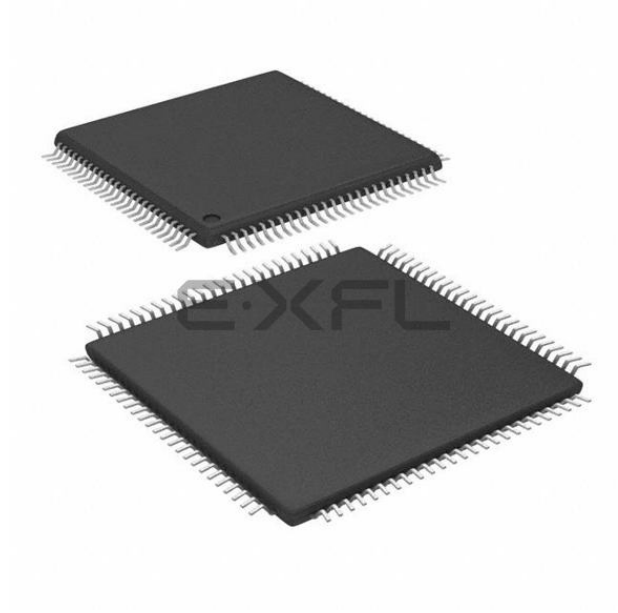


Welcome to E-XFL.COM

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	64MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT
Number of I/O	86
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 24x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f96j94t-i-pt

TABLE 1-3: DEVICE FEATURES FOR THE 100-PIN DEVICES

Features	PIC18F95J94	PIC18F96J94	PIC18F97J94
Operating Frequency	DC – 64 MHz		
Program Memory (Bytes)	32 K	64K	128K
	(Up to 2 Mbytes with Extended Memory)		
Program Memory (Instructions)	16,384	32,768	65,536
Data Memory (Bytes)	4K	4K	4K
Interrupt Sources	42	48	
I/O Ports	Ports A, B, C, D, E, F, G, H, J, K, L		
Parallel Communications	Parallel Slave Port (PSP)		
Timers	8		
Comparators	3		
LCD	480 pixels		
CTMU	Yes		
RTCC	Yes		
Enhanced Capture/Compare/PWM Modules	3 ECCPs and 7 CCPs		
Serial Communications	Two MSSPs, Four Enhanced USARTs (EUSART) and USB		
12-Bit Analog-to-Digital Module	24 Input Channels		
Resets (and Delays)	POR, BOR, CM RESET Instruction, Stack Full, Stack Underflow, $\overline{\text{MCLR}}$, WDT (PWRT, OST)		
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled		
Packages	100-Pin TQFP		

3.1 CPU Clocking Scheme

The system clock source can be provided by one of four sources:

- Primary Oscillator (POSC) on the OSC1 and OSC2 pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins
- Fast Internal RC (FRC) Oscillator
- Low-Power Internal RC (LPRC) Oscillator

The Primary Oscillator and FRC sources have the option of using the internal USB PLL block, which generates both the USB module clock and a separate system clock from the 96 MHz PLL. Refer to **Section 3.8.1 "Oscillator Modes and USB Operation"** for additional information.

The internal FRC provides an 8 MHz clock source. It can optionally be reduced by the programmable clock divider to provide a range of system clock frequencies.

The selected clock source generates the processor and peripheral clock sources. The processor clock source is divided by four to produce the internal instruction cycle clock, FCY. In this document, the instruction cycle clock is also denoted by FOSC/4. The internal instruction cycle clock, FOSC/4, can be provided on the OSC2 I/O pin for some operating modes of the Primary Oscillator. The timing diagram in Figure 3-2 shows the relationship between the processor clock source and instruction execution.

FIGURE 3-2: CLOCK OR INSTRUCTION CYCLE TIMING

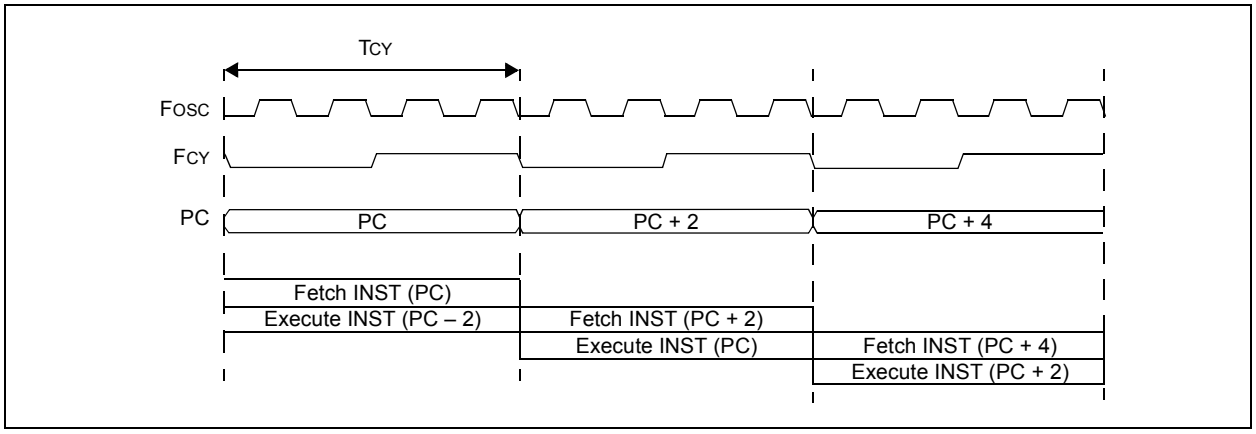


TABLE 6-2: REGISTER FILE SUMMARY (CONTINUED)

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E36h	RPINR24_25	IOC7R<3:0>				IOC6R<3:0>			
E35h	RPINR22_23	IOC5R<3:0>				IOC4R<3:0>			
E34h	RPINR20_21	IOC3R<3:0>				IOC2R<3:0>			
E33h	RPINR18_19	IOC1R<3:0>				IOC0R<3:0>			
E32h	RPINR16_17	ECCP3R<3:0>				ECCP2R<3:0>			
E31h	RPINR14_15	ECCP1R<3:0>				FLT0R<3:0>			
E30h	RPINR12_13	SS2R<3:0>				SDI2R<3:0>			
E2Fh	RPINR10_11	SCK2R<3:0>				SS1R<3:0>			
E2Eh	RPINR8_9	SDI1R<3:0>				SCK1R<3:0>			
E2Dh	RPINR6_7	U4TXR<3:0>				U4RXR<3:0>			
E2Ch	RPINR4_5	U3TXR<3:0>				U3RXR<3:0>			
E2Bh	RPINR2_3	U2TXR<3:0>				U2RXR<3:0>			
E2Ah	RPINR0_1	U1TXR<3:0>				U1RXR<3:0>			
E29h	RPOR46	—	—	—	—	RPO46R<3:0>			
E28h	RPOR44_45	RPO45R<3:0>				RPO44R<3:0>			
E27h	RPOR42_43	RPO43R<3:0>				RPO42R<3:0>			
E26h	RPOR40_41	RPO41R<3:0>				RPO40R<3:0>			
E25h	RPOR38_39	RPO39R<3:0>				RPO38R<3:0>			
E24h	RPOR36_37	RPO37R<3:0>				RPO36R<3:0>			
E23h	RPOR34_35	RPO35R<3:0>				RPO34R<3:0>			
E22h	RPOR32_33	RPO33R<3:0>				RPO32R<3:0>			
E21h	RPOR30_31	RPO31R<3:0>				RPO30R<3:0>			
E20h	RPOR28_29	RPO29R<3:0>				RPO28R<3:0>			
E1Fh	RPOR26_27	RPO27R<3:0>				RPO26R<3:0>			
E1Eh	RPOR24_25	RPO25R<3:0>				RPO24R<3:0>			
E1Dh	RPOR22_23	RPP23R<3:0>				RPO22R<3:0>			
E1Ch	RPOR20_21	RPO21R<3:0>				RPO20R<3:0>			
E1Bh	RPOR18_19	RPO19R<3:0>				RPO18R<3:0>			
E1Ah	RPOR16_17	RPO17R<3:0>				RPO16R<3:0>			
E19h	RPOR14_15	RPO15R<3:0>				RPO14R<3:0>			
E18h	RPOR12_13	RPO13R<3:0>				RPO12R<3:0>			
E17h	RPOR10_11	RPO11R<3:0>				RPO10R<3:0>			
E16h	RPOR8_9	RPO9R<3:0>				RPO8R<3:0>			
E15h	RPOR6_7	RPO7R<3:0>				RPO6R<3:0>			
E14h	RPOR4_5	RPO5R<3:0>				RPO4R<3:0>			
E13h	RPOR2_3	RPO3R<3:0>				RPO2R<3:0>			
E12h	RPOR0_1	RPO1R<3:0>				RPO0R<3:0>			
E11h	UCFG	UTEYE	UOEMON	—	UPUEN	UTRDIS	FSEN	PPB1	PPB0
E10h	UIE	—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE
E0Fh	UEIE	BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE
E0Eh	UEP15	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E0Dh	UEP14	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E0Ch	UEP13	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E0Bh	UEP12	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E0Ah	UEP11	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E09h	UEP10	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E08h	UEP9	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E07h	UEP8	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E06h	UEP7	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E05h	UEP6	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
E04h	UEP5	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL

Legend: — = unimplemented, read as '0'.

REGISTER 10-17: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP	SSP2IP	BCL2IP	USBIP	BCL1IP	HLVDIP	TMR3IP	TMR3GIP
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 **OSCFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **SSP2IP:** Master Synchronous Serial Port 2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **BCL2IP:** Bus Collision Interrupt Priority bit (MSSP)
1 = High priority
0 = Low priority

bit 4 **USBIP:** USB Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **BCL1IP:** Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **HLVDIP:** High/Low-Voltage Detect Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR3GIP:** TMR3 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 11-4: PSPCON: PARALLEL SLAVE PORT CONTROL REGISTER

R-0	R-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
IBF	OBF	IBOV	PSPMODE	—	—	—	—
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 **IBF:** Input Buffer Full Status bit
1 = A word has been received and is waiting to be read by the CPU
0 = No word has been received

bit 6 **OBF:** Output Buffer Full Status bit
1 = The output buffer still holds a previously written word
0 = The output buffer has been read

bit 5 **IBOV:** Input Buffer Overflow Detect bit
1 = A write occurred when a previously input word had not been read (must be cleared in software)
0 = No overflow occurred

bit 4 **PSPMODE:** Parallel Slave Port Mode Select bit
1 = Parallel Slave Port mode
0 = General Purpose I/O mode

bit 3-0 **Unimplemented:** Read as ‘0’

FIGURE 11-4: PARALLEL SLAVE PORT WRITE WAVEFORMS

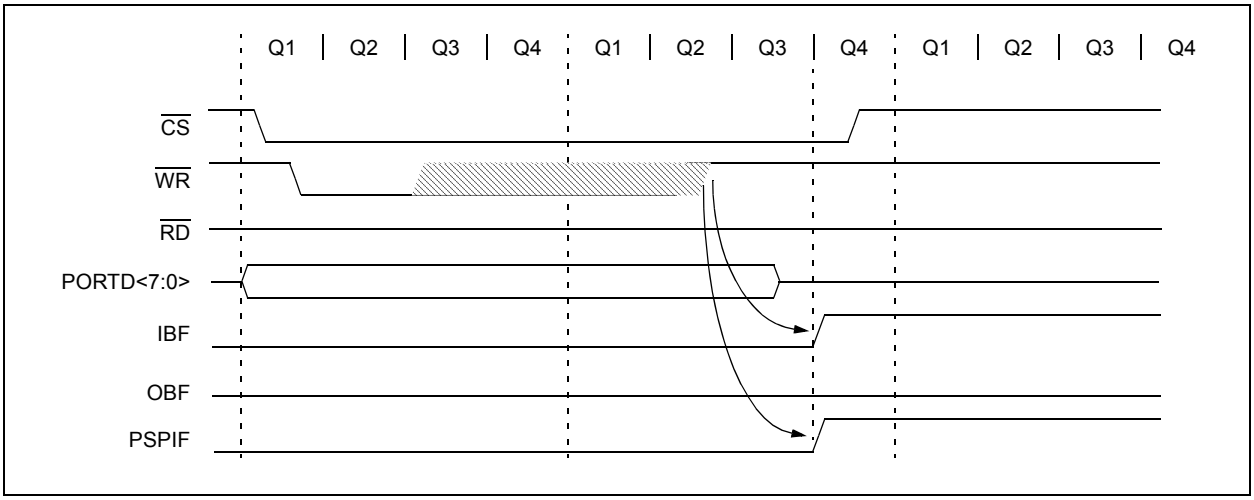


FIGURE 11-6: STRUCTURE OF PORT SHARED WITH PPS PERIPHERALS

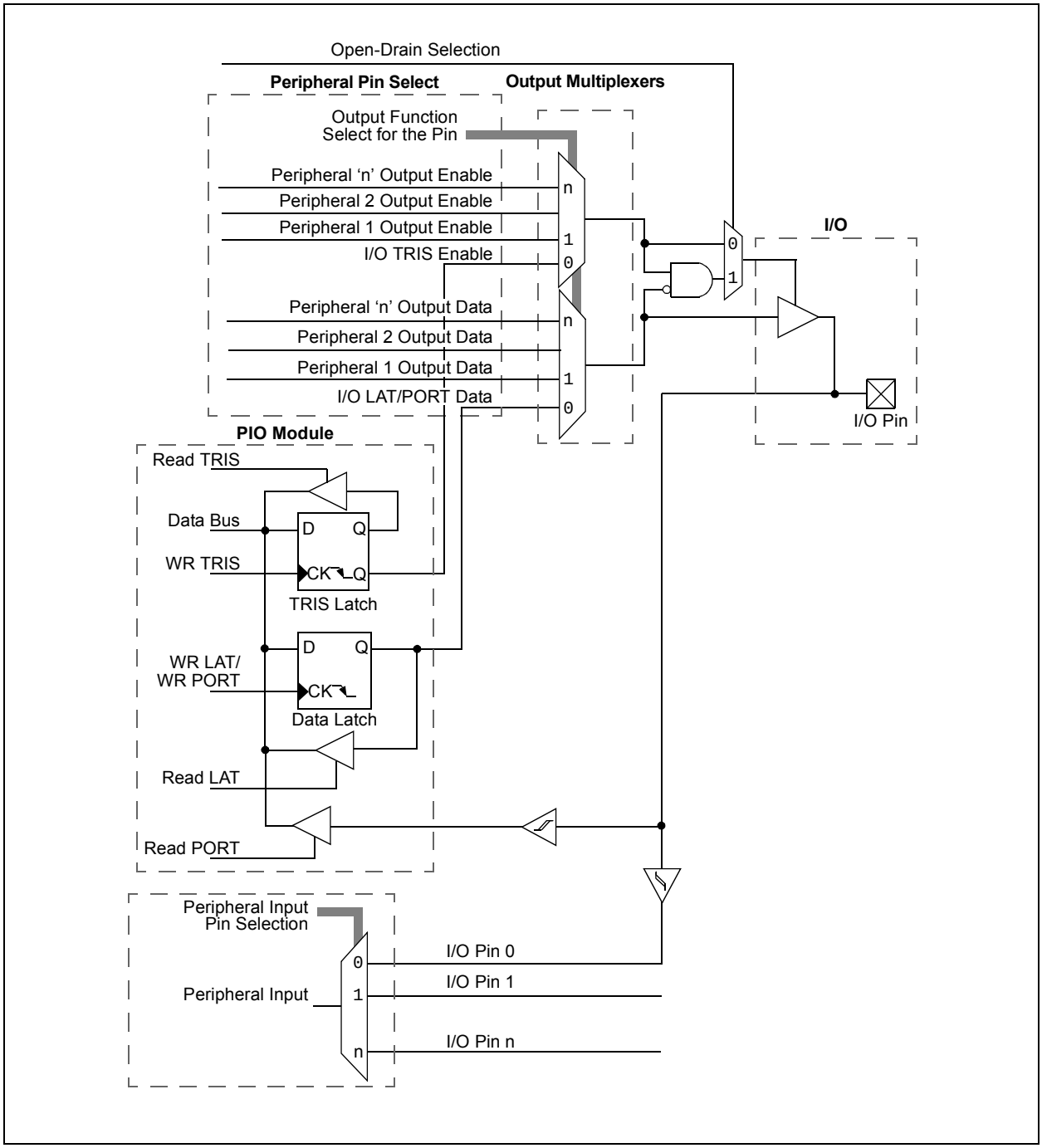


TABLE 11-13: RPIN REGISTERS AND AVAILABLE FUNCTIONS (CONTINUED)

PPS-Lite Input Peripheral Group 4n + 2	
(1) To Map this Signal	(4) to the Associated RPIN Register
$\overline{SS1}$	RPINR10_11<3:0>
INT2	RPINR26_27<7:4>
IOC2	RPINR20_21<3:0>
IOC6	RPINR24_25<3:0>
MDMIN	RPINR28_29<7:4>
U1TX	RPINR0_1<7:4>
U2RX	RPINR2_3<3:0>
SCK2	RPINR10_11<7:4>
ECCP3	RPINR16_17<7:4>
CCP6	RPINR34_35<3:0>
CCP10	RPINR38_39<3:0>
RVP2	RPINR48_49<3:0>
RVP6	RPINR52_53<3:0>
(2) with this RPn Pin	(3) Write this Corresponding Value
RP2	h'0
RP6	h'1
RP10	h'2
RP14	h'3
RP18	h'4
RP22	h'5
RP26	h'6
RP30	h'7
RP34	h'8
RP38	h'9
RP42	h'A
RP46	h'B
—	h'C
—	h'D
—	h'E
VSS	h'F

PPS-Lite Input Peripheral Group 4n + 3	
(1) To Map this Signal	(4) to the Associated RPIN Register
$\overline{SS2}$	RPINR12_13<7:4>
INT3	RPINR28_29<3:0>
IOC3	RPINR20_21<7:4>
IOC7	RPINR24_25<7:4>
U1RX	RPINR0_1<3:0>
U2TX	RPINR2_3<7:4>
SCK1	RPINR8_9<3:0>
ECCP1	RPINR14_15<7:4>
ECCP2	RPINR16_17<3:0>
CCP4	RPINR32_33<3:0>
RVP3	RPINR48_49<7:4>
RVP7	RPINR52_53<7:4>
(2) with this RPn Pin	(3) Write this Corresponding Value
RP3	h'0
RP7	h'1
RP11	h'2
RP15	h'3
RP19	h'4
RP23	h'5
RP27	h'6
RP31	h'7
RP35	h'8
RP39	h'9
RP43	h'A
—	h'B
—	h'C
—	h'D
—	h'E
VSS	h'F

11.15.3.2 Output Mapping

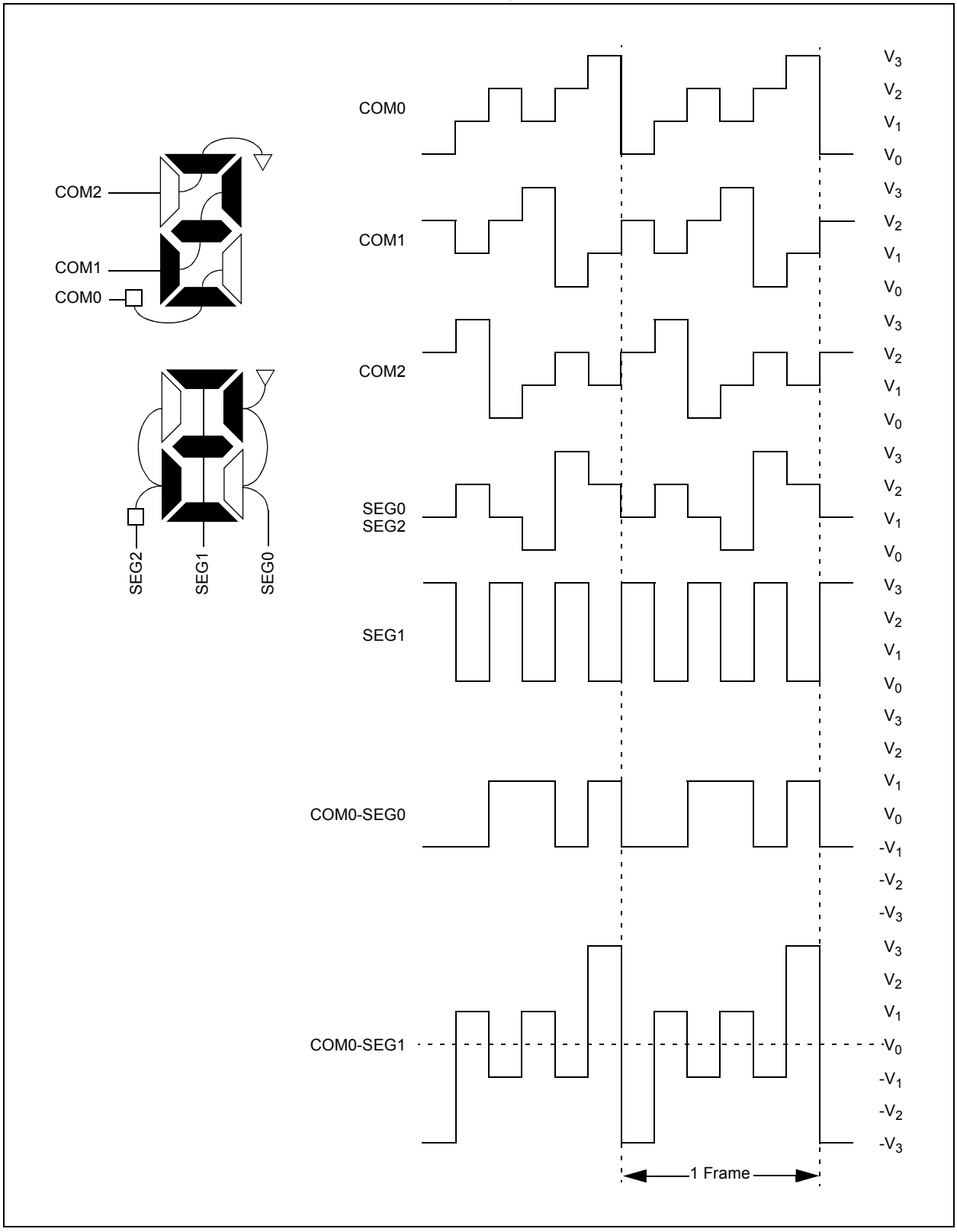
In contrast to the inputs, the outputs of the Peripheral Pin Select options are mapped on the basis of the pin. In this case, a bit field associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers contain sets of 4-bit fields, with each associated with one RPn pin (see Register 11-5). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin. Each pin has a limited set of peripherals to choose from.

The PPS-Lite peripheral outputs and associated RPn pins have been organized into four groups. It is not possible to map a peripheral to an RPn pin which is outside of its group. To map a peripheral output signal to an RPn pin, use the 4-step process, as indicated in Table 11-14. Choose the RPn pin and the signal; the column on the right shows which value to write to the associated RPORx register.

The peripheral outputs that support Peripheral Pin Selection have no default pins. Since the RPORx registers reset to all '0's, the outputs are all disconnected in the device's default (Reset) state.

The list of peripherals for output mapping also includes a null value of b'0000' because of the mapping technique. This allows unused peripherals to not be connected to a pin. Not all peripherals are available on all pins. For example, the "SDO2" signal is only available on RP0, RP4, RP8, etc. The "SDO2" signal is not available on RP1.

FIGURE 13-16: TYPE-A WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE

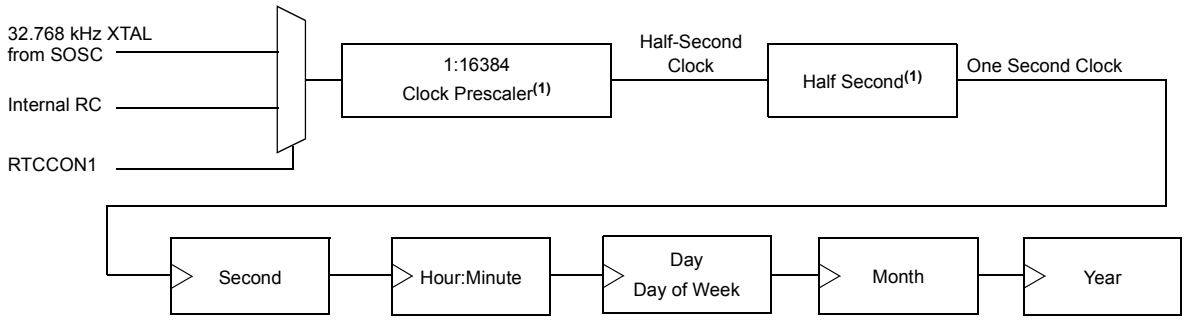


17.2.2 CLOCK SOURCE

As mentioned earlier, the RTCC module is intended to be clocked by an external Real-Time Clock (RTC) crystal, oscillating at 32.768 kHz, but an internal oscillator can be used. The RTCC clock selection is decided by the RTCOSC bit (CONFIG3L<0>).

Calibration of the crystal can be done through this module to yield an error of 3 seconds or less per month. (For further details, see **Section 17.2.9 "Calibration"**.)

FIGURE 17-4: CLOCK SOURCE MULTIPLEXING



Note 1: Writing to the lower half of the MINSEC register resets all counters, allowing fraction of a second synchronization; clock prescaler is held in Reset when RTCEN = 0.

17.2.2.1 Real-Time Clock Enable

The RTCC module can be clocked by an external, 32.768 kHz crystal (SOSC Oscillator) or the LF-INTOSC Oscillator, which can be selected in CONFIG3L<0>.

If the external clock is used, the SOSC Oscillator should be enabled. If LF-INTOSC is providing the clock, the INTOSC clock can be brought out to the RTCC pin by the RTSECSEL<1:0> bits (RTCCON2<1:0>).

17.2.3 DIGIT CARRY RULES

This section explains which timer values are affected when there is a rollover:

- Time of Day: From 23:59:59 to 00:00:00 with a carry to the Day field
- Month: From 12/31 to 01/01 with a carry to the Year field
- Day of Week: From 6 to 0 with no carry (see Table 17-1)
- Year Carry: From 99 to 00; this also surpasses the use of the RTCC

For the day-to-month rollover schedule, see Table 17-2.

Because the following values are in BCD format, the carry to the upper BCD digit occurs at the count of 10, not 16 (SECONDS, MINUTES, HOURS, WEEKDAY, DAYS and MONTHS).

TABLE 17-1: DAY OF WEEK SCHEDULE

Day of Week	
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

TABLE 17-2: DAY TO MONTH ROLLOVER SCHEDULE

Month	Maximum Day Field
01 (January)	31
02 (February)	28 or 29(1)
03 (March)	31
04 (April)	30
05 (May)	31
06 (June)	30
07 (July)	31
08 (August)	31
09 (September)	30
10 (October)	31
11 (November)	30
12 (December)	31

Note 1: See **Section 17.2.4 "Leap Year"**.

20.5.9 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPMx bits in SSPxCON1, and by setting the SSPEN bit. In Master mode, the SCLx and SDAx lines are manipulated by the MSSPx hardware if the TRIS bits are set.

The Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I^2C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options.

1. Assert a Start condition on SDAx and SCLx.
2. Assert a Repeated Start condition on SDAx and SCLx.
3. Write to the SSPxBUF register initiating transmission of data/address.
4. Configure the I^2C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDAx and SCLx.

Note: The MSSPx module, when configured in I^2C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur.

The following events will cause the MSSPx Interrupt Flag bit, SSPxIF, to be set (and MSSPx interrupt if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmitted
- Repeated Start

FIGURE 20-18: MSSPx BLOCK DIAGRAM (I^2C MASTER MODE)

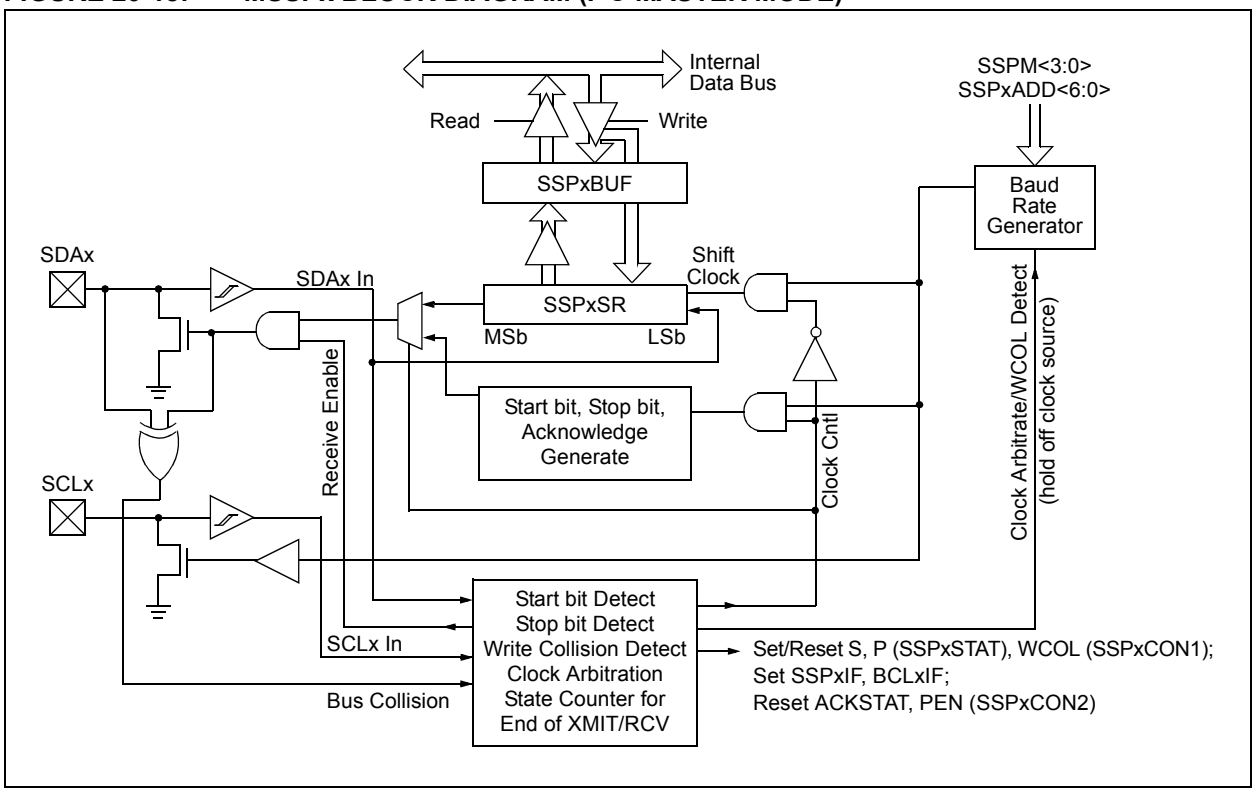


TABLE 21-2: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	FOSC = 40.000 MHz			FOSC = 20.000 MHz			FOSC = 10.000 MHz			FOSC = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	FOSC = 4.000 MHz			FOSC = 2.000 MHz			FOSC = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	FOSC = 40.000 MHz			FOSC = 20.000 MHz			FOSC = 10.000 MHz			FOSC = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	FOSC = 4.000 MHz			FOSC = 2.000 MHz			FOSC = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

21.5.2.3 IrDA Decoder Function

The decoder works by taking the serial data from the RXx pin and replacing it with the decoded data stream. The stream is decoded based on falling edge detection of the RXx input.

Each falling edge of RXx causes the decoded data to be driven low for 16 periods of the 16x Baud Clock. If another falling edge has been detected by the time the 16 periods expire, the decoded data remains low for another 16 periods. If no falling edge was detected, the decoded data is driven high.

Note that the data stream into the device is shifted anywhere from 7 to 8 periods of the 16x Baud Clock from the actual message source. The one clock uncertainty is due to the clock edge resolution. See Figure 21-18 for details.

21.5.2.4 IrDA Receive Polarity

The IrDA receive polarity is selected using the RXDTP bit. This bit only affects the receive encoder and does not affect the transmitter.

When RXDTP = 0, the Idle state of the RXx line is '1' (see Figure 21-18). When RXDTP = 1, the Idle state of the RXx line is '0' (see Figure 21-19).

21.5.2.5 Clock Jitter

Due to jitter or slight frequency differences between devices, it is possible for the next falling bit edge to be missed for one of the 16x periods. In that case, one clock-wide pulse appears on the decoded data stream. Since the EUSARTx performs a majority detect around the bit center, this does not cause erroneous data. See Figure 21-20 for details.

FIGURE 21-18: MACRO VIEW OF IrDA® DECODING SCHEME (RXDTP = 0)

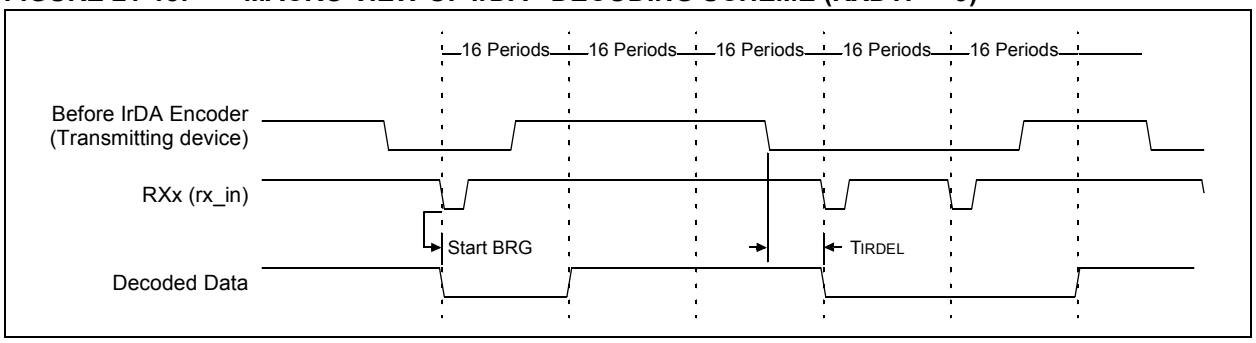


FIGURE 21-19: INVERTED POLARITY DECODING RESULTS (RXDTP = 1)

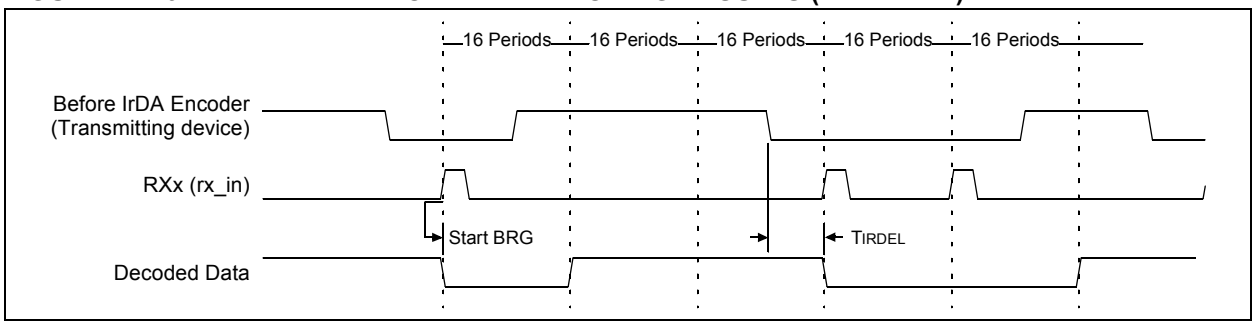


FIGURE 21-20: CLOCK JITTER CAUSING A PULSE BETWEEN CONSECUTIVE ZEROS

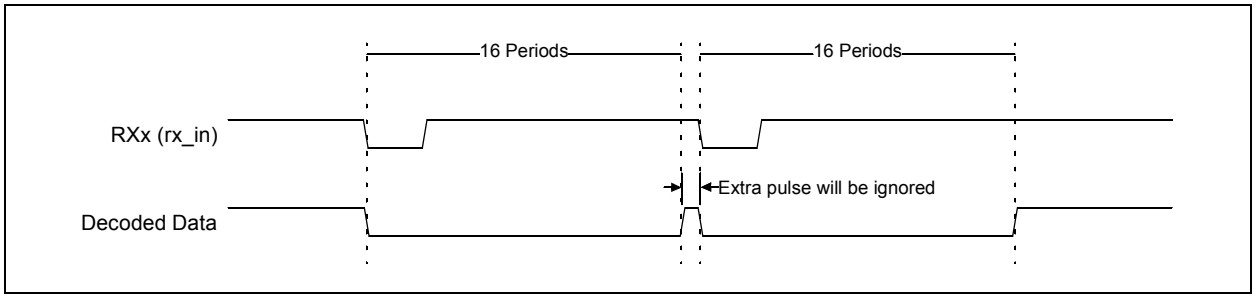
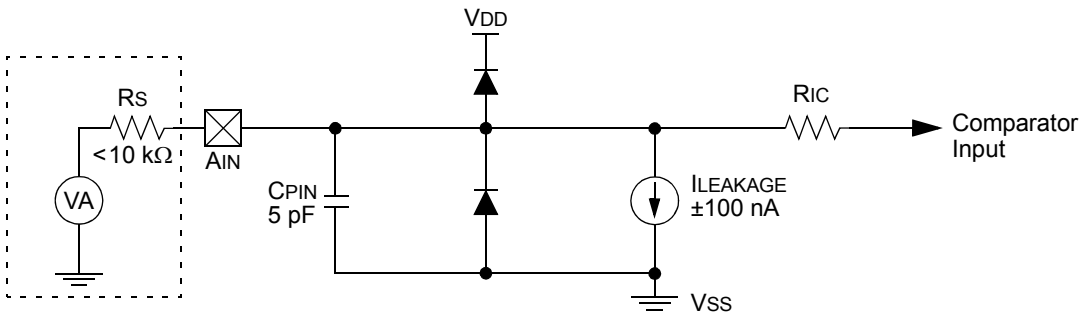


FIGURE 23-3: COMPARATOR ANALOG INPUT MODEL



Legend:

CPIN	=	Input Capacitance
VT	=	Threshold Voltage
ILEAKAGE	=	Leakage Current at the pin due to various junctions
RIC	=	Interconnect Resistance
RS	=	Source Impedance
VA	=	Analog Voltage

23.5 Comparator Control and Configuration

Each comparator has up to eight possible combinations of inputs: up to four external analog inputs and one of two internal voltage references.

All of the comparators allow a selection of the signal from pin, CxINA, or the voltage from the Comparator Reference (CVREF) on the non-inverting channel. This is compared to either CxINB, CxINC, C2IND or the microcontroller's fixed internal reference voltage (VBG, 1.2V nominal) on the inverting channel. The comparator inputs and outputs are tied to fixed I/O pins, defined in Table 23-1. The available comparator configurations and their corresponding bit settings are shown in Figure 23-4.

TABLE 23-1: COMPARATOR INPUTS AND OUTPUTS

Comparator	Input or Output	I/O Pin(†)
1	C1INA (VIN+)	RA5/RF6
	C1INB (VIN-)	RF5
	C1INC (VIN-)	RH6(2)
	C2INB(VIN-)	RF2
	CVREF (VIN+)	RF5
	C1OUT	RPn(1)
2	C2INA (VIN+)	RA5
	C2INB (VIN-)	RF2
	C2INC (VIN-)	RH4(2)
	C2IND (VIN-)	RH5(2)
	CVREF (VIN+)	RF5
	C2OUT	RPn(1)
3	C3INA (VIN+)	RA5/RG2
	C3INB (VIN-)	RG3
	C2INB (VIN-)	RF2
	C3INC (VIN-)	RG4
	CVREF (VIN+)	RF5
	C3OUT	RPn(1)

† The I/O pin is dependent on package type.

Note 1: These pins are remappable I/Os.

2: These pins are not available on 64-pin devices.

26.0 CHARGE TIME MEASUREMENT UNIT (CTMU)

The Charge Time Measurement Unit (CTMU) is a flexible analog module that provides accurate differential time measurement between pulse sources, as well as asynchronous pulse generation. By working with other on-chip analog modules, the CTMU can precisely measure time, capacitance and relative changes in capacitance or generate output pulses with a specific time delay. The CTMU is ideal for interfacing with capacitive-based sensors.

The module includes these key features:

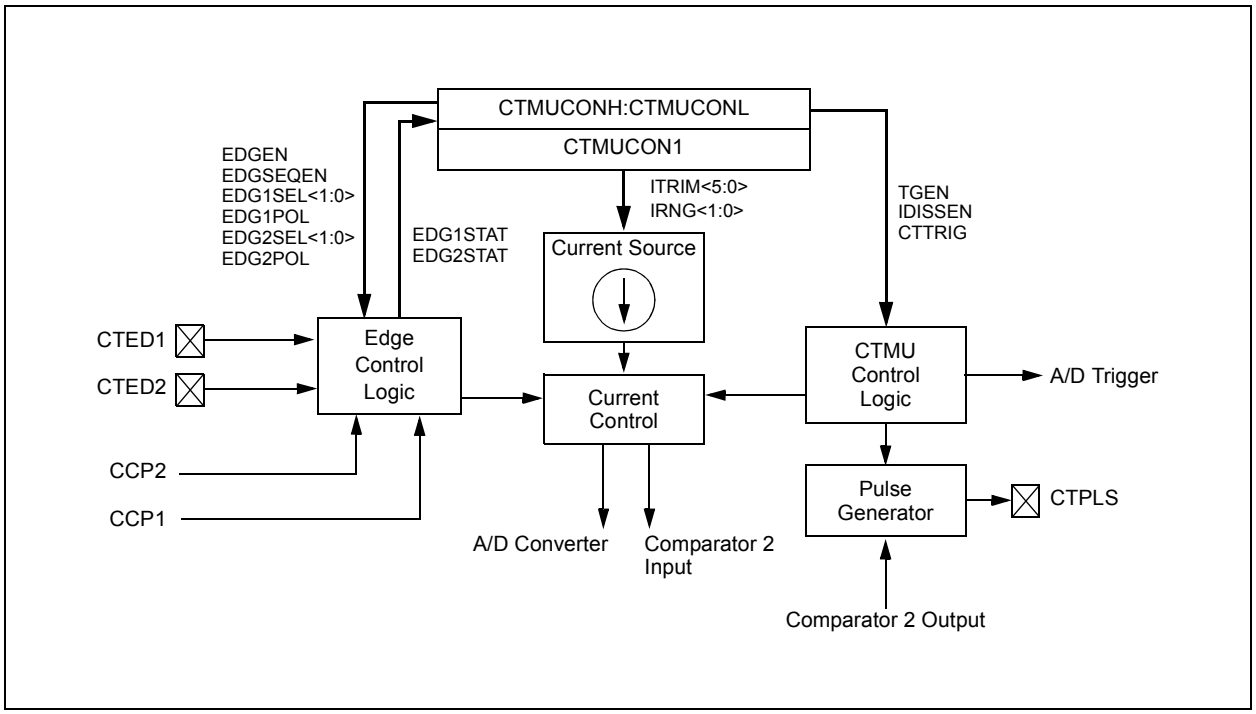
- Up to 24 channels available for capacitive or time measurement input
- Low-cost temperature measurement using on-chip diode channel
- On-chip precision current source
- Sixteen-edge input trigger sources
- Polarity control for each edge source
- Provides a trigger for the A/D Converter
- Control of edge sequence
- Control of response to edges
- Time measurement resolution of 1 nanosecond
- High-precision time measurement
- Time delay of external or internal signal asynchronous to system clock
- Accurate current source suitable for capacitive measurement

The CTMU works in conjunction with the A/D Converter to provide up to 24 channels for time or charge measurement, depending on the specific device and the number of A/D channels available. When configured for time delay, the CTMU is connected to one of the analog comparators. The level-sensitive input edge sources can be selected from four sources: two external inputs or the CCP1/CCP2 Special Event Triggers.

The CTMU special event can trigger the Analog-to-Digital Converter module.

Figure 26-1 provides a block diagram of the CTMU.

FIGURE 26-1: CTMU BLOCK DIAGRAM



27.2.2.2 Internal Pull-up Resistors

The PIC18FXXJ94 devices have built-in pull-up resistors, designed to meet the requirements for low-speed and full-speed USB. The UPUEN bit (UCFG<4>) enables the internal pull-ups. Figure 27-1 shows the pull-ups and their control.

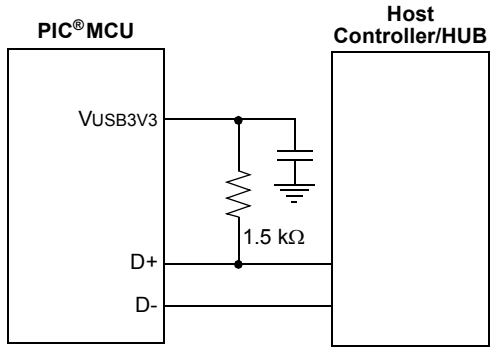
Note: A compliant USB device should never source any current onto the +5V VBUS line of the USB cable. Additionally, USB devices should not source any current on the D+ and D- data lines whenever the +5V VBUS line is less than 1.17V. In order to be USB compliant, applications which are not purely bus-powered should monitor the VBUS line, and avoid turning on the USB module and the D+ or D- pull-up resistor until VBUS is greater than 1.17V. VBUS can be connected to and monitored by a 5V tolerant I/O pin, or if a resistive divider is used, by an analog capable pin.

27.2.2.3 External Pull-up Resistors

External pull-ups may also be used. The VUSB3V3 pin may be used to pull up D+ or D-. The pull-up resistor must be 1.5 kΩ (±5%) as required by the USB specifications.

Figure 27-2 provides an example of external circuitry.

FIGURE 27-2: EXTERNAL CIRCUITRY



Note: The above setting shows a typical connection for a full-speed configuration using an on-chip regulator and an external pull-up resistor.

27.2.2.4 Ping-Pong Buffer Configuration

The usage of ping-pong buffers is configured using the PPB<1:0> bits. Refer to **Section 27.4.4 "Ping-Pong Buffering"** for a complete explanation of the ping-pong buffers.

27.2.2.5 Eye Pattern Test Enable

An automatic eye pattern test can be generated by the module when the UCFG<7> bit is set. The eye pattern output will be observable based on module settings, meaning that the user is first responsible for configuring the SIE clock settings, pull-up resistor and Transceiver mode. In addition, the module has to be enabled.

Once UTEYE is set, the module emulates a switch from a receive to transmit state and will start transmitting a J-K-J-K bit sequence (K-J-K-J for full speed). The sequence will be repeated indefinitely while the Eye Pattern Test mode is enabled.

Note that this bit should never be set while the module is connected to an actual USB system. This Test mode is intended for board verification to aid with USB certification tests. It is intended to show a system developer the noise integrity of the USB signals which can be affected by board traces, impedance mismatches and proximity to other system components. It does not properly test the transition from a receive to a transmit state. Although the eye pattern is not meant to replace the more complex USB certification test, it should aid during first order system debugging.

27.9 Overview of USB

This section presents some of the basic USB concepts and useful information necessary to design a USB device. Although much information is provided in this section, there is a plethora of information provided within the USB specifications and class specifications. Thus, the reader is encouraged to refer to the USB specifications for more information (www.usb.org). If you are very familiar with the details of USB, then this section serves as a basic, high-level refresher of USB.

27.9.1 LAYERED FRAMEWORK

USB device functionality is structured into a layered framework, graphically illustrated in Figure 27-12. Each level is associated with a functional level within the device. The highest layer, other than the device, is the configuration. A device may have multiple configurations. For example, a particular device may have multiple power requirements based on Self-Power Only or Bus Power Only modes.

For each configuration, there may be multiple interfaces. Each interface could support a particular mode of that configuration.

Below the interface is the endpoint(s). Data is directly moved at this level. There can be as many as 16 bidirectional endpoints. Endpoint 0 is always a control endpoint, and by default, when the device is on the bus, Endpoint 0 must be available to configure the device.

27.9.2 FRAMES

Information communicated on the bus is grouped into 1 ms time slots, referred to as frames. Each frame can contain many transactions to various devices and endpoints. See Figure 27-8 for an example of a transaction within a frame.

27.9.3 TRANSFERS

There are four transfer types defined in the USB specification.

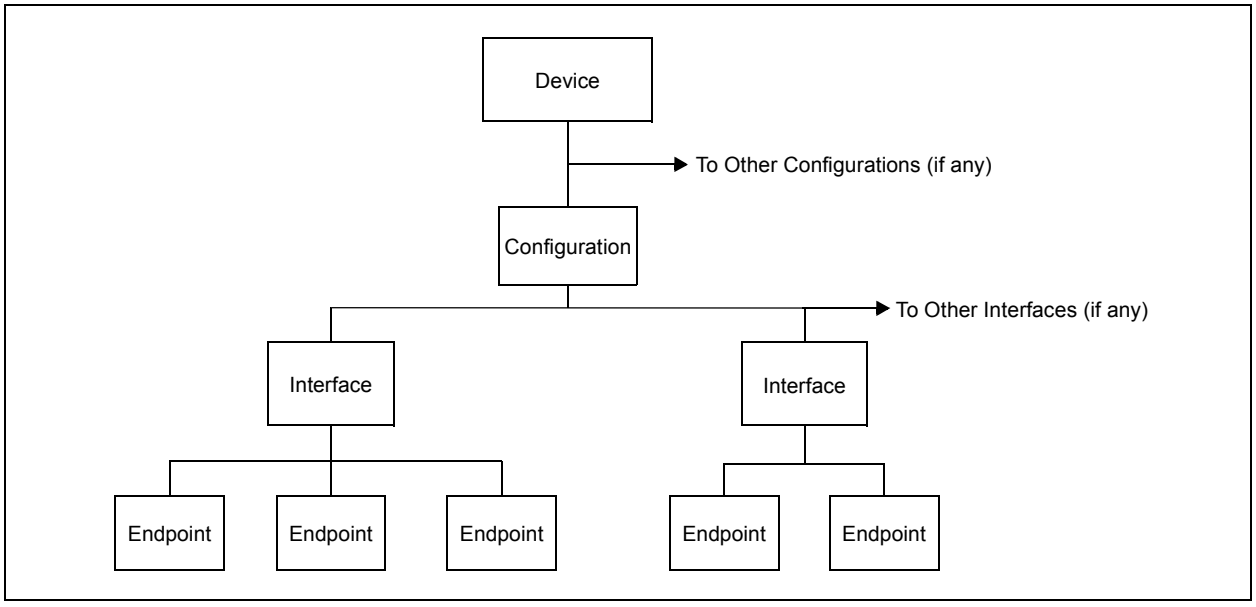
- **Isochronous:** This type provides a transfer method for large amounts of data (up to 1023 bytes) with timely delivery ensured; however, the data integrity is not ensured. This is good for streaming applications where small data loss is not critical, such as audio.
- **Bulk:** This type of transfer method allows for large amounts of data to be transferred with ensured data integrity; however, the delivery timeliness is not ensured.
- **Interrupt:** This type of transfer provides for ensured timely delivery for small blocks of data, plus data integrity is ensured.
- **Control:** This type provides for device setup control.

While full-speed devices support all transfer types, low-speed devices are limited to interrupt and control transfers only.

27.9.4 POWER

Power is available from the USB. The USB specification defines the bus power requirements. Devices may either be self-powered or bus-powered. Self-powered devices draw power from an external source, while bus-powered devices use power supplied from the bus.

FIGURE 27-12: USB LAYERS



DAW **Decimal Adjust W Register**

Syntax: DAW

Operands: None

Operation: If [W<3:0> > 9] or [DC = 1], then
(W<3:0>) + 6 → W<3:0>;
else
(W<3:0>) → W<3:0>

If [W<7:4> > 9] or [C = 1], then
(W<7:4>) + 6 → W<7:4>;
C = 1;
else
(W<7:4>) → W<7:4>

Status Affected: C

Encoding:

0000	0000	0000	0111

Description: DAW adjusts the 8-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register W	Process Data	Write W

Example 1: DAW

Before Instruction
W = A5h
C = 0
DC = 0
After Instruction
W = 05h
C = 1
DC = 0

Example 2:

Before Instruction
W = CEh
C = 0
DC = 0
After Instruction
W = 34h
C = 1
DC = 0

DECF **Decrement f**

Syntax: DECF f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

Operation: (f) – 1 → dest

Status Affected: C, DC, N, OV, Z

Encoding:

0000	01da	ffff	ffff

Description: Decrement register, 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

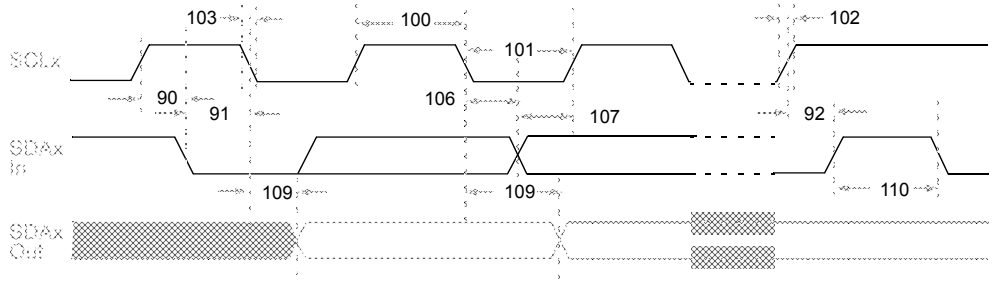
Q Cycle Activity:

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

Example: DECF CNT, 1, 0

Before Instruction
CNT = 01h
Z = 0
After Instruction
CNT = 00h
Z = 1

FIGURE 30-19: MSSPx I²C BUS DATA TIMING



Note: Refer to Figure 30-2 for load conditions.

TABLE 30-37: MSSPx I²C BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	
			400 kHz mode	2(TOSC)(BRG + 1)	—	—	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	—	
101	TLOW	Clock Low Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	
			400 kHz mode	2(TOSC)(BRG + 1)	—	—	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	—	
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(1)	—	300	ns	
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(1)	—	100	ns	
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	Only relevant for Repeated Start condition
			400 kHz mode	2(TOSC)(BRG + 1)	—	—	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	—	
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	After this period, the first clock pulse is generated
			400 kHz mode	2(TOSC)(BRG + 1)	—	—	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	—	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	—	
			400 kHz mode	0	0.9	μs	
			1 MHz mode(1)	—	μs	ns	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
			1 MHz mode(1)	—	—	ns	

Note 1: Maximum pin capacitance = 10 pF for all II²C pins.

2: A Fast mode I²C bus device can be used in a Standard mode I²C bus system, but Parameter #107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, Parameter #102 + Parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCLx line is released.

31.6 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.

31.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

31.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

31.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

31.10 MPLAB PM3 Device Programmer

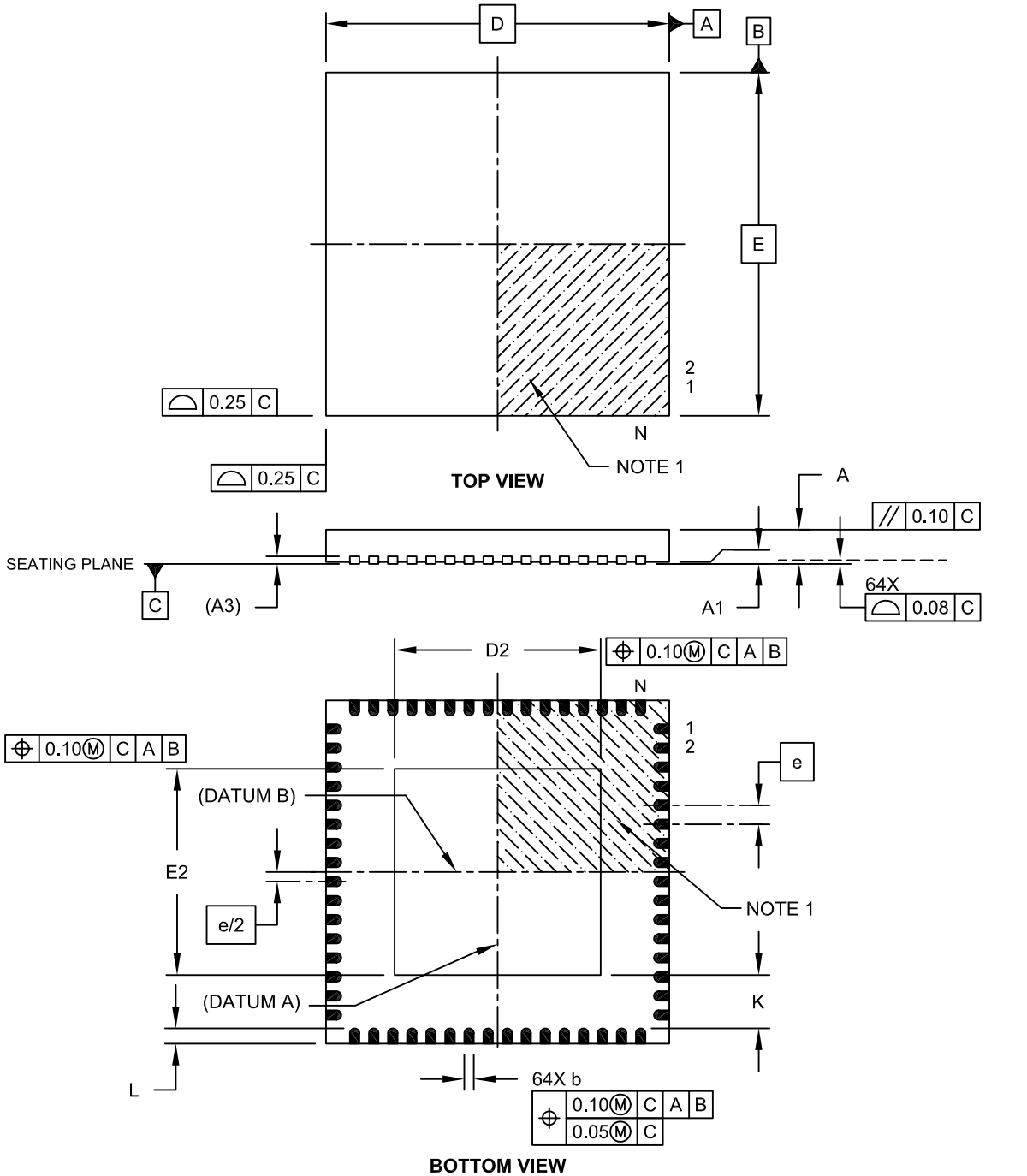
The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

33.2 Package Details

The following sections give the technical details of the packages.

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body with 5.40 x 5.40 Exposed Pad [QFN]

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



Microchip Technology Drawing C04-154A Sheet 1 of 2