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"

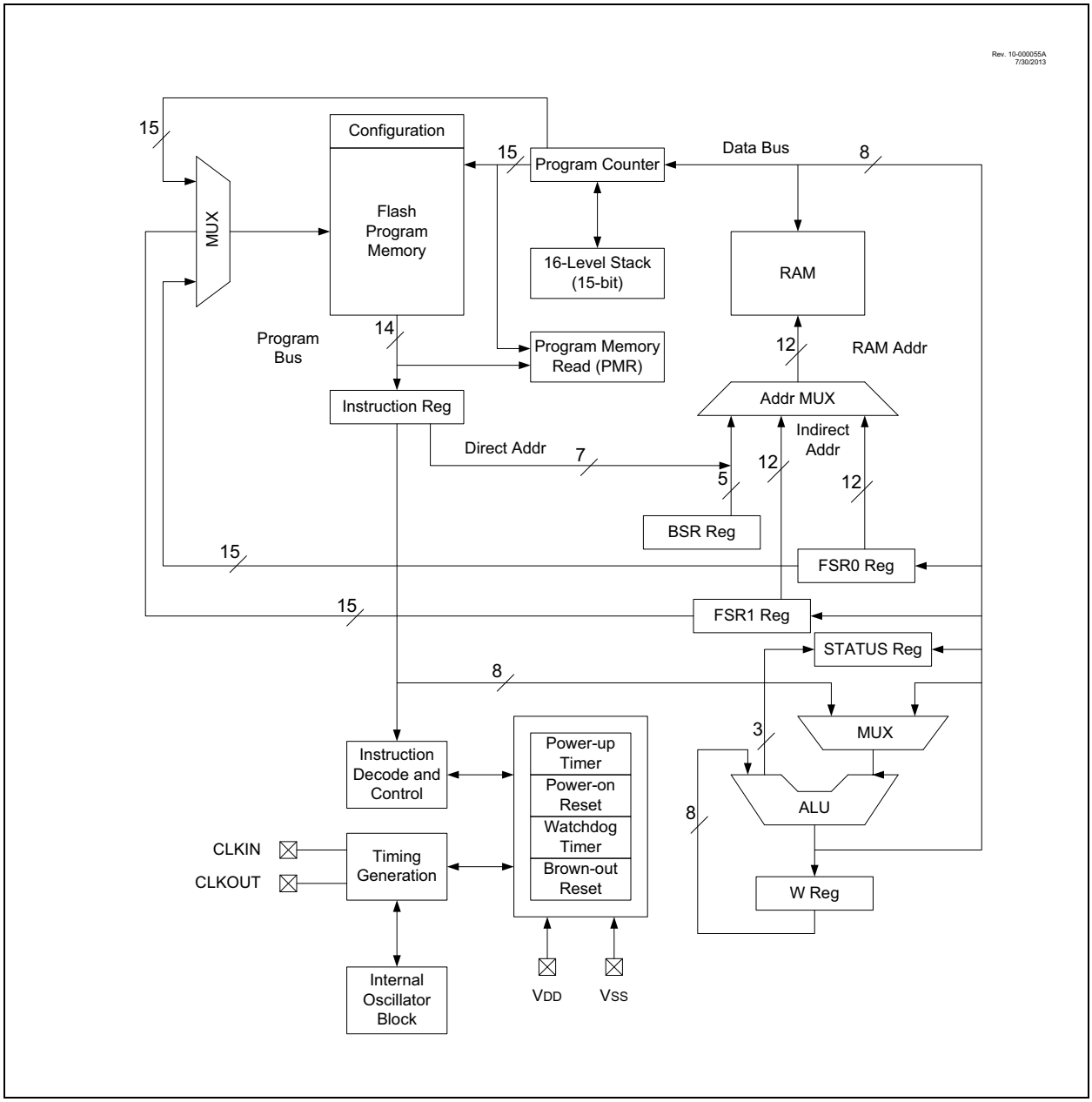
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	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/pic16f1618t-i-so

2.0 ENHANCED MID-RANGE CPU

This family of devices contain an enhanced mid-range 8-bit CPU core. The CPU has 49 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16 levels deep and has Overflow and Underflow Reset capability. Direct, Indirect, and Relative Addressing modes are available. Two File Select Registers (FSRs) provide the ability to read program and data memory.

- Automatic Interrupt Context Saving
- 16-level Stack with Overflow and Underflow
- File Select Registers
- Instruction Set

FIGURE 2-1: CORE BLOCK DIAGRAM



9.1 Independent Clock Source

The WDT can derive its time base from either the 31 kHz LFINTOSC or 31.25 kHz MFINTOSC internal oscillators, depending on the value of either the WDTCCS<2:0> configuration bits or the WDTCS<2:0> bits of WDTCON1. Time intervals in this chapter are based on a minimum nominal interval of 1 ms. See **Section35.0 "Electrical Specifications"** for LFINTOSC and MFINTOSC tolerances.

9.2 WDT Operating Modes

The Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Words. See Table 9-1.

9.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Words are set to '11', the WDT is always on.

WDT protection is active during Sleep.

9.2.2 WDT IS OFF IN SLEEP

When the WDTE bits of Configuration Words are set to '10', the WDT is on, except in Sleep.

WDT protection is not active during Sleep.

9.2.3 WDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Words are set to '01', the WDT is controlled by the SEN bit of the WDTCON0 register.

WDT protection is unchanged by Sleep. See Table 9-1 for more details.

TABLE 9-1: WDT OPERATING MODES

WDTE<1:0>	SEN	Device Mode	WDT Mode
11	X	X	Active
10	X	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0	X	Disabled
00	X	X	Disabled

9.3 Time-Out Period

The WDTPS bits of the WDTCON0 register set the time-out period from 1 ms to 256 seconds (nominal). After a Reset, the default time-out period is two seconds.

9.4 Watchdog Window

The Watchdog Timer has an optional Windowed mode that is controlled by the WDTCWS<2:0> Configuration bits and WINDOW<2:0> bits of the WDTCON1 register. In the Windowed mode, the CLRWDT instruction must occur within the allowed window of the WDT period. Any CLRWDT instruction that occurs outside of this window will trigger a window violation and will cause a WDT Reset, similar to a WDT time out. See Figure 9-2 for an example.

The window size is controlled by the WDTCWS<2:0> Configuration bits, or the WINDOW<2:0> bits of WDTCON1, if WDTCWS<2:0> = 111.

In the event of a window violation, a Reset will be generated and the $\overline{\text{WDTWV}}$ bit of the PCON register will be cleared. This bit is set by a POR or can be set in firmware.

9.5 Clearing the WDT

The WDT is cleared when any of the following conditions occur:

- Any Reset
- Valid CLRWDT instruction is executed
- Device enters Sleep
- Device wakes up from Sleep
- WDT is disabled
- Oscillator Start-up Timer (OST) is running
- Any write to the WDTCON0 or WDTCON1 registers

9.5.1 CLRWDT CONSIDERATIONS (WINDOWED MODE)

When in Windowed mode, the WDT must be armed before a CLRWDT instruction will clear the timer. This is performed by reading the WDTCON0 register. Executing a CLRWDT instruction without performing such an arming action will trigger a window violation.

See Table 9-2 for more information.

9.6 Operation During Sleep

When the device enters Sleep, the WDT is cleared. If the WDT is enabled during Sleep, the WDT resumes counting. When the device exits Sleep, the WDT is cleared again.

The WDT remains clear until the OST, if enabled, completes. See **Section5.0 "Oscillator Module"** for more information on the OST.

When a WDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits in the STATUS register are changed to indicate the event. The $\overline{\text{RWDT}}$ bit in the PCON register can also be used. See **Section3.0 "Memory Organization"** for more information.

REGISTER 12-23: SLRCONC: PORTC SLEW RATE CONTROL 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
SLRC7(1)	SLRC6(1)	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
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 **SLRC<7:0>:** PORTC Slew Rate Enable bits(1)
For RC<7:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

Note 1: SLRC<7:6> on PIC16(L)F1618 only.

REGISTER 12-24: INLVLC: PORTC INPUT LEVEL 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
INLVLC7(1)	INLVLC6(1)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
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 **INLVLC<7:0>:** PORTC Input Level Select bits(1)
For RC<7:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

Note 1: INLVLC<7:6> on PIC16(L)F1618 only.

19.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in **Section35.0 "Electrical Specifications"** for more details.

19.9 Analog Input Connection Considerations

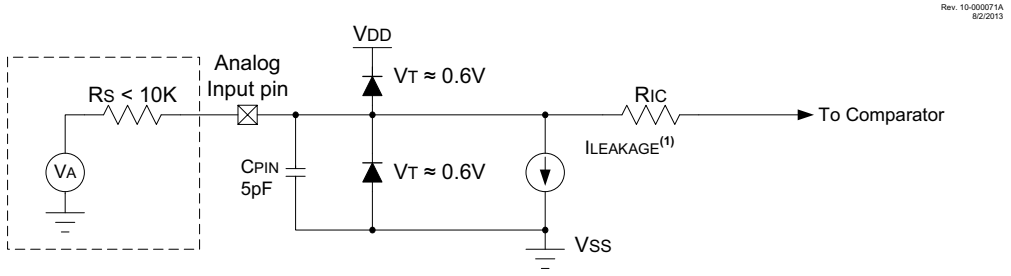
A simplified circuit for an analog input is shown in Figure 19-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS}. The analog input, therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

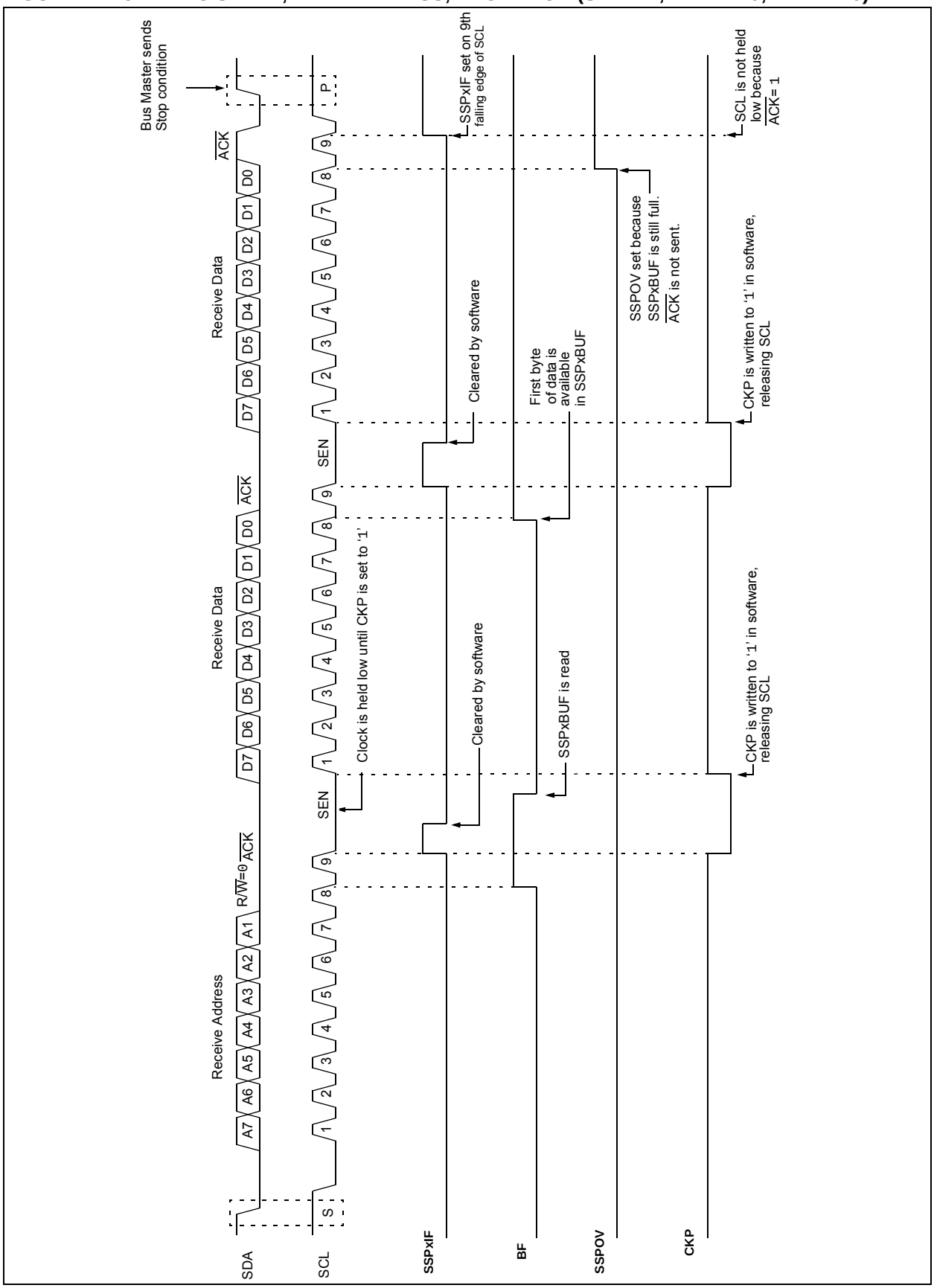
FIGURE 19-3: ANALOG INPUT MODEL



Legend: C_{PIN} = Input Capacitance
$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions
R_{IC} = Interconnect Resistance
R_S = Source Impedance
V_A = Analog Voltage
V_T = Threshold Voltage

Note 1: See **Section35.0 "Electrical Specifications"**.

FIGURE 24-15: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)



25.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 25-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCxREG register.

25.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the CREN bit of the RCxSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXxSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCxSTA register enables the EUSART. The programmer must set the corresponding TRIS bit to configure the RX/DT I/O pin as an input.

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

25.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 25.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCxREG register.

Note:	If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 25.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

25.1.2.3 Receive Interrupts

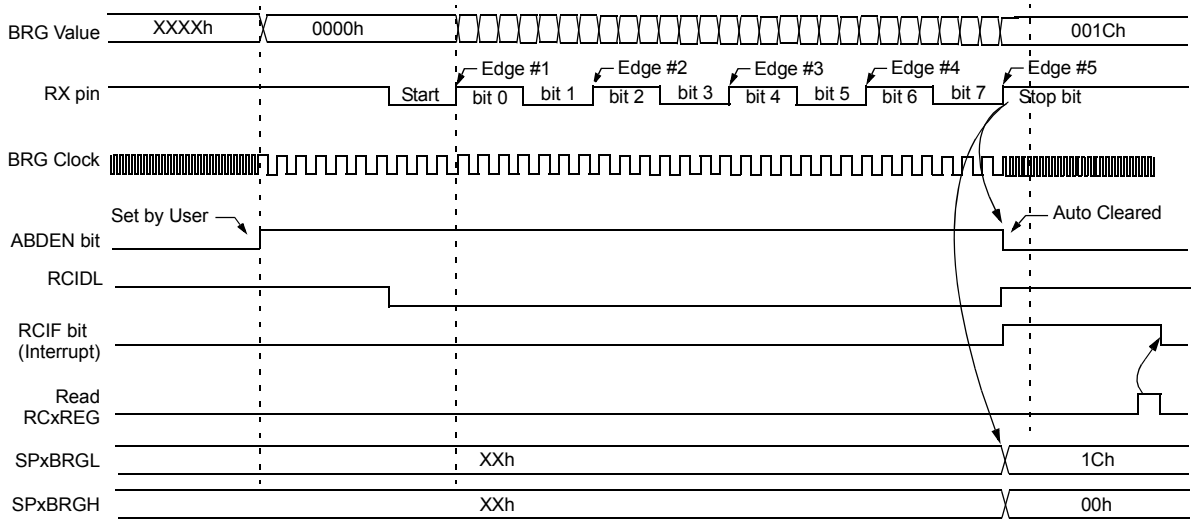
The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting all of the following bits:

- RCIE, Interrupt Enable bit of the PIE1 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

FIGURE 25-6: AUTOMATIC BAUD RATE CALIBRATION



Note 1: The ABD sequence requires the **EUSART** module to be configured in Asynchronous mode.

25.5.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 25.5.1.5 "Synchronous Master Reception"**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never idle
- SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCxREG register. If the RCIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

25.5.2.4 Synchronous Slave Reception Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the ANSEL bit for both the CK and DT pins (if applicable).
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Set the CREN bit to enable reception.
6. The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCxSTA register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCxREG register.
9. 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.

TABLE 25-10: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

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	152
ANSELB(1)	—	—	ANSB5	ANSB4	—	—	—	—	159
ANSELC	ANSC7(1)	ANSC6(1)	—	—	ANSC3	ANSC2	ANSC1	ANSC0	166
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	323
CKPPS	—	—	—	CKPPS<4:0>					174, 172
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
RC1REG	EUSART Receive Data Register								316*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	322
RXPPS	—	—	—	RXPPS<4:0>					174, 172
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	151
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	158
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	321

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

* Page provides register information.

Note 1: PIC16(L)F1618 only.

2: Unimplemented, read as '1'.

26.3.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for all CCP modules.

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to 10 bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2/4/6 registers
- T2CON/T4CON/T6CON registers
- CCPRxH:CCPRxL register pair

Figure shows a simplified block diagram of PWM operation.

Note 1:	The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin.
2:	Clearing the CCPxCON register will relinquish control of the CCPx pin.

26.3.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

1. Disable the CCPx pin output driver by setting the associated TRIS bit.
2. Determine which timer will be used to clock the CCP; Timer2/4/6.
3. Load the associated PR2/4/6 register with the PWM period value.
4. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
5. Load the CCPRxH:CCPRxL register pair with the PWM duty cycle value.
6. Configure and start Timer2/4/6:
   - Clear the TMR2IF/TMR4IF/TMR6IF interrupt flag bit of the PIRx register. See Note below.
   - Configure the CKPS bits of the TxCON register with the Timer prescale value.
   - Enable the Timer by setting the ON bit of the TxCON register.
7. Enable PWM output pin:
   - Wait until the Timer overflows and the TMR2IF/TMR4IF/TMR6IF bit of the PIRx register is set. See Note below.
   - Enable the CCPx pin output driver by clearing the associated TRIS bit.

Note:	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.

REGISTER 26-4: CCPRxH: CCPx HIGH BYTE 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
CCPR<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 Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0

MODE = Capture Mode
CCPRxH<7:0>: MSB of captured TMR1 value
MODE = Compare Mode
CCPRxH<7:0>: MSB compared to TMR1 value
MODE = PWM Mode && FMT = 0
CCPRxH<7:2>: Not used
CCPRxH<1:0>: CCPW<9:8> — Pulse width Most Significant two bits
MODE = PWM Mode && FMT = 1
CCPRxH<7:0>: CCPW<9:2> — Pulse width Most Significant eight bits

REGISTER 26-5: CCPxCAP: CCPx CAPTURE INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CTS<2:0>		
bit 7							bit 0

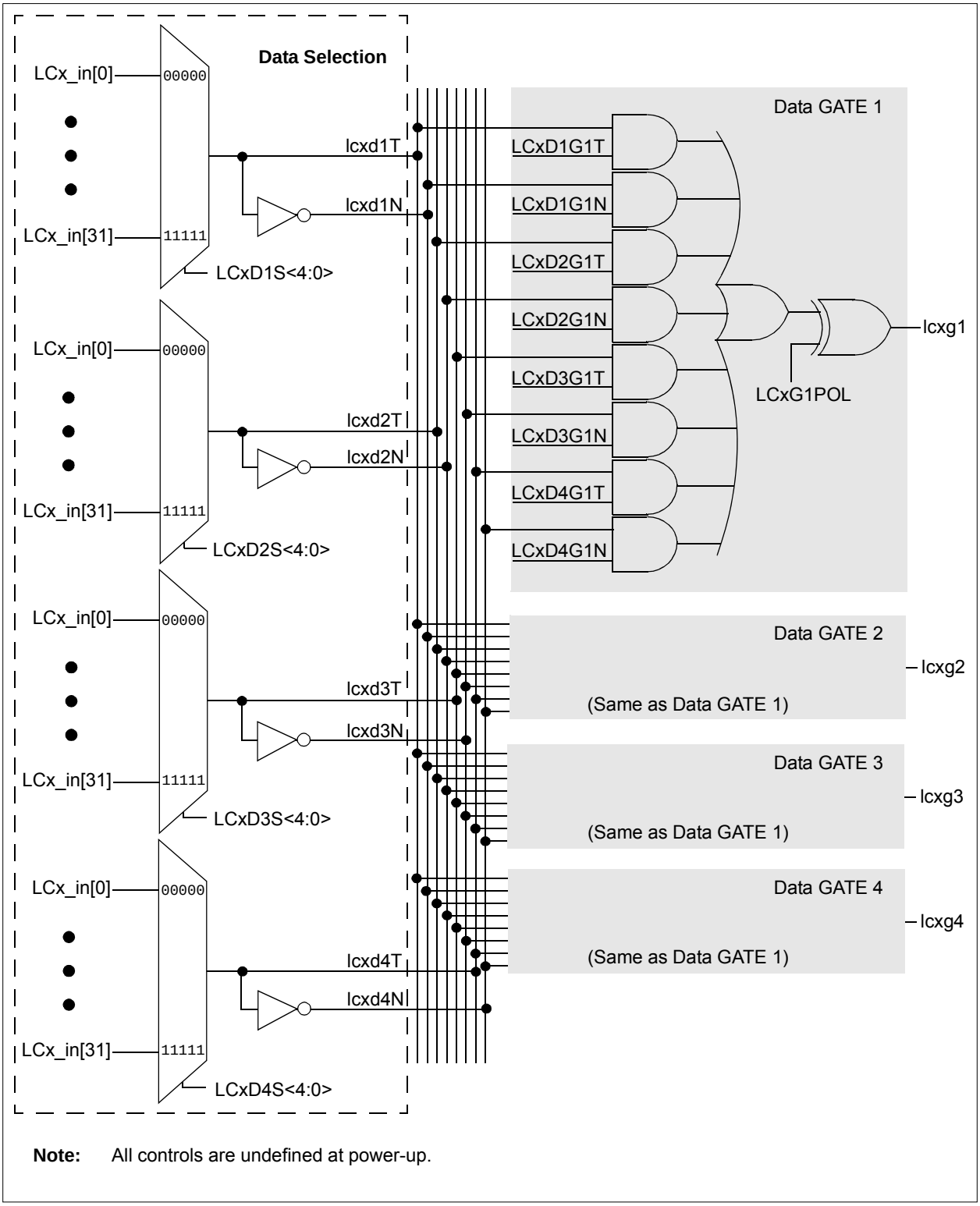
Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-3 **Unimplemented:** Read as '0'

bit 2-0 **CTS<2:0>:** Capture Trigger Input Selection bits

111 = Reserved. No channel connected.
110 = Reserved. No channel connected.
101 = LC2_out
100 = LC1_out
011 = IOC_interrupt
010 = C2_OUT_sync
001 = C1_OUT_sync
000 = CCPx pin

FIGURE 29-2: INPUT DATA SELECTION AND GATING



Note: All controls are undefined at power-up.

REGISTER 29-2: CLCxPOL: SIGNAL POLARITY CONTROL REGISTER

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxPOL	—	—	—	LCxG4POL	LCxG3POL	LCxG2POL	LCxG1POL
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 **LCxPOL:** LCOUT Polarity Control bit
- 1 = The output of the logic cell is inverted
- 0 = The output of the logic cell is not inverted

bit 6-4 **Unimplemented:** Read as ‘0’

bit 3 **LCxG4POL:** Gate 4 Output Polarity Control bit
- 1 = The output of gate 4 is inverted when applied to the logic cell
- 0 = The output of gate 4 is not inverted

bit 2 **LCxG3POL:** Gate 3 Output Polarity Control bit
- 1 = The output of gate 3 is inverted when applied to the logic cell
- 0 = The output of gate 3 is not inverted

bit 1 **LCxG2POL:** Gate 2 Output Polarity Control bit
- 1 = The output of gate 2 is inverted when applied to the logic cell
- 0 = The output of gate 2 is not inverted

bit 0 **LCxG1POL:** Gate 1 Output Polarity Control bit
- 1 = The output of gate 1 is inverted when applied to the logic cell
- 0 = The output of gate 1 is not inverted

30.1 SMT Operation

The core of the module is the 24-bit counter, SMTxTMR combined with a complex data acquisition front-end. Depending on the mode of operation selected, the SMT can perform a variety of measurements summarized in Table 30-1.

30.1.1 CLOCK SOURCES

Clock sources available to the SMT include:

- FOSC
- FOSC/4
- HFINTOSC 16 MHz
- LFINTOSC
- MFINTOSC 31.25 kHz

The SMT clock source is selected by configuring the CSEL<2:0> bits in the SMTxCLK register. The clock source can also be prescaled using the PS<1:0> bits of the SMTxCON0 register. The prescaled clock source is used to clock both the counter and any synchronization logic used by the module.

30.1.2 PERIOD MATCH INTERRUPT

Similar to other timers, the SMT triggers an interrupt when SMTxTMR rolls over to '0'. This happens when SMTxTMR = SMTxPR, regardless of mode. Hence, in any mode that relies on an external signal or a window to reset the timer, proper operation requires that SMTxPR be set to a period larger than that of the expected signal or window.

30.2 Basic Timer Function Registers

The SMTxTMR time base and the SMTxCPW/SMTxPR/SMTxCPR buffer registers serve several functions and can be manually updated using software.

30.2.1 TIME BASE

The SMTxTMR is the 24-bit counter that is the center of the SMT. It is used as the basic counter/timer for measurement in each of the modes of the SMT. It can be reset to a value of 24'h00_0000 by setting the RST bit of the SMTxSTAT register. It can be written to and read from software, but it is not guarded for atomic access, therefore reads and writes to the SMTxTMR should only be made when the GO = 0, or the software should have other measures to ensure integrity of SMTxTMR reads/writes.

30.2.2 PULSE WIDTH LATCH REGISTERS

The SMTxCPW registers are the 24-bit SMT pulse width latch. They are used to latch in the value of the SMTxTMR when triggered by various signals, which are determined by the mode the SMT is currently in. The SMTxCPW registers can also be updated with the current value of the SMTxTMR value by setting the CPWUP bit of the SMTxSTAT register.

30.2.3 PERIOD LATCH REGISTERS

The SMTxCPR registers are the 24-bit SMT period latch. They are used to latch in other values of the SMTxTMR when triggered by various other signals, which are determined by the mode the SMT is currently in.

The SMTxCPR registers can also be updated with the current value of the SMTxTMR value by setting the CPRUP bit in the SMTxSTAT register.

30.3 Halt Operation

The counter can be prevented from rolling-over using the STP bit in the SMTxCON0 register. When halting is enabled, the period match interrupt persists until the SMTxTMR is reset (either by a manual reset, **Section30.2.1 "Time Base"**) or by clearing the SMTxGO bit of the SMTxCON1 register and writing the SMTxTMR values in software.

30.4 Polarity Control

The three input signals for the SMT have polarity control to determine whether or not they are active high/positive edge or active low/negative edge signals.

The following bits apply to Polarity Control:

- WSEL bit (Window Polarity)
- SSEL bit (Signal Polarity)
- CSEL bit (Clock Polarity)

These bits are located in the SMTxCON0 register.

30.5 Status Information

The SMT provides input status information for the user without requiring the need to deal with the polarity of the incoming signals.

30.5.1 WINDOW STATUS

Window status is determined by the WS bit of the SMTxSTAT register. This bit is only used in Windowed Measure, Gated Counter and Gated Window Measure modes, and is only valid when TS = 1, and will be delayed in time by synchronizer delays in non-Counter modes.

30.5.2 SIGNAL STATUS

Signal status is determined by the AS bit of the SMTxSTAT register. This bit is used in all modes except Window Measure, Time of Flight and Capture modes, and is only valid when TS = 1, and will be delayed in time by synchronizer delays in non-Counter modes.

30.5.3 GO STATUS

Timer run status is determined by the TS bit of the SMTxSTAT register, and will be delayed in time by synchronizer delays in non-Counter modes.

FIGURE 30-8: HIGH AND LOW MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

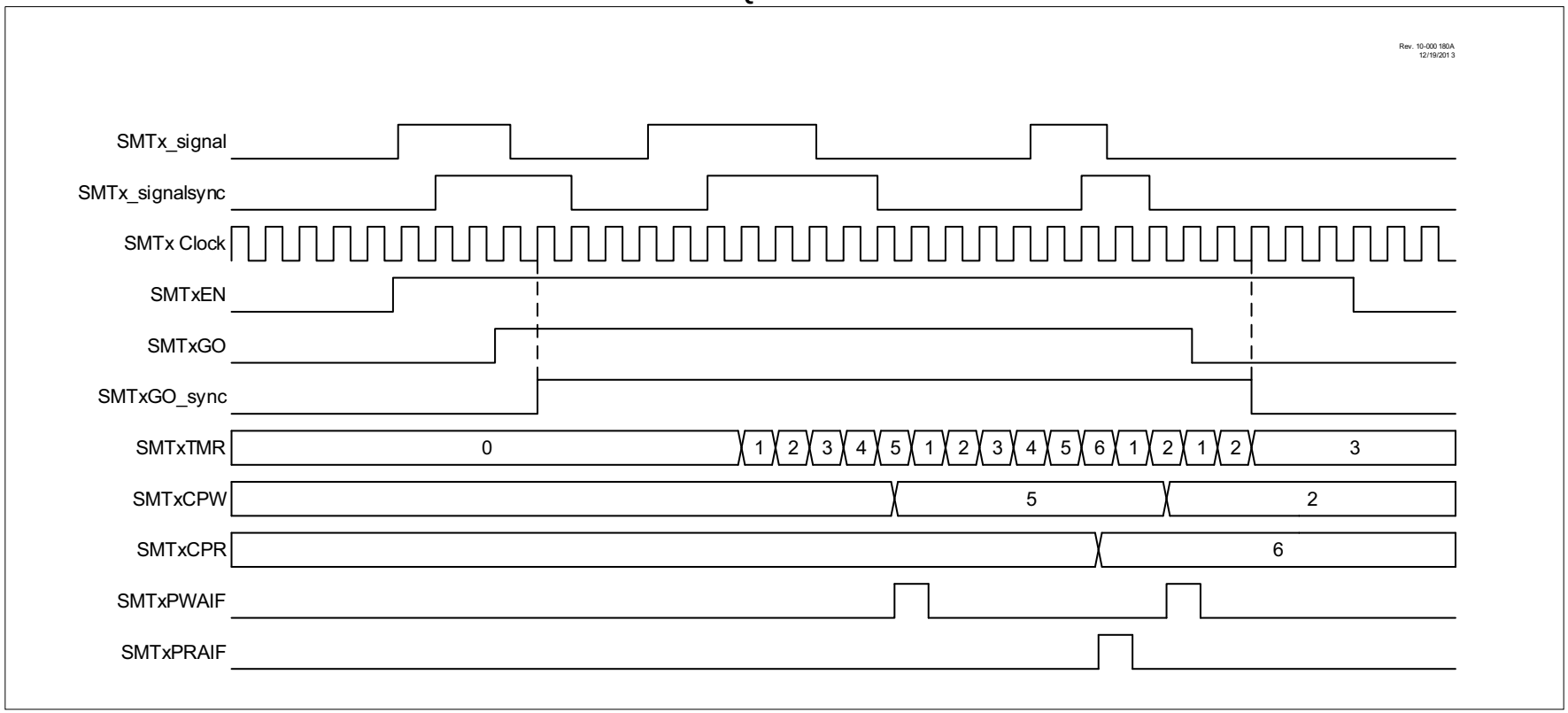


FIGURE 30-17: CAPTURE MODE SINGLE ACQUISITION TIMING DIAGRAM

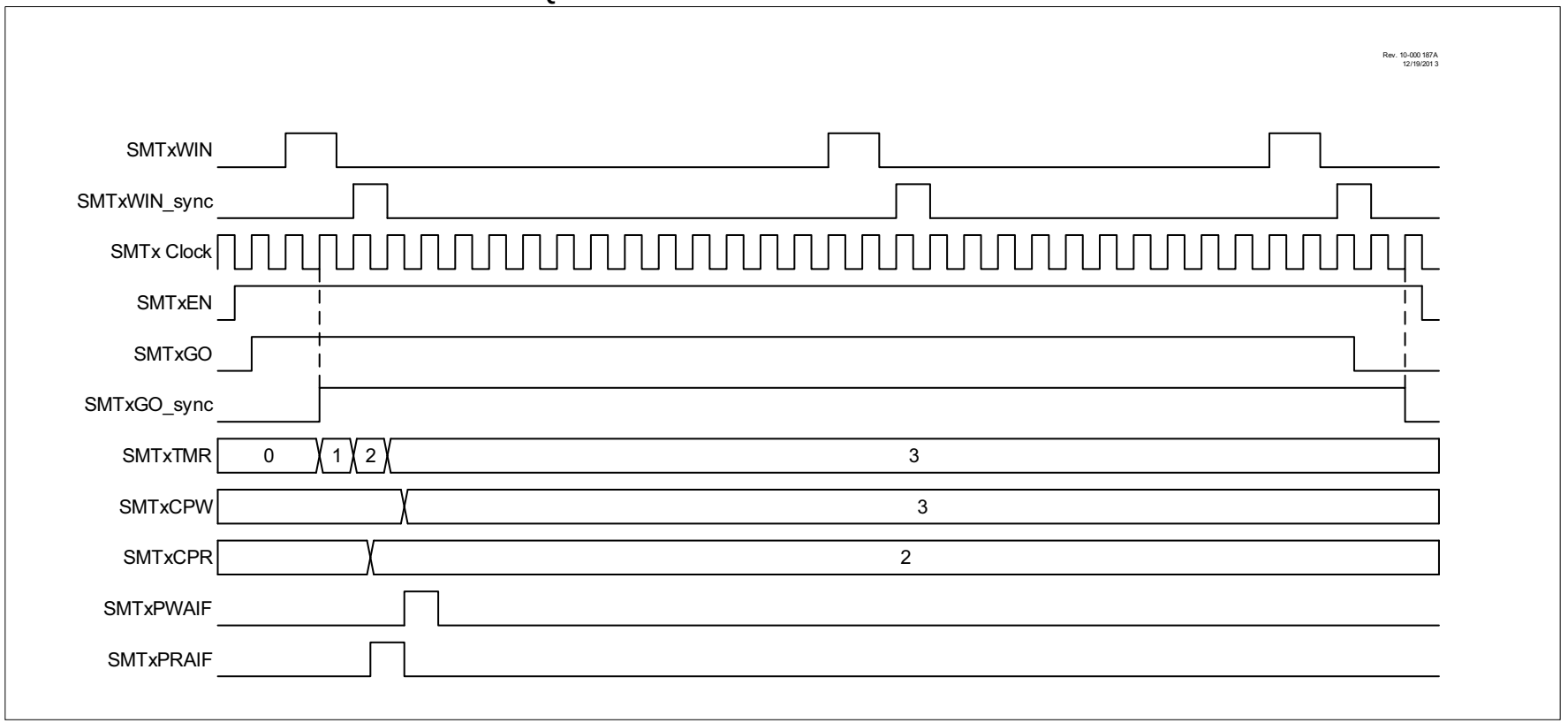
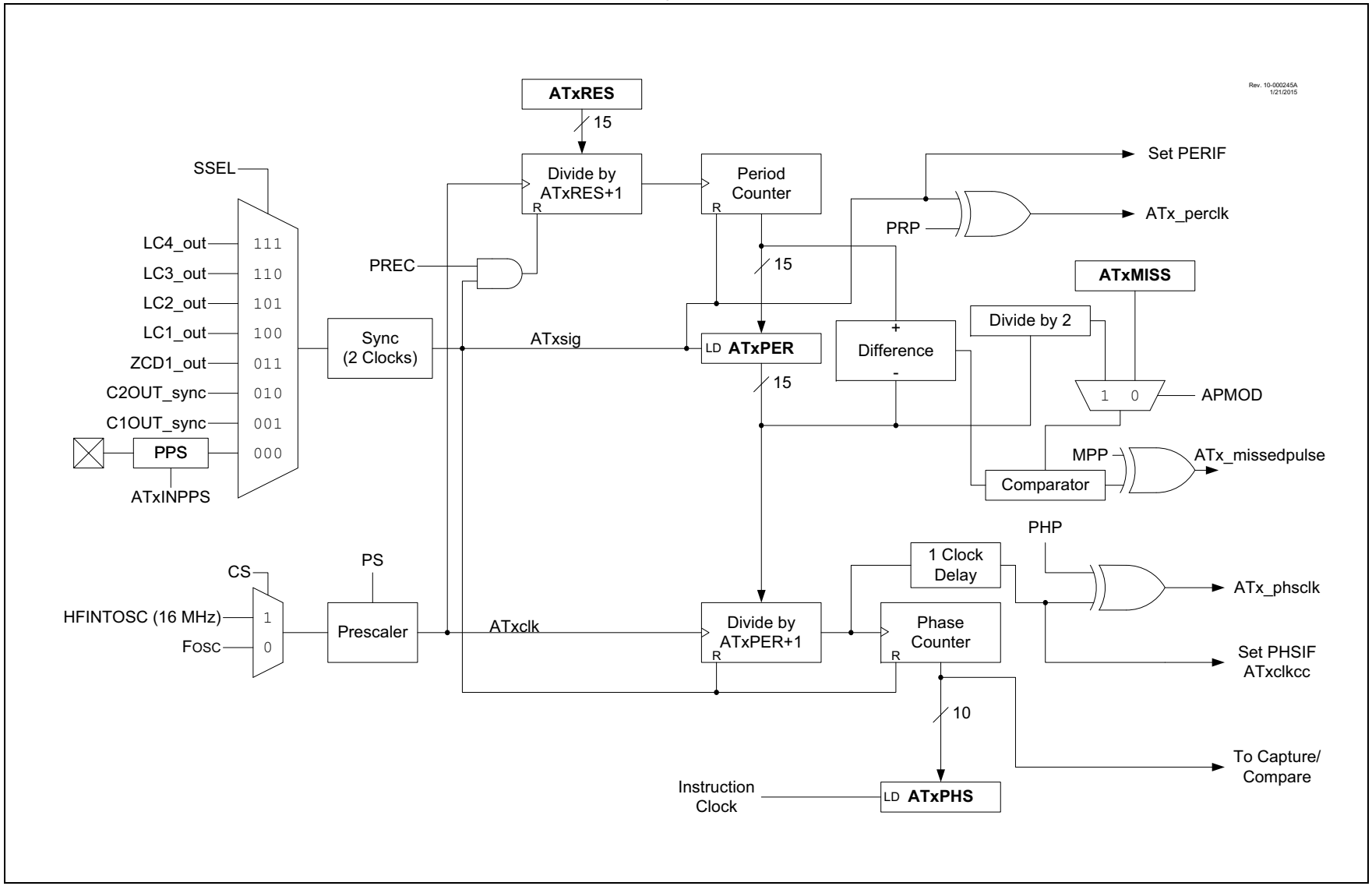


FIGURE 31-1: ANGULAR TIMER SIMPLIFIED BLOCK DIAGRAM, SINGLE-PULSE MODE



REGISTER 31-2: ATxCON1: ANGULAR TIMER CONTROL 1 REGISTER

U-0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R-0/0	R-0/0
—	PHP	—	PRP	—	MPP	ACCS	VALID
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 **Unimplemented:** Read as ‘0’

bit 6 **PHP:** Phase Clock Output Polarity bit
1 = Phase clock output is active-low
0 = Phase clock output is active-high

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **PRP:** Period Clock Output Polarity bit
1 = Period clock output is active-low
0 = Period clock output is active-high

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **MPP:** Missing Pulse Output Polarity bit
1 = Missing pulse output is active-low
0 = Missing pulse output is active-high

bit 1 **ACCS:** Acceleration Sign bit
1 = The value currently in ATxPER is less than the previous value
0 = The value currently in ATxPER is greater than or equal to the previous value

bit 0 **VALID:** Valid Measurement bit
1 = Sufficient input cycles have occurred to make ATxPER and ATxPHS valid.
0 = The values in ATxPER and ATxPHS are not valid; not enough input cycles have occurred

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.

Within the controller, the input is subtracted from a pre-programmed set point to get an error value. This error value, along with the previous two error values (if any), are multiplied by user-input coefficients and the results of these multiplications are added together to make up the output. If the MODE<2:0> bits of the PIDxCON register = 101, the PID output is equal to the current output added to any previous outputs.

The three user-input coefficients (K1, K2, and K3) are derived from the three classic PID coefficients Kp, Ki, and Kd, and must be calculated prior to using the PID module.

1. K1 is the coefficient that is multiplied with the current error (SET-IN). It is defined by the following equation:

EQUATION 32-1:

$$K1 = Kp + Ki \bullet T + \frac{Kd}{T}$$

Note: T is the sampling period.

2. K2 is the coefficient that is multiplied with the previous iteration's error (Z1). Where T is the sampling period, it is defined by the following equation:

EQUATION 32-2:

$$K2 = -\left(Kp + \frac{2Kd}{T}\right)$$

Note: T is the sampling period.

3. K3 is the coefficient that is multiplied with the error that occurred two iterations previous to the current one (Z2). It is defined by the following equation:

EQUATION 32-3:

$$K3 = \frac{Kd}{T}$$

Note: T is the sampling period.

To operate the module in PID controller mode, perform the following steps:

1. Set the MODE<2:0> bits of the PIDxCON register to '101', then set the EN bit of the PIDxCON register.
2. Write the previously calculated K1, K2, and K3 values to the PIDxK1, PIDxK2, and PIDxK3 registers, respectively.
3. Write the desired set point that the input will be compared against to the PIDxSET registers.
4. Write the high byte of the value from the external system to PIDxINH. Then write the low byte of the value from the external system to PIDxINL. This will begin the calculation and set the BUSY bit of the PIDxCON register.
5. Either poll the BUSY bit of the PIDxCON register to check for it clearing or wait for the PIDxDIF interrupt to trigger, indicating that the operation has completed.
6. Read the PIDxOUT registers for the output value. If the PID was in Accumulation mode, PIDxOUT will contain the accumulation of the output added to the previous outputs, otherwise, it will contain only the latest output.
7. For proper PID operation, this output needs to be applied to the external system before the next input to the PID is applied. This is to ensure that the system can adjust based on the PID controller's feedback before the next calculation is made.

Note: The BUSY bit of the PIDxCON register goes high as soon as PIDxINL is written and remains high until all computation is complete. Until the BUSY bit goes low, the PIDxOUT values are not valid, and none of the registers associated with the PID module should be written to, as any such writes will corrupt the calculation.

32.1.2 CONTEXT SAVING

It is possible to save the current state of the PID controller in software and restore it at a later time. In order to perform this, a calculation must not currently be active (BUSY = 0). Saving the PIDxOUT, PIDxZ1, and PIDxZ2 values elsewhere in memory will save the current state of the PID controller, although it may be desirable to also save PIDxK1, PIDxK2, PIDxK3, and/or PIDxSET, depending on the application. At the desired later time, these values can be written back into their respective registers, writing PIDxINL last, and the PID will continue from its previous state.

35.4 AC Characteristics

Timing Parameter Symbology has been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

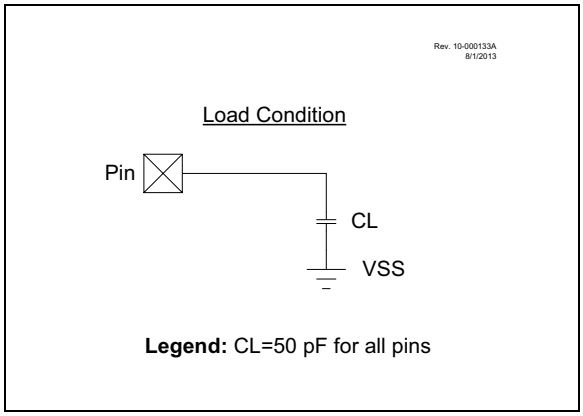
Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	CLKIN
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDIx	sc	SCKx
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O PORT	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 35-4: LOAD CONDITIONS



Legend: CL=50 pF for all pins