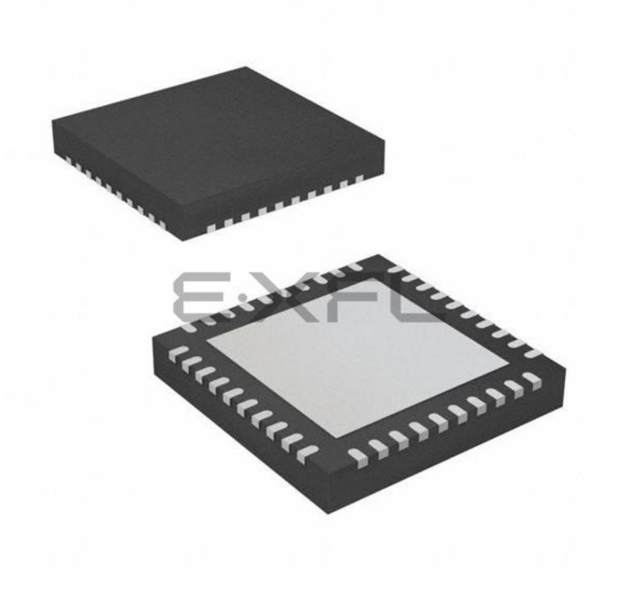


What is "Embedded - Microcontrollers"?

"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, LCD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 31x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UFQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f19175-e-mv

TABLE 5: 48-PIN ALLOCATION TABLE (PIC16(L)F19185/86)

I/O[2]	48-Pin TQFP/QFN	ADC	Reference	Comparator	Zero-Cross Detect	DAC	Timers/SMT	CCP	PWM	CWG	MSSP	EUSART	CLC	RTCC	LCD	Interrupt-on-Change	High Current	Pull-up	Basic
RA0	21	ANA0	—	C1IN0- C2IN0-	—	—	—	—	—	—	—	—	CLCIN0[1]	—	SEG0	IOCA0	—	Y	—
RA1	22	ANA1	—	C1IN1- C2IN1-	—	—	—	—	—	—	—	—	CLCIN1[1]	—	SEG1	IOCA1	—	Y	—
RA2	23	ANA2	—	C1IN0+ C2IN0+	—	DAC1OUT1	—	—	—	—	—	—	—	—	SEG2	IOCA2	—	Y	—
RA3	24	ANA3	VREF+	C1IN1+	—	DAC1REF+	—	—	—	—	—	—	—	—	SEG3	IOCA3	—	Y	—
RA4	25	ANA4	—	—	—	—	T0CKI[1]	—	—	—	—	—	—	—	SEG4 COM3	IOCA4	—	Y	—
RA5	26	—	—	—	—	—	—	—	—	—	$\overline{SS}$[1]	—	—	—	—	IOCA5	—	Y	VBAT
RA6	33	ANA6	—	—	—	—	—	—	—	—	—	—	—	—	SEG6	IOCA6	—	Y	CLKOUT OSC2
RA7	32	ANA7	—	—	—	—	—	—	—	—	—	—	—	—	SEG7	IOCA7	—	Y	OSC1 CLKIN
RB0	8	ANB0	—	C2IN1+	ZCD	—	—	—	—	CWG1IN[1]	—	—	—	—	SEG8	IOCB0	—	Y	INTPPS
RB1	9	ANB1	—	C1IN3- C2IN3-	—	—	—	—	—	—	SCL, SDA[1, 3, 4, 5, 6]	—	—	—	SEG9	IOCB1	HIB1	Y	—
RB2	10	ANB2	—	—	—	—	—	—	—	—	SCL, SDA[1, 3, 4, 5, 6]	—	—	—	SEG10 CFLY1	IOCB2	—	Y	—
RB3	11	ANB3	—	C1IN2- C2IN2-	—	—	—	—	—	—	—	—	—	—	SEG11 CFLY2	IOCB3	—	Y	—
RB4	16	ANB4 ADCACT[1]	—	—	—	—	—	—	—	—	—	—	—	—	COM0	IOCB4	—	Y	—
RB5	17	ANB5	—	—	—	—	T1G[1]	—	—	—	—	—	—	—	SEG13 COM1	IOCB5	—	Y	—
RB6	18	ANB6	—	—	—	—	—	—	—	—	—	TX2[1] CK2[1]	CLCIN2[1]	—	SEG14	IOCB6	—	Y	ICDCLK/ ICSPCLK
RB7	19	ANB7	—	—	—	DAC1OUT2	—	—	—	—	—	RX2[1] DT2[1]	CLCIN3[1]	—	SEG15	IOCB7	—	Y	ICDDAT/ ICSPDAT

Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.
2: All digital output signals shown in this row are PPS remappable. These signals may be mapped to output onto one or more PORTx pin options.
3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.
4: These pins are configured for I^{2}C logic levels. PPS assignments to the other pins will operate, but input logic levels will be standard TTL/ST as selected by INLCVL register, instead of the I^{2}C specific or SMBUS input buffer thresholds.
5: These are alternative I^{2}C logic levels pins.
6: In I^{2}C logic levels configuration, these pins can operate as either SCL and SDA pins.

4.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- Program Memory
  - Configuration Words
  - Device ID
  - User ID
  - Program Flash Memory
  - Device Information Area (DIA)
  - Device Configuration Information (DCI)
  - Revision ID
- Data Memory
  - Core Registers
  - Special Function Registers
  - General Purpose RAM (GPR)
  - Common RAM
- Data EEPROM

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing
- NVMREG access

4.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing 32K x 14 program memory space. Table 4-1 shows the memory sizes implemented. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 4-1).

TABLE 4-1: DEVICE SIZES AND ADDRESSES

Device	Program Flash Memory Size (Words)	Last Program Memory Address
PIC16(L)F19155	8k	1FFFh
PIC16(L)F19175	8k	1FFFh
PIC16(L)F19185	8k	1FFFh
PIC16(L)F19156	16k	3FFFh
PIC16(L)F19176	16k	3FFFh
PIC16(L)F19186	16k	3FFFh

4.2 Memory Access Partition (MAP)

User Flash is partitioned into:

- Application Block
- Boot Block, and
- Storage Area Flash (SAF) Block

The user can allocate the memory usage by setting the $\overline{\text{BBEN}}$ bit, selecting the size of the partition defined by BBSIZE<2:0> bits and enabling the Storage Area Flash by the $\overline{\text{SAFEN}}$ bit of the Configuration Word (see Register 5-4). Refer to Table 4-2 for the different user Flash memory partitions.

4.2.1 APPLICATION BLOCK

Default settings of the Configuration bits ($\overline{\text{BBEN}}$ = 1 and $\overline{\text{SAFEN}}$ = 1) assign all memory in the user Flash area to the Application Block.

4.2.2 BOOT BLOCK

If $\overline{\text{BBEN}}$ = 1, the Boot Block is enabled and a specific address range is alloted as the Boot Block based on the value of the BBSIZE bits of Configuration Word (Register 5-4) and the sizes provided in Table 5-1.

4.2.3 STORAGE AREA FLASH

Storage Area Flash (SAF) is enabled by clearing the $\overline{\text{SAFEN}}$ bit of the Configuration Word in Register 5-4. If enabled, the SAF block is placed at the end of memory and spans 128 words. If the Storage Area Flash (SAF) is enabled, the SAF area is not available for program execution.

4.2.4 MEMORY WRITE PROTECTION

All the memory blocks have corresponding write protection fuses WRTAPP and WRTB bits in the Configuration Word (Register 5-4). If write-protected locations are written from NVMCON registers, memory is not changed and the WRERR bit defined in Register 12-5 is set as explained in **Section 13.4.9 "WRERR Bit"**.

4.2.5 MEMORY VIOLATION

A Memory Execution Violation Reset occurs while executing an instruction that has been fetched from outside a valid execution area, clearing the $\overline{\text{MEMV}}$ bit. Refer to **Section 8.13 "Memory Execution Violation"** for the available valid program execution areas and the PCON1 register definition (Register 8-3) for $\overline{\text{MEMV}}$ bit conditions.

TABLE 4-2: MEMORY ACCESS PARTITION

REG	Address	Partition			
		$\overline{\text{BBEN}}$ = 1 SAFEN = 1	$\overline{\text{BBEN}}$ = 1 SAFEN = 0	$\overline{\text{BBEN}}$ = 0 SAFEN = 1	$\overline{\text{BBEN}}$ = 0 SAFEN = 0
PFM	00 0000h • • • Last Boot Block Memory Address	APPLICATION BLOCK (**Note 4**)	APPLICATION BLOCK (**Note 4**)	BOOT BLOCK (**Note 4**)	BOOT BLOCK (**Note 4**)
	Last Boot Block Memory Address + 1(1) • • • Last Program Memory Address - 80h			APPLICATION BLOCK (**Note 4**)	APPLICATION BLOCK (**Note 4**)
	Last Program Memory Address - 7Fh(2) • • • Last Program Memory Address		SAF (**Note 4**)		SAF (**Note 4**)
CONF IG	Config Memory Address(3)	CONFIG (**Note 4**)			

Note 1: Last Boot Block Memory Address is based on BBSIZE[2:0] given in Table 5-1.
2: Last Program Memory Address is the Flash size given in Table 4-1.
3: Config Memory Address are the address locations of the Configuration Words given in Table 13-2.
4: Each memory block has a corresponding write protection fuse defined by the $\overline{\text{WRTAPP}}$, $\overline{\text{WRTB}}$ and $\overline{\text{WRTSAF}}$ bits in the Configuration Word (Register 5-4).

TABLE 4-10: PIC16(L)F19155/56/75/76/85/86 MEMORY MAP, BANKS 58-63

	Bank 58		Bank 59		Bank 60		Bank 61		Bank 62		Bank 63
1D00h	Core Registers (Table 4-3)	1D80h	Core Registers (Table 4-3)	1E00h	Core Registers (Table 4-3)	1E80h	Core Registers (Table 4-3)	1F00h	Core Registers (Table 4-3)	1F80h	Core Registers (Table 4-3)
1D0Ch	LCDCON	1D8Ch	—	1E0Ch	—	1E8Ch	—	1F0Ch	—	1F8Ch	—
1D0Dh	LCDPS	1D8Dh	—	1E0Dh	—	1E8Dh	—	1F0Dh	—	1F8Dh	—
1D0Eh	LCDSE0	1D8Eh	—	1E0Eh	—	1E8Eh	—	1F0Eh	—	1F8Eh	—
1D0Fh	LCDSE1	1D8Fh	—	1E0Fh	CLCDATA	1E8Fh	PPSLOCK	1F0Fh	—	1F8Fh	—
1D10h	LCDSE2	1D90h	—	1E10h	CLC1CON	1E90h	INTPPS	1F10h	RA0PPS	1F90h	—
1D11h	LCDSE3	1D91h	—	1E11h	CLC1POL	1E91h	T0CKIPPS	1F11h	RA1PPS	1F91h	—
1D12h	LCDSE4	1D92h	—	1E12h	CLC1SEL0	1E92h	T1CKIPPS	1F12h	RA2PPS	1F92h	—
1D13h	LCDSE5	1D93h	—	1E13h	CLC1SEL1	1E93h	T1GPPS	1F13h	RA3PPS	1F93h	—
1D14h	LCDVCON1	1D94h	—	1E14h	CLC1SEL2	1E94h	—	1F14h	RA4PPS	1F94h	—
1D15h	LCDVCON2	1D95h	—	1E15h	CLC1SEL3	1E95h	—	1F15h	RA5PPS	1F95h	—
1D16h	LCDREF	1D96h	—	1E16h	CLC1GLS0	1E96h	—	1F16h	RA6PPS	1F96h	—
1D17h	LCDRL	1D97h	—	1E17h	CLC1GLS1	1E97h	—	1F17h	RA7PPS	1F97h	—
1D18h	LCDDATA0	1D98h	—	1E18h	CLC1GLS2	1E98h	—	1F18h	RB0PPS	1F98h	—
1D19h	LCDDATA1	1D99h	—	1E19h	CLC1GLS3	1E99h	—	1F19h	RB1PPS	1F99h	—
1D1Ah	LCDDATA2	1D9Ah	—	1E1Ah	CLC2CON	1E9Ah	—	1F1Ah	RB2PPS	1F9Ah	—
1D1Bh	LCDDATA3	1D9Bh	—	1E1Bh	CLC2POL	1E9Bh	—	1F1Bh	RB3PPS	1F9Bh	—
1D1Ch	LCDDATA4	1D9Ch	—	1E1Ch	CLC2SEL0	1E9Ch	T2AINPPS	1F1Ch	RB4PPS	1F9Ch	—
1D1Dh	LCDDATA5	1D9Dh	—	1E1Dh	CLC2SEL1	1E9Dh	T4AINPPS	1F1Dh	RB5PPS	1F9Dh	—
1D1Eh	LCDDATA6	1D9Eh	—	1E1Eh	CLC2SEL2	1E9Eh	—	1F1Eh	RB6PPS	1F9Eh	—
1D1Fh	LCDDATA7	1D9Fh	—	1E1Fh	CLC2SEL3	1E9Fh	—	1F1Fh	RB7PPS	1F9Fh	—
1D20h	LCDDATA8	1DA0h	—	1E20h	CLC2GLS0	1EA0h	—	1F20h	RC0PPS	1FA0h	—
1D21h	LCDDATA9	1DA1h	—	1E21h	CLC2GLS1	1EA1h	CCP1PPS	1F21h	RC1PPS	1FA1h	—
1D22h	LCDDATA10	1DA2h	—	1E22h	CLC2GLS2	1EA2h	CCP2PPS	1F22h	RC2PPS	1FA2h	—
1D23h	LCDDATA11	1DA3h	—	1E23h	CLC2GLS3	1EA3h	—	1F23h	RC3PPS	1FA3h	—
1D24h	LCDDATA12	1DA4h	—	1E24h	CLC3CON	1EA4h	—	1F24h	RC4PPS	1FA4h	—
1D25h	LCDDATA13	1DA5h	—	1E25h	CLC3POL	1EA5h	—	1F25h	—	1FA5h	—
1D26h	LCDDATA14	1DA6h	—	1E26h	CLC3SEL0	1EA6h	—	1F26h	RC6PPS	1FA6h	—
1D27h	LCDDATA15	1DA7h	—	1E27h	CLC3SEL1	1EA7h	—	1F27h	RC7PPS	1FA7h	—
1D28h	LCDDATA16	1DA8h	—	1E28h	CLC3SEL2	1EA8h	—	1F28h	RD0PPS(2)	1FA8h	—
1D29h	LCDDATA17	1DA9h	—	1E29h	CLC3SEL3	1EA9h	SMT1WINPPS	1F29h	RD1PPS(2)	1FA9h	—
1D2Ah	LCDDATA18	1DAAh	—	1E2Ah	CLC3GLS0	1EAAh	SMT1SIGPPS	1F2Ah	RD2PPS(2)	1FAAh	—
1D2Bh	LCDDATA19	1DABh	—	1E2Bh	CLC3GLS1	1EABh	—	1F2Bh	RD3PPS(2)	1FABh	—
1D2Ch	LCDDATA20	1DACh	—	1E2Ch	CLC3GLS2	1EACh	—	1F2Ch	RD4PPS(2)	1FACh	—
1D2Dh	LCDDATA21	1DADh	—	1E2Dh	CLC3GLS3	1EADh	—	1F2Dh	RD5PPS(2)	1FADh	—
1D2Eh	LCDDATA22	1DAEh	—	1E2Eh	CLC4CON	1EAEh	—	1F2Eh	RD6PPS(2)	1FAEh	—
1D2Fh	LCDDATA23	1DAFh	—	1E2Fh	CLC4POL	1EAFh	—	1F2Fh	RD7PPS(2)	1FAFh	—
1D30h	LCDDATA24	1DB0h	—	1E30h	CLC4SEL0	1EB0h	—	1F30h	RE0PPS(2)	1FB0h	—
1D31h	LCDDATA25	1DB1h	—	1E31h	CLC4SEL1	1EB1h	CWG1PPS	1F31h	RE1PPS(2)	1FB1h	—
1D32h	LCDDATA26	1DB2h	—	1E32h	CLC4SEL2	1EB2h	—	1F32h	RE2PPS(2)	1FB2h	—
1D33h	LCDDATA27	1DB3h	—	1E33h	CLC4SEL3	1EB3h	—	1F33h	RE3PPS	1FB3h	—
1D34h	LCDDATA28	1DB4h	—	1E34h	CLC4GLS0	1EB4h	—	1F34h	RE4PPS	1FB4h	—
1D35h	LCDDATA29	1DB5h	—	1E35h	CLC4GLS1	1EB5h	—	1F35h	RE5PPS	1FB5h	—
1D36h	LCDDATA30	1DB6h	—	1E36h	CLC4GLS2	1EB6h	—	1F36h	RE6PPS	1FB6h	—
1D37h	LCDDATA31	1DB7h	—	1E37h	CLC4GLS3	1EB7h	—	1F37h	RE7PPS	1FB7h	—
1D38h	LCDDATA32	1DB8h	—	1E38h	RF0PPS(3)	1EB8h	—	1F38h	ANSELA	1FB8h	—
1D39h	LCDDATA33	1DB9h	—	1E39h	RF1PPS(3)	1EB9h	—	1F39h	WPUA	1FB9h	—
1D3Ah	LCDDATA34	1DBAh	—	1E3Ah	RF2PPS(3)	1EBAh	—	1F3Ah	ODCONA	1FBAh	—

Note 1: Unimplemented locations read as '0'.
2: Present only on PIC16(L)F19156/76/86.
3: Present only on PIC16(L)F19185/86.

REGISTER 5-7: REVISIONID: REVISION ID REGISTER

R	R	R	R	R	R	R	R	R	R	R	R	R	R
1	0	MJRREV<5:0>						MNRREV<5:0>					
bit 13													bit 0

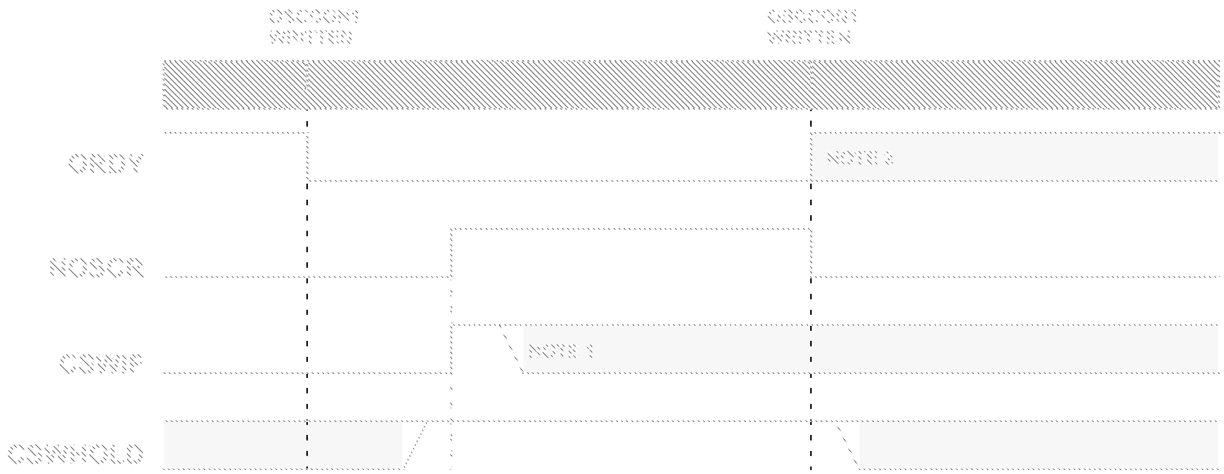
Legend:		
R = Readable bit		
‘0’ = Bit is cleared	‘1’ = Bit is set	x = Bit is unknown

bit 13-12 **Fixed Value**: Read-only bits
These bits are fixed with value ‘10’ for all devices included in this data sheet.

bit 11-6 **MJRREV<5:0>**: Major Revision ID bits
These bits are used to identify a major revision. A major revision is indicated by an all layer revision (B0, C0, etc.)

bit 5-0 **MNRREV<5:0>**: Minor Revision ID bits
These bits are used to identify a minor revision.

FIGURE 9-6: CLOCK SWITCH ABANDONED



Note 1: CSWIF may be cleared before or after rewriting OSCCON1; CSWIF is not automatically cleared.
2: ORDY = `0` if OSCCON1 does not match OSCCON2; a new switch will begin.

REGISTER 12-3: WDTPSL: WDT PRESCALE SELECT LOW BYTE REGISTER

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
PSCNT<7:0>(1)							
bit 7							bit 0

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

bit 7-0 **PSCNT<7:0>**: Prescale Select Low Byte bits(1)

Note 1: The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should be read during normal operation.

REGISTER 12-4: WDTPSH: WDT PRESCALE SELECT HIGH BYTE REGISTER

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
PSCNT<15:8>(1)							
bit 7							bit 0

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

bit 7-0 **PSCNT<15:8>**: Prescale Select High Byte bits(1)

Note 1: The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should be read during normal operation.

REGISTER 12-5: WDTTMR: WDT TIMER REGISTER

U-0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
—	WDTTMR<3:0>				STATE	PSCNT<17:16>(1)	
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **WDTTMR<3:0>**: Watchdog Timer Value bits

bit 2 **STATE**: WDT Armed Status bit
1 = WDT is armed
0 = WDT is not armed

bit 1-0 **PSCNT<17:16>**: Prescale Select Upper Byte bits(1)

Note 1: The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should be read during normal operation.

17.0 INTERRUPT-ON-CHANGE (IOC)

All pins on all PORTA, PORTB, and PORTC, excluding RC5 and RE3 of PORTE, can be configured to operate as Interrupt-On-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual pin, or combination of pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 17-1 is a block diagram of the IOC module.

17.1 Enabling the Module

To allow individual pins to generate an interrupt, the IOCIE bit of the PIE0 register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

17.2 Individual Pin Configuration

For each pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting the associated bits in both of the IOCxP and IOCxN registers.

17.3 Interrupt Flags

The bits located in the IOCxF registers are status flags that correspond to the interrupt-on-change pins of each port. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the PIR0 register reflects the status of all IOCxF bits.

17.4 Clearing Interrupt Flags

The individual status flags, (IOCxF register bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

EXAMPLE 17-1: CLEARING INTERRUPT FLAGS (PORTB EXAMPLE)

```
MOVLW  0xff
XORWF  IOCBF, W
ANDWF  IOCBF, F
```

17.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the affected IOCxF register will be updated prior to the first instruction executed out of Sleep.

24.1 OPERATION

The RTCC consists of a 100-year clock and calendar with automatic leap year detection. The range of the clock is from 00:00:00 (midnight) on January 1st, 2000 to 23:59:59 on December 31st, 2099.

The hours use the 24-hour time format (military time) with no hardware provisions for regular time format (AM/PM). The clock provides a granularity of one second with additional visibility to the half-second.

The user has visibility to the half second field of the counter. This value is read-only and can be reset only by writing to the lower half of the SECONDS register.

24.1.1 REGISTER INTERFACE

The RTCC register set is divided into the following categories:

Control Registers

- RTCCON
- RTCCAL
- ALRMCON
- ALRMRPT

Clock Value Registers

- YEAR
- MONTH
- DAY
- WEEKDAY
- HOURS
- MINUTES
- SECONDS

Alarm Value Registers

- ALRMMNTH
- ALRMDAY
- ALRMWD
- ALRMHR
- ALRMMIN
- ALRMSEC

Note:	The WEEKDAY register is not automatically derived from the date, but it must be correctly set by the user.

The register interface for the RTCC and alarm values is implemented using the Binary Coded Decimal (BCD) format. This simplifies the firmware when using the module, as each of the digits is contained within its own 4-bit value (see **Section 24.1.3 “Clock Sources”** and Figure 24-3).

All timer registers containing a value of seconds or greater are writable. The user can configure the initial start date and time by writing the year, month, day, hour, minutes and seconds into the clock value registers and the timer will then proceed to count from the newly written values.

The RTCC module is enabled by setting the RTCEN bit (RTCCON<7>). Once the RTCC is enabled, the timer will continue incrementing, even while the clock value registers are being re-written. However, any time the SECONDS register is written to, all of the clock value prescalers are reset to ‘0’. This allows lower granularity of timer adjustments.

The Timer registers are updated in the same cycle as the `WRITE` instruction’s execution by the CPU. The user must ensure that when RTCEN = 1, the updated registers will not be incremented at the same time. This can be accomplished in several ways:

- By checking the RTCSYNC bit (RTCCON<4>)
- By checking the preceding digits from which a carry can occur
- By updating the registers immediately following the seconds pulse (or alarm interrupt)

24.1.2 WRITE LOCK

To perform a write to any of the RTCC timer registers, the RTCWREN bit must be set. To avoid accidental writes to the timer, it is recommended that the RTCWREN bit is kept clear at any other time.

The RTCEN bit can only be written to when RTCWREN = 1. A write attempt to this bit while RTCWREN = 0 will be ignored. The RTCC timer registers can be written with RTCEN = 0 or 1.

26.12 Register Definitions: Timer1 Control

REGISTER 26-1: T1CON: TIMER1 CONTROL REGISTER

U-0	U-0	R/W-0/u	R/W-0/u	U-0	R/W-0/u	R/W-0/u	R/W-0/u
—	—	CKPS<1:0>		—	$\overline{\text{SYNC}}$	RD16	ON
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **Unimplemented:** Read as '0'

bit 2 **$\overline{\text{SYNC}}$:** Timer1 Synchronization Control bit
When TMR1CLK = FOSC or FOSC/4
This bit is ignored. The timer uses the internal clock and no additional synchronization is performed.
ELSE
0 = Synchronize external clock input with system clock
1 = Do not synchronize external clock input

bit 1 **RD16:** 16-bit Read/Write Mode Enable bit
0 = Enables register read/write of Timer1 in two 8-bit operation
1 = Enables register read/write of Timer1 in one 16-bit operation

bit 0 **ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1 and clears Timer1 gate flip-flop

29.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral that allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

The Capture/Compare/PWM modules available are shown in Table 29-1.

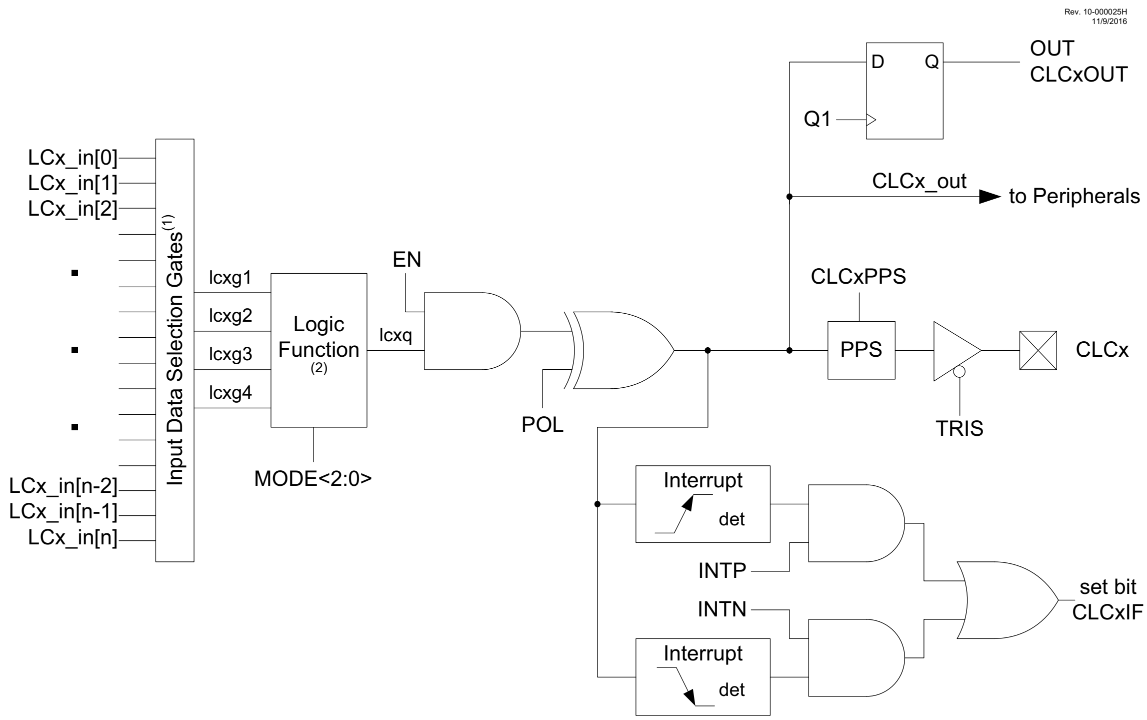
TABLE 29-1: AVAILABLE CCP MODULES

Device	CCP1	CCP2
PIC16(L)F19155/56/75/76/85/86	●	●

The Capture and Compare functions are identical for all CCP modules.

> **Note 1:** In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.
>
> **2:** Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator ‘x’ to indicate the use of a numeral to distinguish a particular module, when required.

FIGURE 32-1: CLCx SIMPLIFIED BLOCK DIAGRAM



Note 1: See Figure 32-2: Input Data Selection and Gating

2: See Figure 32-3: Programmable Logic Functions

33.3.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of clock stretching. An addressed slave device may hold the SCL clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCL line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCL connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

33.3.2 ARBITRATION

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an Idle state.

However, two master devices may try to initiate a transmission on or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDA data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels do not match, loses arbitration, and must stop transmitting on the SDA line.

For example, if one transmitter holds the SDA line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDA line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDA line. If this transmitter is also a master device, it also must stop driving the SCL line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDA line continues with its original transmission.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

33.4 I^{2}C MODE OPERATION

All MSSP I^{2}C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC® microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I^{2}C devices.

33.4.1 BYTE FORMAT

All communication in I^{2}C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

33.4.2 DEFINITION OF I^{2}C TERMINOLOGY

There is language and terminology in the description of I^{2}C communication that have definitions specific to I^{2}C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I^{2}C specification.

33.4.3 SDA AND SCL PINS

When selecting any I^{2}C mode, the SCL and SDA pins should be set by the user to inputs by setting the appropriate TRIS bits.

Note 1:	Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPDATPPS registers. The SCL input is selected with the SSPCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user’s responsibility to make the selections so that both the input and the output for each function is on the same pin.

REGISTER 33-5: SSPxMSK: SSPx MASK REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SSPxMSK<7:0>							
bit 7							bit 0

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

bit 7-1 **SSPxMSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPxADD<n> to detect I^{2}C address match
0 = The received address bit n is not used to detect I^{2}C address match

bit 0 **SSPxMSK<0>:** Mask bit for I^{2}C Slave mode, 10-bit Address
I^{2}C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
1 = The received address bit 0 is compared to SSPxADD<0> to detect I^{2}C address match
0 = The received address bit 0 is not used to detect I^{2}C address match
I^{2}C Slave mode, 7-bit address:
MSK0 bit is ignored.

REGISTER 33-6: SSPxADD: MSSPx ADDRESS AND BAUD RATE REGISTER (I^{2}C MODE)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SSPxADD<7:0>							
bit 7							bit 0

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

Master mode:

bit 7-0 **SSPxADD<7:0>:** Baud Rate Clock Divider bits
SCL pin clock period = ((ADD<7:0> + 1) *4)/FOSC

10-Bit Slave mode – Most Significant Address Byte:

bit 7-3 **Not used:** Unused for Most Significant Address Byte. Bit state of this register is a “don’t care”. Bit pattern sent by master is fixed by I^{2}C specification and must be equal to ‘11110’. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **SSPxADD<2:1>:** Two Most Significant bits of 10-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a “don’t care”.

10-Bit Slave mode – Least Significant Address Byte:

bit 7-0 **SSPxADD<7:0>:** Eight Least Significant bits of 10-bit address

7-Bit Slave mode:

bit 7-1 **SSPxADD<7:1>:** 7-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a “don’t care”.

REGISTER 34-3: BAUDxCON: BAUD RATE CONTROL REGISTER

R/W-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

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

bit 7 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don’t care

bit 6 **RCIDL:** Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is Idle
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don’t care

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **SCKP:** Clock/Transmit Polarity Select bit
Asynchronous mode:
1 = Idle state for transmit (TX) is a low level
0 = Idle state for transmit (TX) is a high level
Synchronous mode:
1 = Idle state for clock (CK) is a high level
0 = Idle state for clock (CK) is a low level

bit 3 **BRG16:** 16-bit Baud Rate Generator bit
1 = 16-bit Baud Rate Generator is used
0 = 8-bit Baud Rate Generator is used

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = USART will continue to sample the Rx pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge.
0 = RX pin not monitored nor rising edge detected
Synchronous mode:
Unused in this mode – value ignored

bit 0 **ABDEN:** Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Enable baud rate measurement on the next character – requires reception of a SYNCH field (55h);
cleared in hardware upon completion
0 = Baud rate measurement disabled or completed
Synchronous mode:
Unused in this mode – value ignored

REGISTER 35-7: LCDRL: LCD INTERNAL REFERENCE LADDER CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LRLAP1	LRLAP0	LRLBP1	LRLBP0	LCDIRI	LRLAT2	LRLAT1	LRLAT0
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 **LRLAP<1:0>**: LCD Reference Ladder A Time Power Control bits

During Time Interval A:
11 = Internal LCD reference is the High-Power (HP) ladder
10 = Internal LCD reference ladder is powered in Medium Power mode
01 = Internal LCD reference ladder is powered in Low-Power mode
00 = Internal LCD reference ladder is powered down and unconnected

bit 5-4 **LRLBP<1:0>**: LCD Reference Ladder B Time Power Control bits

During Time Interval B:
11 = Internal LCD reference ladder is powered in High-Power mode
10 = Internal LCD reference ladder is powered in Medium Power mode
01 = Internal LCD reference ladder is powered in Low-Power mode
00 = Internal LCD reference ladder is powered down and unconnected

bit 3 **LCDIRI:** LCD Internal Reference Buffer Idle Enable bit

Allows the Internal reference band gap buffer to shut down when the LCD Reference Ladder is in Power mode 'B'
1 = When the LCD Reference Ladder is in power mode 'B', the LCD Internal Reference Band Gap buffer is disabled
0 = The LCD Internal Reference Buffer ignores the LCD Reference Ladder power mode

bit 2-0 **LRLAT<2:0>**: LCD Reference Ladder A Time Interval Control bits

Sets the number of 32 clock counts when the A Time Interval Power mode is active.

For Type-A Waveforms (WFT = 0):
111 = Internal LCD reference ladder is in A Power mode for 7 clocks and B Power mode for 9 clocks
110 = Internal LCD reference ladder is in A Power mode for 6 clocks and B Power mode for 10 clocks
101 = Internal LCD reference ladder is in A Power mode for 5 clocks and B Power mode for 11 clocks
100 = Internal LCD reference ladder is in A Power mode for 4 clocks and B Power mode for 12 clocks
011 = Internal LCD reference ladder is in A Power mode for 3 clocks and B Power mode for 13 clocks
010 = Internal LCD reference ladder is in A Power mode for 2 clocks and B Power mode for 14 clocks
001 = Internal LCD reference ladder is in A Power mode for 1 clock and B Power mode for 15 clocks
000 = Internal LCD reference ladder is always in B Power mode

For Type-B Waveforms (WFT = 1):
111 = Internal LCD reference ladder is in A Power mode for 7 clocks and B Power mode for 25 clocks
110 = Internal LCD reference ladder is in A Power mode for 6 clocks and B Power mode for 26 clocks
101 = Internal LCD reference ladder is in A Power mode for 5 clocks and B Power mode for 27 clocks
100 = Internal LCD reference ladder is in A Power mode for 4 clocks and B Power mode for 28 clocks
011 = Internal LCD reference ladder is in A Power mode for 3 clocks and B Power mode for 29 clocks
010 = Internal LCD reference ladder is in A Power mode for 2 clocks and B Power mode for 30 clocks
001 = Internal LCD reference ladder is in A Power mode for 1 clock and B Power mode for 31 clocks
000 = Internal LCD reference ladder is always in B Power mode

LSLF — Logical Left Shift

Syntax:	[*label*] LSLF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>) → C (f<6:0>) → dest<7:1> 0 → dest<0>
Status Affected:	C, Z
Description:	The contents of register ‘f’ are shifted one bit to the left through the Carry flag. A ‘`0`’ is shifted into the LSb. If ‘d’ is ‘`0`’, the result is placed in W. If ‘d’ is ‘`1`’, the result is stored back in register ‘f’.

LSRF — Logical Right Shift

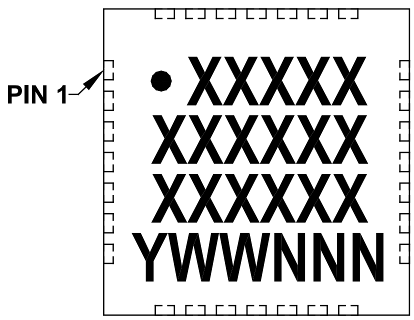
Syntax:	[*label*] LSRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	0 → dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register ‘f’ are shifted one bit to the right through the Carry flag. A ‘`0`’ is shifted into the MSb. If ‘d’ is ‘`0`’, the result is placed in W. If ‘d’ is ‘`1`’, the result is stored back in register ‘f’.

MOVF — Move f

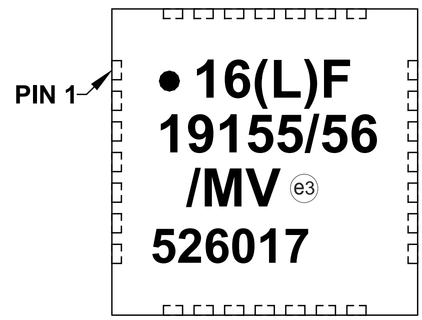
Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [`0`,`1`]
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register f is moved to a destination dependent upon the status of d. If d = `0`, destination is W register. If d = `1`, the destination is file register f itself. d = `1` is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
<u>Example:</u>	`MOVF FSR, 0`
	After Instruction W = value in FSR register Z = `1`

42.1 Package Marking Information (Continued)

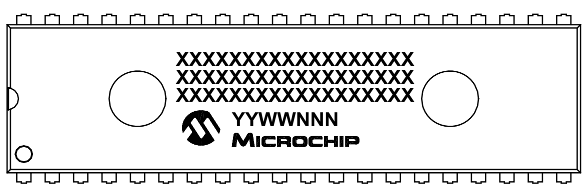
28-Lead UQFN (4x4x0.5 mm)



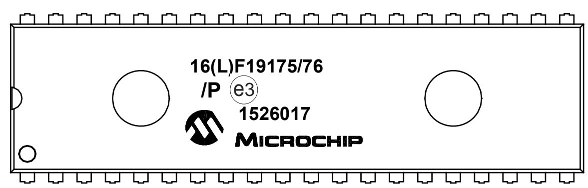
Example



40-Lead PDIP (600 mil)



Example

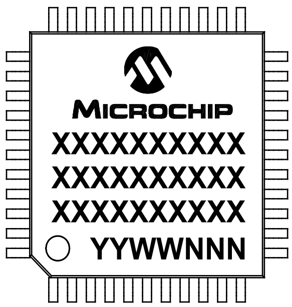


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free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

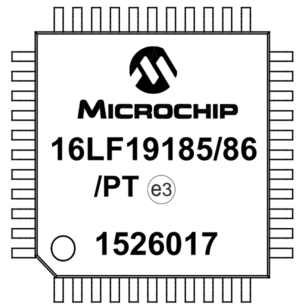
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

42.1 Package Marking Information (Continued)

48-Lead TQFP (7x7x1 mm)



Example

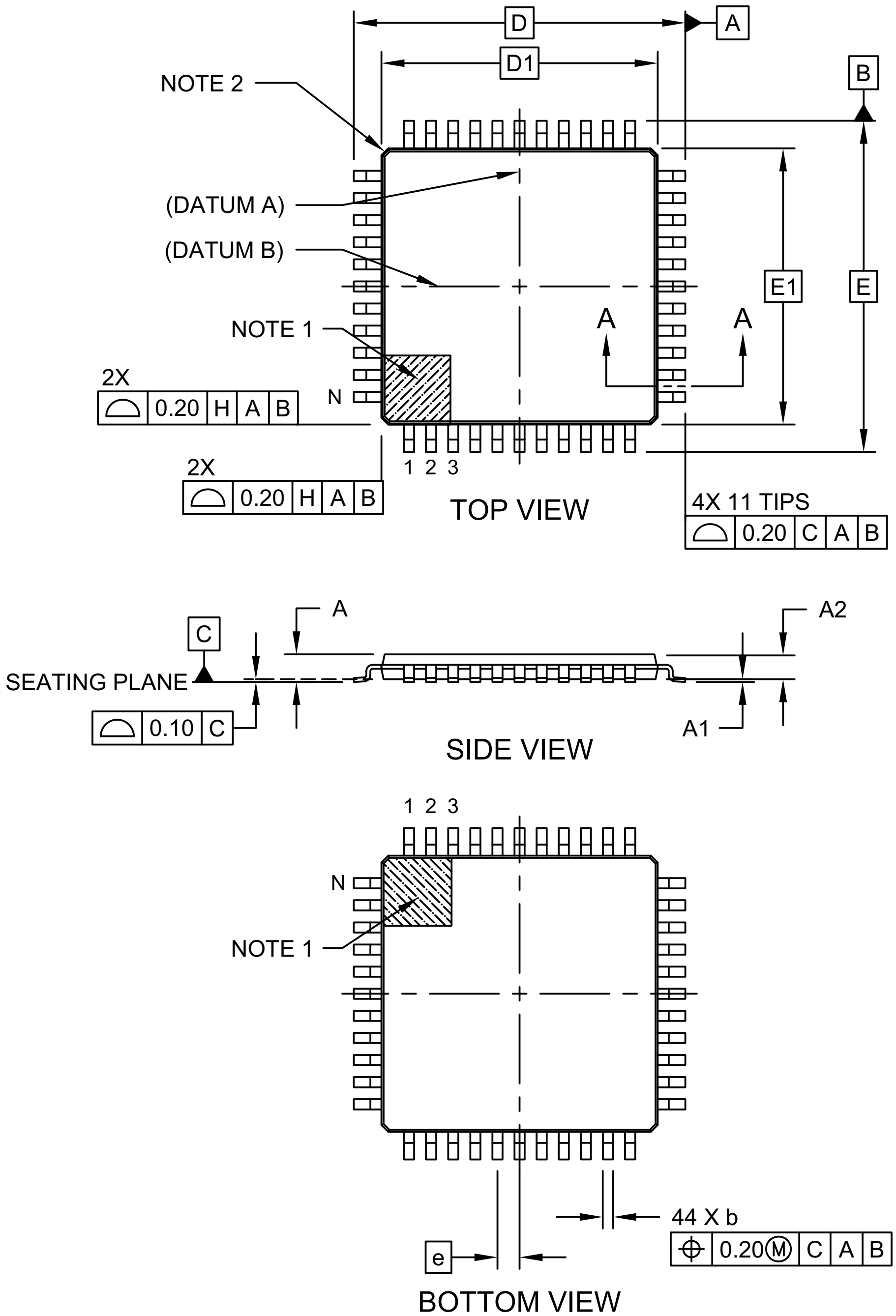


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The 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.

44-Lead Plastic Thin Quad Flatpack (PT) - 10x10x1.0 mm Body [TQFP]

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



Microchip Technology Drawing C04-076C Sheet 1 of 2