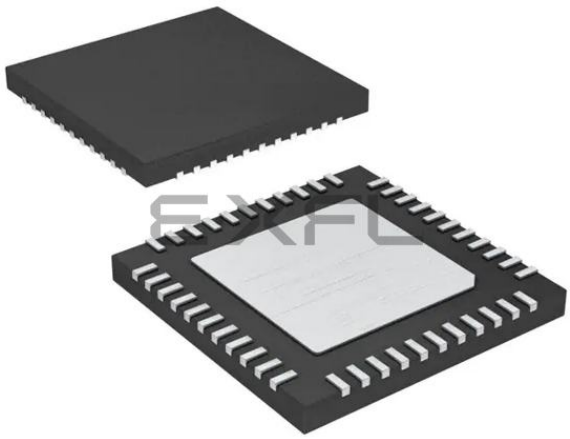


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

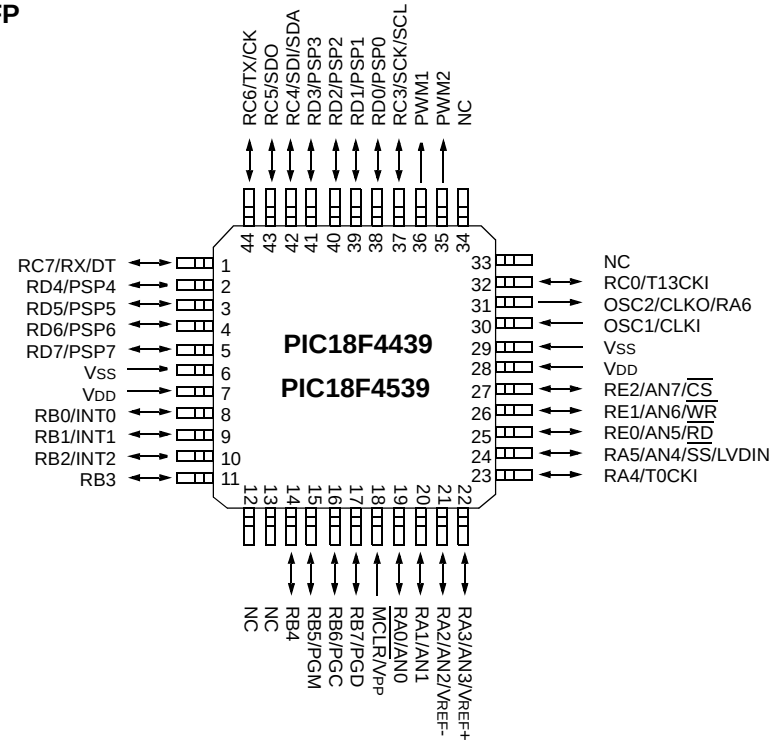
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	32
Program Memory Size	24KB (12K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1408 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf4539t-i-ml

Pin Diagrams

44-Pin TQFP



44-Pin QFN

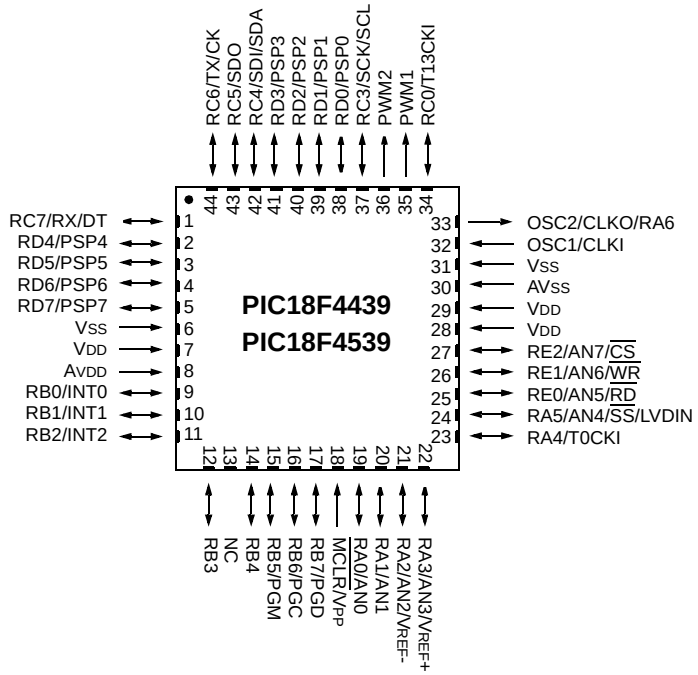


TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
ADRESH	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	2439	4439	2539	4539	0000 00-0	0000 00-0	uuuu uu-u
ADCON1	2439	4439	2539	4539	00-- 0000	00-- 0000	uu-- uuuu
CCPR1H	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L*	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON*	2439	4439	2539	4539	--00 0000	--00 0000	--uu uuuu
CCPR2H	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L*	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON*	2439	4439	2539	4539	--00 0000	--00 0000	--uu uuuu
TMR3H	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	2439	4439	2539	4539	0000 0000	uuuu uuuu	uuuu uuuu
SPBRG	2439	4439	2539	4539	0000 0000	0000 0000	uuuu uuuu
RCREG	2439	4439	2539	4539	0000 0000	0000 0000	uuuu uuuu
TXREG	2439	4439	2539	4539	0000 0000	0000 0000	uuuu uuuu
TXSTA	2439	4439	2539	4539	0000 -010	0000 -010	uuuu -uuu
RCSTA	2439	4439	2539	4539	0000 000x	0000 000x	uuuu uuuu
EEADR	2439	4439	2539	4539	0000 0000	0000 0000	uuuu uuuu
EEDATA	2439	4439	2539	4539	0000 0000	0000 0000	uuuu uuuu
EECON1	2439	4439	2539	4539	xx-0 x000	uu-0 u000	uu-0 u000
EECON2	2439	4439	2539	4539	---- ----	---- ----	---- ----

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

* These registers are retained to maintain compatibility with PIC18FXX2 devices; however, one or more bits are reserved. Users should not modify the value of these bits. See Section 4.9.2 for details.

Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 3-2 for RESET value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other Oscillator modes, they are disabled and read ‘0’.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they are read ‘0’.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
IPR2	2439	4439	2539	4539	---1 1111	---1 1111	---u uuuu
PIR2	2439	4439	2539	4539	---0 0000	---0 0000	---u uuuu(3)
PIE2	2439	4439	2539	4539	---0 0000	---0 0000	---u uuuu
IPR1	2439	4439	2539	4539	1111 1111	1111 1111	uuuu uuuu
	2439	4439	2539	4539	-111 1111	-111 1111	-uuu uuuu
PIR1	2439	4439	2539	4539	0000 0000	0000 0000	uuuu uuuu(3)
	2439	4439	2539	4539	-000 0000	-000 0000	-uuu uuuu(3)
PIE1	2439	4439	2539	4539	0000 0000	0000 0000	uuuu uuuu
	2439	4439	2539	4539	-000 0000	-000 0000	-uuu uuuu
TRISE	2439	4439	2539	4539	0000 -111	0000 -111	uuuu -uuu
TRISD	2439	4439	2539	4539	1111 1111	1111 1111	uuuu uuuu
TRISC*	2439	4439	2539	4539	1111 1111	1111 1111	uuuu uuuu
TRISB	2439	4439	2539	4539	1111 1111	1111 1111	uuuu uuuu
TRISA(5,6)	2439	4439	2539	4539	-111 1111(5)	-111 1111(5)	-uuu uuuu(5)
LATE	2439	4439	2539	4539	---- -xxx	---- -uuu	---- -uuu
LATD	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC*	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA(5,6)	2439	4439	2539	4539	-xxx xxxx(5)	-uuu uuuu(5)	-uuu uuuu(5)
PORTE	2439	4439	2539	4539	---- -000	---- -000	---- -uuu
PORTD	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC*	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	2439	4439	2539	4539	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA(5,6)	2439	4439	2539	4539	-x0x 0000(5)	-u0u 0000(5)	-uuu uuuu(5)

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

* These registers are retained to maintain compatibility with PIC18FXX2 devices; however, one or more bits are reserved. Users should not modify the value of these bits. See Section 4.9.2 for details.

Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 3-2 for RESET value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other Oscillator modes, they are disabled and read ‘0’.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they are read ‘0’.

4.3 Fast Register Stack

For PIC18FXX39 devices, a “fast interrupt return” option is available for high priority interrupts. A single level Fast Register Stack is provided for the STATUS, WREG and BSR registers; it is not readable or writable. When the processor vectors for an interrupt, the stack is loaded with the current value of the corresponding register. If the `FAST RETURN` instruction is used to return from the interrupt, the values in the registers are then loaded back into the working registers.

Note:	The fast interrupt return for PIC18FXX39 devices is reserved for use by the ProMPT kernel and the Timer2 match interrupt. It is not available to the user for any other interrupts or returns from subroutines.

4.4 PCL, PCLATH and PCLATU

The program counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21-bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<15:8> bits and is not directly readable or writable. Updates to the PCH register may be performed through the PCLATH register. The upper byte is called PCU. This register contains the PC<20:16> bits and is not directly readable or writable. Updates to the PCU register may be performed through the PCLATU register.

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the LSB of PCL is fixed to a value of ‘0’. The PC increments by 2 to address sequential instructions in the program memory.

The `CALL`, `RCALL`, `GOTO` and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter.

The contents of PCLATH and PCLATU will be transferred to the program counter by an operation that writes PCL. Similarly, the upper two bytes of the program counter will be transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see Section 4.8.1).

4.5 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 4-3.

FIGURE 4-3: CLOCK/INSTRUCTION CYCLE

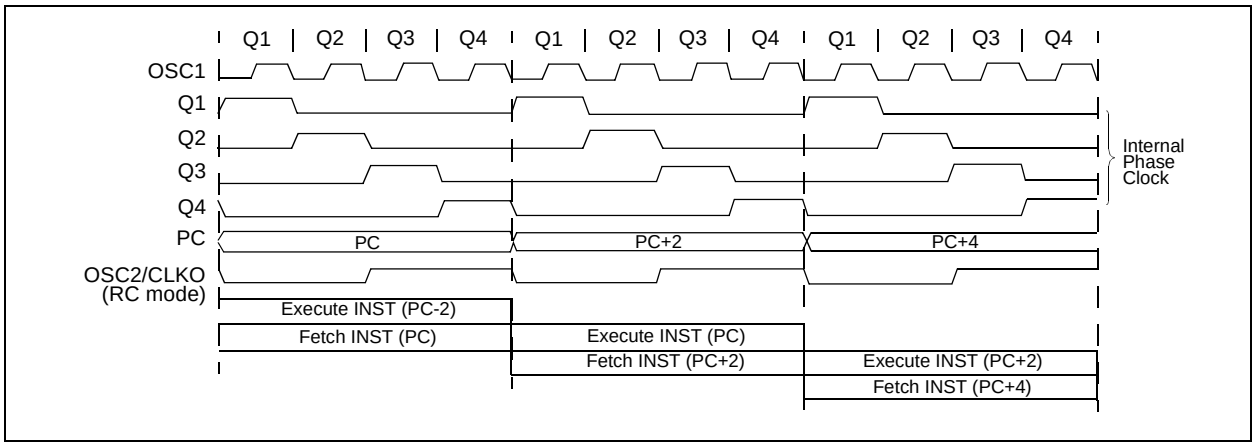


TABLE 4-1: SPECIAL FUNCTION REGISTER MAP

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2(3)	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2(3)	FBEh	CCPR1L*	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2(3)	FBDh	CCP1CON*	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2(3)	FBCh	CCPR2H	F9Ch	—
FFBh	PCLATU	FDBh	PLUSW2(3)	FBBh	CCPR2L*	F9Bh	—
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON*	F9Ah	—
FF9h	PCL	FD9h	FSR2L	FB9h	—	F99h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	—	F98h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	—	F97h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	—	F96h	TRISE(2)
FF5h	TABLAT	FD5h	T0CON	FB5h	—	F95h	TRISD(2)
FF4h	PRODH	FD4h	—	FB4h	—	F94h	TRISC(4)
FF3h	PRODL	FD3h	OSCCON*	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	—
FF0h	INTCON3	FD0h	RCON	FB0h	—	F90h	—
FEFh	INDF0(3)	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—
FEEh	POSTINC0(3)	FCEh	TMR1L	FAEh	RCREG	F8Eh	—
FEDh	POSTDEC0(3)	FCDh	T1CON	FADh	TXREG	F8Dh	LATE(2)
FECh	PREINC0(3)	FCCh	TMR2*	FACh	TXSTA	F8Ch	LATD(2)
FEBh	PLUSW0(3)	FCBh	PR2*	FABh	RCSTA	F8Bh	LATC(4)
FEAh	FSR0H	FCAh	T2CON*	FAAh	—	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	—
FE7h	INDF1(3)	FC7h	SSPSTAT	FA7h	EECON2	F87h	—
FE6h	POSTINC1(3)	FC6h	SSPCON1	FA6h	EECON1	F86h	—
FE5h	POSTDEC1(3)	FC5h	SSPCON2	FA5h	—	F85h	—
FE4h	PREINC1(3)	FC4h	ADRESH	FA4h	—	F84h	PORTE(2)
FE3h	PLUSW1(3)	FC3h	ADRESL	FA3h	—	F83h	PORTD(2)
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC(4)
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	—	FA0h	PIE2	F80h	PORTA

* These registers are retained to maintain compatibility with PIC18FXX2 devices; however, one or more bits are reserved in PIC18FXX39 devices. Users should not alter the values of these bits.

Note 1: Unimplemented registers are read as '0'.

2: This register is not available on PIC18F2X39 devices.

3: This is not a physical register.

4: Bits 1 and 2 are reserved; users should not alter their values.

5.0 FLASH PROGRAM MEMORY

The FLASH Program Memory is readable, writable, and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation may not be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

5.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

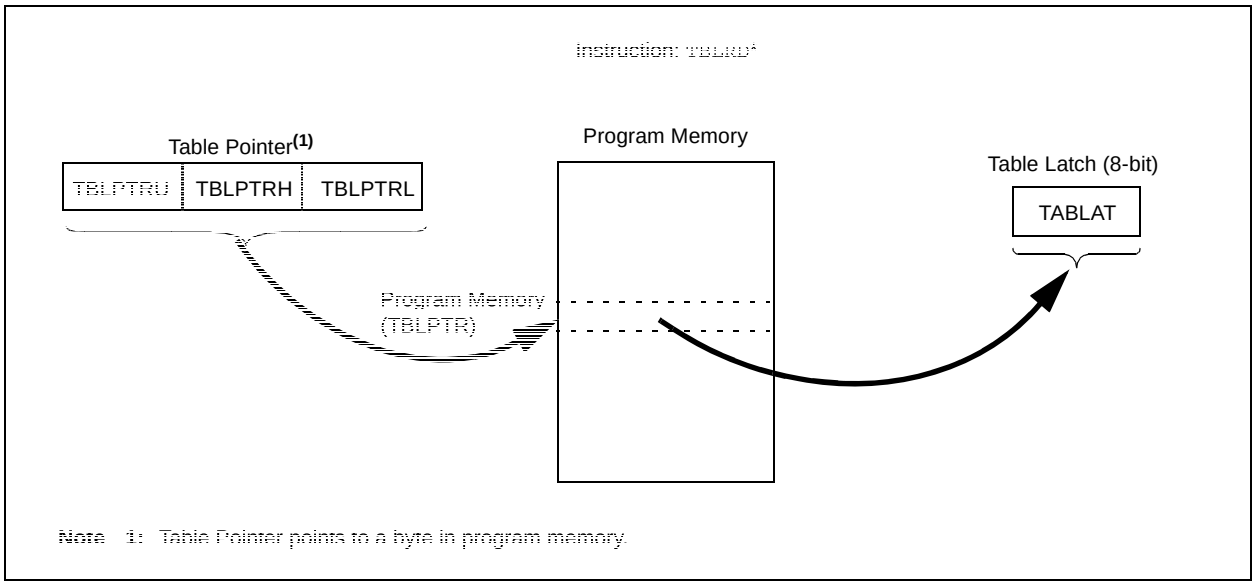
The program memory space is 16-bits wide, while the data RAM space is 8-bits wide. Table Reads and Table Writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table Read operations retrieve data from program memory and places it into the data RAM space. Figure 5-1 shows the operation of a Table Read with program memory and data RAM.

Table Write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in Section 5.5, “Writing to FLASH Program Memory”. Figure 5-2 shows the operation of a Table Write with program memory and data RAM.

Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word aligned. Therefore, a table block can start and end at any byte address. If a Table Write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 5-1: TABLE READ OPERATION



NOTES:

NOTES:

FIGURE 16-27: BUS COLLISION DURING START CONDITION (SCL = 0)

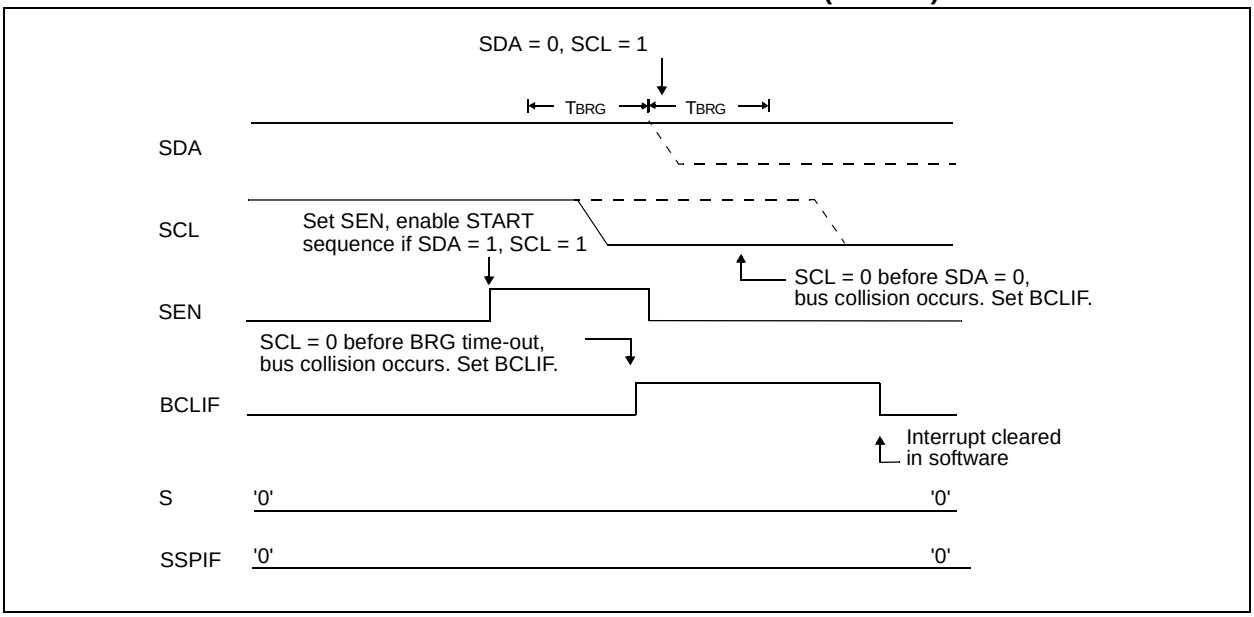
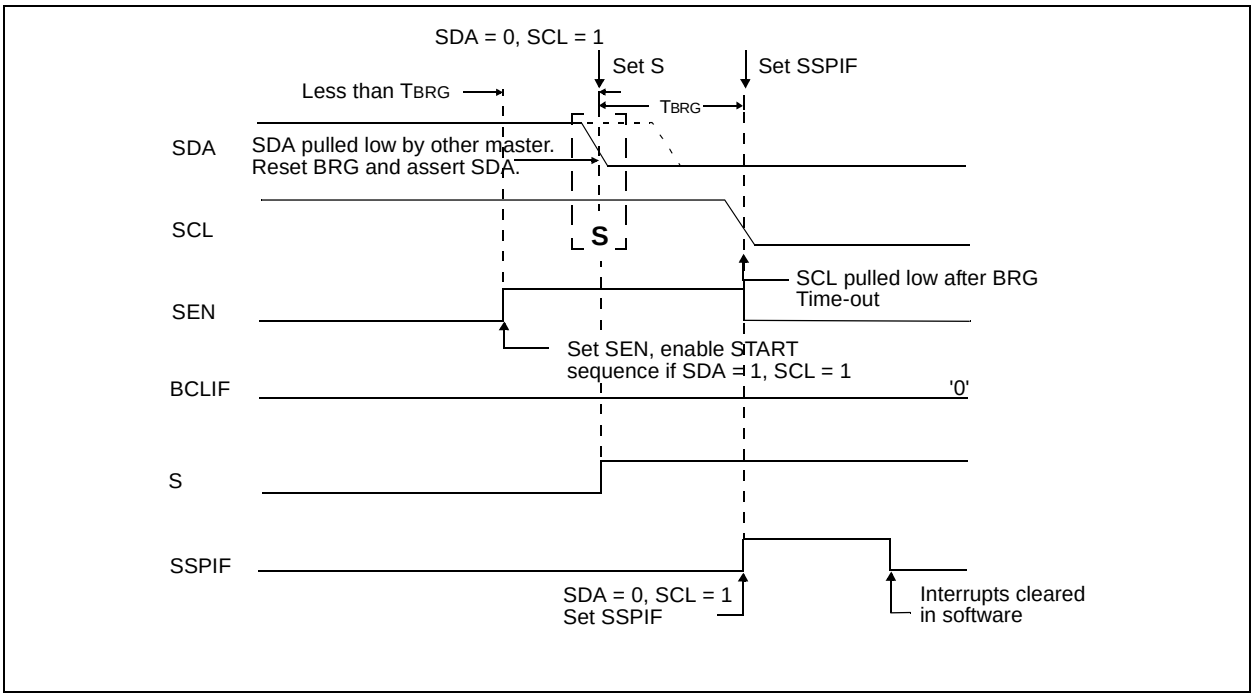


FIGURE 16-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION



REGISTER 17-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

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

bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data
Can be Address/Data bit or a parity bit

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

FIGURE 17-2: ASYNCHRONOUS TRANSMISSION

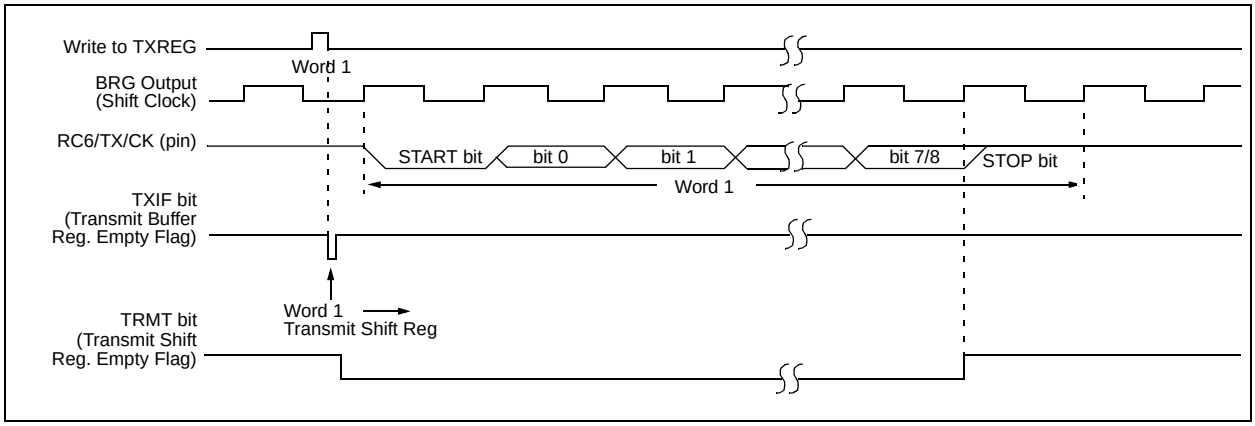
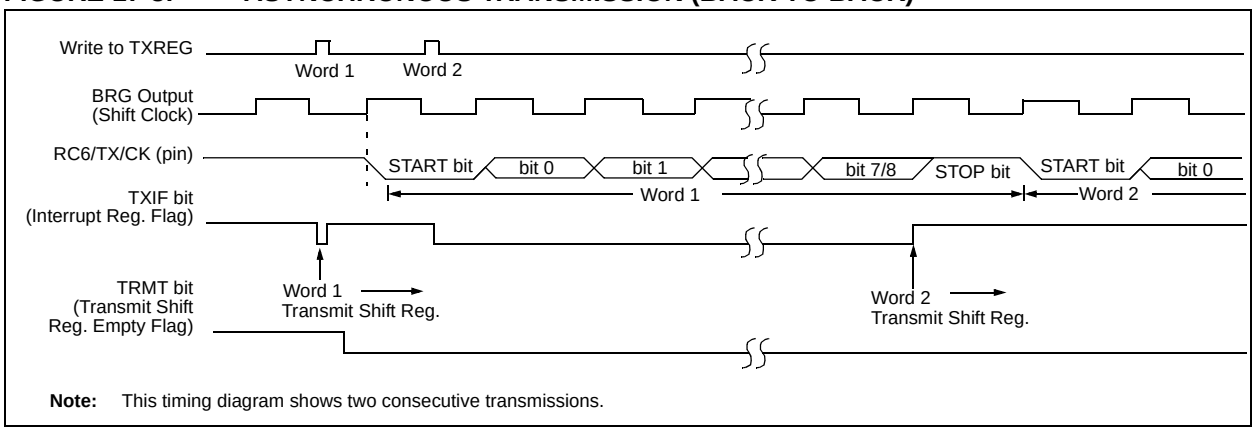


FIGURE 17-3: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 17-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on All Other RESETS
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	—	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	—	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	—	TMR2IP	TMR1IP	0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	USART Transmit Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'.
Shaded cells are not used for Asynchronous Transmission.

Note 1: The PSPIF, PSPIE and PSPIP bits are reserved on the PIC18F2X39 devices; always maintain these bits clear.

To calculate the minimum acquisition time, Equation 18-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

EQUATION 18-1: ACQUISITION TIME

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \end{aligned}$$

EQUATION 18-2: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \bullet (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$

or

$$T_C = -(120\ pF)(1\ k\Omega + R_{SS} + R_S) \ln(1/2048)$$

Example 18-1 shows the calculation of the minimum required acquisition time, TACQ. This calculation is based on the following application system assumptions:

- CHOLD = 120 pF
- Rs = 2.5 kΩ
- Conversion Error ≤ 1/2 LSb
- VDD = 5V → Rss = 7 kΩ
- Temperature = 50°C (system max.)
- VHOLD = 0V @ time = 0

EXAMPLE 18-1: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$

Temperature coefficient is only required for temperatures > 25°C.

$T_{ACQ} = 2\ \mu s + T_C + [(Temp - 25°C)(0.05\ \mu s/°C)]$

$$\begin{aligned} T_C &= -C_{HOLD}(R_{IC} + R_{SS} + R_S) \ln(1/2048) \\ &= -120\ pF\ (1\ k\Omega + 7\ k\Omega + 2.5\ k\Omega) \ln(0.0004883) \\ &= -120\ pF\ (10.5\ k\Omega) \ln(0.0004883) \\ &= -1.26\ \mu s\ (-7.6246) \\ &= 9.61\ \mu s \end{aligned}$$

$$\begin{aligned} T_{ACQ} &= 2\ \mu s + 9.61\ \mu s + [(50°C - 25°C)(0.05\ \mu s/°C)] \\ &= 11.61\ \mu s + 1.25\ \mu s \\ &= 12.86\ \mu s \end{aligned}$$

18.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 12 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable. The seven possible options for TAD are:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal A/D module RC oscillator (2-6 μs)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 μs.

Table 18-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

18.3 Configuring Analog Port Pins

The ADCON1, TRISA and TRISE registers control the operation of the A/D port pins. The port pins, that are desired as analog inputs, must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

Note 1: When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

2: Analog levels on any pin that is defined as a digital input (including the AN4:AN0 pins) may cause the input buffer to consume current that is out of the device's specification.

TABLE 18-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)		Maximum Device Frequency	
Operation	**ADCS2:ADCS0**	**PIC18FXX39**	**PIC18LFXX39**
2 TOSC	000	1.25 MHz	666 kHz
4 TOSC	100	2.50 MHz	1.33 MHz
8 TOSC	001	5.00 MHz	2.67 MHz
16 TOSC	101	10.00 MHz	5.33 MHz
32 TOSC	010	20.00 MHz	10.67 MHz
64 TOSC	110	40.00 MHz	21.33 MHz
RC	011	—	—

20.2 Watchdog Timer (WDT)

The Watchdog Timer is a free running on-chip RC oscillator, which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKI pin. That means that the WDT will run, even if the clock on the OSC1/CLKI and OSC2/CLKO/RA6 pins of the device has been stopped, for example, by execution of a SLEEP instruction.

During normal operation, a WDT time-out generates a device RESET (Watchdog Timer Reset). If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The $\overline{TO}$ bit in the RCON register will be cleared upon a WDT time-out.

The Watchdog Timer is enabled/disabled by a device configuration bit. If the WDT is enabled, software execution may not disable this function. When the WDTEN configuration bit is cleared, the SWDTEN bit enables/disables the operation of the WDT.

The WDT time-out period values may be found in the Electrical Specifications (Section 23.0) under parameter D031. Values for the WDT postscaler may be assigned using the configuration bits.

Note 1:	The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT and prevent it from timing out and generating a device RESET condition.
2:	When a CLRWDT instruction is executed and the postscaler is assigned to the WDT, the postscaler count will be cleared, but the postscaler assignment is not changed.

20.2.1 CONTROL REGISTER

Register 20-13 shows the WDTCON register. This is a readable and writable register, which contains a control bit that allows software to override the WDT enable configuration bit, only when the configuration bit has disabled the WDT.

REGISTER 20-13: WDTCON REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN
bit 7							bit 0

bit 7-1 **Unimplemented**: Read as ’0’

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit
1 = Watchdog Timer is on
0 = Watchdog Timer is turned off if the WDTEN configuration bit in the configuration register = 0

Legend:	
R = Readable bit	W = Writable bit
U = Unimplemented bit, read as ‘0’	- n = Value at POR

21.0 INSTRUCTION SET SUMMARY

The PIC18FXXX instruction set adds many enhancements to the previous PIC MCU instruction sets, while maintaining an easy migration from these PIC MCU instruction sets.

Most instructions are a single program memory word (16-bits), but there are three instructions that require two program memory locations.

Each single word instruction is a 16-bit word divided into an OPCODE, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18FXXX instruction set summary in Table 21-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 21-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The destination of the result (specified by ‘d’)
3. The accessed memory (specified by ‘a’)

The file register designator 'f' specifies which file register is to be used by the instruction.

The destination designator ‘d’ specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the WREG register. If 'd' is one, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The bit in the file register (specified by ‘b’)
3. The accessed memory (specified by ‘a’)

The bit field designator 'b' selects the number of the bit affected by the operation, while the file register designator 'f' represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by ‘k’)
- The desired FSR register to load the literal value into (specified by ‘f’)
- No operand required (specified by ‘—’)

The **control** instructions may use some of the following operands:

- A program memory address (specified by ‘n’)
- The mode of the Call or Return instructions (specified by ‘s’)
- The mode of the Table Read and Table Write instructions (specified by ‘m’)
- No operand required (specified by ‘—’)

All instructions are a single word, except for three double-word instructions. These three instructions were made double-word instructions, so that all the required information is available in these 32 bits. In the second word, the 4 MSbs are ‘1’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs. Two-word branch instructions (if true) would take 3 μs.

Figure 21-1 shows the general formats that the instructions can have.

All examples use the format ‘nnh’ to represent a hexadecimal number, where ‘h’ signifies a hexadecimal digit.

The Instruction Set Summary, shown in Table 21-2, lists the instructions recognized by the Microchip Assembler (MPASM™).

Section 21.1 provides a description of each instruction.

DECFSZ **Decrement f, skip if 0**

Syntax: [*label*] DECFSZ f [,d [,a]]

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

Operation: (f) – 1 → dest,
skip if result = 0

Status Affected: None

Encoding:

0010	11da	ffff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0, the result is placed in W. If 'd' is 1, the result is placed back in register 'f' (default). If the result is 0, the next instruction, which is already fetched, is discarded, and a NOP is executed instead, making it a two-cycle instruction. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

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

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    DECFSZ  CNT, 1, 1
        GOTO    LOOP
CONTINUE
```

Before Instruction
PC = Address (HERE)

After Instruction
CNT = CNT - 1
If CNT = 0;
PC = Address (CONTINUE)
If CNT ≠ 0;
PC = Address (HERE+2)

DCFSNZ **Decrement f, skip if not 0**

Syntax: [*label*] DCFSNZ f [,d [,a]

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

Operation: (f) – 1 → dest,
skip if result ≠ 0

Status Affected: None

Encoding:

0100	11da	ffff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0, the result is placed in W. If 'd' is 1, the result is placed back in register 'f' (default). If the result is not 0, the next instruction, which is already fetched, is discarded, and a NOP is executed instead, making it a two-cycle instruction. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

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

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    DCFSNZ  TEMP, 1, 0
ZERO    :
NZERO   :
```

Before Instruction
TEMP = ?

After Instruction
TEMP = TEMP - 1,
If TEMP = 0;
PC = Address (ZERO)
If TEMP ≠ 0;
PC = Address (NZERO)

TABLE 23-22: A/D CONVERSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
130	TAD	A/D clock period	PIC18**F**XXXX	1.6	20(4)	µs	TOSC based
			PIC18**LF**XXXX	2.0	6.0	µs	A/D RC mode
131	TCNV	Conversion time (not including acquisition time) **(Note 1)**		11	12	TAD	
132	TACQ	Acquisition time **(Note 2)**		5 10	— —	µs µs	VREF = VDD = 5.0V VREF = VDD = 2.5V
135	TSWC	Switching Time from convert → sample		—	**(Note 3)**		

Note 1: ADRES register may be read on the following TCY cycle.

2: The time for the holding capacitor to acquire the "New" input voltage, when the new input value has not changed by more than 1 LSB from the last sampled voltage. The source impedance (*RS*) on the input channels is 50Ω. See Section 18.0 for more information on acquisition time consideration.

3: On the next Q4 cycle of the device clock.

4: The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.

24.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

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

"Typical" represents the mean of the distribution at 25°C. "Maximum" or "minimum" represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over the whole temperature range.

FIGURE 24-1: TYPICAL IDD vs. FOSC OVER VDD (HS MODE)

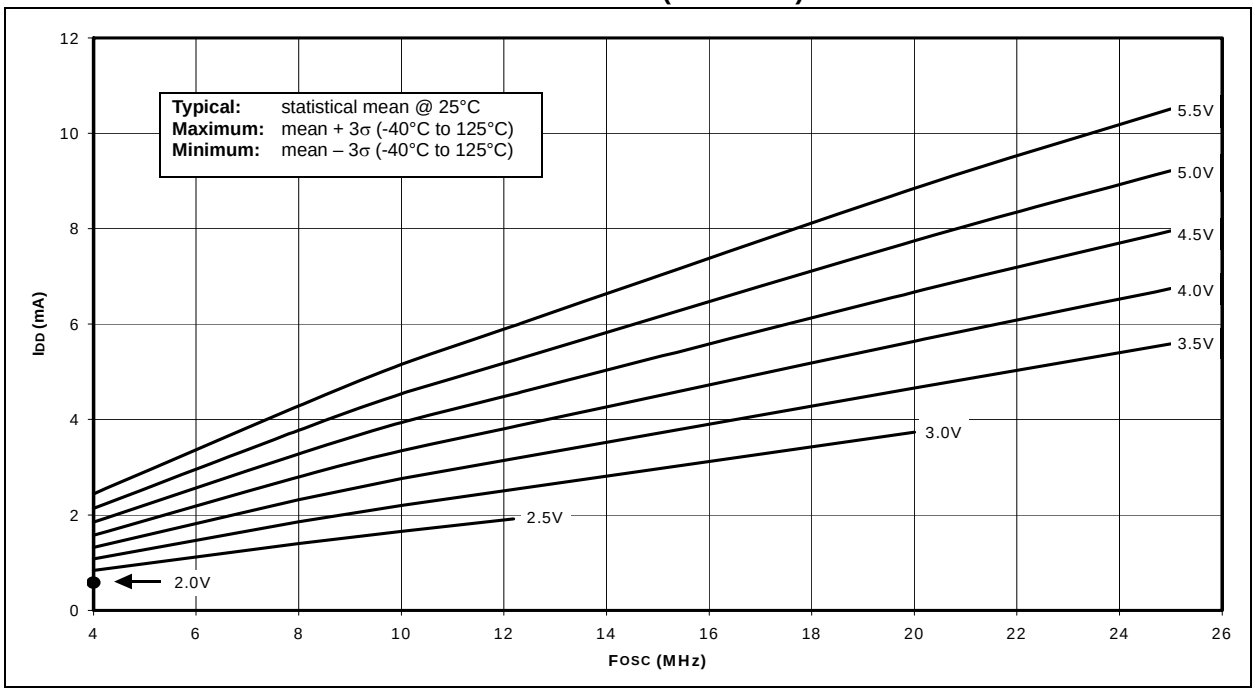


FIGURE 24-2: MAXIMUM IDD vs. FOSC OVER VDD (HS MODE)

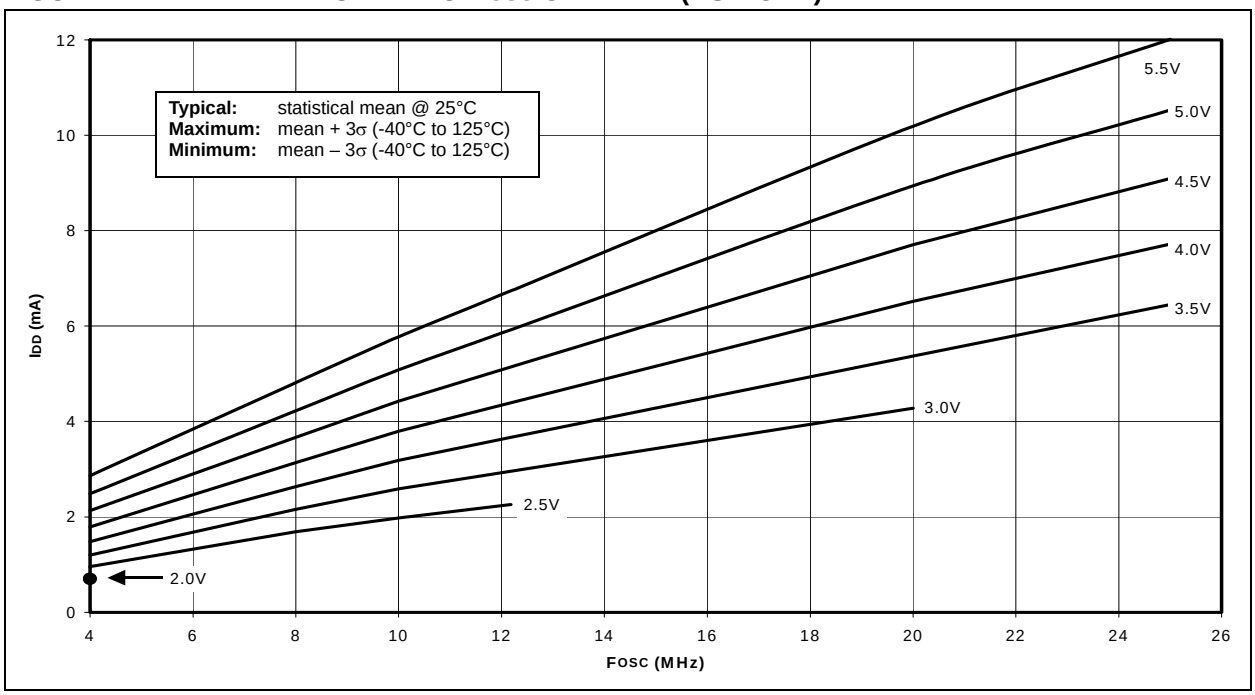


FIGURE 24-17: MINIMUM AND MAXIMUM VIN vs. VDD (TTL INPUT, -40°C TO +125°C)

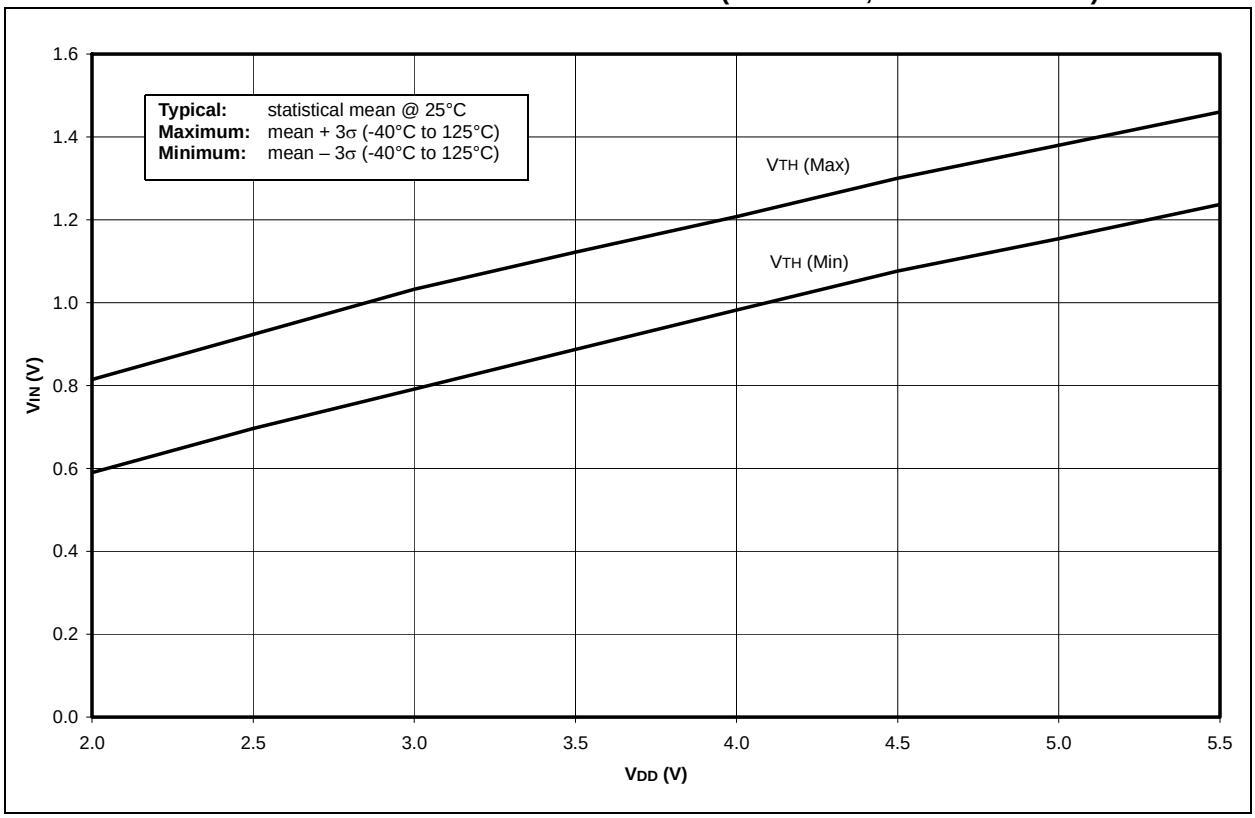
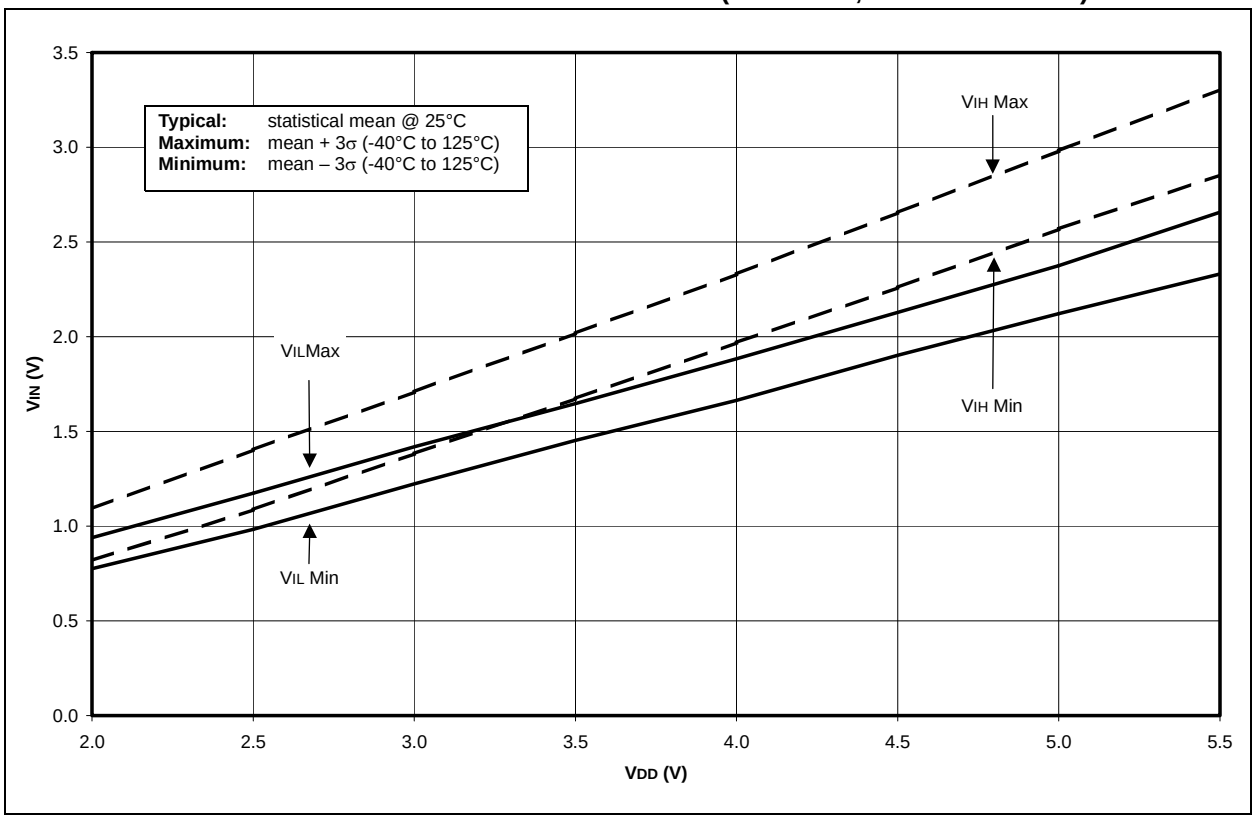
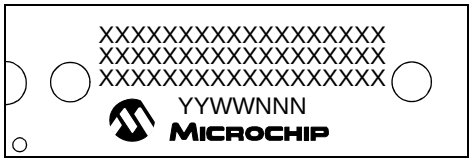


FIGURE 24-18: MINIMUM AND MAXIMUM VIN vs. VDD (I^2C INPUT, -40°C TO +125°C)

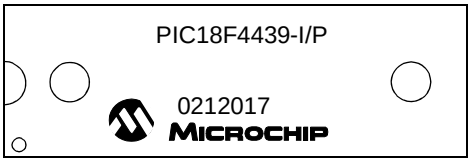


Package Marking Information (Cont'd)

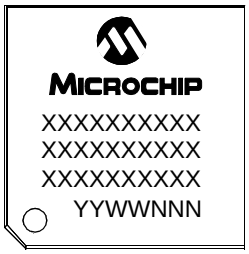
40-Lead PDIP



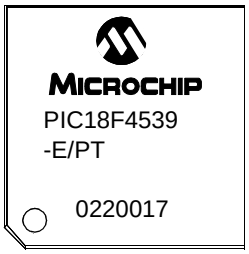
Example



44-Lead TQFP



Example



44-Lead QFN

Example