	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

**Example:** GOTO THERE  
 After Instruction  
 PC = Address (THERE)

## INCF Increment f

Syntax:	INCF f{,d{,a}}
Operands:	$0 \leq f \leq 255$ $d \in [0, 1]$ $a \in [0, 1]$
Operation:	$(f) + 1 \rightarrow \text{dest}$
Status Affected:	C, DC, N, OV, Z
Encoding:	0010 10da ffff ffff
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). 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 <b>Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"</b> for details.
Words:	1
Cycles:	1

Q Cycle Activity:

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

**Example:** INCF CNT, 1, 0

Before Instruction  
 CNT = FFh  
 Z = 0  
 C = ?  
 DC = ?  
 After Instruction  
 CNT = 00h  
 Z = 1  
 C = 1  
 DC = 1

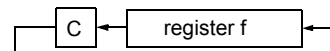


# PIC18F2221/2321/4221/4321 FAMILY

RETURN	Return from Subroutine												
Syntax:	RETURN {s}												
Operands:	$s \in [0, 1]$												
Operation:	(TOS) $\rightarrow$ PC; if $s = 1$ , (WS) $\rightarrow$ W, (STATUS) $\rightarrow$ STATUS, (BSRS) $\rightarrow$ BSR, PCLATU, PCLATH are unchanged												
Status Affected:	None												
Encoding:	<table><tr><td>0000</td><td>0000</td><td>0001</td><td>001s</td></tr></table>	0000	0000	0001	001s								
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. If 's' = 1, the contents of the shadow registers, WS, STATUS and BSRS, are loaded into their corresponding registers, W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default).												
Words:	1												
Cycles:	2												
Q Cycle Activity:	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>No operation</td><td>Process Data</td><td>POP PC from stack</td></tr><tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr></table>	Q1	Q2	Q3	Q4	Decode	No operation	Process Data	POP PC from stack	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4										
Decode	No operation	Process Data	POP PC from stack										
No operation	No operation	No operation	No operation										

**Example:** RETURN  
After Instruction:  
PC = TOS

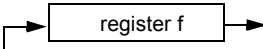
RLCF	Rotate Left f through Carry				
Syntax:	RLCF f {,d {,a}}				
Operands:	$0 \leq f \leq 255$ $d \in [0, 1]$ $a \in [0, 1]$				
Operation:	$(f \ll n) \rightarrow \text{dest} \ll n + 1$ , $(f \ll 7) \rightarrow C$ , $(C) \rightarrow \text{dest} \ll 0$				
Status Affected:	C, N, Z				
Encoding:	<table><tr><td>0011</td><td>01da</td><td>ffff</td><td>ffff</td></tr></table>	0011	01da	ffff	ffff
0011	01da	ffff	ffff		
Description:	<p>The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default).</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> <div><div>C</div><div>← register f ←</div></div>				
Words:	1				
Cycles:	1				
Q Cycle Activity:					



**Example:** RLCF REG, 0, 0

Before Instruction  
REG = 1110 0110  
C = 0  
After Instruction  
REG = 1110 0110  
W = 1100 1100  
C = 1

# PIC18F2221/2321/4221/4321 FAMILY

RRNCF		Rotate Right f (No Carry)											
Syntax:	RRNCF f {,d {,a}}												
Operands:	$0 \leq f \leq 255$ $d \in [0, 1]$ $a \in [0, 1]$												
Operation:	$(f < n) \rightarrow \text{dest} < n - 1 >$ , $(f < 0) \rightarrow \text{dest} < 7 >$												
Status Affected:	N, Z												
Encoding:	<table><tr><td>0100</td><td>00da</td><td>ffff</td><td>ffff</td></tr></table>					0100	00da	ffff	ffff				
0100	00da	ffff	ffff										
Description:	<p>The contents of register 'f' are rotated one bit to the right. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> <div></div>												
Words:	1												
Cycles:	1												
Q Cycle Activity:	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr></table>					Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4										
Decode	Read register 'f'	Process Data	Write to destination										

**Example 1:** RRNCF REG, 1, 0

Before Instruction  
 REG = 1101 0111  
 After Instruction  
 REG = 1110 1011

**Example 2:** RRNCF REG, 0, 0

Before Instruction  
 W = ?  
 REG = 1101 0111  
 After Instruction  
 W = 1110 1011  
 REG = 1101 0111

SETF	Set f								
Syntax:	SETF f {,a}								
Operands:	$0 \leq f \leq 255$ $a \in [0, 1]$								
Operation:	FFh $\rightarrow$ f								
Status Affected:	None								
Encoding:	<table><tr><td>0110</td><td>100a</td><td>ffff</td><td>ffff</td></tr></table>	0110	100a	ffff	ffff				
0110	100a	ffff	ffff						
Description:	<p>The contents of the specified register are set to FFh.</p> <p>If 'a' is '0', the Access Bank is selected.</p> <p>If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p>								
Words:	1								
Cycles:	1								
Q Cycle Activity:									
	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write register 'f'</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write register 'f'						

**Example:** SETF REG, 1

Before Instruction  
 REG = 5Ah  
 After Instruction  
 REG = FFh

# PIC18F2221/2321/4221/4321 FAMILY

SUBLW	Subtract W from Literal				
Syntax:	SUBLW k				
Operands:	$0 \leq k \leq 255$				
Operation:	$k - (W) \rightarrow W$				
Status Affected:	N, OV, C, DC, Z				
Encoding:	<table><tr><td>0000</td><td>1000</td><td>kkkk</td><td>kkkk</td></tr></table>	0000	1000	kkkk	kkkk
0000	1000	kkkk	kkkk		
Description	W is subtracted from the eight-bit literal 'k'. The result is placed in W.				
Words:	1				
Cycles:	1				

Q Cycle Activity:

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

**Example 1:** SUBLW 02h

Before Instruction	
W	= 01h
C	= ?
After Instruction	
W	= 01h
C	= 1 ; result is positive
Z	= 0
N	= 0

**Example 2:** SUBLW 02h

Before Instruction	
W	= 02h
C	= ?
After Instruction	
W	= 00h
C	= 1 ; result is zero
Z	= 1
N	= 0

**Example 3:** SUBLW 02h

Before Instruction	
W	= 03h
C	= ?
After Instruction	
W	= FFh ; (2's complement)
C	= 0 ; result is negative
Z	= 0
N	= 1

SUBWF	Subtract W from f				
Syntax:	SUBWF f {,d {,a}}				
Operands:	$0 \leq f \leq 255$ $d \in [0, 1]$ $a \in [0, 1]$				
Operation:	$(f) - (W) \rightarrow \text{dest}$				
Status Affected:	N, OV, C, DC, Z				
Encoding:	<table><tr><td>0101</td><td>11da</td><td>ffff</td><td>ffff</td></tr></table>	0101	11da	ffff	ffff
0101	11da	ffff	ffff		
Description:	Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1' the				

Q Cycle Activity:

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

**Example 1:** SUBWF REG, 1, 0

Before Instruction	
REG	= 3
W	= 2
C	= ?
After Instruction	
REG	= 1
W	= 2
C	= 1 ; result is positive
Z	= 0
N	= 0

**Example 2:** SUBWF REG, 0, 0

Before Instruction	
REG	= 2
W	= 2
C	= ?
After Instruction	
REG	= 2
W	= 0
C	= 1 ; result is zero
Z	= 1
N	= 0

**Example 3:** SUBWF REG, 1, 0

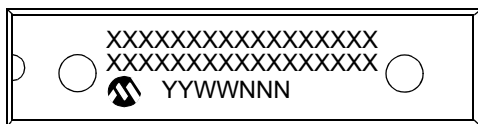
Before Instruction	
REG	= 1
W	= 2
C	= ?
After Instruction	
REG	= FFh ; (2's complement)
W	= 2
C	= 0 ; result is negative
Z	= 0
N	= 1

# PIC18F2221/2321/4221/4321 FAMILY

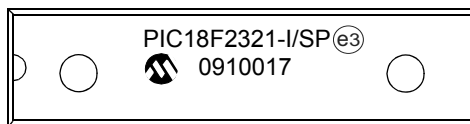
## 28.0 PACKAGING INFORMATION

### 28.1 Package Marking Information

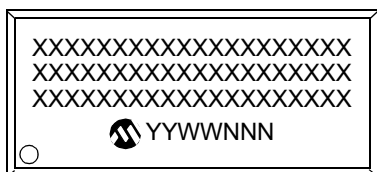
28-Lead SPDIP



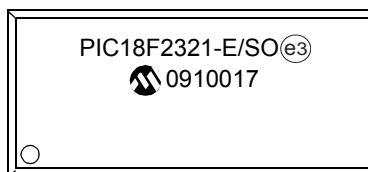
Example



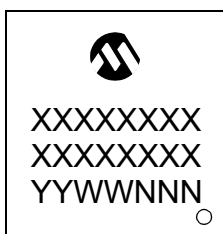
28-Lead SOIC



Example



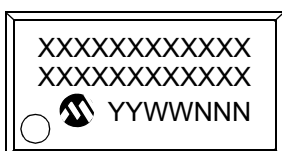
28-Lead QFN



Example



28-Lead SSOP



Example



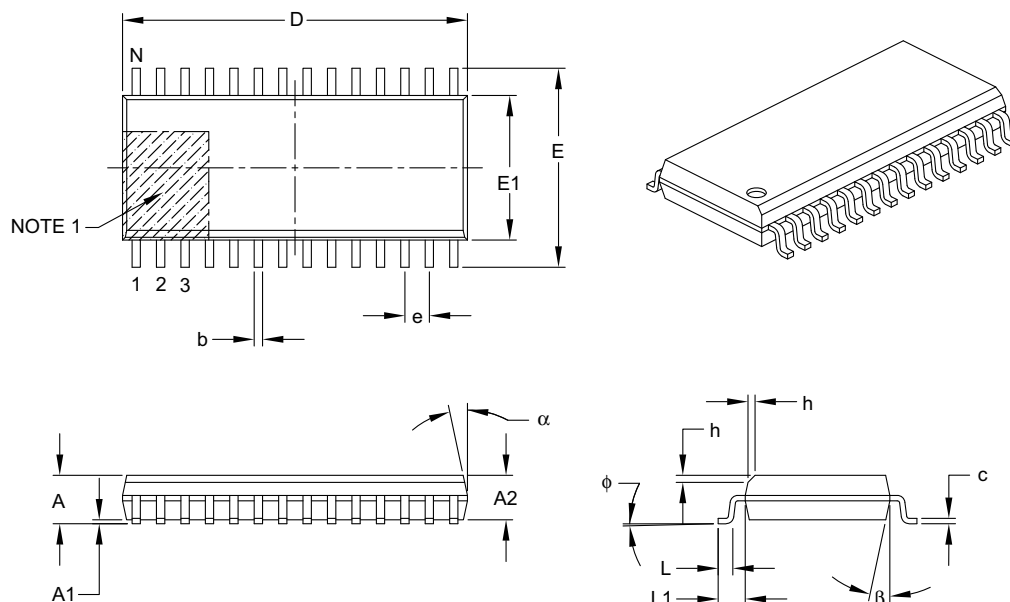
<b>Legend:</b>	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

# PIC18F2221/2321/4221/4321 FAMILY

## 28-Lead Plastic Small Outline (SO) – Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packages>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	1.27 BSC		
Overall Height	A	–	–	2.65
Molded Package Thickness	A2	2.05	–	–
Standoff §	A1	0.10	–	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	17.90 BSC		
Chamfer (optional)	h	0.25	–	0.75
Foot Length	L	0.40	–	1.27
Footprint	L1	1.40 REF		
Foot Angle Top	φ	0°	–	8°
Lead Thickness	c	0.18	–	0.33
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-052B