Welcome to E-XFL.COM

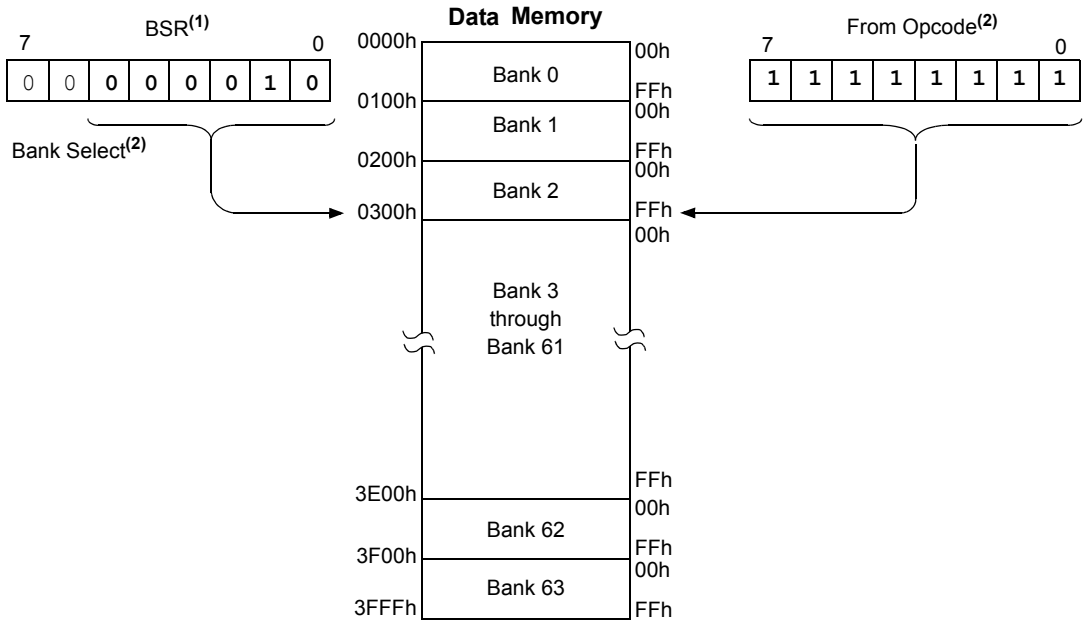
What is "Embedded - Microcontrollers"?

"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, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 43x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP Exposed Pad
Supplier Device Package	48-TQFP-EP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f57k42t-i-pt

FIGURE 4-5: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



Note 1: The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<5:0>) to the registers of the Access Bank.

REGISTER 5-3: CONFIGURATION WORD 2L (30 0002h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
BOREN<1:0>		$\overline{\text{LPBOREN}}$	IVT1WAY	MVECEN	PWRTS<1:0>		MCLRE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '1'	
-n = Value for blank device	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **BOREN<1:0>:** Brown-out Reset Enable bits

When enabled, Brown-out Reset Voltage (V_{BOR}) is set by the BORV bit.
11 = Brown-out Reset is enabled, SBOREN bit is ignored
10 = Brown-out Reset is enabled while running, disabled in Sleep; SBOREN is ignored
01 = Brown-out Reset is enabled according to SBOREN
00 = Brown-out Reset is disabled

bit 5 **$\overline{\text{LPBOREN}}$:** Low-Power BOR Enable bit

1 = Low-Power BOR is disabled
0 = Low-Power BOR is enabled

bit 4 **IVT1WAY:** IVTLOCK bit One-Way Set Enable bit

1 = IVTLOCKED bit can be cleared and set only once; IVT registers remain locked after one clear/set cycle
0 = IVTLOCK ED bit can be set and cleared multiple times (subject to the unlock sequence)

bit 3 **MVECEN:** Multi-vector Enable bit

1 = Multi-vector enabled; Vector table used for interrupts
0 = Legacy interrupt behavior

bit 2-1 **PWRTS<1:0>:** Power-up Timer Selection bits

11 = PWRT is disabled
10 = PWRT set at 64 ms (2048 LFINTOSC Cycles)
01 = PWRT set at 16 ms (512 LFINTOSC Cycles)
00 = PWRT set at 1 ms (32 LFINTOSC Cycles)

bit 0 **MCLRE:** Master Clear ($\overline{\text{MCLR}}$) Enable bit

If LVP = 1:
RE3 pin function is $\overline{\text{MCLR}}$
If LVP = 0:
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$
0 = $\overline{\text{MCLR}}$ pin function is a port defined function

REGISTER 9-22: PIE8: PERIPHERAL INTERRUPT ENABLE REGISTER 8

R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0	U-0
TMR5GIE	TMR5IE	—	—	—	—	—	—
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 **TMR5GIE:** TMR5 Gate Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **TMR5IE:** TMR5 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5-0 **Unimplemented:** Read as ‘0’

REGISTER 9-23: PIE9: PERIPHERAL INTERRUPT ENABLE REGISTER 9

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CLC3IE	CWG3IE	CCP3IE	TMR6IE
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 7-4 **Unimplemented:** Read as ‘0’

bit 3 **CLC3IE:** CLC3 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **CWG3IE:** CWG3 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **CCP3IE:** CCP3 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **TMR6IE:** TMR6 Interrupt Enable bit
1 = Enabled
0 = Disabled

REGISTER 13-2: NVMCON2: NONVOLATILE MEMORY CONTROL 2 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMCON2<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
x = Bit is unknown	'0' = Bit is cleared	'1' = Bit is set
-n = Value at POR		

bit 7-0 **NVMCON2<7:0>:**
Refer to **Section 13.1.4 "NVM Unlock Sequence"**.

Note 1: This register always reads zeros, regardless of data written.

Register 13-3: NVMADRL: Data EEPROM Memory Address Low

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
ADR<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
x = Bit is unknown	'0' = Bit is cleared	'1' = Bit is set
-n = Value at POR		

bit 7-0 **ADR<7:0>:** EEPROM Read Address bits

REGISTER 13-4: NVMADRH: DATA EEPROM MEMORY ADDRESS HIGH(1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	ADR<9:8>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
x = Bit is unknown	'0' = Bit is cleared	'1' = Bit is set
-n = Value at POR		

bit 7-2 **Unimplemented:** Read as '0'
bit 1-0 **ADR<9:8>:** EEPROM Read Address bits

Note 1: The NVMADRH register is not implemented on PIC18(L)F45/55K42.

REGISTER 14-18: SCANTRIG: SCAN TRIGGER SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	TSEL<3:0>			
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **TSEL<3:0>**: Scanner Data Trigger Input Selection bits

1111 = Reserved
- •
- •
- •

1010 = Reserved
1001 = SMT1_output
1000 = TMR6_postscaled
0111 = TMR5_output
0110 = TMR4_postscaled
0101 = TMR3_output
0100 = TMR2_postscaled
0011 = TMR1_output
0010 = TMR0_output
0001 = CLKREF_output
0000 = LFINTOSC

20.8 Register Definitions: Timer0 Control

REGISTER 20-1: T0CON0: TIMER0 CONTROL REGISTER 0

R/W-0/0	U-0	R-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	OUT	MD16	OUTPS<3:0>			
bit 7							bit 0

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

bit 7 **EN:** TMR0 Enable bit
1 = The module is enabled and operating
0 = The module is disabled and in the lowest power mode

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

bit 5 **OUT:** TMR0 Output bit (read-only)
TMR0 output bit

bit 4 **MD16:** TMR0 Operating as 16-Bit Timer Select bit
1 = TMR0 is a 16-bit timer
0 = TMR0 is an 8-bit timer

bit 3-0 **OUTPS<3:0>:** TMR0 Output Postscaler (Divider) Select bits
1111 = 1:16 Postscaler
1110 = 1:15 Postscaler
1101 = 1:14 Postscaler
1100 = 1:13 Postscaler
1011 = 1:12 Postscaler
1010 = 1:11 Postscaler
1001 = 1:10 Postscaler
1000 = 1:9 Postscaler
0111 = 1:8 Postscaler
0110 = 1:7 Postscaler
0101 = 1:6 Postscaler
0100 = 1:5 Postscaler
0011 = 1:4 Postscaler
0010 = 1:3 Postscaler
0001 = 1:2 Postscaler
0000 = 1:1 Postscaler

22.6 Timer2 Operation During Sleep

When PSYNC = 1, Timer2 cannot be operated while the processor is in Sleep mode. The contents of the T2TMR 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.

REGISTER 23-5: CCPRxH: CCPx REGISTER HIGH BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
RH<7:0>							
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 7-0

MODE = Capture Mode:
RH<7:0>: MSB of captured TMR1 value
MODE = Compare Mode:
RH<7:0>: MSB compared to TMR1 value
MODE = PWM Mode && FMT = 0:
RH<7:2>: Not used
RH<1:0>: CCPW<9:8> – Pulse-Width MS 2 bits
MODE = PWM Mode && FMT = 1:
RH<7:0>: CCPW<9:2> – Pulse-Width MS 8 bits

TABLE 23-4: SUMMARY OF REGISTERS ASSOCIATED WITH CCPx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPxCON	EN	—	OUT	FMT	MODE<3:0>				350
CCPxCAP	—	—	—	—	—	—	CTS<1:0>		352
CCPRxL	CCPRx<7:0>								352
CCPRxH	CCPRx<15:8>								353
CCPTMRS0	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		351

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the CCP module.

REGISTER 26-7: CWGxAS1: CWG AUTO-SHUTDOWN CONTROL REGISTER 1

U-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
—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E
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 **AS6E:** CWG Auto-shutdown Source 6 Enable bit

1 = Auto-shutdown for Source 6 is enabled

CWG Module	CWG1	CWG2	CWG3
Auto-shutdown Source 6	CLC2 OUT	CLC3 OUT	CLC4 OUT

0 = Auto-shutdown for Source 6 is disabled

bit 5 **AS5E:** CWG Auto-shutdown Source 5 (CMP2 OUT) Enable bit

1 = Auto-shutdown for CMP2 OUT is enabled

0 = Auto-shutdown for CMP2 OUT is disabled

bit 4 **AS4E:** CWG Auto-shutdown Source 4 (CMP1 OUT) Enable bit

1 = Auto-shutdown for CMP1 OUT is enabled

0 = Auto-shutdown for CMP1 OUT is disabled

bit 3 **AS3E:** CWG Auto-shutdown Source 3 (TMR6_Postscaled) Enable bit

1 = Auto-shutdown for TMR6_Postscaled is enabled

0 = Auto-shutdown for TMR6_Postscaled is disabled

bit 2 **AS2E:** CWG Auto-shutdown Source 2 (TMR4_Postscaled) Enable bit

1 = Auto-shutdown for TMR4_Postscaled is enabled

0 = Auto-shutdown for TMR4_Postscaled is disabled

bit 1 **AS1E:** CWG Auto-shutdown Source 1 (TMR2_Postscaled) Enable bit

1 = Auto-shutdown for TMR2_Postscaled is enabled

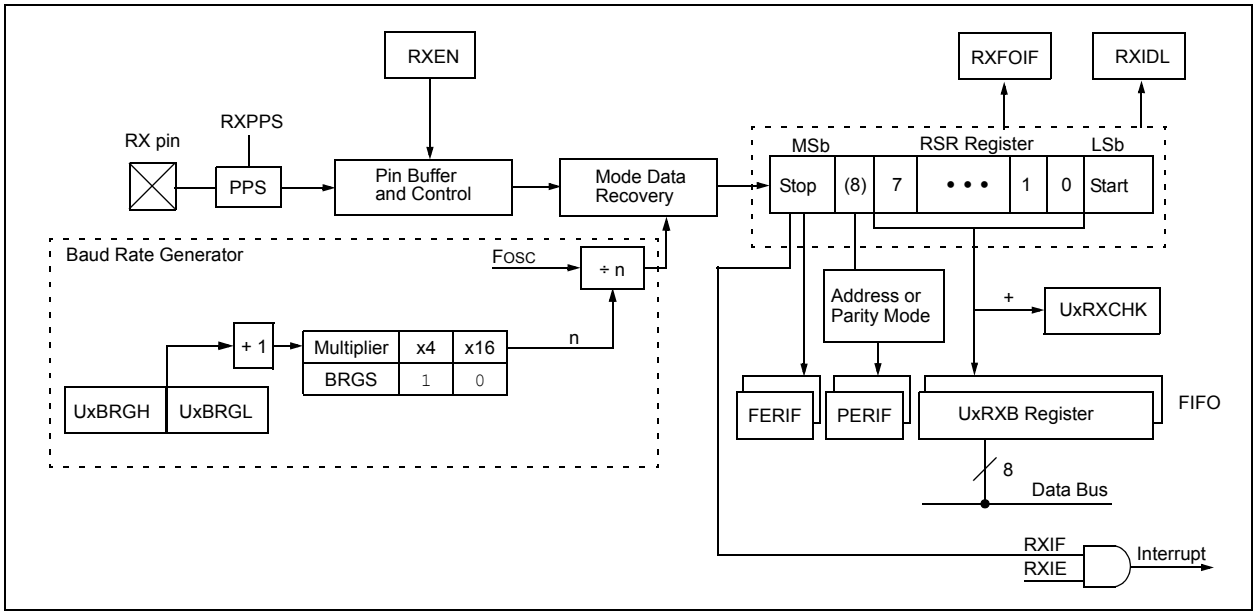
0 = Auto-shutdown for TMR2_Postscaled is disabled

bit 0 **AS0E:** CWG Auto-shutdown Source 0 (Pin selected by CWGxPPS) Enable bit

1 = Auto-shutdown for CWGxPPS Pin is enabled

0 = Auto-shutdown for CWGxPPS Pin is disabled

FIGURE 31-2: UART RECEIVE BLOCK DIAGRAM



The operation of the UART module is controlled through nineteen registers:

- Three control registers (UxCON0-UxCON2)
- Error enable and status (UxERRIE, UxERRIR, UxUIR)
- UART buffer status and control (UxFIFO)
- Three 9-bit protocol parameters (UxP1-UxP3)
- 16-bit baud rate generator (UxBRGH:L)
- Transmit buffer write (UxTXB)
- Receive buffer read (UxRXB)
- Receive checksum (UxRXCHK)
- Transmit checksum (UxTXCHK)

These registers are detailed in **Section 31.21 "Register Definitions: UART Control"**.

31.1 UART I/O Pin Configuration

The RX input pin is selected with the UxRPPS register. The TX output pin is selected with each pin's RxyPPS register. When the TRIS control for the pin corresponding to the TX output is cleared, then the UART will maintain control and the logic level on the TX pin. Changing the TXPOL bit in UxCON2 will immediately change the TX pin logic level regardless of the value of EN or TXEN.

31.2 UART Asynchronous Modes

The UART has five asynchronous modes:

- 7-bit
- 8-bit
- 8-bit with even parity in the 9th bit
- 8-bit with odd parity in the 9th bit
- 8-bit with address indicator in the 9th bit

The UART transmits and receives data using the standard Non-Return-to-Zero (NRZ) format. NRZ is implemented with two levels: a VOH mark state, which represents a '1' data bit, and a VOL space state, which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by seven or eight data bits, one optional parity or address bit, and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits with no parity. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See **Section 31.17 "UART Baud Rate Generator (BRG)"** for more information.

In all the asynchronous modes, the UART transmits and receives the LSb first. The UART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is supported by the hardware by even and odd parity modes.

31.2.1 UART ASYNCHRONOUS TRANSMITTER

The UART transmitter block diagram is shown in Figure 31-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the UxTXB register.

31.17.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit in the UxERRIR register will be set if the baud rate counter overflows before the fifth falling edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the UxBRGH:UxBRGL register pair. After the ABDOVF bit has been set, the state machine continues to search until the fifth falling edge is detected on the RX pin. Upon detecting the fifth falling RX edge, the hardware will set the ABDIF interrupt flag and clear the ABDEN bit in the UxCON0 register. The UxBRGH and UxBRGL register values retain their previous value. The ABDIF flag in the UxUIR register and ABDOVF flag in the UxERRIR register can be cleared by software directly. To generate an interrupt on an auto-baud overflow condition, all the following bits must be set:

- ABDOVE bit in the UxERRIE register
- UxEIE bit in the PIEx register
- PIE and GIE bits in the INTCON register

To terminate the auto-baud process before the ABDIF flag is set, clear the ABDEN bit, then clear the ABDOVF bit in the UxERRIR register.

31.17.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the UART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX line.

The Auto-Wake-up feature is enabled by setting both the WUE bit in the UxCON1 register and the UxIE bit in the PIEx register. Once set, the normal receive sequence on RX is disabled, and the UART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a transition out of the Idle state on the RX line. (This coincides with the start of a Break or a wake-up signal character for the LIN protocol.)

The UART module generates a WUIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 31-13), and asynchronously, if the device is in Sleep mode (Figure 31-14). The interrupt condition is cleared by clearing the WUIF bit in the UxUIR register. To generate an interrupt on a wake-up event, all the following bits must be set:

- UxIE bit in the PIEx register
- PIE and GIE bits in the INTCON register

The WUE bit is automatically cleared by the transition to the Idle state on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the UART module is in Idle mode, waiting to receive the next character.

31.17.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled, the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits of the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character of the transmission must be all zeros. This must be eleven or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL modes). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the UART.

WUE Bit

To ensure that no actual data is lost, check the RXIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

REGISTER 32-9: SPIxCON2: SPI CONFIGURATION REGISTER 2

R-0/0	R-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
BUSY	SSFLT	—	—	—	SSET	TXR(1)	RXR(1)
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

bit 7 **BUSY**: SPI Module Busy Status bit

1 = Data exchange is busy

0 = Data exchange is not taking place

bit 6 **SSFLT**: SS(in) Fault Status bit

If SSET = 0

1 = SS(in) ended the transaction unexpectedly, and the data byte being received was lost

0 = SS(in) ended normally

If SSET = 1

This bit is unchanged.

bit 5-3 **Unimplemented**: Read as '0'

bit 2 **SSET**: Slave Select Enable bit

Master mode:

1 = SS(out) is driven to the active state continuously

0 = SS(out) is driven to the active state while the transmit counter is not zero

Slave mode:

1 = SS(in) is ignored and data is clocked on all SCK(in) (as though SS = TRUE at all times)

0 = SS(in) enables/disables data input and tri-states SDO if the TRIS bit associated with the SDO pin is set (see Table 32-2 for details)

bit 1 **TXR**: Transmit Data-Required Control bit(1)

1 = TxFIFO data is required for a transfer

0 = TxFIFO data is not required for a transfer

bit 0 **RXR**: Receive FIFO Space-Required Control bit(1)

1 = Data transfers are suspended if the RxFIFO is full

0 = Received data is not stored in the FIFO

Note 1: See Table 32-1 as well as **Section 32.5 "Master mode"** and **Section 32.6 "Slave Mode"** for more details pertaining to TXR and RXR function.

2: This register should not be written to while a transfer is in progress (BUSY bit of SPIxCON2 is set).

REGISTER 33-5: I2CxBTO: I²C BUS TIMEOUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	BTO<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 Resets
'1' = Bit is set	'0' = Bit is cleared	HS = Hardware set HC = Hardware clear

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

bit 2-0 **BTO<2:0>:** I²C Bus Timeout Selection bits

BTO<2:0>	**I²Cx Bus Timeout Selection**
111	CLC4OUT
110	CLC3OUT
101	CLC2OUT
100	CLC1OUT
011	TMR6 post scaled output
010	TMR4 post scaled output
001	TMR2 post scaled output
000	Reserved

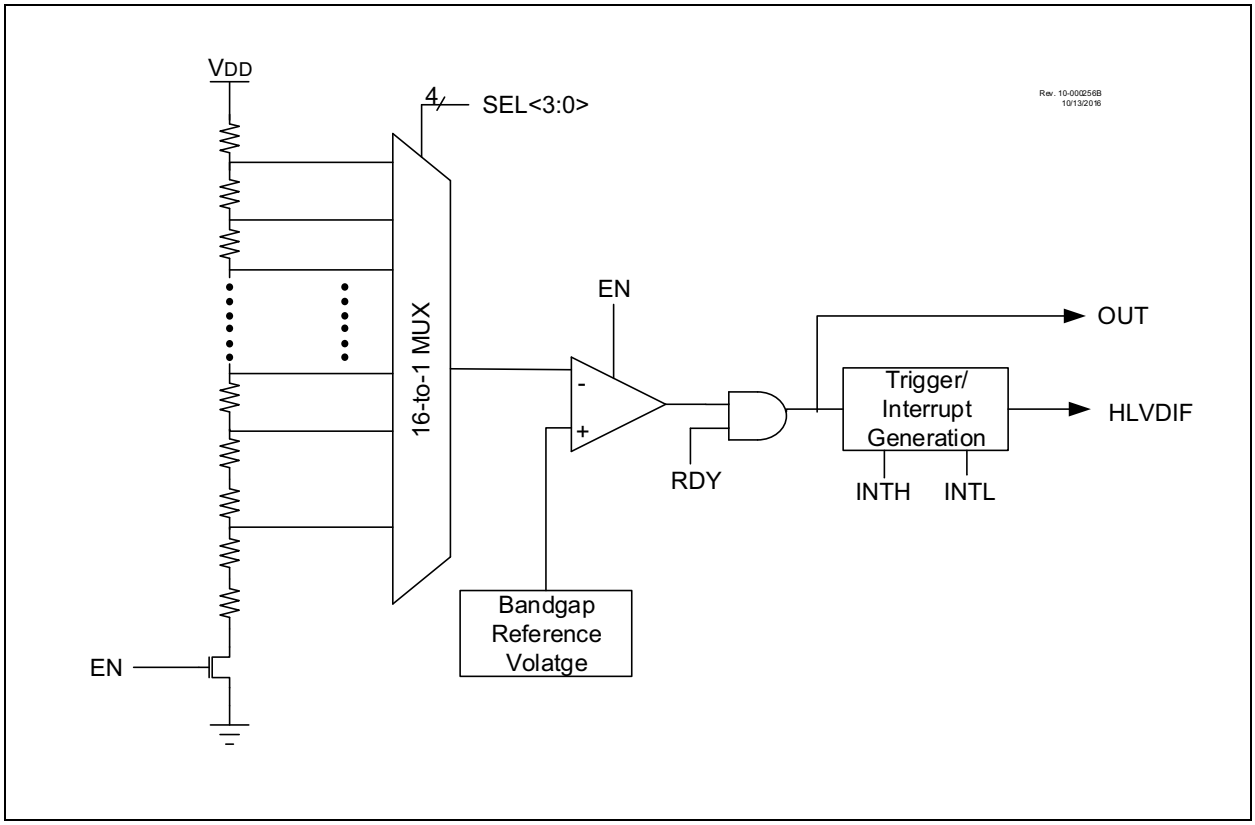
39.1 Operation

When the HLVD module is enabled, a comparator uses an internally generated voltage reference as the set point. The set point is compared with the trip point, where each node in the resistor divider represents a trip point voltage. The “trip point” voltage is the voltage level at which the device detects a high or low-voltage event, depending on the configuration of the module.

When the supply voltage is equal to the trip point, the voltage tapped off of the resistor array is equal to the internal reference voltage generated by the voltage reference module. The comparator then generates an interrupt signal by setting the HLVDIF bit.

The trip point voltage is software programmable to any of SEL<3:0> bits (HLVDCON1<3:0>).

FIGURE 39-1: HLVD MODULE BLOCK DIAGRAM



41.1.1 STANDARD INSTRUCTION SET

ADDFSR **Add Literal to FSR**

Syntax: ADDFSR f, k

Operands: $0 \le k \le 63$
$f \in [0, 1, 2]$

Operation: FSR(f) + k → FSR(f)

Status Affected: None

Encoding:

1110	1000	ffkk	kkkk

Description: The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR
Decode	Read literal 'k'	Process Data	Write to FSR

Example: `ADDFSR 2, 23h`

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 0422h

ADDLW **ADD literal to W**

Syntax: ADDLW k

Operands: $0 \le k \le 255$

Operation: (W) + k → W

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

Encoding:

0000	1111	kkkk	kkkk

Description: The contents of W are added to the 8-bit literal 'k' and the result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: `ADDLW 15h`

Before Instruction
W = 10h
After Instruction
W = 25h

ADDWF **ADD W to f**

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

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (W) + (f) → dest

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

Encoding:

0010	01da	ffff	ffff

Description: Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
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 $f \le 95$ (5Fh). See **Section 41.2.3 "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: `ADDWF REG, 0, 0`

Before Instruction
W = 17h
REG = 0C2h
After Instruction
W = 0D9h
REG = 0C2h

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

BZ **Branch if Zero**

Syntax: BZ n

Operands: -128 ≤ n ≤ 127

Operation: if ZERO bit is '1'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0000	nnnn	nnnn

Description: If the ZERO bit is '1', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BZ Jump`

Before Instruction
PC = address `(HERE)`
After Instruction
If ZERO = 1;
PC = address `(Jump)`
If ZERO = 0;
PC = address `(HERE + 2)`

CALL **Subroutine Call**

Syntax: CALL k {,s}

Operands: 0 ≤ k ≤ 1048575
s ∈ [0,1]

Operation: (PC) + 4 → TOS,
k → PC<20:1>,
if s = 1
(W) → WS,
(Status) → STATUSS,
(BSR) → BSRS

Status Affected: None

Encoding:

1st word (k<7:0>)	1110	110s	$k_7$kkk	kkkk$_0$
2nd word(k<19:8>)	1111	k_{19}kkk	kkkk	kkkk$_8$

Description: Subroutine call of entire 2-Mbyte memory range. First, return address (PC + 4) is pushed onto the return stack. If 's' = 1, the W, Status and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into PC<20:1>. `CALL` is a 2-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	PUSH PC to stack	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: `HERE CALL THERE, 1`

Before Instruction
PC = address `(HERE)`
After Instruction
PC = address `(THERE)`
TOS = address `(HERE + 4)`
WS = W
BSRS = BSR
STATUSS = Status

SUBWF — Subtract W from f

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

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f) – (W) → dest

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

Encoding:

0101	11da	ffff	ffff

Description: Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
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 $f \le 95$ (5Fh). See **Section 41.2.3 "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 1:
```
SUBWF   REG, 1, 0
```
Before Instruction
REG = 3
W = 2
C = ?
After Instruction
REG = 1
W = 2
C = 1 ; result is positive
Z = 0
N = 0

Example 2:
```
SUBWF   REG, 0, 0
```
Before Instruction
REG = 2
W = 2
C = ?
After Instruction
REG = 2
W = 0
C = 1 ; result is zero
Z = 1
N = 0

Example 3:
```
SUBWF   REG, 1, 0
```
Before Instruction
REG = 1
W = 2
C = ?
After Instruction
REG = FFh ;(2's complement)
W = 2
C = 0 ; result is negative
Z = 0
N = 1

SUBWFB — Subtract W from f with Borrow

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

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f) – (W) – $(\overline{C})$ → dest

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

Encoding:

0101	10da	ffff	ffff

Description: Subtract W and the CARRY flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
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 $f \le 95$ (5Fh). See **Section 41.2.3 "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 1:
```
SUBWFB  REG, 1, 0
```
Before Instruction
REG = 19h (0001 1001)
W = 0Dh (0000 1101)
C = 1
After Instruction
REG = 0Ch (0000 1100)
W = 0Dh (0000 1101)
C = 1
Z = 0
N = 0 ; result is positive

Example 2:
```
SUBWFB  REG, 0, 0
```
Before Instruction
REG = 1Bh (0001 1011)
W = 1Ah (0001 1010)
C = 0
After Instruction
REG = 1Bh (0001 1011)
W = 00h
C = 1
Z = 1 ; result is zero
N = 0

Example 3:
```
SUBWFB  REG, 1, 0
```
Before Instruction
REG = 03h (0000 0011)
W = 0Eh (0000 1110)
C = 1
After Instruction
REG = F5h (1111 0101)
; [2's comp]
W = 0Eh (0000 1110)
C = 0
Z = 0
N = 1 ; result is negative

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
388Dh	FSR2L_SHAD	FSR2L								125
388Ch	FSR1H_SHAD	—	—	FSR1H						125
388Bh	FSR1L_SHAD	FSR1L								125
388Ah	FSR0H_SHAD	—	—	FSR0H						125
3889h	FSR0L_SHAD	FSR0L								125
3888h	PCLATU_SHAD	—	—	—	PCU					125
3887h	PCLATH_SHAD	PCH								125
3886h	BSR_SHAD	—	—	BSR						125
3885h	WREG_SHAD	WREG								125
3884h	STATUS_SHAD	—	$\overline{TO}$	$\overline{PD}$	N	OV	Z	DC	C	125
3883h	SHADCON	—	—	—	—	—	—	—	SHADLO	168
3882h	BSR_CSHAD	—	—	BSR						57
3881h	WREG_CSHAD	WREG								57
3880h	STATUS_C-SHAD	—	$\overline{TO}$	$\overline{PD}$	N	OV	Z	DC	C	57
387Fh - 3800h	—	Unimplementeds								

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.

2: Unimplemented in PIC18(L)F26/27K42.

3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.

4: Unimplemented in PIC18(L)F45/55K42.

TABLE 44-23: SPI MODE REQUIREMENTS (MASTER MODE)

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	TSCK	SCK Cycle Time (2x Prescaled)	61	—	—	ns	Transmit only mode
			—	16(1)	—	MHz	
			95	—	—	ns	Full duplex mode
			—	10(1)	—	MHz	
SP70*	TSSL2SCH, TSSL2SCL	SDO to SCK↓ or SCK↑ input	TSCK	—	—	ns	FST = 0
			0	—	—	ns	FST = 1
SP71*	TSCH	SCK output high time	0.5 TSCK - 12	—	0.5 TSCK + 12	ns	
SP72*	TSCL	SCK output low time	0.5 TSCK - 12	—	0.5 TSCK + 12	ns	
SP73*	TDIV2SCH, TDIV2SCL	Setup time of SDI data input to SCK edge	85	—	—	ns	
SP74*	TSCH2DIL, TSCL2DIL	Hold time of SDI data input to SCK edge	0	—	—	ns	
		Hold time of SDI data input to final SCK	0.5 TSCK	—	—	ns	CKE = 0, SMP = 1
SP75*	TDOR	SDO data output rise time	—	10	25	ns	CL = 50 pF
SP76*	TDOF	SDO data output fall time	—	10	25	ns	CL = 50 pF
SP78*	TSCR	SCK output rise time	—	10	25	ns	CL = 50 pF
SP79*	TSCF	SCK output fall time	—	10	25	ns	CL = 50 pF
SP80*	TSCH2DOV, TSCL2DOV	SDO data output valid after SCK edge	- 15	—	15	ns	CL = 20 pF
SP81*	TDOV2SCH, TDOV2SCL	SDO data output valid to first SCK edge	TSCK - 10	—	—	ns	CL = 20 pF CKE = 1
SP82*	TSSL2DOV	SDO data output valid after $\overline{SS}$↓ edge	—	—	50	ns	CL = 20 pF
SP83*	TSCH2SSH, TSCL2SSH	$\overline{SS}$ ↑ after last SCK edge	0.5 TSCK - 10	—	—	ns	
SP84*	TSSH2SSL	SS↑ to SS↓ edge	0.5 TSCK - 10	—	—	ns	

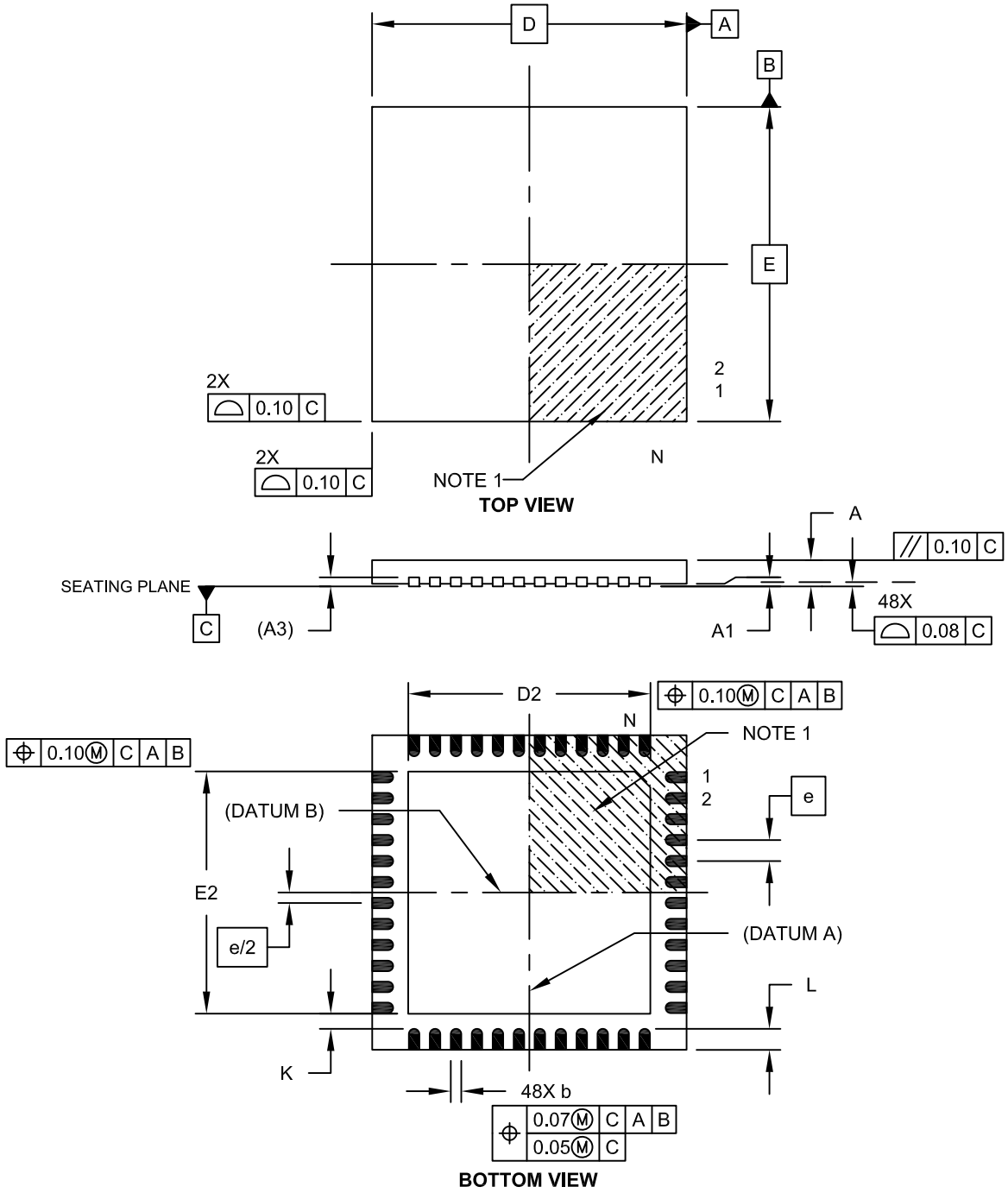
* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: SPIxCON1.SMP bit must be set and the slew rate control must be disabled on the clock and data pins (clear the corresponding bits in SLRCONx register) for SPI to operate over 4 MHz.

48-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 6x6x0.5 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 C04-153A Sheet 1 of 2