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	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1619-i-so

TABLE 3-6: PIC16(L)F1615/9 MEMORY MAP, BANK 24-31

BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31	
C00h	Core Registers (Table 3-1)	C80h	Core Registers (Table 3-1)	D00h	Core Registers (Table 3-1)	D80h	Core Registers (Table 3-1)	E00h	Core Registers (Table 3-1)	E80h	Core Registers (Table 3-1)	F00h	Core Registers (Table 3-1)	F80h	Core Registers (Table 3-1)
C0Bh		C8Bh		D0Bh		D8Bh		E0Bh		E8Bh		F0Bh		F8Bh	
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch	See Table 3-7 for register mapping details	E0Ch	See Table 3-8 for register mapping details	E8Ch	See Table 3-9 and Table 3-10 for register mapping details	F0Ch	See Table 3-12 for register mapping details	F8Ch	See Table 3-12 for register mapping details
C0Dh	—	C8Dh	—	D0Dh	—	D8Dh		E0Dh		E8Dh		F0Dh		F8Dh	
C0Eh	—	C8Eh	—	D0Eh	—	D8Eh		E0Eh		E8Eh		F0Eh		F8Eh	
C0Fh	—	C8Fh	—	D0Fh	—	D8Fh		E0Fh		E8Fh		F0Fh		F8Fh	
C10h	—	C90h	—	D10h	—	D90h		E10h		E90h		F10h		F90h	
C11h	—	C91h	—	D11h	—	D91h		E11h		E91h		F11h		F91h	
C12h	—	C92h	—	D12h	—	D92h		E12h		E92h		F12h		F92h	
C13h	—	C93h	—	D13h	—	D93h		E13h		E93h		F13h		F93h	
C14h	—	C94h	—	D14h	—	D94h		E14h		E94h		F14h		F94h	
C15h	—	C95h	—	D15h	—	D95h		E15h		E95h		F15h		F95h	
C16h	—	C96h	—	D16h	—	D96h		E16h		E96h		F16h		F96h	
C17h	—	C97h	—	D17h	—	D97h		E17h		E97h		F17h		F97h	
C18h	—	C98h	—	D18h	—	D98h		E18h		E98h		F18h		F98h	
C19h	—	C99h	—	D19h	—	D99h		E19h		E99h		F19h		F99h	
C1Ah	—	C9Ah	—	D1Ah	—	D9Ah		E1Ah		E9Ah		F1Ah		F9Ah	
C1Bh	—	C9Bh	—	D1Bh	—	D9Bh		E1Bh		E9Bh		F1Bh		F9Bh	
C1Ch	—	C9Ch	—	D1Ch	—	D9Ch		E1Ch		E9Ch		F1Ch		F9Ch	
C1Dh	—	C9Dh	—	D1Dh	—	D9Dh		E1Dh		E9Dh		F1Dh		F9Dh	
C1Eh	—	C9Eh	—	D1Eh	—	D9Eh		E1Eh		E9Eh		F1Eh		F9Eh	
C1Fh	—	C9Fh	—	D1Fh	—	D9Fh		E1Fh		E9Fh		F1Fh		F9Fh	
C20h	Unimplemented Read as '0'	CA0h	Unimplemented Read as '0'	D20h	Unimplemented Read as '0'	DA0h		E20h		EA0h		F20h		FA0h	
C6Fh		CEFh		D6Fh		DEFh		E6Fh		EEFh		F6Fh		FEFh	
C70h	Accesses 70h – 7Fh	CF0h	Accesses 70h – 7Fh	D70h	Accesses 70h – 7Fh	DF0h	Accesses 70h – 7Fh	E70h	Accesses 70h – 7Fh	EF0h	Accesses 70h – 7Fh	F70h	Accesses 70h – 7Fh	FF0h	Accesses 70h – 7Fh
CFFh		CFFh		D7Fh		DFFh		E7Fh		EFFh		F7Fh		FFFh	

Legend: = Unimplemented data memory locations, read as '0'.

12.1.7 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after Reset. Other functions are selected with the peripheral pin select logic. See **Section13.0 “Peripheral Pin Select (PPS) Module”** for more information. Analog input functions, such as ADC inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELA register. Digital output functions may continue to control the pin when in Analog mode.

TABLE 16-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		118

Legend: Shaded cells are unused by the temperature indicator module.

22.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively. Table 22-1 displays the Timer1 enable selections.

TABLE 22-1: TIMER1 ENABLE SELECTIONS

TMR1ON	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	Always On
1	1	Count Enabled

22.2 Clock Source Selection

The TMR1CS<1:0> bits of the T1CON register are used to select the clock source for Timer1. Table 22-2 displays the clock source selections.

22.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

When the FOSC internal clock source is selected, the Timer1 register value will increment by four counts every instruction clock cycle. Due to this condition, a 2 LSB error in resolution will occur when reading the Timer1 value. To utilize the full resolution of Timer1, an asynchronous input signal must be used to gate the Timer1 clock input.

The following asynchronous sources may be used:

- Asynchronous event on the T1G pin to Timer1 gate
- C1 or C2 comparator input to Timer1 gate

22.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input T1CKI. The external clock source can be synchronized to the microcontroller system clock or it can run asynchronously.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0) when T1CKI is high then Timer1 is enabled (TMR1ON=1) when T1CKI is low.

TABLE 22-2: CLOCK SOURCE SELECTIONS

TMR1CS<1:0>	Clock Source
11	LFINTOSC
10	External Clocking on T1CKI Pin
01	System Clock (FOSC)
00	Instruction Clock (FOSC/4)

23.6 Timer2 Operation During Sleep

When PSYNC = 1, Timer2 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and T2PR registers will remain unchanged while processor is in Sleep mode.

When PSYNC = 0, Timer2 will operate in Sleep as long as the clock source selected is also still running. Selecting the LFINTOSC, MFINTOSC, or HFINTOSC oscillator as the timer clock source will keep the selected oscillator running during Sleep.

FIGURE 24-14: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 0, DHEN = 0)

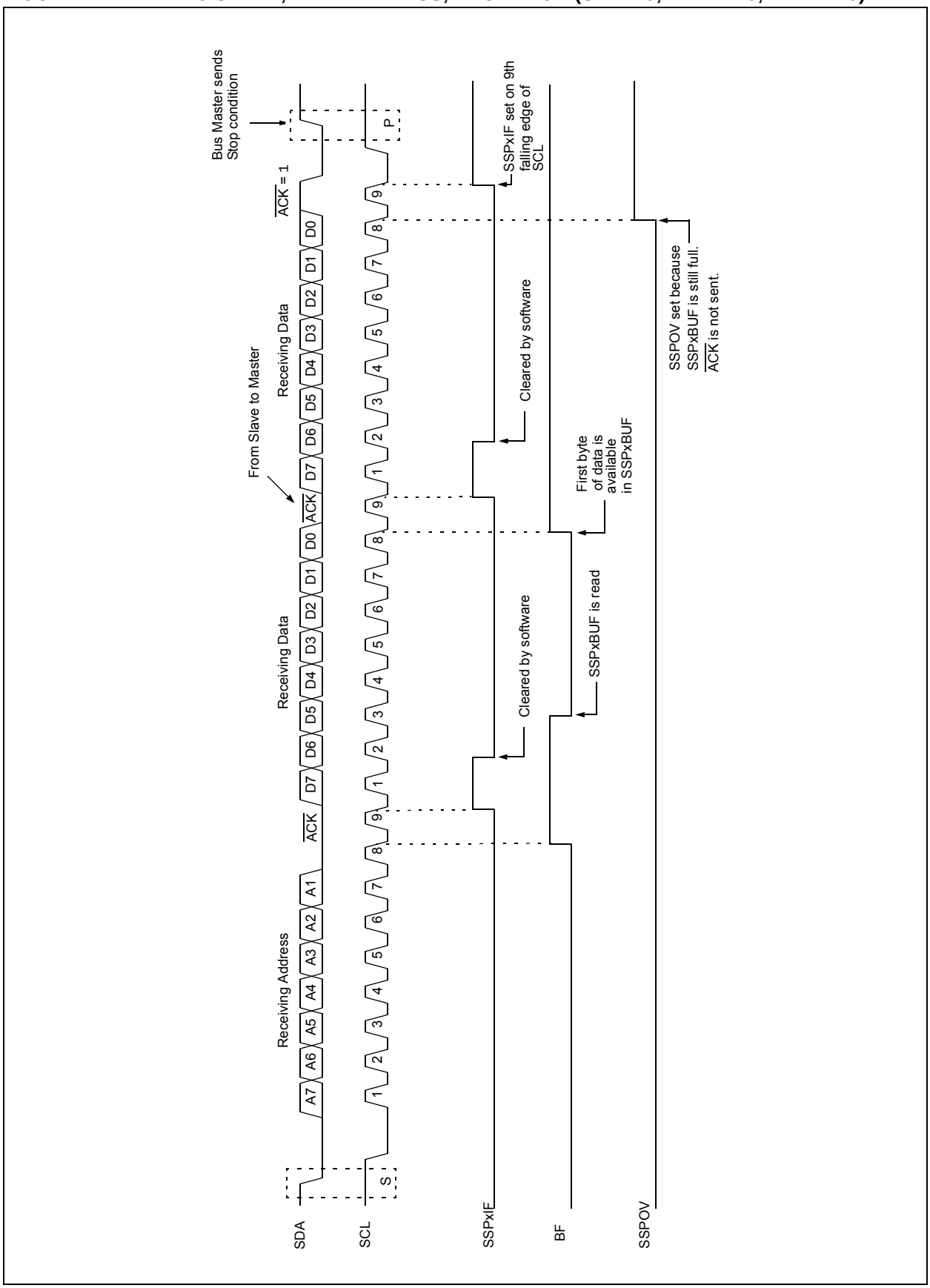


FIGURE 24-20: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)

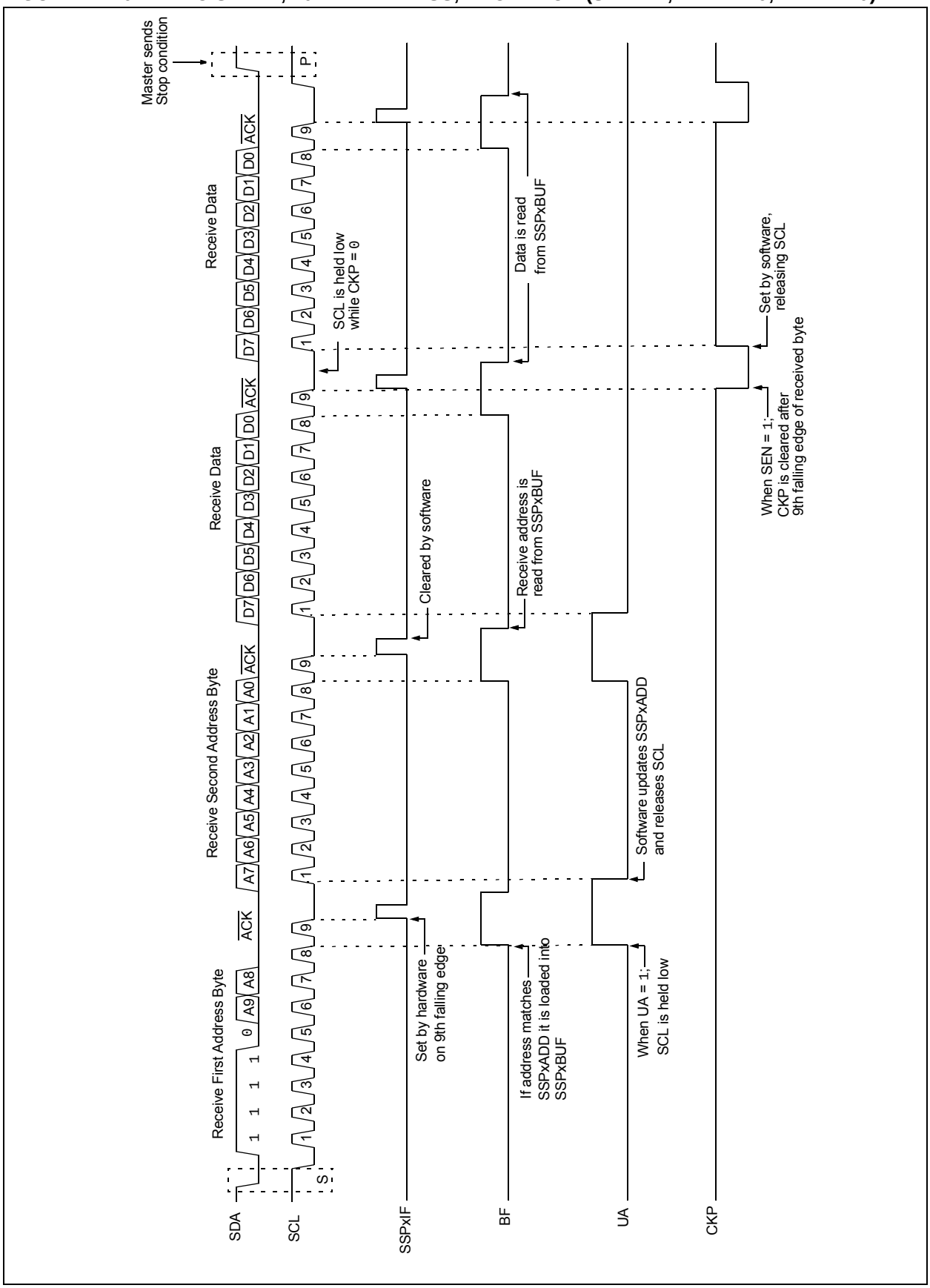
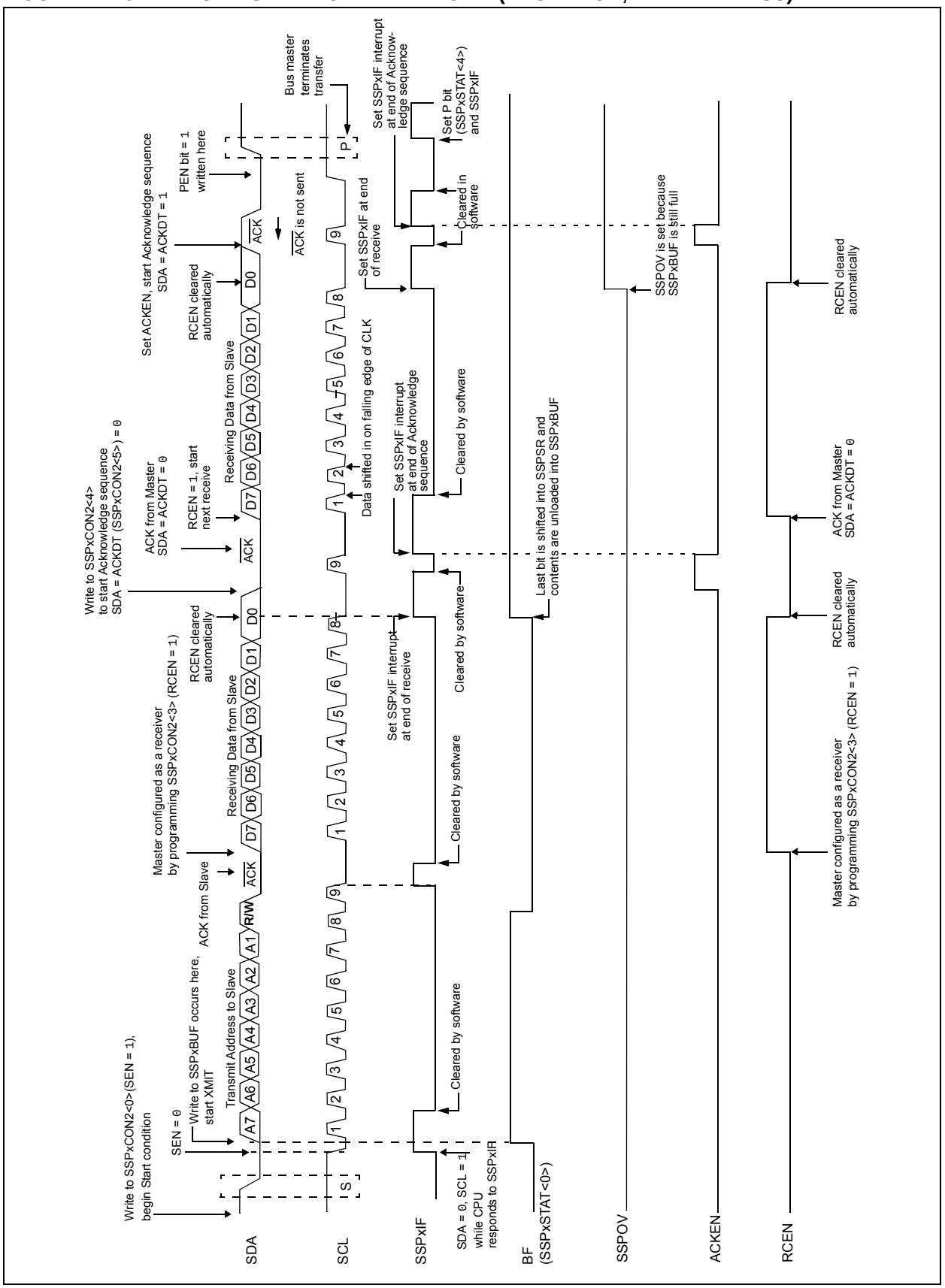


FIGURE 24-29: I²C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



25.1.1.5 TSR Status

The TRMT bit of the TXxSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXxREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

25.1.1.6 Transmitting 9-Bit Characters

The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXxSTA register is set, the EUSART will shift nine bits out for each character transmitted. The TX9D bit of the TXxSTA register is the ninth, and Most Significant data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the eight Least Significant bits into the TXxREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXxREG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section 25.1.2.7 "Address Detection"** for more information on the Address mode.

25.1.1.7 Asynchronous Transmission Set-up:

1. Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 25.4 "EUSART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
4. Set SCKP bit if inverted transmit is desired.
5. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
6. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
7. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
8. Load 8-bit data into the TXxREG register. This will start the transmission.

FIGURE 25-3: ASYNCHRONOUS TRANSMISSION

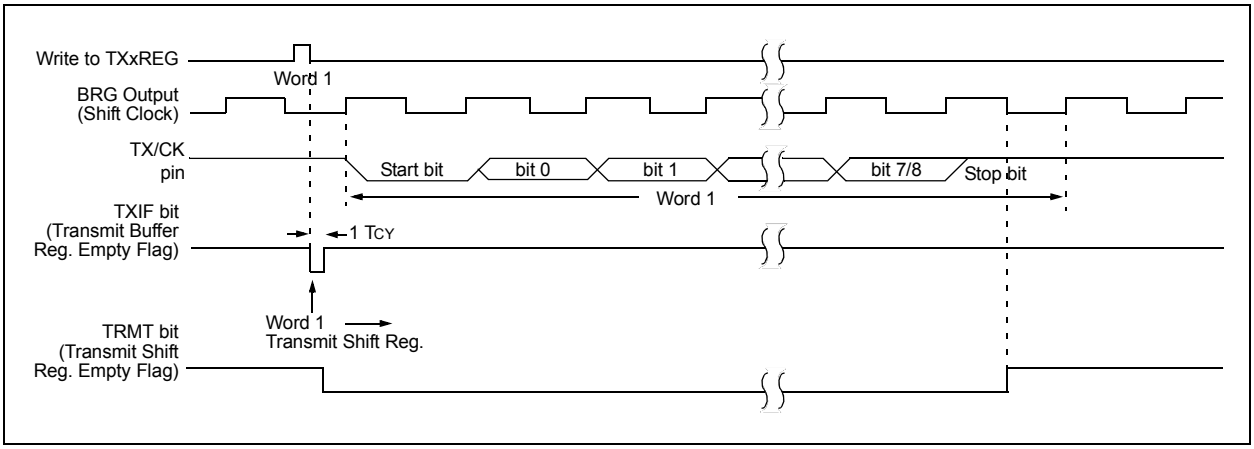
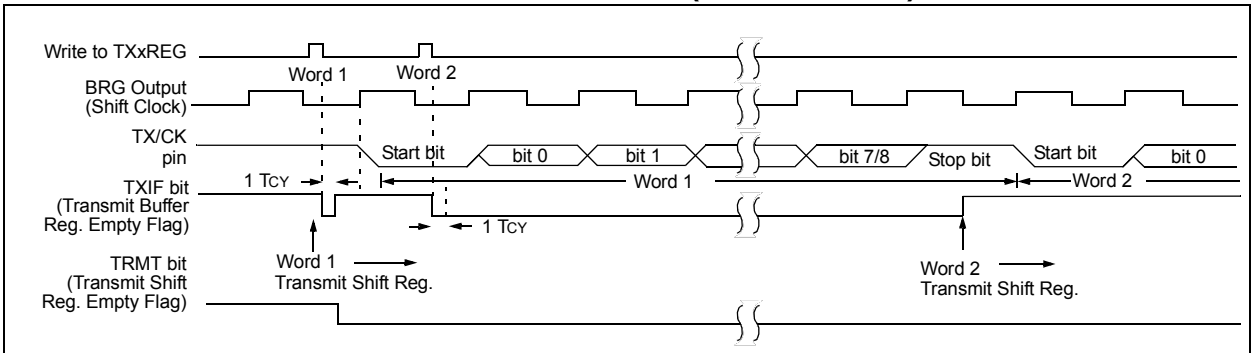


FIGURE 25-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 25-9: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	160
ANSELB(1)	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7(1)	ANSC6(1)	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	331
CKPPS	—	—	—	CKPPS<4:0>					182, 180
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	330
RXPPS	—	—	—	RXPPS<4:0>					182, 180
RxyPPS	—	—	—	RxyPPS<4:0>					180
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	159
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
TX1REG	EUSART Transmit Data Register								321*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	329

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous slave transmission.

* Page provides register information.

Note 1: PIC16(L)F1619 only.

2: Unimplemented, read as '1'.

29.1 CLCx Setup

Programming the CLCx 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 CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

29.1.1 DATA SELECTION

There are 41 signals available as inputs to the configurable logic. Four 41 input multiplexers are used to select the inputs to pass on to the next stage. This allows for any of the possible input signals to be used as any of the four inputs to the CLC module.

Data selection is through four multiplexers as indicated on the left side of Figure 29-2. Data inputs in the figure are identified by a generic numbered input name.

Table 29-1 correlates the generic input name to the actual signal for each CLC module. The column labeled CLCxSELy refers to the value of any of the four registers associated with the four multiplexers, CLCxSEL0 through CLCxSEL3.

Data inputs for each multiplexer are selected with their respective CLCxSELy registers.

Note:	Data selections are undefined at power-up.

TABLE 29-1: CLCx DATA INPUT SELECTION

Data Input	CLCxSELy	CLC Input Signal
LCx_in[0]	000000	CLCIN0
LCx_in[1]	000001	CLCIN1
LCx_in[2]	000010	CLCIN2
LCx_in[3]	000011	CLCIN3
LCx_in[4]	000100	LC1_out
LCx_in[5]	000101	LC2_out
LCx_in[6]	000110	LC3_out
LCx_in[7]	000111	LC4_out
LCx_in[8]	001000	C1OUT_sync
LCx_in[9]	001001	C2OUT_sync
LCx_in[10]	001010	CWGOUTA
LCx_in[11]	001011	CWGOUTB
LCx_in[12]	001100	CCP1_out
LCx_in[13]	001101	CCP2_out
LCx_in[14]	001110	PWM3_out
LCx_in[15]	001111	PWM4_out
LCx_in[16]	010000	AT1_cmp1
LCx_in[17]	010001	AT1_cmp2
LCx_in[18]	010010	AT1_cmp3
LCx_in[19]	010011	SMT1_match
LCx_in[20]	010100	SMT2_match
LCx_in[21]	010101	ZCD1_output
LCx_in[22]	010110	TMR0_overflow
LCx_in[23]	010111	TMR1_overflow
LCx_in[24]	011000	TMR2_postscaled
LCx_in[25]	011001	TMR3_overflow
LCx_in[26]	011010	TMR4_postscaled
LCx_in[27]	011011	TMR5_overflow
LCx_in[28]	011100	TMR6_postscaled
LCx_in[29]	011101	IOC_interrupt
LCx_in[30]	011110	ADC_rc
LCx_in[31]	011111	LFINTOSC
LCx_in[32]	100000	HFINTOSC
LCx_in[33]	100001	FOSC
LCx_in[34]	100010	AT1_missedpulse
LCx_in[35]	100011	AT1_perclk
LCx_in[36]	100100	AT1_phsclk
LCx_in[37]	100101	TX
LCx_in[38]	100110	RX
LCx_in[39]	100111	SCK
LCx_in[40]	101000	SDO

29.2.3 CLCx SETUP STEPS

The following steps should be followed when setting up the CLCx:

- Disable CLCx by clearing the LCxEN bit.
- Select desired inputs using CLCxSEL0, CLCxSEL1, CLCxSEL2 and CLCxSEL3 registers (See Table 29-1).
- Clear any associated ANSEL bits.
- Set all TRIS bits associated with inputs.
- Clear all TRIS bits associated with outputs.
- Enable the chosen inputs through the four gates using CLCxGLS0, CLCxGLS1, CLCxGLS2, and CLCxGLS3 registers.
- Select the gate output polarities with the LCxPOLy bits of the CLCxPOL register.
- Select the desired logic function with the LCxMODE<2:0> bits of the CLCxCON register.
- Select the desired polarity of the logic output with the LCxPOL bit of the CLCxPOL register. (This step may be combined with the previous gate output polarity step).
- If driving a device, set the desired pin PPS control register and also clear the TRIS bit corresponding to that output.
- If interrupts are desired, configure the following bits:
  - Set the LCxINTP bit in the CLCxCON register for rising event.
  - Set the LCxINTN bit in the CLCxCON register or falling event.
  - Set the CLCxIE bit of the associated PIE registers.
  - Set the GIE and PEIE bits of the INTCON register.
- Enable the CLCx by setting the LCxEN bit of the CLCxCON register.

29.3 CLCx Interrupts

An interrupt will be generated upon a change in the output value of the CLCx when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in each CLC for this purpose.

The CLCxIF bit of the associated PIR registers will be set when either edge detector is triggered and its associated enable bit is set. The LCxINTP enables rising edge interrupts and the LCxINTN bit enables falling edge interrupts. Both are located in the CLCxCON register.

To fully enable the interrupt, set the following bits:

- LCxON bit of the CLCxCON register
- CLCxIE bit of the associated PIE registers
- LCxINTP bit of the CLCxCON register (for a rising edge detection)
- LCxINTN bit of the CLCxCON register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The CLCxIF bit of the associated PIR registers, must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

29.4 Output Mirror Copies

Mirror copies of all LCxCON output bits are contained in the CLCxDATA register. Reading this register reads the outputs of all CLCs simultaneously. This prevents any reading skew introduced by testing or reading the CLCxOUT bits in the individual CLCxCON registers.

29.5 Effects of a Reset

The CLCxCON register is cleared to zero as the result of a Reset. All other selection and gating values remain unchanged.

29.6 Operation During Sleep

The CLC module operates independently from the system clock and will continue to run during Sleep, provided that the input sources selected remain active.

The HFINTOSC remains active during Sleep when the CLC module is enabled and the HFINTOSC is selected as an input source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and as a CLC input source, when the CLC is enabled, the CPU will go idle during Sleep, but the CLC will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

31.7 Interrupts

The angular timer and its capture/compare features can generate multiple interrupt conditions. To accommodate all of these interrupt sources, the module is provided with its own interrupt logic structure, similar to that of the micro controller. Angular timer interrupts are enabled by the ATxIE0 register (Register 31-13) and their respective flags are located in the ATxIR0 register (Register 31-14). The capture/compare interrupts are enabled by the ATxIE1 register (Register 31-15) with flags in the ATxIR1 register (Register 31-16). All sources are funneled into a single Angular Timer Interrupt Flag bit, ATxIF of the PIR5 register (Register 7-11). This means that upon a triggered interrupt, the ATxIR0 and ATxIR1 register bits will indicate the source of the triggered interrupt. It also means that in order for specific interrupts to generate a microcontroller interrupt, both the ATxIE bit of the PIE register and the desired enable bit in either ATxIE0 or ATxIE1 must be set.

Note:	Due to the nature of the angular timer interrupts, the ATxIF flag bit of the PIR5 register is read-only.

31.7.1 ANGULAR TIMER PERIOD INTERRUPT

This interrupt is triggered when the AT module detects a period event. In Single-Pulse mode, a period event occurs on every input signal edge. In Multi-Pulse mode, a period event occurs on the input signal edge following a missed pulse. The period interrupt generation matches with the pulses on the period clock output of the timer. It is enabled by the ATPERIE bit of the ATxIE0 register and the status is indicated by the PERIF bit of the ATxIR0 register.

31.7.2 ANGULAR TIMER PHASE CLOCK INTERRUPT

This interrupt is triggered on each pulse of the phase clock output of the timer. It is enabled by the ATPHIE bit of the ATxIE0 register and the status is indicated by the PHSIF bit of the ATxIR0 register.

31.7.3 ANGULAR TIMER MISSING PULSE INTERRUPT

This interrupt is triggered upon the output of a missing pulse detection signal. Refer to **Section 31.2.3 “Missing Pulse Detection”** for more information. This interrupt is enabled by the ATMISSIE bit of the ATxIE0 register and its status is indicated by the ATMISSIF bit of the ATxIR0 register.

31.7.4 ANGULAR TIMER CAPTURE/COMPARE INTERRUPTS

Capture and compare interrupts are triggered by the capture/compare functions of the module. If configured for Capture mode, the interrupt will trigger after the capture signal has successfully latched the value of the phase counter into the capture registers. If configured for Compare mode, the interrupt will trigger when a match is detected between the value placed in the compare register and the value of the phase counter. These interrupts are controlled by the CC1IE, CC2IE, and CC3IE bits of the ATxIE1 register, respectively, and are similarly indicated by the CC1IF, CC2IF, and CC3IF bits of the ATxIR1 register.

TABLE 31-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE ANGULAR TIMER MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
AT1CC1H	—	—	—	—	—	—	CC1<9:8>		485
AT1CC1L	CC1<7:0>								485
AT1CCON1	CC1EN	—	—	CC1POL	CAP1P	—	—	CC1MODE	483
AT1CCON2	CC2EN	—	—	CC2POL	CAP2P	—	—	CC2MODE	483
AT1CCON3	CC3EN	—	—	CC3POL	CAP3P	—	—	CC3MODE	483
AT1CLK	—	—	—	—	—	—	—	CS0	473
AT1CON0	EN	PREC	PS<1:0>		POL	—	APMOD	MODE	471
AT1CON1	—	PHP	—	PRP	—	MPP	ACCS	VALID	472
AT1CSEL1	—	—	—	—	—	CP1S<2:0>			484
AT1CSEL2	—	—	—	—	—	CP2S<2:0>			484
AT1CSEL3	—	—	—	—	—	CP3S<2:0>			484
AT1ERRH	ERR<15:8>								482
AT1ERRL	ERR<7:0>								482
AT1IE0	—	—	—	—	—	PHSIE	MISSIE	PERIE	478
AT1IR0	—	—	—	—	—	PHSIF	MISSIF	PERIF	478
AT1IE1	—	—	—	—	—	CC3IE	CC2IE	CC1IE	479
AT1IR1	—	—	—	—	—	CC3IF	CC2IF	CC1IF	480
AT1MISSH	MISS<15:8>								475
AT1MISSL	MISS<7:0>								475
AT1PERH	POV	PER<14:8>							476
AT1PERL	PER<7:0>								476
AT1PHSH	—	—	—	—	—	—	PHS<9:8>		477
AT1PHSL	PHS<7:0>								477
AT1RESH	—	—	—	—	—	—	RES<9:8>		474
AT1RESL	RES<7:0>								474
AT1SIG	—	—	—	—	—	SSEL<2:0>			473
AT1STPTH	—	STPT<14:8>							481
AT1STPTL	STPT<7:0>								481
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	—	AT1IE	PID1EIE	PID1DIE	110
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF	115

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the AT module.

32.0 MATH ACCELERATOR WITH PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) MODULE

The math accelerator module is a mathematics module that can perform a variety of operations, most prominently acting as a PID (Proportional-Integral-Derivative) controller. A PID controller is an algorithm that uses the present error (proportional), the sum of the present and all previous errors (integral), and the difference between the present and previous change (derivative) to correct errors and provide stability in a system. It provides feedback to a system through a series of iterations, using the present error as well as previous errors to calculate a new input to the controller. The data flow for both PID modes is illustrated in Figure 32-1.

The module accomplishes the task of calculating the PID algorithm by utilizing user-provided coefficients along with a multiplier and accumulator. As such, this multiplier and accumulator can also be configured to quickly and efficiently perform signed and unsigned multiply-and-add calculations both with and without accumulation. The data flow for these modes is illustrated in Figure 32-2.

Features of this module include:

- Signed multiplier
- 35-bit signed accumulator
- PID controller support with user inputs for K1, K2, K3, system error and desired set point
- Completion and Error interrupts
- Multiple user modes allowing for PID with or without accumulation as well as several multiplication operations

32.1 PID Module Setup Summary

The PID module can be configured either as a PID controller or as a multiply and accumulate module. Multiply and accumulate can be performed in four modes:

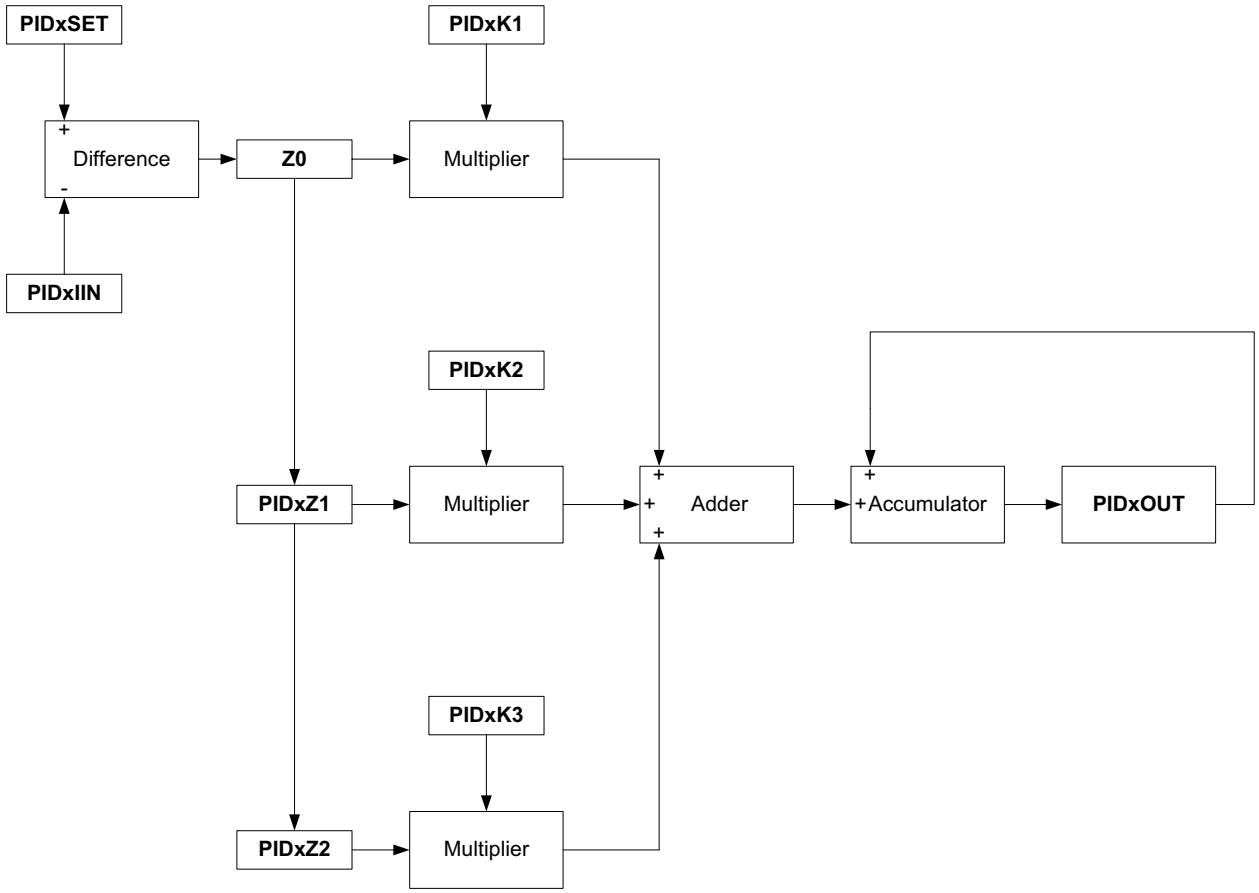
- Unsigned multiply and add, without accumulation
- Unsigned multiply and accumulate
- Signed multiply and add, without accumulation
- Signed multiply and accumulate

All of the modes are selected by the MODE<2:0> bits of the PIDxCON register.

32.1.1 PID MODE SETUP AND OPERATION

When the MODE<2:0> bits of the PIDxCON register are equal to ‘101’, the module is in PID controller mode. The operation of the module in PID controller mode is generally performed as a loop. The input from an external system is fed into the controller, and the controller’s output is fed back into the external system. This will produce a new response from the system that is then looped back into the PID controller. The data flow for the PID operation is illustrated in Figure 32-1.

FIGURE 32-1: PID MODULE BASIC DATA FLOW BLOCK DIAGRAM, PID MODES



Note 1: After the results of PIDxZ2 are multiplied by PIDxK3 and the result is added to the accumulator, the current value from PIDxZ1 is loaded into PIDxZ2. The same is true for PIDxZ1 and the current SET-IN value.

REGISTER 32-14: PIDxOUTHL: PID OUTPUT HIGH LOW 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
OUT<23:16>							
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	q = Value depends on condition

bit 7-0 **OUT<23:16>:** Bits <23:16> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

REGISTER 32-15: PIDxOUTLH: PID OUTPUT LOW HIGH 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
OUT<15:8>							
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	q = Value depends on condition

bit 7-0 **OUT<15:8>:** Bits <15:8> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

REGISTER 32-16: PIDxOUTLL: PID OUTPUT LOW LOW 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
OUT<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	q = Value depends on condition

bit 7-0 **OUT<7:0>:** Bits <7:0> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

TABLE 35-9: PLL CLOCK TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
F10	FOSC	Oscillator Frequency Range	4	—	8	MHz	
F11	FSYS	On-Chip VCO System Frequency	16	—	32	MHz	
F12	TRC	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13*	ΔCLK	CLKOUT Stability (Jitter)	-0.25%	—	+0.25%	%	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

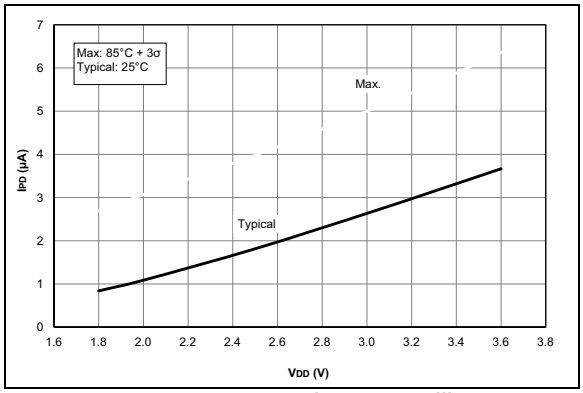


FIGURE 36-37: *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC16LF1615/9 Only.*

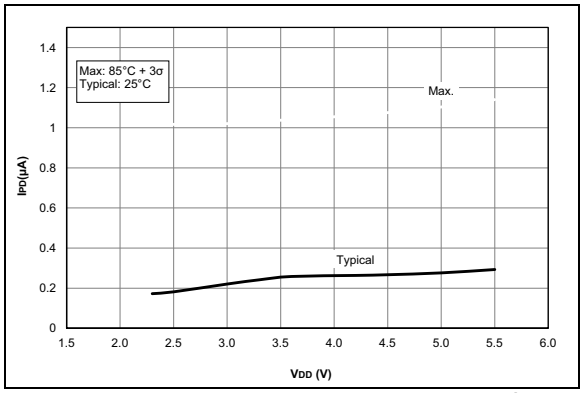


FIGURE 36-40: *IPD, ADC Non-Converting, PIC16F1615/9 Only.*

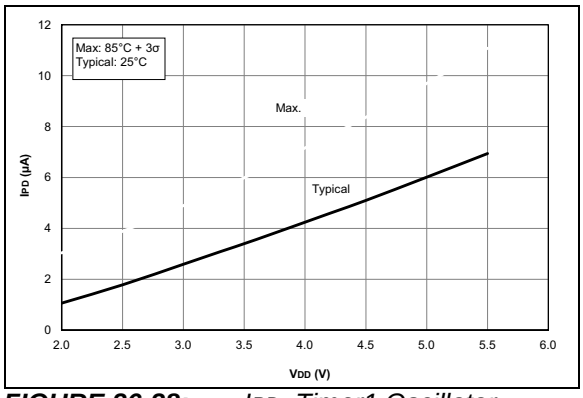


FIGURE 36-38: *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC16F1615/9 Only.*

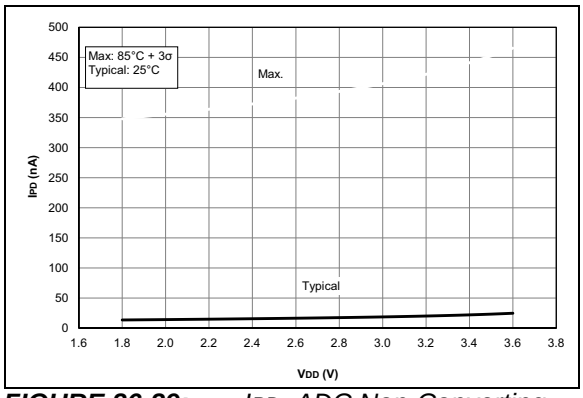
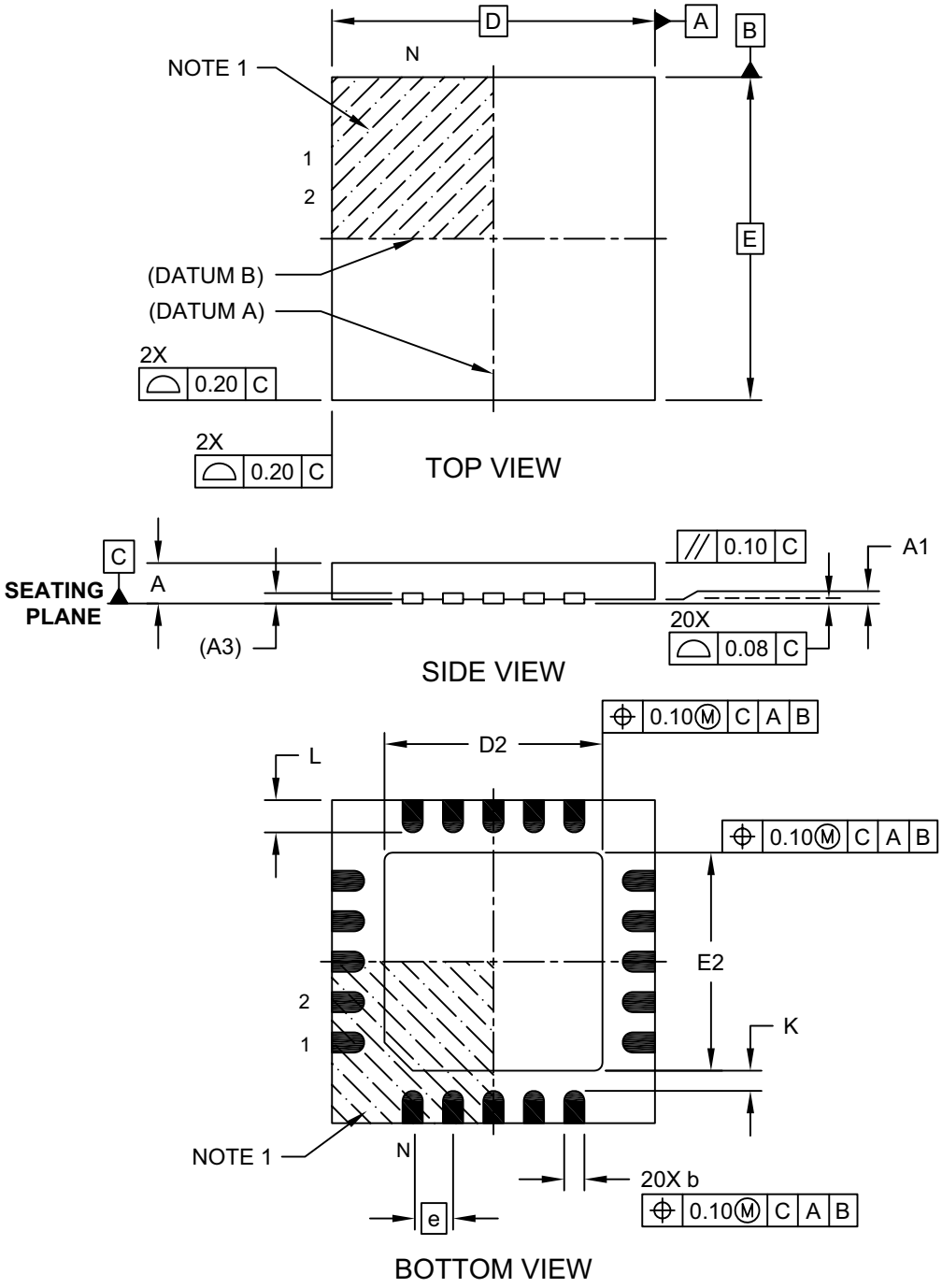


FIGURE 36-39: *IPD, ADC Non-Converting, PIC16LF1615/9 Only.*

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



Microchip Technology Drawing C04-255A Sheet 1 of 2