Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 17x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18446-e-ss

7. 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
  - Common RAM
- EEPROM

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

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

7.1 Program Memory Organization

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

Table 7-1. Device Sizes And Addresses

Device	Program Memory Size (Words)	Last Program Memory Address
PIC16(L)F18426	16384	0x3FFF
PIC16(L)F18446	16384	0x3FFF

7.9 Register Summary: Shadow Registers

Offset	Name	Bit Pos.								
0x1FE4	STATUS_SHAD	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x1FE5	WREG_SHAD	7:0	WREG[7:0]							
0x1FE6	BSR_SHAD	7:0			BSR[5:0]					
0x1FE7	PCLATH_SHAD	7:0		PCLATH[6:0]						
0x1FE8	FSR0_SHAD	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1FEA	FSR1_SHAD	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							

7.10 Register Definitions: Shadow Registers

10. Interrupts

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

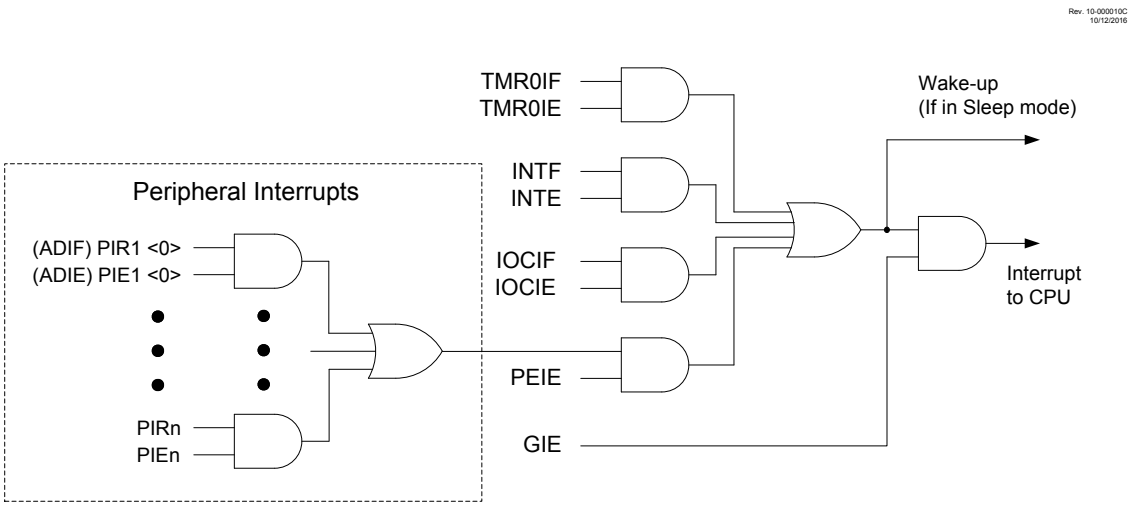
This chapter contains the following information for Interrupts:

- Operation
- Interrupt Latency
- Interrupts During Sleep
- INT Pin
- Automatic Context Saving

Many peripherals produce interrupts. Refer to the corresponding chapters for details.

A block diagram of the interrupt logic is shown below.

Figure 10-1. Interrupt Logic



10.1 Operation

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

- GIE bit of the INTCON register
- Interrupt Enable bit(s) of the PIEx[y] registers for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIEx registers)

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are 9 PIR registers.

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

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack

10.7.13 PIR2

Name: PIR2
Address: 0x70E

Peripheral Interrupt Request (Flag) Register 2

Bit	7	6	5	4	3	2	1	0
		ZCDIF					C2IF	C1IF
Access		R/W/HS					R/W/HS	R/W/HS
Reset		0					0	0

Bit 6 – ZCDIF Zero-Cross Detect Interrupt Flag bit

Value	Description
1	An enabled rising and/or falling ZCD1 event has been detected (must be cleared in software)
0	No ZCD1 event has occurred

Bits 0, 1 – CnIF Comparator ‘n’ Interrupt Flag bit

Value	Description
1	Comparator Cn interrupt asserted (must be cleared in software)
0	Comparator Cn interrupt not asserted

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON appropriate interrupt flag bits are clear prior to enabling an interrupt.

18.4.1 FVRCON

Name: FVRCON
Address: 0x90C

Fixed Voltage Reference Control Register

Bit	7	6	5	4	3	2	1	0
	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR[1:0]		ADFVR[1:0]	
Access	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	0

Bit 7 – FVREN Fixed Voltage Reference Enable bit

Value	Description
1	Fixed Voltage Reference is enabled
0	Fixed Voltage Reference is disabled

Bit 6 – FVRRDY Fixed Voltage Reference Ready Flag bit(1)
Reset States: POR/BOR = x
All Other Resets = q

Value	Description
1	Fixed Voltage Reference output is ready for use
0	Fixed Voltage Reference output is not ready or not enabled

Bit 5 – TSEN Temperature Indicator Enable bit(3)

Value	Description
1	Temperature Indicator is enabled
0	Temperature Indicator is disabled

Bit 4 – TSRNG Temperature Indicator Range Selection bit(3)

Value	Description
1	Temperature in High Range V_{OUT} = 3 V_T
0	Temperature in Low Range V_{OUT} = 2 V_T

Bits 3:2 – CDAFVR[1:0] Comparator FVR Buffer Gain Selection bits

Value	Description
11	Comparator FVR Buffer Gain is 4x, (4.096V)(2)
10	Comparator FVR Buffer Gain is 2x, (2.048V)(2)
01	Comparator FVR Buffer Gain is 1x, (1.024V)
00	Comparator FVR Buffer is off

Bits 1:0 – ADFVR[1:0] ADC FVR Buffer Gain Selection bit

Value	Description
11	ADC FVR Buffer Gain is 4x, (4.096V)(2)
10	ADC FVR Buffer Gain is 2x, (2.048V)(2)

20.8.3 ADCON2

Name: ADCON2
Address: 0x113

ADC Control Register 2

Bit	7	6	5	4	3	2	1	0
	PSIS	CRS[2:0]			ACLR	MD[2:0]		
Access	R/W	R/W	R/W	R/W	R/W/HC	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – PSIS ADC Previous Sample Input Select bits

Value	Description
1	FLTR is transferred to PREV at start-of-conversion
0	ADRES is transferred to PREV at start-of-conversion

Bits 6:4 – CRS[2:0] ADC Accumulated Calculation Right Shift Select bits

Value	Condition	Description
0 to 7	MD = b'100'	Low-pass filter time constant is 2^{CRS}, filter gain is 1:1
0 to 7	MD = b'011' to b'001'	The accumulated value is right-shifted by CRS (divided by 2^{CRS})**(1,2)**
x	MD = b'000' to b'001'	These bits are ignored

Bit 3 – ACLR A/D Accumulator Clear Command bit**(3)**

Value	Description
1	ACC, AOV and CNT registers are cleared
0	Clearing action is complete (or not started)

Bits 2:0 – MD[2:0] ADC Operating Mode Selection bits**(4)**

Value	Description
111-101	Reserved
100	Low-pass Filter mode
011	Burst Average mode
010	Average mode
001	Accumulate mode
000	Basic (Legacy) mode

Note:

1. To correctly calculate an average, the number of samples (set in RPT) must be 2^{CRS}.
2. CRS = `3'b111` is a reserved option.
3. This bit is cleared by hardware when the accumulator operation is complete; depending on oscillator selections, the delay may be many instructions.
4. See Computation Modes for Full mode descriptions.

20.8.16 ADPREV

Name: ADPREV
Address: 0x09B

ADC Previous Result Register

Bit	15	14	13	12	11	10	9	8
	PREVH[7:0]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	PREVL[7:0]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	x	x	x	x	x	x	x	x

Bits 15:8 – PREVH[7:0] Previous ADC Result Most Significant bits

Value	Condition	Description
0 to 0xFF	PSIS = 1	Upper byte of ADFLTR at the start of current ADC conversion
varies	PSIS = 0	Upper bits of ADRES at the start of current ADC conversion(1)

Bits 7:0 – PREVL[7:0] Previous ADC Result Least Significant bits

Value	Condition	Description
0 to 0xFF	PSIS = 1	Lower byte of ADFLTR at the start of current ADC conversion
varies	PSIS = 0	Lower bits of ADRES at the start of current ADC conversion(1)

Note: If PSIS = 0, PREVH and PREVL are formatted the same way as ADRES is, depending on the FRM bit.

26.13 Register Summary - Timer1

Offset	Name	Bit Pos.								
0x020C	TMR1	7:0	TMRxL[7:0]							
		15:8	TMRxH[7:0]							
0x020E	T1CON	7:0			CKPS[1:0]			SYNC	RD16	ON
0x020F	T1GCON	7:0	GE	GPOL	GTM	GSPM	GGO/DONE	GVAL		
0x0210	TMR1GATE	7:0				GSS[4:0]				
0x0211	TMR1CLK	7:0				CS[4:0]				
0x0212	TMR3	7:0	TMRxL[7:0]							
		15:8	TMRxH[7:0]							
0x0214	T3CON	7:0			CKPS[1:0]			SYNC	RD16	ON
0x0215	T3GCON	7:0	GE	GPOL	GTM	GSPM	GGO/DONE	GVAL		
0x0216	TMR3GATE	7:0				GSS[4:0]				
0x0217	TMR3CLK	7:0				CS[4:0]				
0x0218	TMR5	7:0	TMRxL[7:0]							
		15:8	TMRxH[7:0]							
0x021A	T5CON	7:0			CKPS[1:0]			SYNC	RD16	ON
0x021B	T5GCON	7:0	GE	GPOL	GTM	GSPM	GGO/DONE	GVAL		
0x021C	TMR5GATE	7:0				GSS[4:0]				
0x021D	TMR5CLK	7:0				CS[4:0]				

26.14 Register Definitions: Timer1

Long bit name prefixes for the odd numbered timers is shown in the following table. Refer to the *"Long Bit Names"* section for more information.

Table 26-5. Timer1 prefixes

Peripheral	Bit Name Prefix
Timer1	T1
Timer3	T3
Timer5	T5

Related Links

Long Bit Names

28.1 Register Summary - Timer Selection Registers for CCP/PWM

Offset	Name	Bit Pos.								
0x021E	CCPTMRS0	7:0	C4TSEL[1:0]		C3TSEL[1:0]		C2TSEL[1:0]		C1TSEL[1:0]	
0x021F	CCPTMRS1	7:0			P7TSEL[1:0]		P6TSEL[1:0]			

28.2 Register Definitions: CCP/PWM Timer Selection

6. Enable PWM output pin:
   – Wait until the Timer overflows and the TMR2IF bit of the PIRx register is set. See Note below.
   – Enable the CCPx pin output driver by clearing the associated TRIS bit.

Important: In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

Related Links

TxCON

29.4.3 Timer2 Timer Resource

The PWM standard mode makes use of the 8-bit Timer2 timer resources to specify the PWM period.

29.4.4 PWM Period

The PWM period is specified by the T2PR register of Timer2. The PWM period can be calculated using the formula in the equation below.

Equation 29-1. PWM Period

$$PWMPeriod = [(T2PR + 1)] \bullet 4 \bullet T_{OSC} \bullet \left(TMR2\mathrm{Pr}escaleValue\right)$$

where $T_{OSC} = 1/F_{OSC}$

When T2TMR is equal to T2PR, the following three events occur on the next increment cycle:

- T2TMR is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is transferred from the CCPRx register into a 10-bit buffer.

Important: The Timer postscaler (see *"Timer2 Interrupt"*) is not used in the determination of the PWM frequency.

Related Links

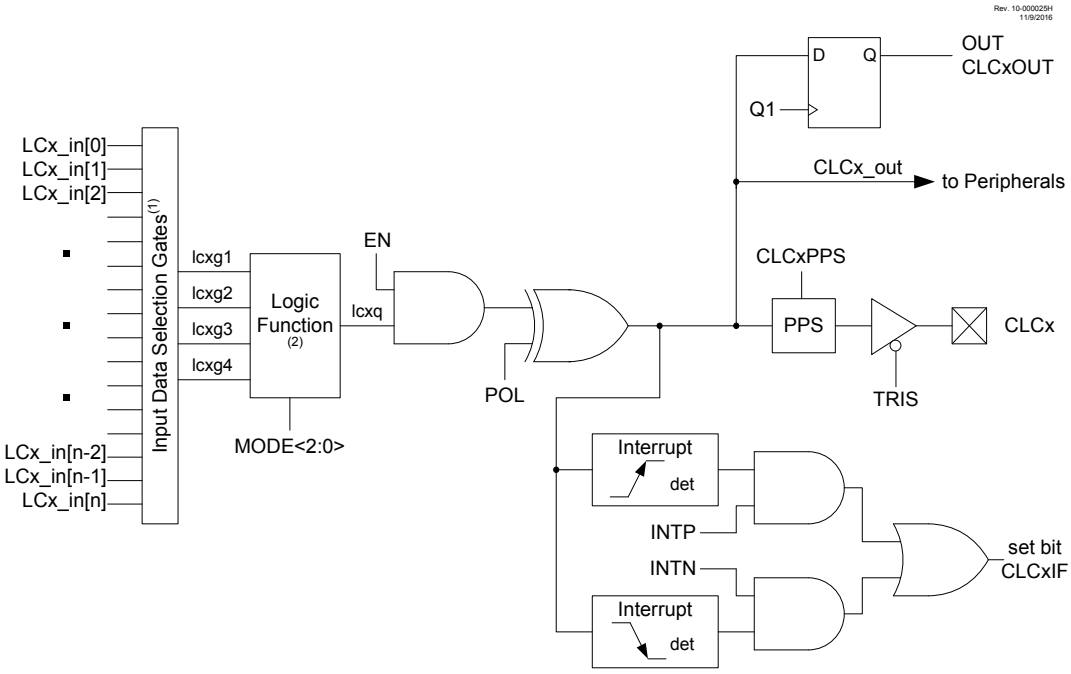
Timer2 Interrupt

29.4.5 PWM Duty Cycle

The PWM duty cycle is specified by writing a 10-bit value to the CCPRx register. The alignment of the 10-bit value is determined by the FMT bit (see Figure 29-6). The CCPRx register can be written to at any time. However, the duty cycle value is not latched into the 10-bit buffer until after a match between T2PR and T2TMR.

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

Figure 33-1. CLC Simplified Block Diagram



Note:

1. See Figure 33-2 for input data selection and gating.
2. See Figure 33-3 for programmable logic functions.

33.1 CLC Setup

Programming the CLC module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLC Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

33.1.1 Data Selection

There are 64 signals available as inputs to the configurable logic. Four 64-input multiplexers are used to select the inputs to pass on to the next stage.

Data selection is through four multiplexers as indicated on the left side of the following diagram. Data inputs in the figure are identified by a generic numbered input name.

33.8.1 CLCxCON

Name: CLCxCON
Address: 0x1E10,0x1E1A,0x1E24,0x1E2E

Configurable Logic Cell Control Register

Bit	7	6	5	4	3	2	1	0
	EN		OUT	INTP	INTN	MODE[2:0]		
Access	R/W		RO	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – EN
CLC Enable bit

Value	Description
1	Configurable logic cell is enabled and mixing signals
0	Configurable logic cell is disabled and has logic zero output

Bit 5 – OUT
Logic cell output data, after LCPOL. Sampled from CLCxOUT

Bit 4 – INTP
Configurable Logic Cell Positive Edge Going Interrupt Enable bit

Value	Description
1	CLCxIF will be set when a rising edge occurs on CLCxOUT
0	Rising edges on CLCxOUT have no effect on CLCxIF

Bit 3 – INTN
Configurable Logic Cell Negative Edge Going Interrupt Enable bit

Value	Description
1	CLCxIF will be set when a falling edge occurs on CLCxOUT
0	Falling edges on CLCxOUT have no effect on CLCxIF

Bits 2:0 – MODE[2:0]
Configurable Logic Cell Functional Mode Selection bits

Value	Description
111	Cell is 1-input transparent latch with Set and Reset
110	Cell is J-K flip-flop with Reset
101	Cell is 2-input D flip-flop with Reset
100	Cell is 1-input D flip-flop with Set and Reset
011	Cell is S-R latch
010	Cell is 4-input AND
001	Cell is OR-XOR
000	Cell is AND-OR

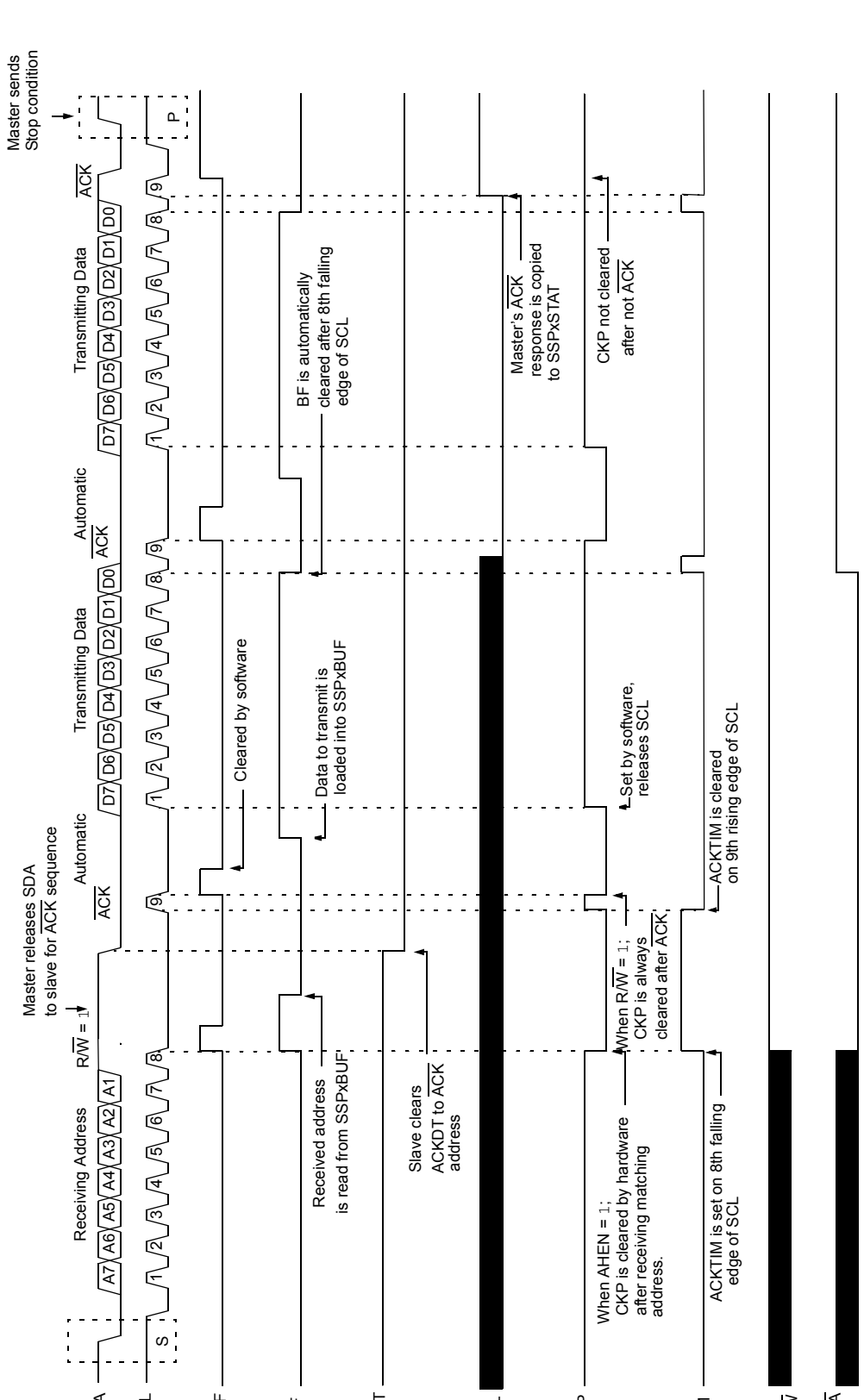


Figure 35-19. I²C Slave, 7-bit Address, Transmission (AHEN = 1)

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

36.3.1.6 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCxREG is read to access the FIFO. When this happens the OERR bit of the RCxSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCxREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

36.3.1.7 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCxREG.

36.3.1.8 Synchronous Master Reception Setup

1. Initialize the SPxBRGH:SPxBRGL register pair and set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Select the receive input pin by writing the appropriate values to the RxyPPS register and RXxPPS register. Both selections should enable the same pin.
3. Select the clock output pin by writing the appropriate values to the RxyPPS register and CKxPPS register. Both selections should enable the same pin.
4. Clear the ANSEL bit for the RXx pin (if applicable).
5. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
6. Ensure bits CREN and SREN are clear.
7. If interrupts are desired, set the RCxIE bit of the PIEx register and the GIE and PEIE bits of the INTCON register.
8. If 9-bit reception is desired, set bit RX9.
9. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
10. Interrupt flag bit RCxIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCxIE was set.
11. Read the RCxSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
12. Read the 8-bit received data by reading the RCxREG register.
13. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

Offset	Name	Bit Pos.								
0x150A	PCLATH	7:0		PCLATH[6:0]						
0x150B	INTCON	7:0	GIE	PEIE						INTEDG
0x150C ... 0x157F	Reserved									
0x1580	INDF0	7:0	INDF0[7:0]							
0x1581	INDF1	7:0	INDF1[7:0]							
0x1582	PCL	7:0	PCL[7:0]							
0x1583	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x1584	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1586	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1588	BSR	7:0			BSR[5:0]					
0x1589	WREG	7:0	WREG[7:0]							
0x158A	PCLATH	7:0		PCLATH[6:0]						
0x158B	INTCON	7:0	GIE	PEIE						INTEDG
0x158C ... 0x15FF	Reserved									
0x1600	INDF0	7:0	INDF0[7:0]							
0x1601	INDF1	7:0	INDF1[7:0]							
0x1602	PCL	7:0	PCL[7:0]							
0x1603	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x1604	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1606	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1608	BSR	7:0			BSR[5:0]					
0x1609	WREG	7:0	WREG[7:0]							
0x160A	PCLATH	7:0		PCLATH[6:0]						
0x160B	INTCON	7:0	GIE	PEIE						INTEDG
0x160C ... 0x167F	Reserved									
0x1680	INDF0	7:0	INDF0[7:0]							
0x1681	INDF1	7:0	INDF1[7:0]							
0x1682	PCL	7:0	PCL[7:0]							
0x1683	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x1684	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1686	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1688	BSR	7:0			BSR[5:0]					
0x1689	WREG	7:0	WREG[7:0]							
0x168A	PCLATH	7:0		PCLATH[6:0]						

- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

41.4 MPLINK Object Linker/MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

41.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

41.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

Table 42-10. I/O and CLKOUT Timing Specifications

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ. †**	**Max.**	**Units**	**Conditions**
IO1*	$T_{CLKOUTH}$	CLKOUT rising edge delay (rising edge F_{OSC} (Q1 cycle) to falling edge CLKOUT	—	—	70	ns	
IO2*	$T_{CLKOUTL}$	CLKOUT falling edge delay (rising edge F_{OSC} (Q3 cycle) to rising edge CLKOUT	—	—	72	ns	
IO3*	T_{IO_VALID}	Port output valid time (rising edge F_{OSC} (Q1 cycle) to port valid)	—	50	70	ns	
IO4*	T_{IO_SETUP}	Port input setup time (Setup time before rising edge F_{OSC} – Q2 cycle)	20	—	—	ns	
IO5*	T_{IO_HOLD}	Port input hold time (Hold time after rising edge F_{OSC} – Q2 cycle)	50	—	—	ns	
IO6*	T_{IOR_SLREN}	Port I/O rise time, slew rate enabled	—	25	—	ns	V_{DD}=3.0V
IO7*	T_{IOR_SLRDIS}	Port I/O rise time, slew rate disabled	—	5	—	ns	V_{DD}=3.0V
IO8*	T_{IOF_SLREN}	Port I/O fall time, slew rate enabled	—	25	—	ns	V_{DD}=3.0V
IO9*	T_{IOF_SLRDIS}	Port I/O fall time, slew rate disabled	—	5	—	ns	V_{DD}=3.0V
IO10*	T_{INT}	INT pin high or low time to trigger an interrupt	25	—	—	ns	
IO11*	T_{IOC}	Interrupt-on-Change minimum high or low time to trigger interrupt	25	—	—	ns	
* - These parameters are characterized but not tested.							

43. DC and AC Characteristics Graphs and Tables

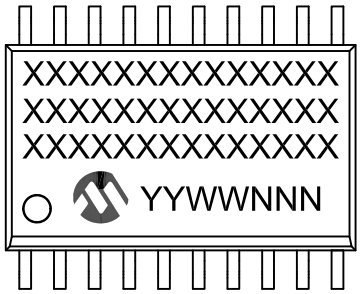
The graphs and tables provided in this section are for design guidance and are not tested. In some graphs or tables, the data presented are outside specified operating range (i.e., outside specified V_{DD} range). This is for information only and devices are ensured to operate properly only within the specified range. Unless otherwise noted, all graphs apply to both the L and LF devices.

Note:
The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

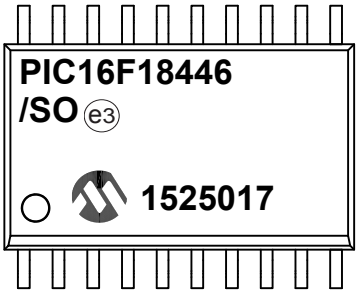
Note:
"Typical" represents the mean of the distribution at 25°C. "Maximum", "Max.", "Minimum" or "Min." represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.

Rev. 30-009020C
09/21/2017

20-Lead SOIC (7.50 mm)

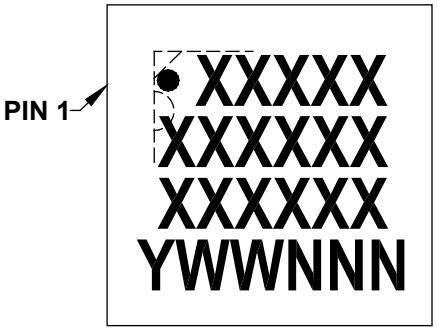


Example

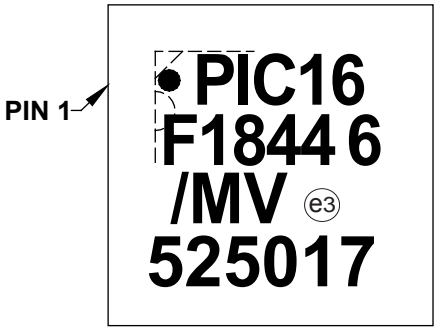


Rev. 30-009020D
09/21/2017

20-Lead UQFN (4x4x0.5 mm)



Example

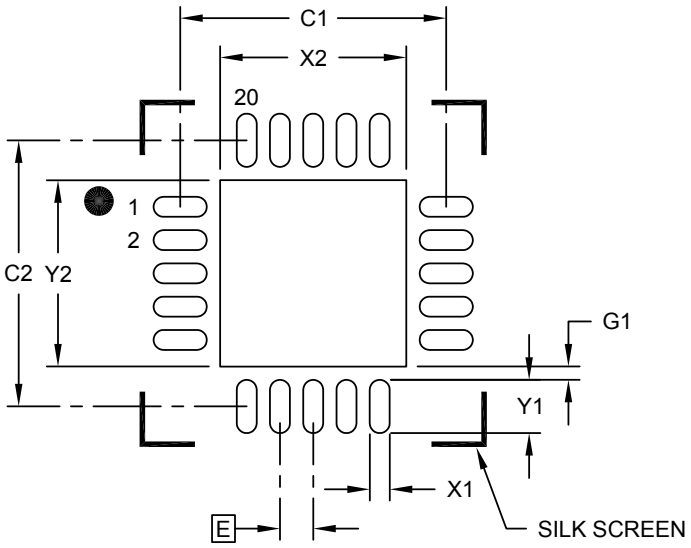


44.1 Package Details

The following sections give the technical details of the packages.

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	X2			2.80
Optional Center Pad Length	Y2			2.80
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X20)	X1			0.30
Contact Pad Length (X20)	Y1			0.80
Contact Pad to Center Pad (X20)	G1	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2255A