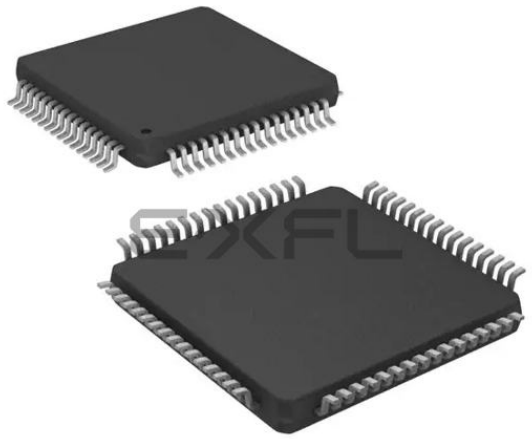


Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	50
Program Memory Size	16KB (8K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	454 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c752t-16i-pt

NOTES:

4.0 ON-CHIP OSCILLATOR CIRCUIT

The internal oscillator circuit is used to generate the device clock. Four device clock periods generate an internal instruction clock (TCY).

There are four modes that the oscillator can operate in. They are selected by the device configuration bits during device programming. These modes are:

- LF Low Frequency (FOSC ≤ 2 MHz)
- XT Standard Crystal/Resonator Frequency (2 MHz ≤ FOSC ≤ 33 MHz)
- EC External Clock Input (Default oscillator configuration)
- RC External Resistor/Capacitor (FOSC ≤ 4 MHz)

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 96 ms (nominal) on POR and BOR. The PWRT is designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external RESET circuitry.

SLEEP mode is designed to offer a very low current power-down mode. The user can wake from SLEEP through external RESET, Watchdog Timer Reset, or through an interrupt.

Several oscillator options are made available to allow the part to better fit the application. The RC oscillator option saves system cost while the LF crystal option saves power. Configuration bits are used to select various options.

4.1 Oscillator Configurations

4.1.1 OSCILLATOR TYPES

The PIC17CXXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1:FOSC0) to select one of these four modes:

- LF Low Power Crystal
- XT Crystal/Resonator
- EC External Clock Input
- RC Resistor/Capacitor

The main difference between the LF and XT modes is the gain of the internal inverter of the oscillator circuit, which allows the different frequency ranges.

For more details on the device configuration bits, see Section 17.0.

4.1.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT or LF modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 4-2). The PIC17CXXX oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications.

For frequencies above 24 MHz, it is common for the crystal to be an overtone mode crystal. Use of overtone mode crystals require a tank circuit to attenuate the gain at the fundamental frequency. Figure 4-3 shows an example circuit.

4.1.3 OSCILLATOR/RESONATOR START-UP

As the device voltage increases from Vss, the oscillator will start its oscillations. The time required for the oscillator to start oscillating depends on many factors. These include:

- Crystal/resonator frequency
- Capacitor values used (C1 and C2)
- Device VDD rise time
- System temperature
- Series resistor value (and type) if used
- Oscillator mode selection of device (which selects the gain of the internal oscillator inverter)

Figure 4-1 shows an example of a typical oscillator/resonator start-up. The peak-to-peak voltage of the oscillator waveform can be quite low (less than 50% of device VDD) when the waveform is centered at VDD/2 (refer to parameter #D033 and parameter #D043 in the electrical specification section).

FIGURE 4-1: OSCILLATOR/RESONATOR START-UP CHARACTERISTICS

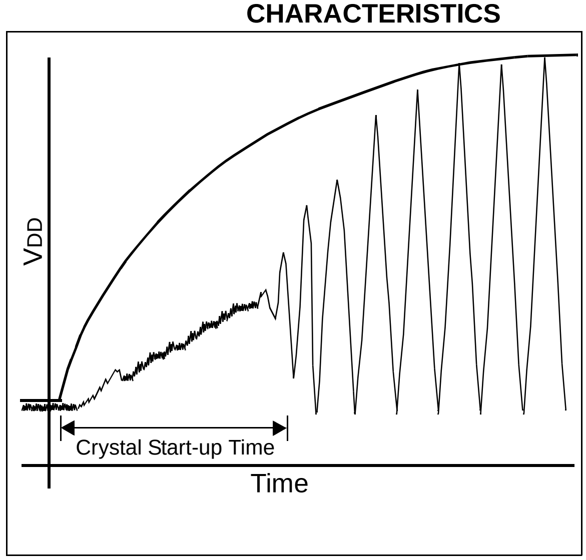
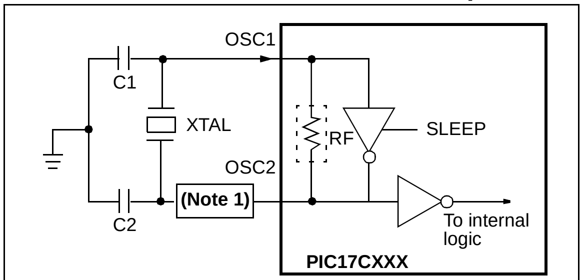


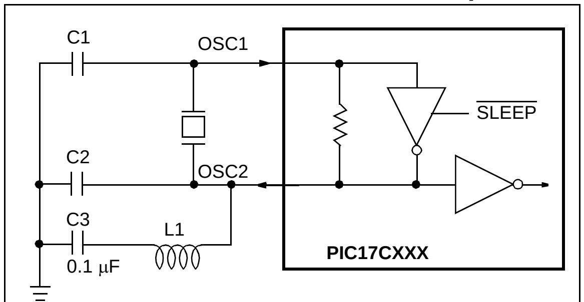
FIGURE 4-2: CRYSTAL OR CERAMIC RESONATOR OPERATION (XT OR LF OSC CONFIGURATION)



See Table 4-1 and Table 4-2 for recommended values of C1 and C2.

Note 1: A series resistor (Rs) may be required for AT strip cut crystals.

FIGURE 4-3: CRYSTAL OPERATION, OVERTONE CRYSTALS (XT OSC CONFIGURATION)



To filter the fundamental frequency:

$$\frac{1}{L1*C2} = (2\pi f)^2$$

Where f = tank circuit resonant frequency. This should be midway between the fundamental and the 3rd overtone frequencies of the crystal.
C3 blocks DC current to ground.

TABLE 4-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Oscillator Type	Resonator Frequency	Capacitor Range C1 = C2(1)
LF	455 kHz 2.0 MHz	15 - 68 pF 10 - 33 pF
XT	4.0 MHz 8.0 MHz 16.0 MHz	22 - 68 pF 33 - 100 pF 33 - 100 pF

Higher capacitance increases the stability of the oscillator, but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Note 1: These values include all board capacitances on this pin. Actual capacitor value depends on board capacitance.

Resonators Used:

455 kHz	Panasonic EFO-A455K04B	± 0.3%
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%

Resonators used did not have built-in capacitors.

TABLE 4-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Freq	C1(2)	C2(2)
LF	32 kHz 1 MHz 2 MHz	100-150 pF 10-68 pF 10-68 pF	100-150 pF 10-68 pF 10-68 pF
XT	2 MHz 4 MHz 8 MHz 16 MHz 24 MHz(1) 32 MHz(1)	47-100 pF 15-68 pF 15-47 pF 15-47 pF 15-47 pF 10-47 pF	47-100 pF 15-68 pF 15-47 pF 15-47 pF 15-47 pF 10-47 pF

Higher capacitance increases the stability of the oscillator, but also increases the start-up time and the oscillator current. These values are for design guidance only. Rs may be required in XT mode to avoid overdriving the crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values for external components.

Note 1: Overtone crystals are used at 24 MHz and higher. The circuit in Figure 4-3 should be used to select the desired harmonic frequency.

2: These values include all board capacitances on this pin. Actual capacitor value depends on board capacitance.

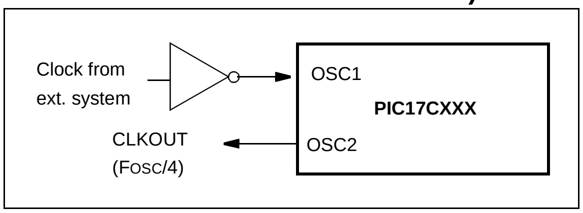
Crystals Used:

32.768 kHz	Epson C-001R32.768K-A	± 20 PPM
1.0 MHz	ECS-10-13-1	± 50 PPM
2.0 MHz	ECS-20-20-1	± 50 PPM
4.0 MHz	ECS-40-20-1	± 50 PPM
8.0 MHz	ECS ECS-80-S-4 ECS-80-18-1	± 50 PPM
16.0 MHz	ECS-160-20-1	± 50 PPM
25 MHz	CTS CTS25M	± 50 PPM
32 MHz	CRYSTEK HF-2	± 50 PPM

4.1.4 EXTERNAL CLOCK OSCILLATOR

In the EC oscillator mode, the OSC1 input can be driven by CMOS drivers. In this mode, the OSC1/CLKIN pin is hi-impedance and the OSC2/CLKOUT pin is the CLKOUT output (4 TOSC).

FIGURE 4-4: EXTERNAL CLOCK INPUT OPERATION (EC OSC CONFIGURATION)



4.1.5 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used, or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with series resonance, or one with parallel resonance.

Figure 4-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 4-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

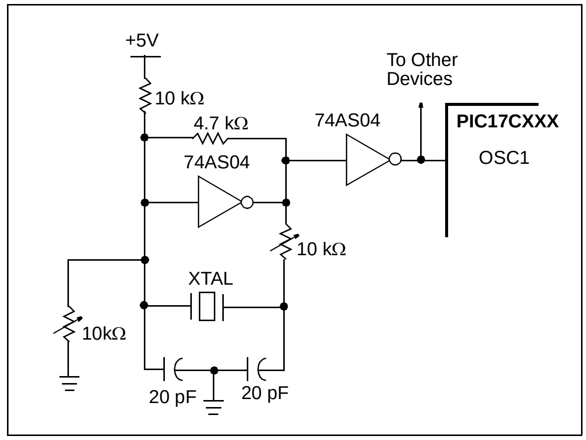


Figure 4-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 4-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT

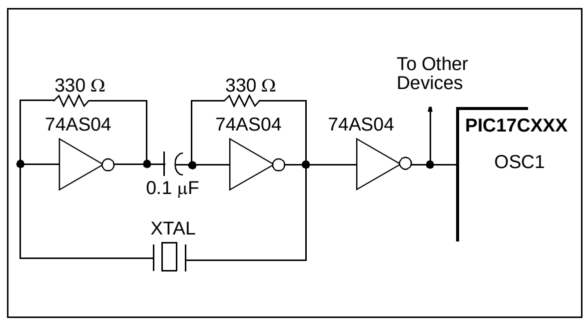


TABLE 5-4: INITIALIZATION CONDITIONS FOR SPECIAL FUNCTION REGISTERS (CONTINUED)

Register	Address	Power-on Reset Brown-out Reset	$\overline{\text{MCLR}}$ Reset WDT Reset	Wake-up from SLEEP through Interrupt
Bank 4				
PIR2	10h	000- 0010	000- 0010	uuu- uuuu(1)
PIE2	11h	000- 0000	000- 0000	uuu- uuuu
Unimplemented	12h	---- ----	---- ----	---- ----
RCSTA2	13h	0000 -00x	0000 -00u	uuuu -uuu
RCREG2	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXSTA2	15h	0000 --1x	0000 --1u	uuuu --uu
TXREG2	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
SPBRG2	17h	0000 0000	0000 0000	uuuu uuuu
Bank 5				
DDRF	10h	1111 1111	1111 1111	uuuu uuuu
PORTF(4)	11h	0000 0000	0000 0000	uuuu uuuu
DDRG	12h	1111 1111	1111 1111	uuuu uuuu
PORTG(4)	13h	xxxx 0000	uuuu 0000	uuuu uuuu
ADCON0	14h	0000 -0-0	0000 -0-0	uuuu uuuu
ADCON1	15h	000- 0000	000- 0000	uuuu uuuu
ADRESL	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESH	17h	xxxx xxxx	uuuu uuuu	uuuu uuuu
Bank 6				
SSPADD	10h	0000 0000	0000 0000	uuuu uuuu
SSPCON1	11h	0000 0000	0000 0000	uuuu uuuu
SSPCON2	12h	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	13h	0000 0000	0000 0000	uuuu uuuu
SSPBUF	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
Unimplemented	15h	---- ----	---- ----	---- ----
Unimplemented	16h	---- ----	---- ----	---- ----
Unimplemented	17h	---- ----	---- ----	---- ----

Legend: u = unchanged, x = unknown, - = unimplemented, read as '0', q = value depends on condition

Note 1: One or more bits in INTSTA, PIR1, PIR2 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GLINTD bit is cleared, the PC is loaded with the interrupt vector.

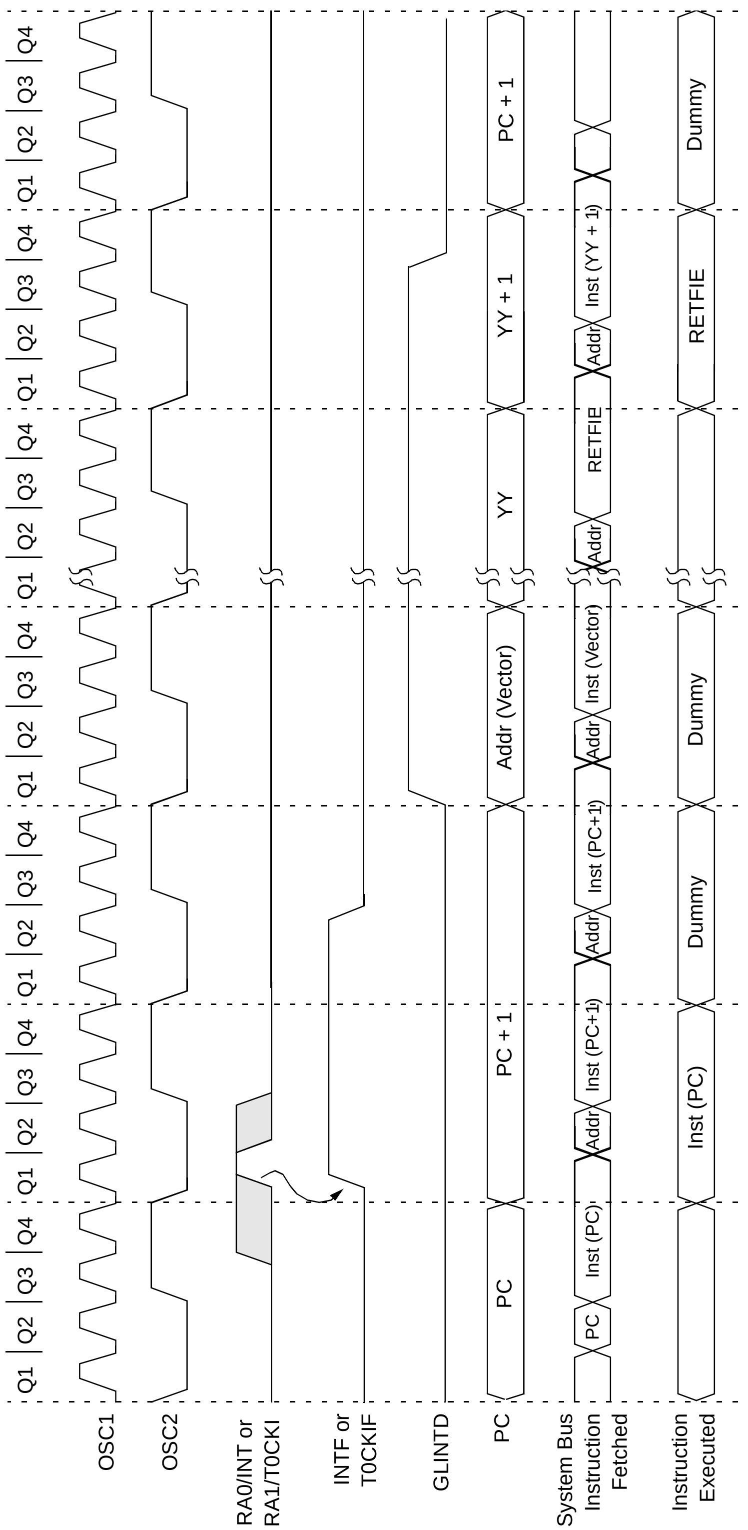
3: See Table 5-3 for RESET value of specific condition.

4: This is the value that will be in the port output latch.

5: When the device is configured for Microprocessor or Extended Microcontroller mode, the operation of this port does not rely on these registers.

6: On any device RESET, these pins are configured as inputs.

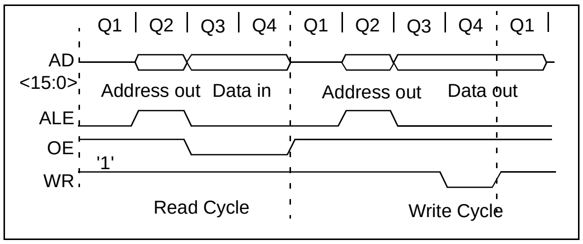
FIGURE 6-2: INT PIN/T0CKI PIN INTERRUPT TIMING



7.1.2 EXTERNAL MEMORY INTERFACE

When either Microprocessor or Extended Microcontroller mode is selected, PORTC, PORTD and PORTE are configured as the system bus. PORTC and PORTD are the multiplexed address/data bus and PORTE<2:0> is for the control signals. External components are needed to demultiplex the address and data. This can be done as shown in Figure 7-4. The waveforms of address and data are shown in Figure 7-3. For complete timings, please refer to the electrical specification section.

FIGURE 7-3: EXTERNAL PROGRAM MEMORY ACCESS WAVEFORMS



The system bus requires that there is no bus conflict (minimal leakage), so the output value (address) will be capacitively held at the desired value.

As the speed of the processor increases, external EPROM memory with faster access time must be used. Table 7-2 lists external memory speed requirements for a given PIC17C7XX device frequency.

In Extended Microcontroller mode, when the device is executing out of internal memory, the control signals will continue to be active. That is, they indicate the action that is occurring in the internal memory. The external memory access is ignored.

The following selection is for use with Microchip EPROMs. For interfacing to other manufacturers memory, please refer to the electrical specifications of the desired PIC17C7XX device, as well as the desired memory device to ensure compatibility.

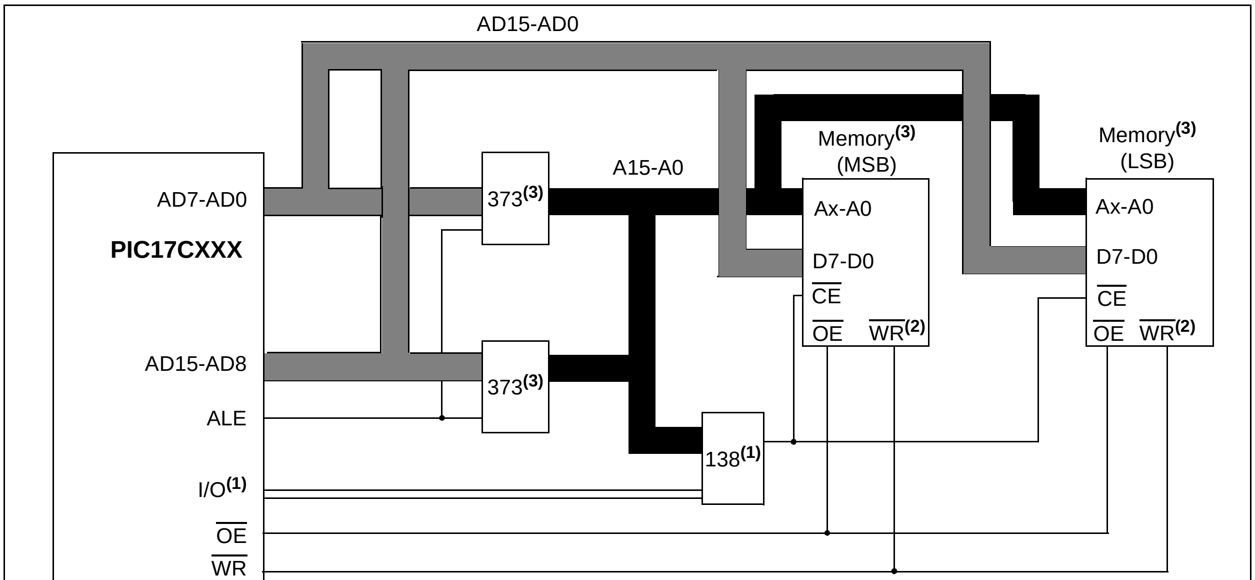
TABLE 7-2: EPROM MEMORY ACCESS TIME ORDERING SUFFIX

PIC17C7XX Oscillator Frequency	Instruction Cycle Time (TCY)	EPROM Suffix
8 MHz	500 ns	-25
16 MHz	250 ns	-15
20 MHz	200 ns	-10
25 MHz	160 ns	-70

Note: The access times for this requires the use of fast SRAMs.

The electrical specifications now include timing specifications for the memory interface with PIC17**LC**XXX devices. These specifications reflect the capability of the device by characterization. Please validate your design with these timings.

FIGURE 7-4: TYPICAL EXTERNAL PROGRAM MEMORY CONNECTION DIAGRAM



Note 1: Use of I/O pins is only required for paged memory.
2: This signal is unused for ROM and EPROM devices.
3: 16-bit wide devices are now common and could be used instead of 8-bit wide devices.

7.3 Stack Operation

PIC17C7XX devices have a 16 x 16-bit hardware stack (Figure 7-1). The stack is not part of either the program or data memory space, and the stack pointer is neither readable nor writable. The PC (Program Counter) is "PUSH'd" onto the stack when a `CALL` or `LCALL` instruction is executed, or an interrupt is acknowledged. The stack is "POP'd" in the event of a `RETURN`, `RETLW`, or a `RETFIE` instruction execution. PCLATH is not affected by a "PUSH" or a "POP" operation.

The stack operates as a circular buffer, with the stack pointer initialized to '0' after all RESETS. There is a stack available bit (STKAV) to allow software to ensure that the stack will not overflow. The STKAV bit is set after a device RESET. When the stack pointer equals Fh, STKAV is cleared. When the stack pointer rolls over from Fh to 0h, the STKAV bit will be held clear until a device RESET.

Note 1: There is not a status bit for stack underflow. The STKAV bit can be used to detect the underflow which results in the stack pointer being at the Top-of-Stack.

2: There are no instruction mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW` and `RETFIE` instructions, or the vectoring to an interrupt vector.

3: After a RESET, if a "POP" operation occurs before a "PUSH" operation, the STKAV bit will be cleared. This will appear as if the stack is full (underflow has occurred). If a "PUSH" operation occurs next (before another "POP"), the STKAV bit will be locked clear. Only a device RESET will cause this bit to set.

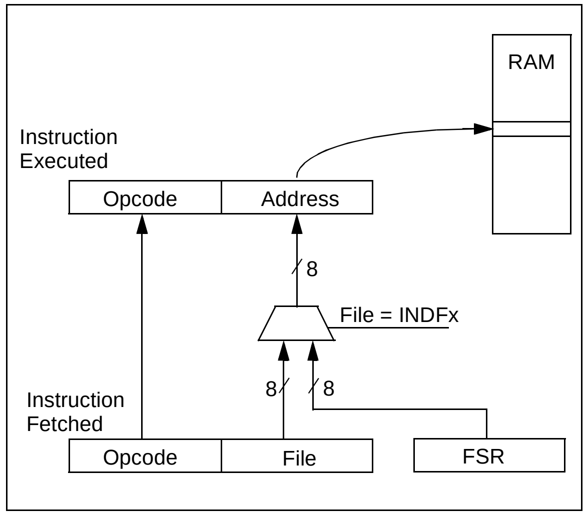
After the device is "PUSH'd" sixteen times (without a "POP"), the seventeenth push overwrites the value from the first push. The eighteenth push overwrites the second push (and so on).

7.4 Indirect Addressing

Indirect addressing is a mode of addressing data memory where the data memory address in the instruction is not fixed. That is, the register that is to be read or written can be modified by the program. This can be useful for data tables in the data memory. Figure 7-6 shows the operation of indirect addressing. This depicts the moving of the value to the data memory address specified by the value of the FSR register.

Example 7-1 shows the use of indirect addressing to clear RAM in a minimum number of instructions. A similar concept could be used to move a defined number of bytes (block) of data to the USART transmit register (TXREG). The starting address of the block of data to be transmitted could easily be modified by the program.

FIGURE 7-6: INDIRECT ADDRESSING



7.4.1 INDIRECT ADDRESSING REGISTERS

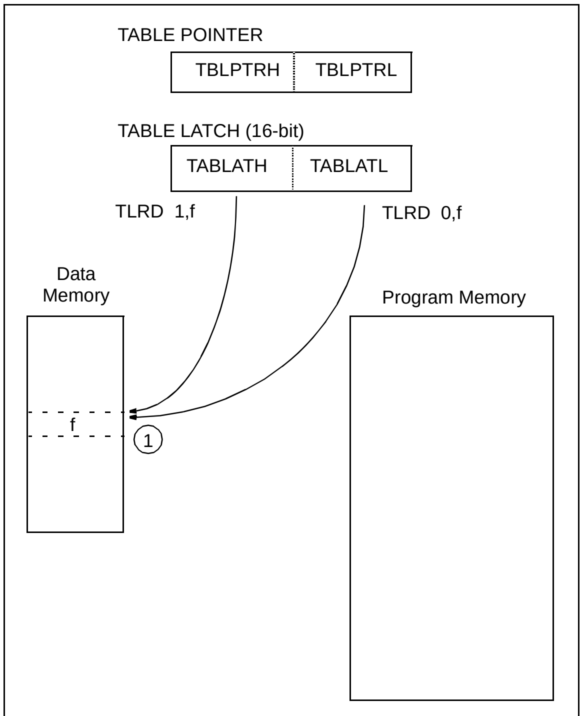
The PIC17C7XX has four registers for indirect addressing. These registers are:

- INDF0 and FSR0
- INDF1 and FSR1

Registers INDF0 and INDF1 are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. The FSR is an 8-bit register and allows addressing anywhere in the 256-byte data memory address range. For banked memory, the bank of memory accessed is specified by the value in the BSR.

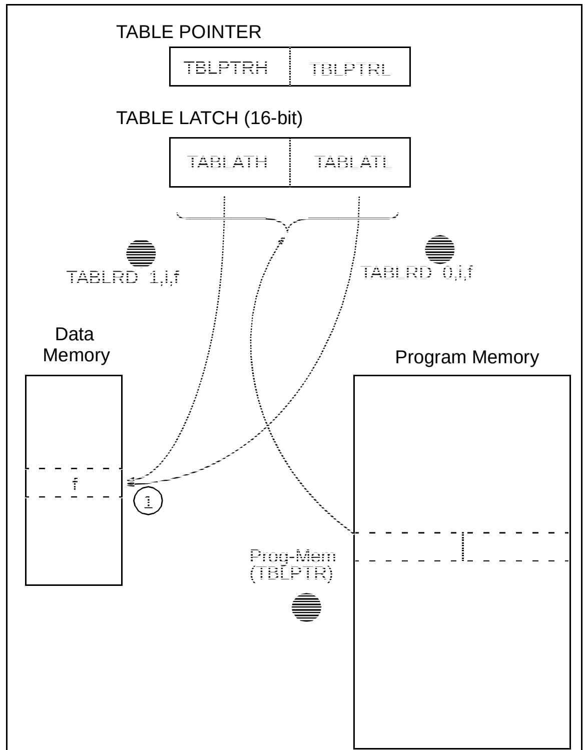
If file INDF0 (or INDF1) itself is read indirectly via an FSR, all '0's are read (Zero bit is set). Similarly, if INDF0 (or INDF1) is written to indirectly, the operation will be equivalent to a `NOP`, and the status bits are not affected.

FIGURE 8-3: TLRD INSTRUCTION OPERATION



Step 1: 8-bit value from TABLAT (16-bit) high or low byte, loaded into register 'f'.

FIGURE 8-4: TABLRD INSTRUCTION OPERATION



Step 1: 8-bit value from TABLAT (16-bit) high or low byte, loaded into register 'f'.
2: 16-bit value at Program Memory (TBLPTR), loaded into TABLAT register.
3: If "i" = 1, then TBLPTR = TBLPTR + 1,
If "i" = 0, then TBLPTR is unchanged.

10.2 PORTB and DDRB Registers

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is DDRB. A '1' in DDRB configures the corresponding port pin as an input. A '0' in the DDRB register configures the corresponding port pin as an output. Reading PORTB reads the status of the pins, whereas writing to PORTB will write to the port latch.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (PORTA<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are enabled on any RESET.

PORTB also has an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB0 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB0) are compared with the value in the PORTB data latch. The “mismatch” outputs of RB7:RB0 are OR’d together to set the PORTB Interrupt Flag bit, RBIF (PIR1<7>).

This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt by:

a) Read-Write PORTB (such as: `MOVPF PORTB, PORTB`). This will end the mismatch condition.

b) Then, clear the RBIF bit.

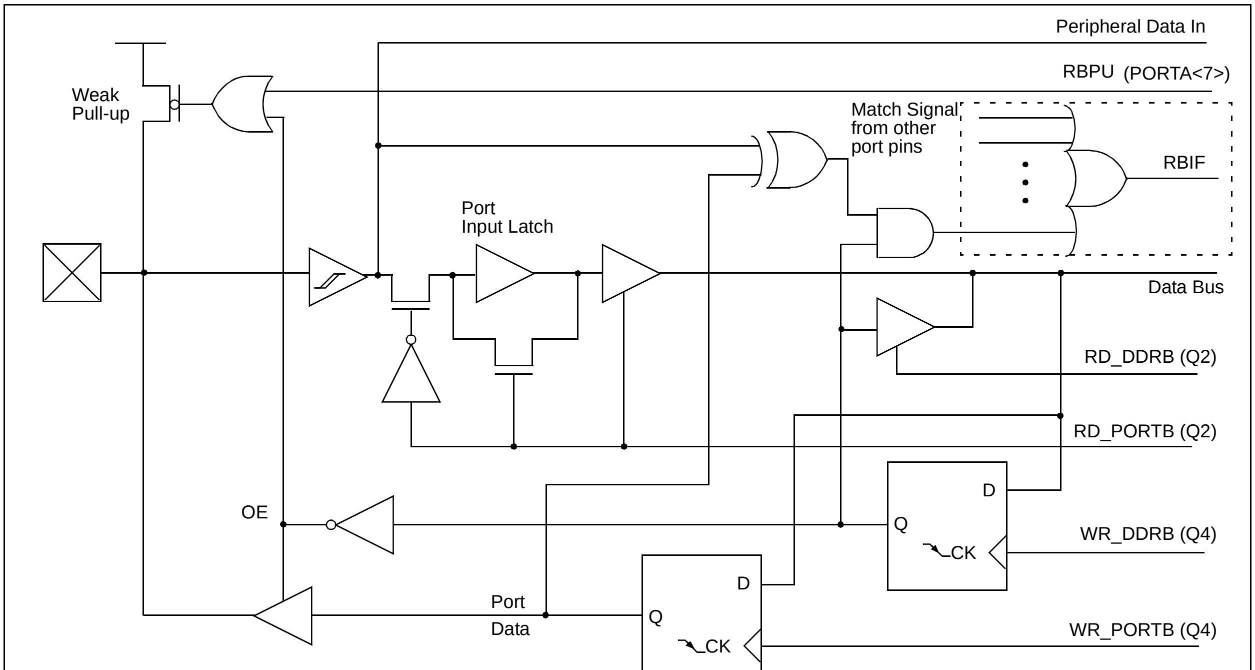
A mismatch condition will continue to set the RBIF bit. Reading, then writing PORTB, will end the mismatch condition and allow the RBIF bit to be cleared.

This interrupt-on-mismatch feature, together with software configurable pull-ups on this port, allows easy interface to a keypad and makes it possible for wake-up on key depression. For an example, refer to Application Note AN552, “Implementing Wake-up on Keystroke.”

The interrupt-on-change feature is recommended for wake-up on operations, where PORTB is only used for the interrupt-on-change feature and key depression operations.

Note:	On a device RESET, the RBIF bit is indeterminate, since the value in the latch may be different than the pin.

FIGURE 10-5: BLOCK DIAGRAM OF RB5:RB4 AND RB1:RB0 PORT PINS



Note: I/O pins have protection diodes to VDD and VSS.

NOTES:

14.2 USART Asynchronous Mode

In this mode, the USART uses standard nonreturn-to-zero (NRZ) format (one START bit, eight or nine data bits, and one STOP bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock x64 of the bit shift rate. Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

The Asynchronous mode is selected by clearing the SYNC bit (TXSTA<4>).

The USART Asynchronous module consists of the following components:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

14.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 14-1. The heart of the transmitter is the transmit shift register (TSR). The shift register obtains its data from the read/write transmit buffer (TXREG). TXREG is loaded with data in software. The TSR is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once TXREG transfers the data to the TSR (occurs in one TCY at the end of the current BRG cycle), the TXREG is empty and an interrupt bit, TXIF, is set. This interrupt can be enabled/disabled by setting/clearing the TXIE bit. TXIF will be set, regardless of TXIE and cannot be reset in software. It will reset only when new data is loaded into TXREG. While TXIF indicates the status of the TXREG, the TRMT (TXSTA<1>) bit shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty.

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

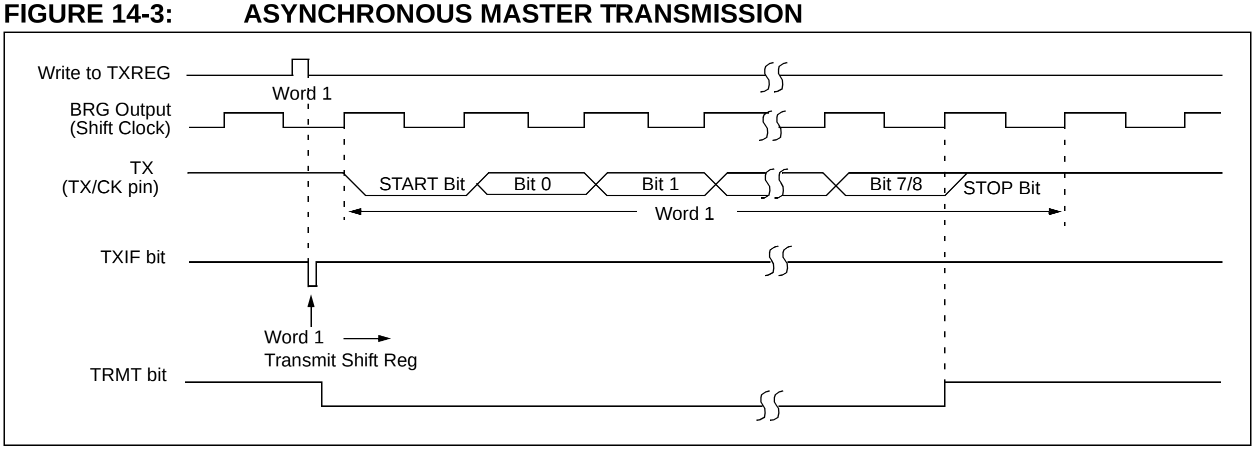
Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until TXREG has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 14-3). The transmission can also be started by first loading TXREG and then setting TXEN. Normally, when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 14-4). Clearing TXEN during a transmission will cause the transmission to be aborted. This will reset the transmitter and the TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, the TX9 (TXSTA<6>) bit should be set and the ninth bit value should be written to TX9D (TXSTA<0>). The ninth bit value must be written before writing the 8-bit data to the TXREG. This is because a data write to TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty).

Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are desired, then set the TXIE bit.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. If 9-bit transmission is selected, the ninth bit should be loaded in TX9D.
6. Load data to the TXREG register.
7. Enable the transmission by setting TXEN (starts transmission).

FIGURE 14-3: ASYNCHRONOUS MASTER TRANSMISSION



15.2.10 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I²C module is in the idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the baud rate generator is loaded with the contents of SSPADD<6:0> and begins counting. The SDA pin is released (brought high) for one baud rate generator count (TBRG). When the baud rate generator times out, if SDA is sampled high, the SCL pin will be de-asserted (brought high). When SCL is sampled high the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA is low) for one TBRG while SCL is high. Following this, the RSEN bit in the SSPCON2 register will be automatically cleared and the baud rate generator is not reloaded, leaving the SDA pin held low. As soon as a START condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the baud rate generator has timed out.

> **Note 1:** If the RSEN is programmed while any other event is in progress, it will not take effect.
>
> **2:** A bus collision during the Repeated Start condition occurs if:
> - SDA is sampled low when SCL goes from low to high.
> - SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data “1”.

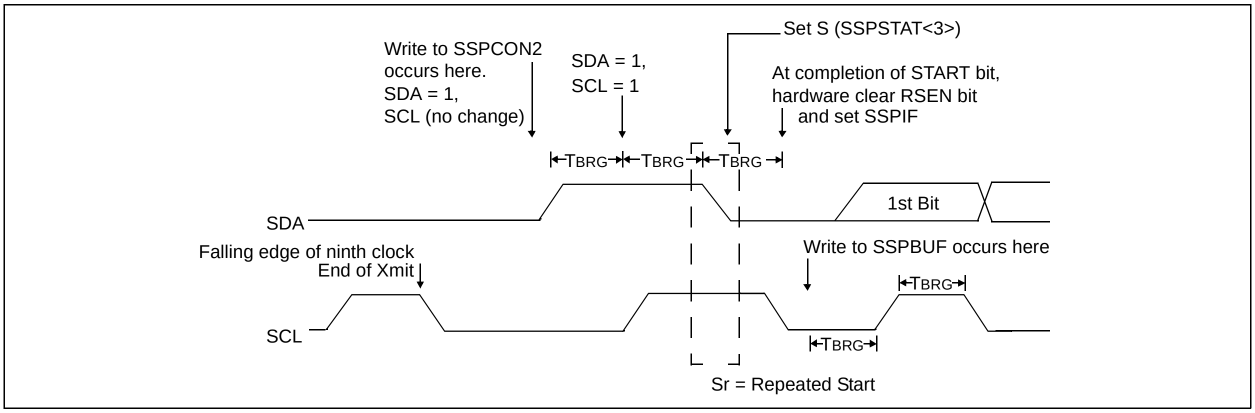
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode), or eight bits of data (7-bit mode).

15.2.10.1 WCOL status flag

If the user writes the SSPBUF when a Repeated Start sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn’t occur).

> **Note:** Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

FIGURE 15-22: REPEAT START CONDITION WAVEFORM



CPFSLT — Compare f with WREG, skip if f < WREG

Syntax: [*label*] CPFSLT f

Operands: $0 \le f \le 255$

Operation: (f) – (WREG),
skip if (f) < (WREG)
(unsigned comparison)

Status Affected: None

Encoding:

0011	0000	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of WREG by performing an unsigned subtraction.

If the contents of 'f' are less than the contents of WREG, then the fetched instruction is discarded and a `NOP` is executed instead, making this a two-cycle instruction.

Words: 1

Cycles: 1 (2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

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

Example:
```
HERE    CPFSLT REG
NLESS   :
LESS    :
```

Before Instruction
PC = Address (`HERE`)
W = ?

After Instruction
If REG < WREG;
PC = Address (`LESS`)
If REG ≥ WREG;
PC = Address (`NLESS`)

DAW — Decimal Adjust WREG Register

Syntax: [*label*] DAW f,s

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

Operation: If [[WREG<7:4> > 9].OR.[C = 1]].AND.
[WREG<3:0> > 9]
then
WREG<7:4> + 7→ f<7:4>, s<7:4>;

If [WREG<7:4> > 9].OR.[C = 1]
then
WREG<7:4> + 6→ f<7:4>, s<7:4>;
else
WREG<7:4>→ f<7:4>, s<7:4>;

If [WREG<3:0> > 9].OR.[DC = 1]
then
WREG<3:0> + 6→ f<3:0>, s<3:0>;
else
WREG<3:0>→ f<3:0>, s<3:0>

Status Affected: C

Encoding:

0010	111s	ffff	ffff

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

s = 0: Result is placed in Data memory location 'f' and WREG.

s = 1: Result is placed in Data memory location 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f' and other specified register

Example: `DAW REG1, 0`

Before Instruction
WREG = 0xA5
REG1 = ??
C = 0
DC = 0

After Instruction
WREG = 0x05
REG1 = 0x05
C = 1
DC = 0

RRNCF — Rotate Right f (no carry)

Syntax: [*label*] RRNCF f,d

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

Operation: f<n> → d<n-1>;
f<0> → d<7>

Status Affected: None

Encoding:

`0010`	`000d`	`ffff`	`ffff`

Description: The contents of register 'f' are rotated one bit to the right. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1: `RRNCF REG, 1`

Before Instruction
WREG = ?
REG = `1101 0111`

After Instruction
WREG = `0`
REG = `1110 1011`

Example 2: `RRNCF REG, 0`

Before Instruction
WREG = ?
REG = `1101 0111`

After Instruction
WREG = `1110 1011`
REG = `1101 0111`

SETF — Set f

Syntax: [*label*] SETF f,s

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

Operation: FFh → f;
FFh → d

Status Affected: None

Encoding:

`0010`	`101s`	`ffff`	`ffff`

Description: If 's' is 0, both the data memory location 'f' and WREG are set to FFh. If 's' is 1, only the data memory location 'f' is set to FFh.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f' and other specified register

Example1: `SETF REG, 0`

Before Instruction
REG = 0xDA
WREG = 0x05

After Instruction
REG = 0xFF
WREG = 0xFF

Example2: `SETF REG, 1`

Before Instruction
REG = 0xDA
WREG = 0x05

After Instruction
REG = 0xFF
WREG = 0x05

FIGURE 20-20: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

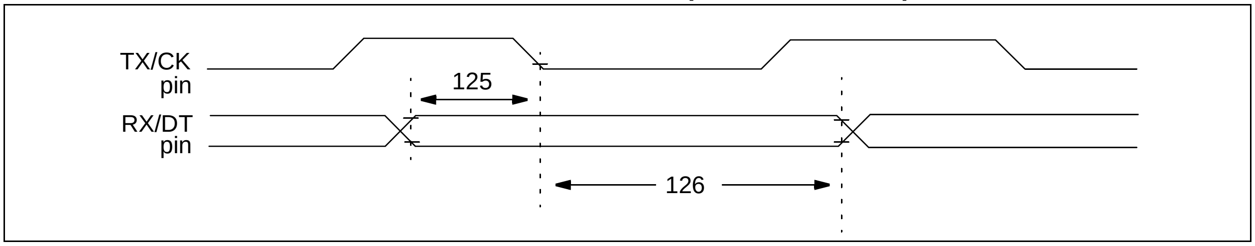


TABLE 20-15: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data setup before CK↓ (DT setup time)	15	—	—	ns	
126	TckL2dtl	Data hold after CK↓ (DT hold time)	15	—	—	ns	

† Data in “Typ” column is at 5V, 25°C unless otherwise stated.

FIGURE 21-11: TYPICAL AND MAXIMUM IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED, -40°C to +125°C)

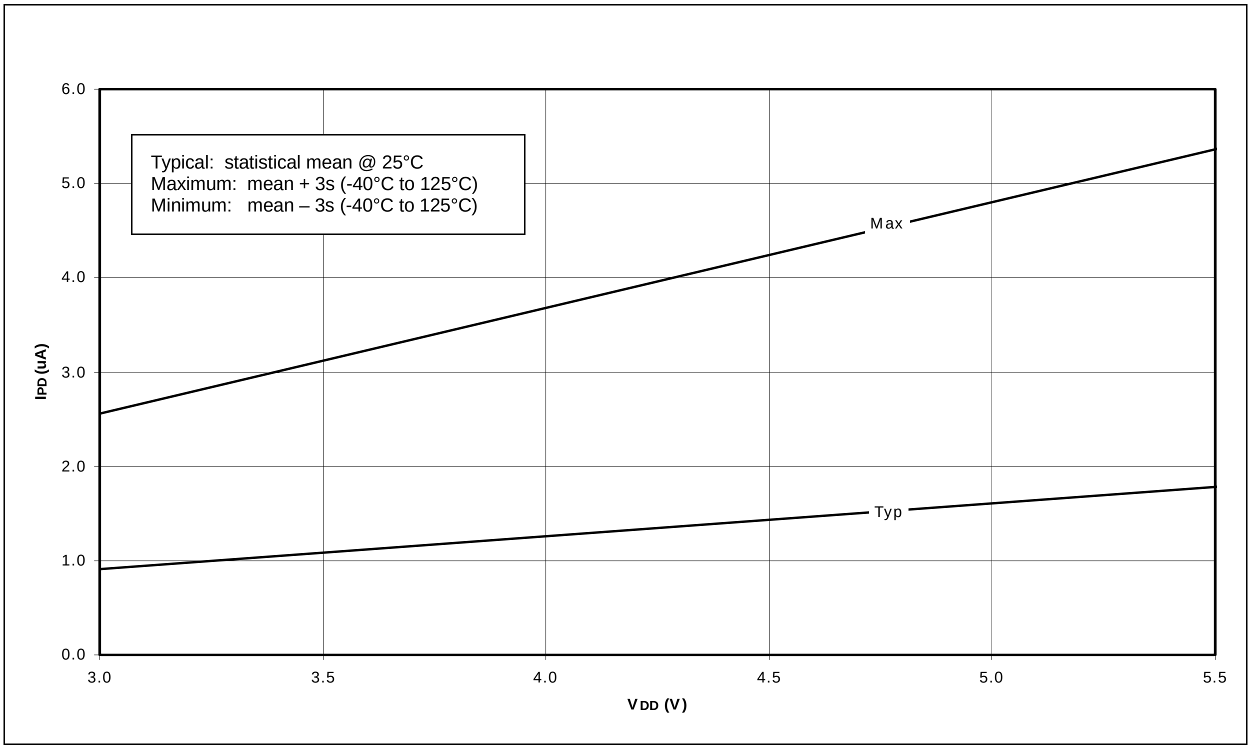
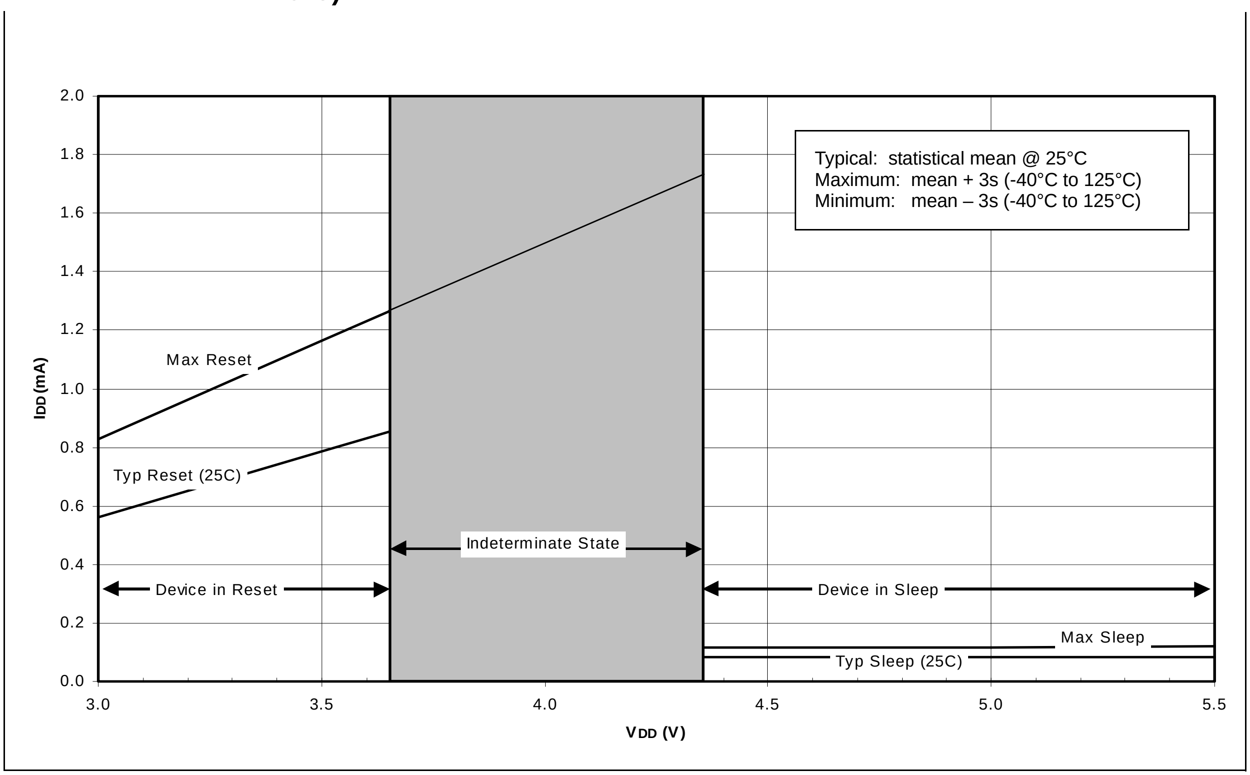
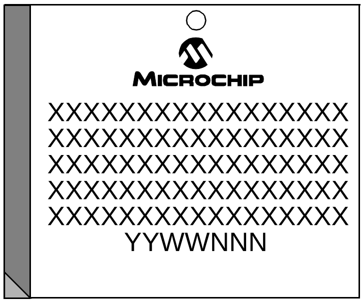


FIGURE 21-12: TYPICAL AND MAXIMUM IPD vs. VDD (SLEEP MODE, BOR ENABLED, -40°C to +125°C)

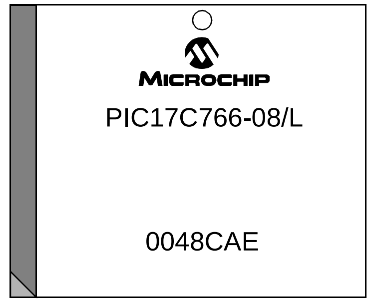


Package Marking Information (Cont.)

84-Lead PLCC



Example



T

NOTES:

