

Power-up Trouble Shooting

*Author: Mark Palmer
Microchip Technology Inc.*

*Contributions: Richard Hull & Randy Yach
Microchip Technology Inc.*

INTRODUCTION

For any application to begin proper operation, the application must power-up properly. Many criteria must be taken into account to ensure this. The PICmicro™ devices integrate several features to simplify the design for the power-up sequence. These integrated features also reduce the total system cost.

This application note describes the requirements for the device to properly power-up, common pitfalls that designers encounter, and methods to assist in solving power-up problems.

THE POWER-UP SEQUENCE

There are several factors that determine the actual power-up sequence that a device will go through. These factors are:

- The Processor Family
 - PIC16C5X (Baseline)
 - PIC16CXXX (Midrange)
 - PIC17CXXX (High-end)
- Oscillator Configuration
- Device Configuration
- MCLR pin

Note: The PIC16CXXX family refers to devices with a 14-bit instruction word. This does not include the PIC16C5X family.

The Power-on Reset (POR) signal generation is discussed, followed by the power-up sequence for the specific device families.

Power-on Reset (POR) signal

The data sheets show a Power-on Reset (POR) pulse, as in Figure 1. The POR signal is a level triggered signal. This representation may help in the understanding of future devices, which may have a brown-out reset capability.

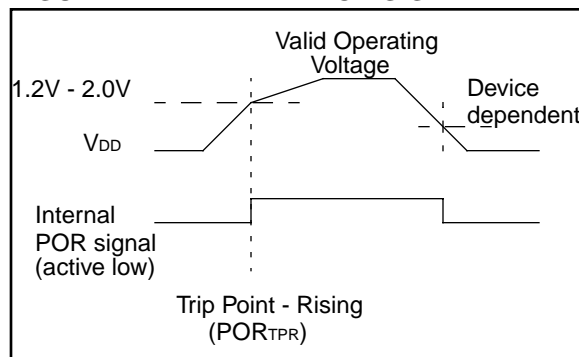
The power-up sequence begins by increasing the voltage on the V_{DD} pin (from 0V). If the slope of the V_{DD} rise time is faster than 0.05 V/ms, the internal circuitry is

capable of generating an internal reset signal. Depending on the device family, different power-up sequences will occur after this POR signal.

If the slope is less than 0.05 V/ms, the \overline{MCLR} pin should be held low, by external circuitry, until a valid operating V_{DD} level is reached.

The V_{DD} rise time specification needs to be met, until the POR signal is generated. After the POR signal is generated the slope of the V_{DD} rise can change (to a faster or slower rise). This may have other ramifications, see the "Power-up Consideration" section. In general, the POR signal will trip (POR_{TP}) somewhere between 1.2V to 2.0V (Figure 1).

FIGURE 1: INTERNAL POR SIGNAL



When V_{DD} is falling, the voltage at which the internal POR signal returns to a low level is processor/device dependent. To ensure that a device will have a POR, the device voltage must return to V_{SS} before power is re-applied.

Note: Some devices (with EPROM program memory) have a newer POR circuit that does not require V_{DD} to return to V_{SS} . See the device data sheet for the complete specification on the POR operation.

The POR will be generated regardless of the level of the \overline{MCLR} pin. The PICmicro device families are different on what triggers the power-up sequence. Table 1 describes the events that cause the POR sequence to occur.

After reaching the POR trip point (POR_{TPR}), the POR sequence holds the device in reset for a given time. Once this time has elapsed, the device voltage must be valid or the \overline{MCLR} pin must be low. The time from the POR rising edge to the time that V_{DD} must be valid level is the $T_{POR2VDDV}$ time.

TABLE 1: EVENTS THAT TRIGGER POR SEQUENCE

Device	Events
PIC16C5X	Both the POR signal rising edge and any $\overline{\text{MCLR}}$ rising edge ⁽¹⁾
PIC16CXXX	The POR signal rising edge
PIC17CXXX	Either the POR signal rising edge or the first $\overline{\text{MCLR}}$ rising edge (if $\overline{\text{MCLR}}$ is low when the POR occurs). After this event, all following $\overline{\text{MCLR}}$ rising edges ⁽¹⁾ cause the device to start program execution immediately.

Note 1: The POR low-to-high transition onfigures Special Function Register (SFR) bits/registers to a specified value. The SFR bits/register are not identically affected by the $\overline{\text{MCLR}}$ signal. Refer to the device data sheet to see how the bits are affected by these two conditions.

The POR sequence for each of the PICmicro families is described in the following three sections:

PIC16C5X Family

PIC16CXXX Family

PIC17CXXX Family

PIC16C5X Family

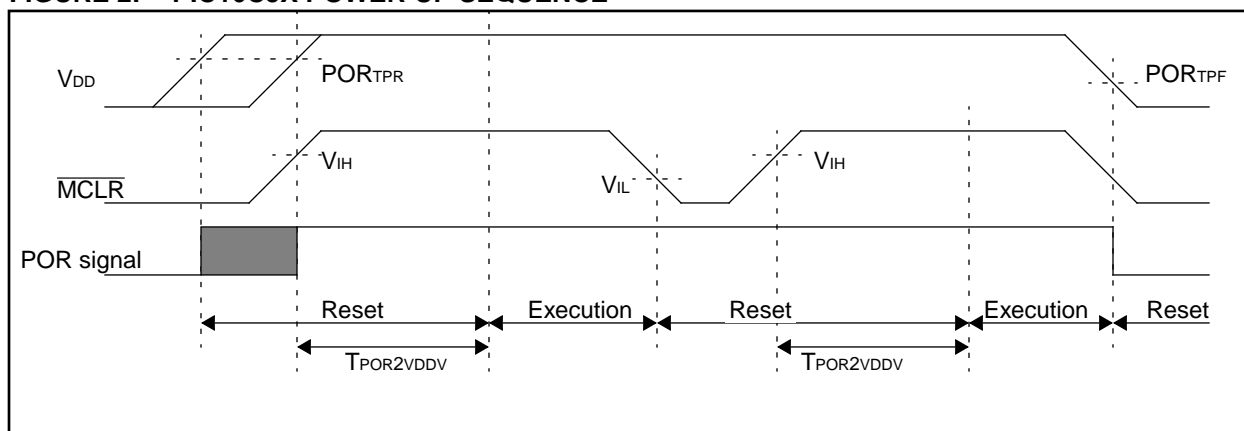
After the $\overline{\text{MCLR}}$ pin has reached a high level, the device is held in reset for typically 18 ms. This time is determined by an on-chip RC oscillator and 8-bit ripple counter. This Device Reset Timer (DRT), allows most crystals (except low frequency crystals) to start-up and stabilize. Due to the characteristics of resistors and capacitors, this time is extremely variable over temperature and voltage. There is also a device to device variation. See the data sheet for the range of this time-out.

TABLE 2: TIME-OUT IN VARIOUS SITUATIONS (TYPICAL)

Oscillator Configuration	Power-up	Wake-up from SLEEP
XT, HS, LP ⁽¹⁾	18 ms	18 ms
RC	18 ms	18 ms

Note 1: 32 kHz crystals have a typical start-up time of 1-2 seconds. Crystals >100 kHz have a typical start-up time of 10-20 ms. Resonators are typically <1 ms. All these times are voltage dependent.

FIGURE 2: PIC16C5X POWER-UP SEQUENCE



PIC16CXXX Family

After the POR rising edge has occurred, the device can have up to 2 time-out sequences that occur in series. The first being the Power-up Timer (PWRT), the second being the Oscillator Start-up Timer (OST).

The Power-up Timer time-out will occur if enable bit PWRTE is read as a '1'. The PWRT uses a 10-bit counter, with the clock from an internal RC. Due to the characteristics of resistors and capacitors, this time is extremely variable over temperature and voltage. There is also a device to device variation. See the data sheet for the range of this time-out.

Note: Future devices will change the polarity of the PWRTE configuration bit. Refer to the specific data sheet for the polarity of this bit.

The OST will occur on power-up/wake-up when the device has oscillator mode selected. This allows the oscillator to stabilize before program execution begins. The OST uses a 10-bit counter, with the clock from the OSC pin. The time is dependent on the frequency of the input clock. This timer is disabled if the oscillator is configured as RC.

Figure 3 shows how the two timers work in the power-up sequence. V_{DD} must be valid when program execution starts. The $T_{PWRT} + T_{OST}$ times can be thought of as the time that the device gives for the V_{DD} to become valid ($T_{POR2VDDV}$). Figure 4 shows when device execution begins for the case of the MCLR pin going high before $T_{POR2VDDV}$ times out. Figure 5 shows when the MCLR pin is held low longer than the $T_{POR2VDDV}$ time. The device starts execution immediately when MCLR goes high. Table 3 gives typical reset times.

TABLE 3: TIME-OUT IN VARIOUS SITUATIONS (TYPICAL)

Oscillator Configuration	Power-up		Wake-up from SLEEP
	PWRTE = 1 (2)	PWRTE = 0 (2)	
XT, HS, LP ⁽¹⁾	72 ms + 1024 T_{OSC}	1024 T_{OSC}	1024 T_{OSC}
RC	72 ms	—	—

Note 1: 32 kHz crystals have a typical start-up time of 1-2 seconds. Crystals >100 kHz have a typical start-up time of 10-20 ms. Resonators are typically <1 ms. All these times are voltage dependent.

Note 2: Future devices will change the polarity of this configuration bit. Refer to the specific data sheet for the polarity of the PWRT Configuration Bit.

FIGURE 3: PIC16CXX POWER-UP SEQUENCE

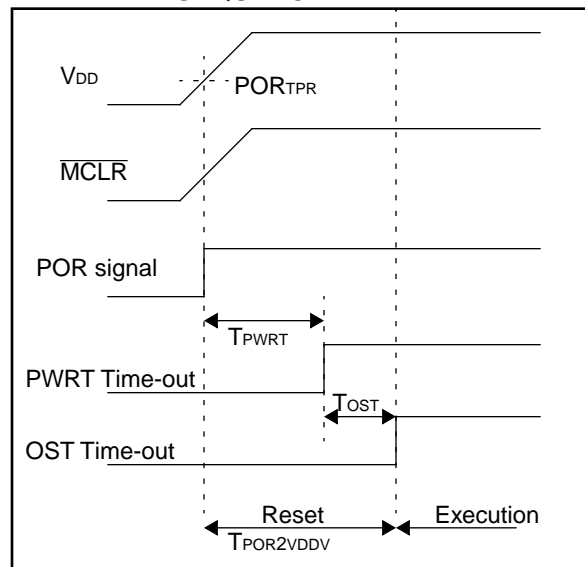


FIGURE 4: START OF DEVICE OPERATION ($\overline{MCLR} < T_{POR2VDDV}$)

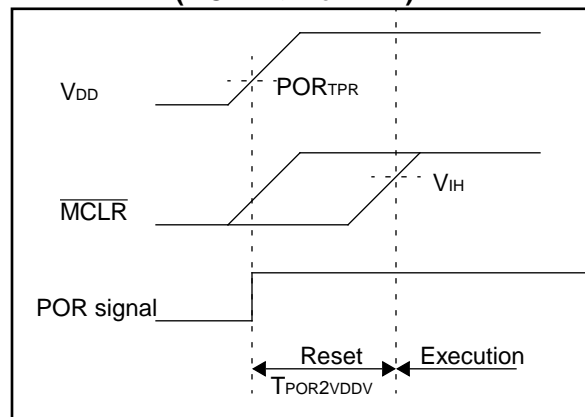
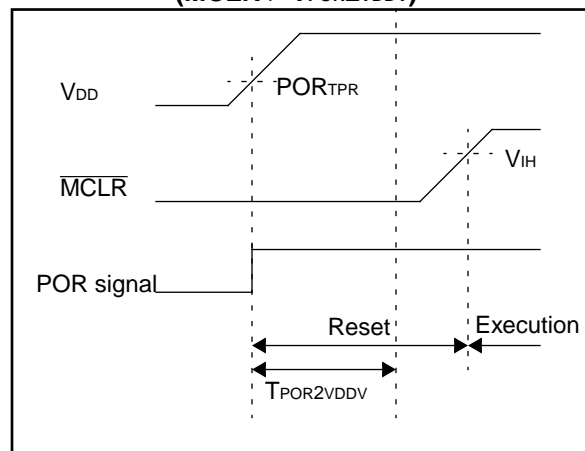


FIGURE 5: START OF DEVICE OPERATION ($\overline{MCLR} > T_{POR2VDDV}$)



AN607

PIC17CXXX Family

When the $\overline{\text{MCLR}}$ pin comes to a high level, after the POR rising edge, the device has 2 time-out sequences that occur in parallel. One is the Power-up Timer (PWRT), the other is the Oscillator Start-up Timer (OST). The timer with the greater time holds the device in reset. Figure 6 shows the sequence with $\overline{\text{MCLR}}$ tied to V_{DD} . Figure 7 shows the time-out when $\overline{\text{MCLR}}$ is independent of V_{DD} . The PWRT time is generally longer, except for low frequency crystals/resonators. The OST time does not include the start-up time of the oscillator/resonator.

The PWRT uses a 10-bit counter, with the clock from an internal RC. The characteristics of the RC vary from device to device and over temperature and voltage. The specification for the time-out range can be found in the electrical specification of the data sheet.

The OST uses a 10-bit counter, with the clock from the OSC pin. The time is dependent on the frequency of the input clock.

Until $\overline{\text{MCLR}}$ has reached a high level, the POR sequence will not start. While the POR signal remains high, all following $\overline{\text{MCLR}}$ pulses will not cause the POR sequences to occur (Figure 8).

TABLE 4: TIME-OUT IN VARIOUS SITUATIONS (TYPICAL)

Oscillator Configuration	Power-up	Wake-up from Sleep
RC, EC	Greater of 80 ms and $1024 T_{osc}$	—
XT, LF ⁽¹⁾	Greater of 80 ms and $1024 T_{osc}$	$1024 T_{osc}$

Note 1: 32 kHz crystals have a typical start-up time of 1-2 seconds. Crystals >100 kHz have a typical start-up time of 10-20 ms. Resonators are typically <1 ms. All these times are voltage dependent.

FIGURE 6: PIC17CXX POWER-UP SEQUENCE ($\overline{\text{MCLR}}$ TIED TO V_{DD})

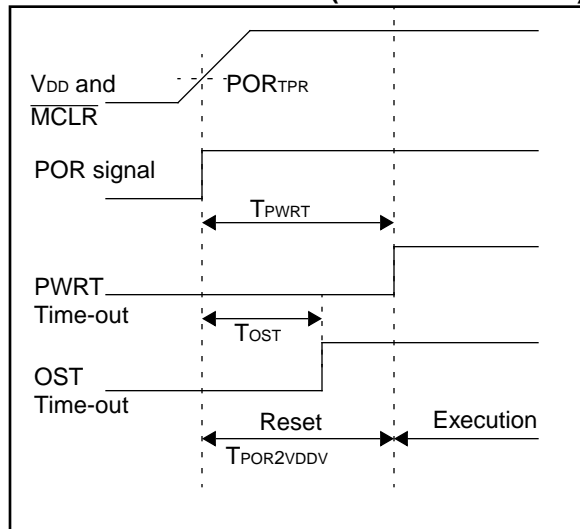


FIGURE 7: PIC17CXX POWER-UP SEQUENCE ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD})

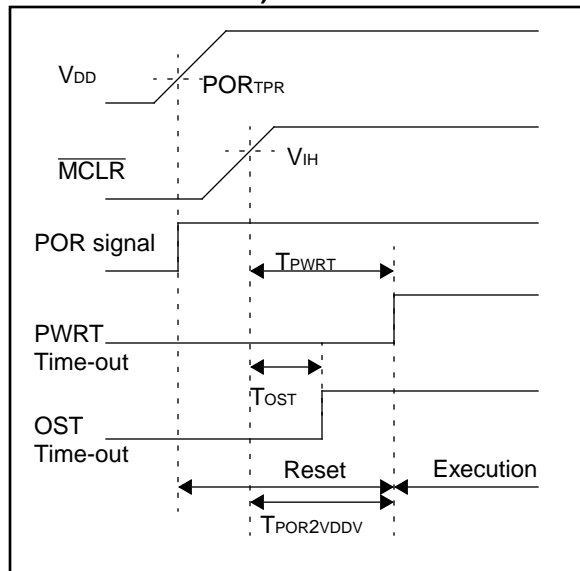
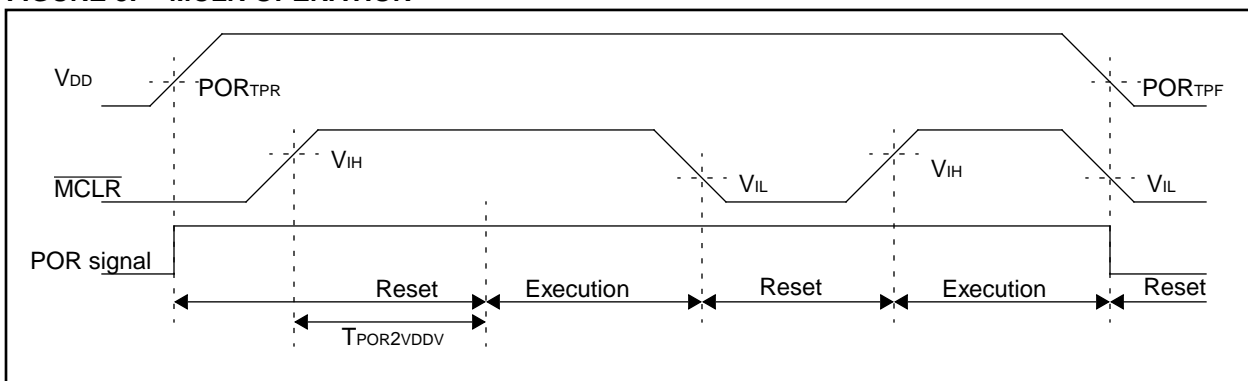


FIGURE 8: $\overline{\text{MCLR}}$ OPERATION



POWER-UP CONSIDERATIONS

The device must be at a valid operating voltage when the device exits reset. This can be done by ensuring that the power supply rise time is fast enough to guarantee an operating V_{DD} level, or by using an external reset circuit which will hold \overline{MCLR} low until the operating V_{DD} level is reached.

When the rise time of V_{DD} is very fast, there will be a time delay before the Power-on Reset (POR) signal will rise to a logic high ($T_{TP2PORH}$). This delay is in the 1-5 μ s range, as shown in Figure 9.

Figure 10, Figure 11, and Figure 12 show the maximum time from the POR sequence beginning to the device having a valid operating voltage. Table 5 gives the $T_{POR2VDDV}$ times. When determining the time at which V_{DD} must be valid, the POR trip point must be assumed to be at the minimum POR voltage trip point.

How Crystal Frequencies affect Start-up time

Both the PIC16CXXX and PIC17CXXX families may have start-up times that include the contributions of the oscillator. Table 5 shows how the oscillator can affect each mode of operation, with Table 6 giving the reset time that an oscillator generates. This time can be used in the equation to calculate the total reset time, at the given frequency. This time may vary slightly due to the initial start-up characteristics of the crystal/oscillator circuit.

Note 1: The rise time specification does not ensure that a valid V_{DD} operating voltage will be reached before the device exits reset. The device's V_{DD} must be within the specified operating range for proper device operation.

Note 2: The start-up characteristics of the crystal/oscillator must also be taken into account when determining the time that the device must be held in reset.

TABLE 5: MAXIMUM TIME FROM POR RISING EDGE TO VALID V_{DD} VOLTAGE

	Osc Mode	Maximum Time	Conditions
PIC16C5X	LP, XT, HS, and RC	9 ms	
PIC16CXXX	RC	28 ms	
	LP, XT, and HS	28 ms + $1024 T_{osc}$	PWRTE = 1
	LP, XT, and HS	$1024 T_{osc}$	PWRTE = 0
PIC17CXXX	LF, XT, EC, and RC	Greater of (40 ms or $1024 T_{osc}$)	

FIGURE 9: POR DELAY FOR FAST V_{DD} RISE TIME

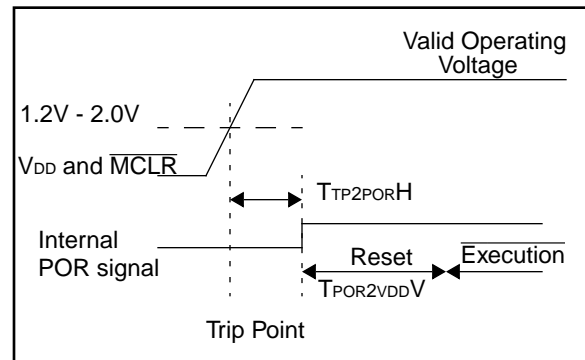


TABLE 6: RESET TIME DUE TO OSCILLATOR

	Clock Frequency								
	32 kHz	1 MHz	2 MHz	4 MHz	8 MHz	10 MHz	16 MHz	20 MHz	25 MHz
$1024 T_{osc}$	32 ms	1.0 ms	512 μ s	256 μ s	128 μ s	102.4 μ s	64 μ s	51.2 μ s	41 μ s

AN607

FIGURE 10: MAXIMUM POWER-UP TIME, $\overline{\text{MCLR}}$ TIED TO V_{DD} (PIC16C5X, PIC16CXXX, PIC17CXXX)

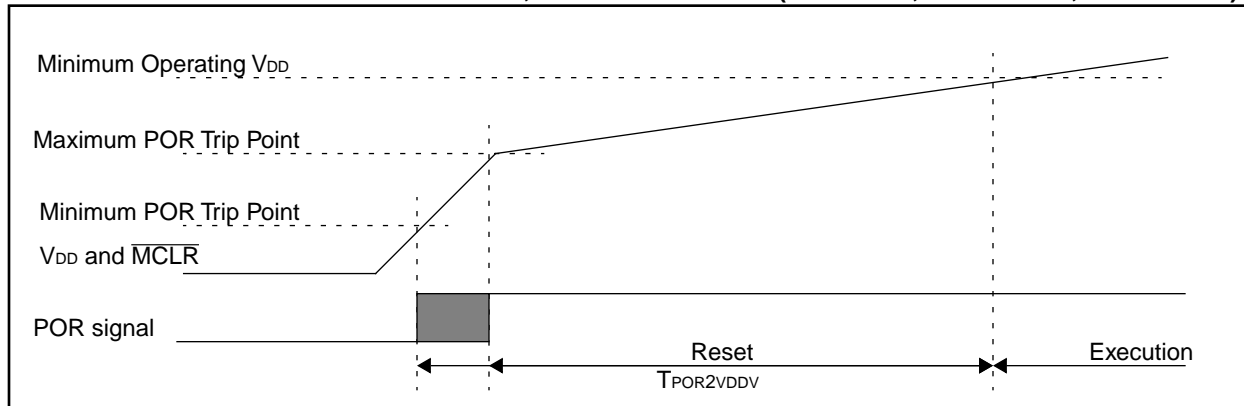


FIGURE 11: MAXIMUM POWER-UP TIME, $\overline{\text{MCLR}}$ NOT TIED TO V_{DD} (PIC16CXXX)

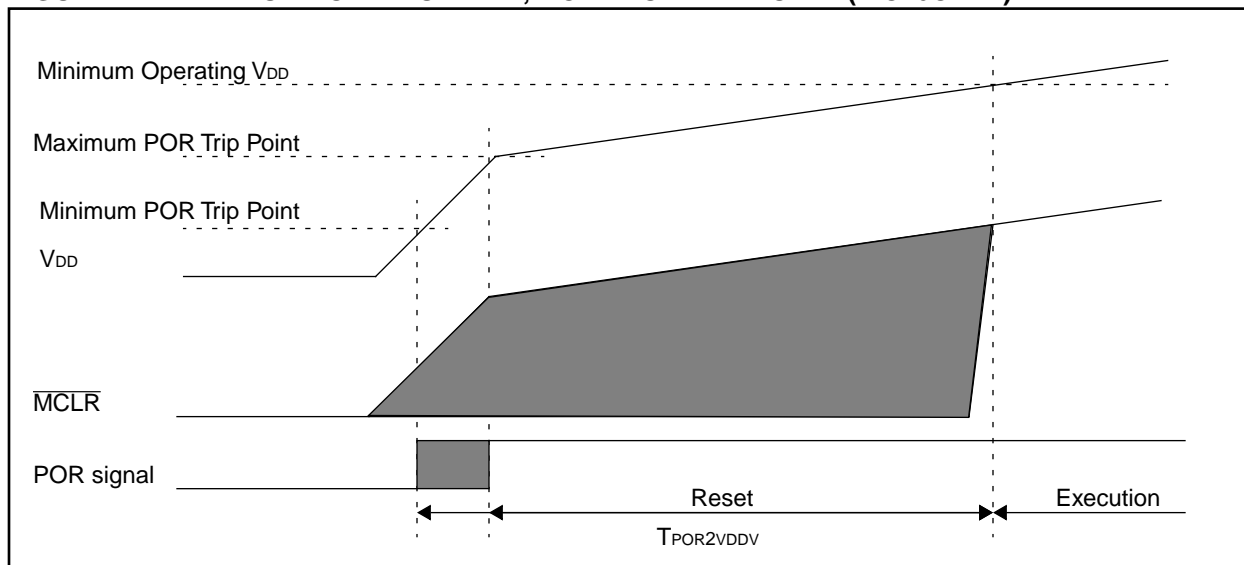
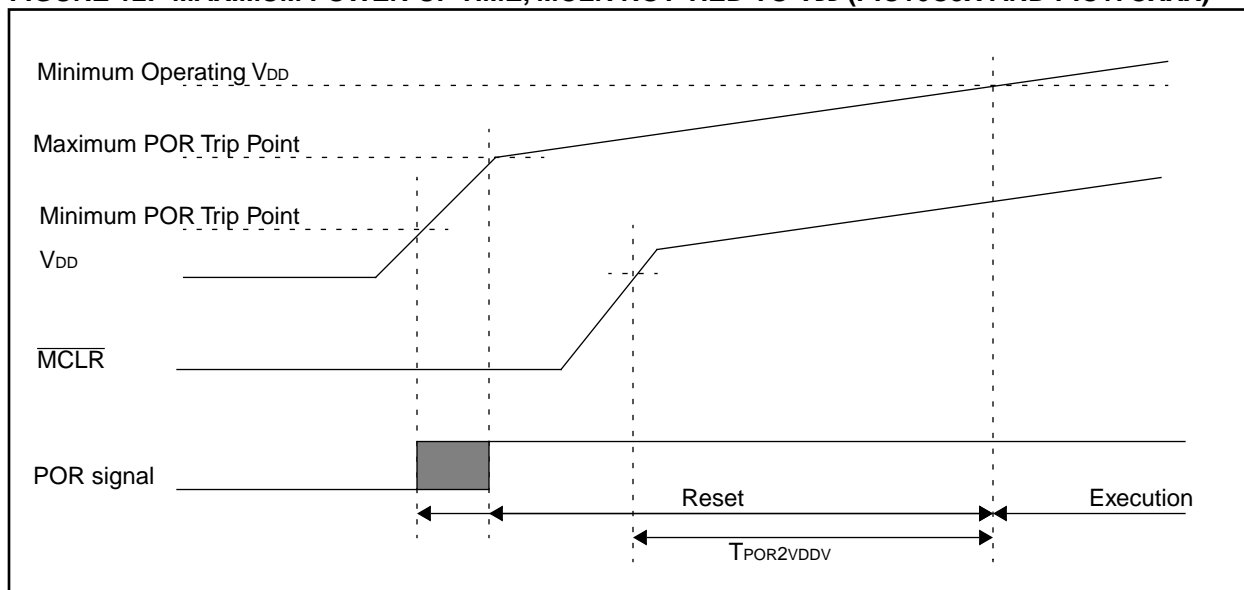


FIGURE 12: MAXIMUM POWER-UP TIME, $\overline{\text{MCLR}}$ NOT TIED TO V_{DD} (PIC16C5X AND PIC17CXXX)



Oscillator and Resonator Considerations

Oscillators and resonators from different manufacturers may have different characteristics. The recommended capacitor selection can be found in each device's data sheet. When we do the capacitor selection, during the oscillator/resonator characterization, we are currently using devices from one of several manufacturers. Generally we use oscillators from either ECS, CTS, FOX or Epson, and ceramic resonators from either Murata Erie or Panasonic. Other manufacturers may be used in the future, depending on availability and other factors.

Other manufacturers devices may have significantly different characteristics. To ensure proper oscillator operation, the circuit should be verified at the lowest temperature/highest V_{DD} (to ensure that the crystal is not overdriven), and with the highest temperature/lowest V_{DD} (to ensure the device still starts up) that the device will be subjected to while in the application. This ensures a stable start-up and frequency for this device, at the extreme conditions of the application.

For production purposes, the above testing should be done with many different samples of the components selected. This is so the part to part variation of the capacitors, resistors, crystals/resonators, and PICmicro devices are taken into account. All PICmicro final data sheets supply the characterization information on the transconductance of the oscillator (measurement of gain). This information can be used to check part to part variations of the PICmicro.

When selecting the crystal, the designer must ensure that it is a parallel cut type. Failure to use a parallel cut crystal may cause:

- Frequency operation out of the specified range of the crystal.
- Unreliable oscillator start-up.
- Device or crystal damage.

RAM and Special Function Register Initialization

After a successful Power-up Reset, the device will begin to execute the firmware program. To have expected operation, ALL RAM should be initialized by the program. This includes the Special Function Registers (SFR) and the general purpose data memory. The use (read) of an uninitialized RAM location will cause the program to do exactly what you told it, with the unexpected RAM value. It should not be expected that all devices will power-up with the same uninitialized device values.

There are many factors that contribute to how a RAM cell powers up, but the most common "gotcha" is between the Windowed and OTP device types. Many times a user forgets to cover the window after erasing the Windowed device. When the device is powering up, and the light is able to shine onto the device die, the transistor characteristics will shift. This can cause the

device RAM to have a different power-up value than a device where no light can shine onto the die (OTP or covered).

Note: RAM locations should be initialized before they are used. Use of an uninitialized location will cause proper device operation with the improper values. That is, it will do what you told it to do, not what you wanted it to do.

Valid Operating Voltage Levels

When the device is operating, the device voltage must be within the specified Min/Max limits. Operation of the device outside these limits may cause unexpected device operation.

One of the primary functional failure modes of a device is when the applied voltage is lower than the specified minimum requirement. This functional failure is called Brown-out. Brown-out causes the program memory not to be read correctly. For example, the program counter may be pointing to a `MOVE` instruction, but the device reads it as a `GOTO` instruction (with a random destination). This can have disastrous affects to the operation of the application. If brown-out conditions are possible, the application needs to be protected by using a brown-out circuit. A brown-out circuit works with the `MCLR` pin to put the device in `RESET` before the device's actual voltage violates the minimum limit.

Figure 13 shows a low cost external brown-out protection circuit. The voltage at which the circuit causes a reset is dependent upon the tolerances of the components. Figure 14 shows the use of a Dallas Semiconductor EconoReset. This device monitors the status of the power supply, and generates a reset when an out-of-tolerance condition is detected. Motorola also makes some 3-terminal devices to monitor the power supply, such as the MC34164, MC34064, MC33064. Their data sheets should be reviewed to ensure that the device is suitable for your application.

FIGURE 13: LOW COST EXTERNAL BROWN-OUT PROTECTION CIRCUIT

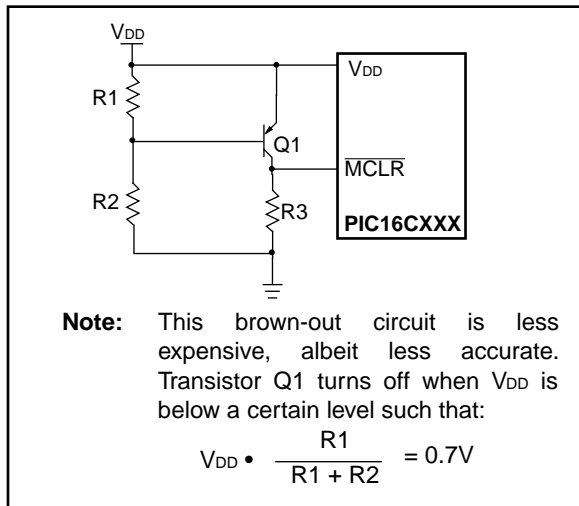
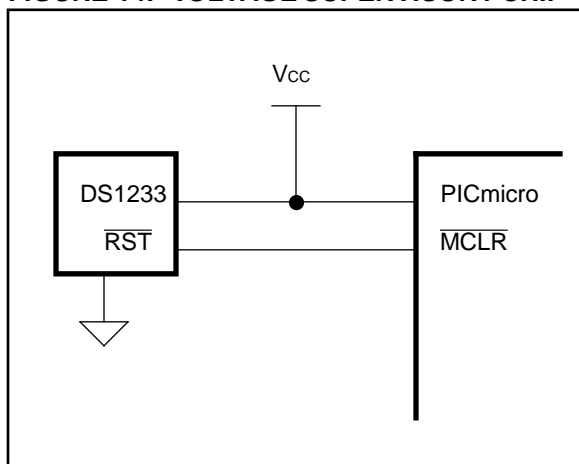


FIGURE 14: VOLTAGE SUPERVISORY CHIP



Brown-out and the WDT

The recommended solution for brown-out conditions is the use of a brown-out circuit. The brown-out circuit will keep the device in reset until a valid operating voltage is present. In some applications the additional cost of the external brown-out circuit, can be traded-off with system recovery from brown-out. Use of the Watchdog Timer (WDT) can enhance the probability of system recovery from a brown-out condition.

Note: If I/O drive conflicts can cause critical problems, this technique should not be used. This is due to the indeterminate time before a device reset could occur, which would reset all pins to inputs to eliminate any I/O conflict.

When using the WDT in brown-out conditions, care must be taken. Brown-outs may cause an unrecoverable condition, but with good design practice the probability of this can be significantly reduced.

During a brown-out, improper program execution can occur due to an EPROM read failure. This program execution can also corrupt data memory locations, which include the Special Function Registers (SFRs). Corrupting the control registers may cause hardware conflicts. For example, an input may become an output. Other conflicts are possible, but the situation will be application dependent.

As the device voltage gets lower, internal logic can become corrupted. This can include the Program Counter (PC) value, Stack Pointer and contents, State machines, Data Memory, etc.

When a valid voltage is returned, the device may be at an unexpected program location, possibly using corrupted values. In this situation, the device would not be expected to operate as intended and could get into a state that appears locked-up.

For the PIC17C42 in code protected microcontroller mode, once the Program Counter (PC) exceeds the 32K-word boundary, the device will become locked-up. The PC can exceed the 32K-word boundary from the execution of incorrect instructions (due to failure reading the EPROM) or by the PC becoming corrupted.

If the WDT is to be used to reset the device, care must be used in structuring the program. Optimally, only one CLRWDT instruction should be used. This minimizes the possibility of program execution returning to a loop which clears the WDT. This loop could then lock-up the device, since other control registers are corrupted and the device is not configured as expected. An example is; if the loop was waiting for an interrupt, but the bit that enables global interrupts was disabled, the device would no longer respond to the interrupts and would appear locked-up.

Example 1 shows a simple implementation of using the WDT reset for system recovery. The program loops, waiting for a WDT time-out (which clears the \overline{TO} bit). After the WDT reset, the \overline{TO} bit needs to be set (by executing a CLRWDT instruction). The program should then initialize the device. Then application code can start executing. There is a possibility of the \overline{TO} bit being corrupted by low voltage, and the device not being in a reset state when the software initializes the device.

The WDT example in Appendix B: uses a different method, independent of the \overline{TO} bit. This uses RAM locations which get loaded with a value. A WDT time-out (or other reset) needs to occur. The RAM locations are verified to contain the same values. Once the RAM is verified, it is cleared, and the device should be initialized. These RAM locations can be used by the application program.

EXAMPLE 1: USING WDT RESET

```

org    Reset_Address
      GOTO    TO_TEST      ;At any reset,
                          ;test the  $\overline{TO}$  bit

org    TO_TEST
      BTFSC   STATUS, TO   ;WDT Time-Out?
      GOTO    HERE         ;NO, Wait for  $\overline{TO}$ 
Time_Out:
      GOTO    HERE         ;YES, Good Reset
      CLRWDT
      :
      :                   ;Start here
      :                   ;Initialize Device
      :                   ;Application Code
  
```

False Power-down

In applications where power is removed from the device's supply lines, but voltage is still applied to an I/O pin, unexpected operation may occur. Power is able to be supplied to the device through this I/O pin. Since the device is still partially powered, the internal logic is never completely powered down. Figure 15 shows the general structure of an I/O pin. Figure 16 depicts the internal voltage level that is actually applied to some device logic, versus what is seen at the pin.

To guarantee a Power-on Reset (POR) rising edge, the device voltage (V_{DD}) must start from V_{SS} . When the device is inadvertently powered from an I/O pin, the voltage at the V_{DD} pin may appear to be near ground but may actually be higher in the device. With some of the internal logic powered, the characteristics of the device can be similar to a brown-out situation. Similar design practices to brown-out should be implemented.

A method for protecting the device from being powered from an I/O pin is shown in Figure 17.

FIGURE 15: TYPICAL ELECTRICAL STRUCTURE OF I/O PIN

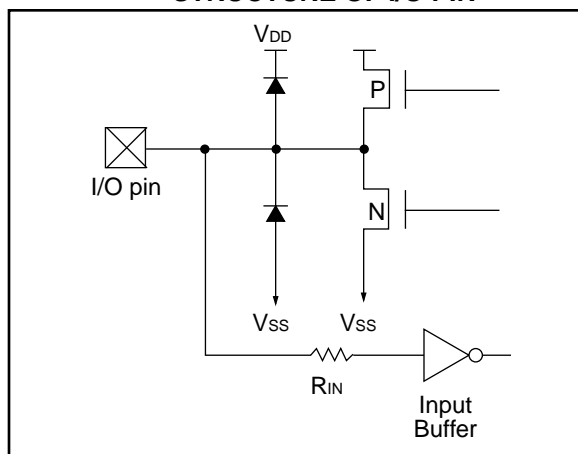


FIGURE 16: FALSE POWER-DOWN

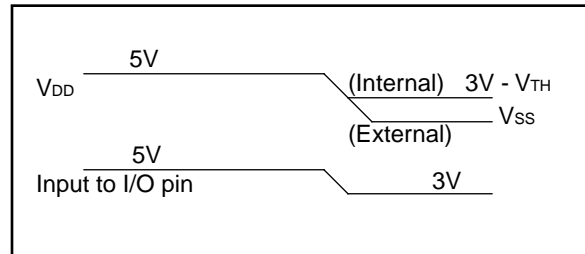
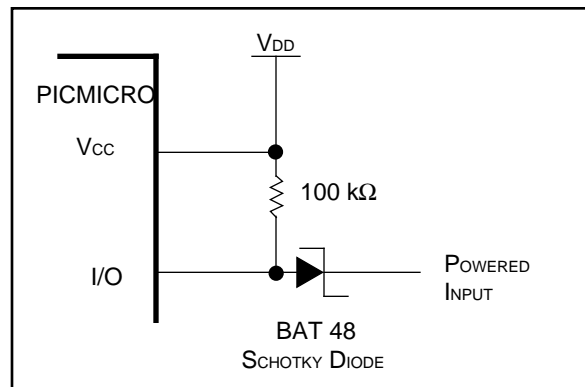


FIGURE 17: POWERED INPUT PROTECTION



In general, a brown-out detect circuit should cause the PICmicro to RESET (\overline{MCLR} forced low). This ensures that the internal logic is in a known state until a valid device voltage level is reached. The actual brown-out circuit depends on the voltage range of the device and the application requirements. A comprehensive brown-out circuit would use a dedicated device to monitor the voltage and force the \overline{MCLR} pin low when the voltage becomes lower than specified.

Another case of false power-down situations is when the power is removed from the system, but the capacitor loading keeps a non-zero voltage on the V_{DD} pin. When power is reapplied, the device never powered down so no Power-on-Reset will occur. A simple brown-out circuit should fix this.

TROUBLESHOOTING

There are several techniques that can be used to troubleshoot problems related to powering up. First it is important to try to locate the source of the problem. These sources could be:

- No oscillation on OSC1/OSC2 pins
- Improper/no Program Execution

In cases where there is no oscillation on the OSC1/OSC2 pins, some of the following should be tried:

- a) Verify that there are good connections/the components are good.
- b) Verify that the crystal/resonator manufacturer is one that has been tested, if not try other capacitor values.
- c) See if an external clock (from a function generator) causes device operation to begin.
- d) Verify that all components are well grounded.
- e) If a scope probe is connected to the oscillator output, it must be a low capacitance/high impedance probe. If it is not, the oscillator may stop.

In cases where program execution is not as expected:

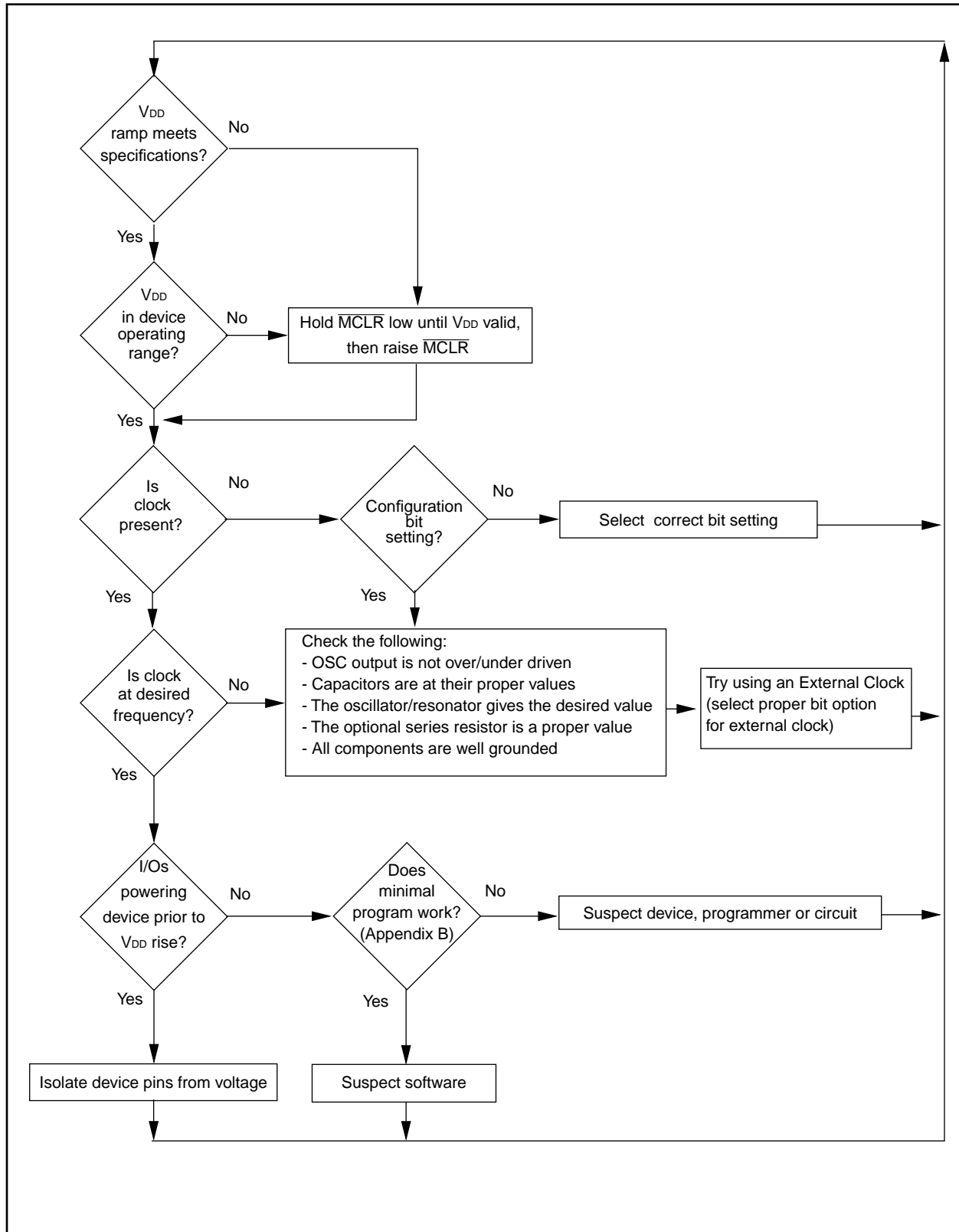
- a) Use a minimal program with external clock input.
- b) Tie $\overline{\text{MCLR}}$ to ground until solid power is applied to the device then release $\overline{\text{MCLR}}$ (bring high).
- c) Measure V_{DD} rise time to determine if an external reset circuit is needed, and, if so, what type of reset circuit should be used.
- d) Verify that the device program memory and configuration bits are programmed to their expected states.

The flowchart shown in Figure 18 can be used to troubleshoot power-up problems. This flowchart is only intended to be the first level diagnostic in trying to solve a power-up problem. Many other flowcharts can be used, depending on the characteristics of the problem and the set-up of the application.

CONCLUSION

Understanding the criteria for the powering up of a device will allow you to make better design choices. If device power-up problems are still encountered, many techniques can be used to solve the problem. Appendix B contains example code which can be used to verify that a device is operating (powered-up correctly). This eliminates the possibility of the program as the cause, and allows debug on the hardware.

FIGURE 18: TROUBLESHOOTING FLOWCHART



APPENDIX A: Q & As

Q. *When I use a windowed device (JW), my application works as expected. When I program an OTP device, it no longer works as expected. Why is this?*

A. The silicon is the same between the OTP and windowed devices. If the windowed device's window is not covered (with black tape), light shines onto the silicon. The light causes the potential levels of gates to shift. This in turn can cause RAM to be initialized to an unknown state, which could be different than in the OTP device. If RAM is not initialized by the program before it is used, these different power-up states of the RAM could be the cause of the problem. Ensure that all RAM is initialized in the device. This includes the SFRs.

Q. *My oscillator is not oscillating, what could be wrong?*

A. There are several possibilities, some which include:

1. The wrong oscillator bit setting is selected. The erased (default) state is RC oscillator mode.
2. The wrong capacitor values are installed. Refer to the most current data sheet for recommended values.
3. The characteristics of your manufacturers crystal are different than those that are characterized by Microchip. Generally our tests have been done with one of the following manufacturers' crystals/resonators: ECS, FOX, Murata Erie, or Panasonic.
4. The external connections to the device are wrong. Verify that all connections to the device are correct and that good signals / levels are being applied.
5. The cut of the crystal is a series type, as opposed to the specified parallel type.
6. No bypassing capacitors were used on the device. The noise on V_{DD} could be affecting the oscillator circuitry.

Q. *The device was powered-down and then powered back up, but the device does not operate. What could be wrong.*

A. Possibilities include:

1. If power was applied to an I/O pin when the device was "powered-down", the device would be powered through the I/O pin. The internal logic is not actually powered-down, and Power-on Reset (POR) will not occur.
2. When V_{DD} was powered-down, V_{DD} was not given enough time to settle to 0V.
3. The V_{DD} ramp rate is too slow.

Q. *My oscillator is oscillating, but the device is not working. What could be wrong?*

A. There are several possibilities, some which include:

1. Slow V_{DD} rise time, which was too slow to cause a Power-on Reset (POR). The rise time should not exceed the minimum device specification. For most devices this is 0.05 V/ms. Also the device must be at the minimum operating V_{DD} of the processor when reset is exited.
2. Ensure that the \overline{MCLR} pin is not low. This holds the device in RESET.
3. A brown-out has occurred, and has corrupted the internal state machines (including the WDT). An external brown-out circuit is recommended to hold the device in RESET during the brown-out condition.
4. The $CLRWDT$ instruction is not being used (often enough) when the WDT is enabled.

Q. *When I power-up the device, it does not operate and it gets hot.*

A. Your design is probably permitting fast high voltage signals (spike) onto one of the device pins. This sudden high voltage (and associated current) is in excess of the protection diode limit. The device must be powered-down (to V_{SS}) to release this condition. This condition may cause a functional failure or affect device reliability. All Microchip devices meet or exceed the Human Body Model (HBM) and Machine Model (MM) for ESD and latch-up.

Q. ***My oscillator is oscillating, but not at the expected frequency. What could be wrong.***

A. For many designers, working with oscillators and their related issues are a “black magic”, since the characteristics can vary widely between manufacturers. I suggest that you read all the application notes that we have available on oscillators. Some quick possibilities are:

1. The cut of the crystal is a series type, as opposed to the specified parallel type.
2. No bypassing capacitors were used on the device. The noise on V_{DD} could affect the oscillator circuitry.
3. The capacitor values used are causing the oscillator to operate in one of the harmonic frequencies.

Note: This is not an all inclusive list. You may need to investigate other design aspects.

Q. ***The device seems to never exit reset, or is continually resetting.***

A. The $CLRWDT$ instruction is not being used (often enough) when the WDT is enabled.

Q. ***The device was powered-down and back up again, but it does not reset. It just starts operating immediately.***

A. Possibilities include:

1. If power was applied to an I/O pin when the device was “powered-down”, the device would be powered through the I/O pin. The internal logic is not actually powered-down, and a Power-on Reset (POR) will not occur.
2. When V_{DD} was powered down, V_{DD} was not given enough time to settle to 0V.

Q. ***The oscillator is operating (I check it with a scope), yet when I look at other pins the program is not executing. Why?***

A. One possible reason is that when the oscilloscope probe is placed on the OSC2 pin, the additional capacitance is enough to cause oscillation to start. Removing the capacitive load of the probe causes the oscillation to stop.

Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).

APPENDIX B: TEST PROGRAMS

PIC16C5X BIT TOGGLE

MPASM 01.02.04 Intermediate C5X_B0T.ASM 12-20-1994 9:25:7

PAGE 1

LOC	OBJECT CODE	VALUE	LINE	SOURCE TEXT
			0001	LIST P = 16C54, F = INHX8M, n = 66
			0002	;
			0003	*****
			0004	;
			0005	; This program is a minimum program to toggle a single I/O port pin for the
			0006	; 16C5x family of devices. The only initialization is that of the data
			0007	; direction register (TRIS) of the I/O pin and the Toggling of the pin.
			0008	; The waveform will be 1 unit high and 3 units low.
			0009	;
			0010	Program: C5X_B0T.ASM
			0011	Revision Date: 12-20-94
			0012	;
			0013	*****
			0014	;
			0015	;
			0016	HARDWARE SETUP
			0017	None
			0018	;
			0019	;
			0020	INCLUDE <p16C5x.inc>
			0002	; P16C5X.INC Standard Header File, Version 0.1 Microchip Technology, Inc.
			0020	;
			0021	;
0FF9			0022	__FUSES (__CP_OFF & __WDT_OFF & __XT_OSC)
			0023	;
			0024	*****
			0025	***** Start program here.
			0026	*****
			0027	;
0000			0028	START ; POWER_ON Reset (Beginning of program)
0000 0063			0029	CLRF STATUS ; Do initialization (Bank 0)
0001 0C00			0030	MOVLW 0x00 ; Specify value for PortB output latch
0002 0026			0031	MOVWF PORTB ;
0003 0C00			0032	MOVLW 0x00 ; Specify which PortB pins are inputs / outputs
0004 0006			0033	TRIS PORTB ;
			0034	;
0005 0506			0035	lzz BSF PORTB, 0 ; B0 is High
0006 0406			0036	BCF PORTB, 0 ; B0 is Low
0007 0A05			0037	GOTO lzz ; Loop
			0038	;
			0039	;
			0040	;
			0041	;
			0042	; Reset address. Determine type of RESET
			0043	;
			0044	IFDEF __16C54
01FF			0045	RESET_V EQU 0x1FF
			0046	ENDIF
			0047	;
			0048	IFDEF __16C54A
			0049	RESET_V EQU 0x1FF
			0050	ENDIF
			0051	;
			0052	IFDEF __16C55
			0053	RESET_V EQU 0x1FF
			0054	ENDIF
			0055	;
			0056	IFDEF __16C56
			0057	RESET_V EQU 0x3FF
			0058	ENDIF
			0059	;
			0060	IFDEF __16C57
			0061	RESET_V EQU 0x7FF
			0062	ENDIF

```
0063 ;
0064     IFDEF      __16C58A
0065 RESET_V      EQU      0x7FF
0066     ENDIF
0067 ;
01FE      0068 PROG_MEM_END      EQU      RESET_V - 1
0069 ;
0070 ;
0071             org      PROG_MEM_END      ; End of Program Memory
01FE 0BFE      0072 ERR_LP_1      GOTO      ERR_LP_1      ; If you get here your program was lost
0073 ;
0074             org      RESET_V      ; RESET vector location
01FF 0A00      0075 R_VECTOR      GOTO      START      ;
0076 ;
0077 ;
0078     end
0079
0080
0081
```

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```
0000 : XXXXXXXX-----
0040 : -----
0180 : -----
01C0 : -----XX
```

All other memory blocks unused.

```
Errors   :    0
Warnings :    0
Messages :    0
```

PIC16CXXX BIT TOGGLE

MPASM 01.02.04 Intermediate CXX_B0T.ASM 12-20-1994 10:18:22

PAGE 1

LOC	OBJECT CODE	LINE	SOURCE TEXT
	VALUE		
		0001	LIST P = 16C74, F = INHX8M, n = 66
		0002	;
		0003	;*****
		0004	;
		0005	; This program is a minimum program to toggle a single I/O port pin for the
		0006	; 16Cxx family of devices. The only initialization is that of the data
		0007	; direction register (TRIS) of the I/O pin and the Toggling of the pin.
		0008	; The waveform will be 1 unit high and 3 units low.
		0009	;
		0010	Program: CXX_B0T.ASM
		0011	Revision Date: 12-20-94
		0012	;
		0013	;*****
		0014	;
		0015	;
		0016	HARDWARE SETUP
		0017	None
		0018	;
		0019	;
		0020	INCLUDE <p16Cxx.inc>
		0002	P16CXX.INC Standard Header File, Version 0.2 Microchip Technology, Inc.
		0020	;
		0021	;
3FF9		0022	__FUSES (_CP_OFF & _WDT_OFF & _XT_OSC & _PWRTE_ON)
		0023	;
		0024	;*****
		0025	Start program here.
		0026	;*****
		0027	;
0000		0028	START ; POWER_ON Reset (Beginning of program)
0000 0183		0029	CLRF STATUS ; Do initialization (Bank 0)
0001 3000		0030	MOVLW 0x00 ; Specify value for PortB output latch
0002 0086		0031	MOVWF PORTB ;
0003 1683		0032	BSF STATUS, RP0 ; Bank 1
0004 3000		0033	MOVLW 0x00 ; Specify which PortB pins are inputs / outputs
0005 0086		0034	MOVWF TRISB ;
0006 1283		0035	BCF STATUS, RP0 ; Bank 0
		0036	;
0007 1406		0037	lzz BSF PORTB, 0 ; B0 is High
0008 1006		0038	BCF PORTB, 0 ; B0 is Low
0009 2807		0039	GOTO lzz ; Loop
		0040	;
		0041	;
		0042	;
		0043	;
		0044	End of Program Memory
		0045	;
		0046	IFDEF __16C71
		0047	PROG_MEM_END EQU 0x3FF
		0048	ENDIF
		0049	;
		0050	IFDEF __16C71A
		0051	PROG_MEM_END EQU 0x3FF
		0052	ENDIF
		0053	;
		0054	IFDEF __16C73
		0055	PROG_MEM_END EQU 0xFFF
		0056	ENDIF
		0057	;
		0058	IFDEF __16C74
0FFF		0059	PROG_MEM_END EQU 0xFFF
		0060	ENDIF
		0061	;
		0062	IFDEF __16C61
		0063	PROG_MEM_END EQU 0x3FF
		0064	ENDIF
		0065	;
		0066	IFDEF __16C63
		0067	PROG_MEM_END EQU 0x7FF
		0068	ENDIF
		0069	;
		0070	IFDEF __16C64


```

0071 PROG_MEM_END    EQU    0x7FF
0072     ENDIF
0073 ;
0074     IFDEF    __16C65
0075 PROG_MEM_END    EQU    0xFFFF
0076     ENDIF
0077 ;
0078     IFDEF    __16C84
0079 PROG_MEM_END    EQU    0x3FF
0080     ENDIF
0081 ;
0082     IFDEF    __16C84A
0083 PROG_MEM_END    EQU    0x3FF
0084     ENDIF
0085 ;
0086 ;
0087     org    PROG_MEM_END    ; End of Program Memory
0088 ERR_LP_1    GOTO    ERR_LP_1    ; If you get here your program was lost
0089 ;
0090 ;
0091     end
0092
0093
0094
0095

```

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```

0000 : XXXXXXXXXXXX-----
0040 : -----
0F80 : -----
0FC0 : -----X

```

All other memory blocks unused.

```

Errors   :    0
Warnings :    0
Messages :    0

```

Note: Special Function Register data memory locations, in Bank 1, are specified by their true address in the file PIC16CXXX.INC. The use of the MPASM assembler will generate a warning message, when those labels are used with direct addressing. Warning messages can be turned off with an assembler option.

PIC17CXXX BIT TOGGLE

MPASM 01.02.04 Intermediate P17_B0T.ASM 12-19-1994 17:15:3

PAGE 1

LOC	OBJECT	CODE	LINE	SOURCE	TEXT
	VALUE		0001	LIST	P = 17C42, F = INHX32, n = 66
			0002		;
			0003		*****
			0004		;
			0005		; This program is a minimum program to toggle a single I/O port pin for the
			0006		; 17Cxx family of devices. The only initialization is that of the data
			0007		; direction register (DDR) of the I/O pin and the Toggling of the pin.
			0008		; The waveform will be 1 unit high and 1 unit low.
			0009		;
			0010	Program:	P17_B0T.ASM
			0011	Revision Date:	12-20-94
			0012		;
			0013		*****
			0014		;
			0015		;
			0016	HARDWARE SETUP	
			0017	None	
			0018		;
			0019		;
			0020	INCLUDE <p17Cxx.inc>	
			0002	P17CXX.INC	Standard Header File, Version 0.2 Microchip Technology, Inc.
			0020		
			0021		;
FFE2			0022	__FUSES (_MC_MODE & _WDT_NORM & _XT_OSC)
			0023		;
			0024		*****
			0025	*****	Start program here.
			0026		*****
			0027		;
0000			0028	START	; POWER_ON Reset (Beginning of program)
0000 2904			0029	CLRF	ALUSTA ; Do initialization
0001 290F			0030	CLRF	BSR ; Bank 0
0002 B000			0031	MOVLW	0x00 ; Specify value for PortB output latch
0003 0112			0032	MOVWF	PORTB ;
0004 B000			0033	MOVLW	0x00 ; Specify which PortB pins are inputs / outputs
0005 0111			0034	MOVWF	DDRB ;
			0035		;
0006 3812			0036	lzz	BTG PORTB, 0 ; Toggle level on B0
0007 C006			0037	GOTO	lzz ; Loop
			0038		;
			0039		;
			0040		
			0041		;
			0042		; End of Program Memory
			0043		;
			0044	IFDEF	__17C42
07FF			0045	PROG_MEM_END	EQU 0x7FF
			0046	ENDIF	
			0047		;
			0048		;
			0049	org	PROG_MEM_END ; End of Program Memory
07FF C7FF			0050	ERR_LP_1	GOTO ERR_LP_1 ; If you get here your program was lost
			0051		;
			0052		;
			0053	end	
			0054		
			0055		

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```

0000 : XXXXXXXX-----
0040 : -----
0780 : -----
07C0 : -----X

```

All other memory blocks unused.

Errors : 0
Warnings : 0
Messages : 0

WDT RESET WITH RAM VERIFY

MPASM 01.20 Released

BO_RAMT.ASM 6-30-1995 16:04:36

PAGE 1

LOC	OBJECT CODE	LINE	SOURCE TEXT
VALUE			
		00001	LIST P = 17C44, F = INHX32, n = 66
		00002	;
		00003	*****
		00004	;
		00005	; This program is a minimum program to recover from a brown-out condition thru
		00006	; the use of the WDT. The method is to load RAM locations with a known value
		00007	; and compare these locations after any RESET. If the RAM location matches the
		00008	; expected value then program flow can continue. The longer this RAM string
		00009	; is, the greater the probability that the RAM would NOT power up in that state.
		00010	;
		00011	;
		00012	; NOTE: This does not Guarantee device recovery, due to the random start-up
		00013	; point after brown-out. This point could be a loop with a CLRWD
		00014	; instruction. The recommended solution is to always use a brown-out
		00015	; circuit.
		00016	;
		00017	Program: B0_RAMT.ASM
		00018	Revision Date: 06-29-95
		00019	;
		00020	*****
		00021	;
		00022	;
		00023	; HARDWARE SETUP
		00024	; None
		00025	;
		00026	;
0001		00027	TRUE EQU 1
0000		00028	FALSE EQU 0
		00029	;
0001		00030	Debug EQU TRUE
		00031	#define __CONFIG __FUSES
		00032	;
		00033	INCLUDE <DEV_FAM.inc>
		00102	list
		00034	;
		00035	if (P16C5X)
		00036	INCLUDE <p16C5x.inc>
		00037	__CONFIG (_CP_OFF & _WDT_ON & _XT_OSC)
		00038	endif
		00039	;
		00040	if (P16CXX)
		00041	INCLUDE <p16Cxx.inc>
		00042	__CONFIG (_CP_OFF & _WDT_ON & _XT_OSC & _PWRT_ON)
		00043	endif
		00044	;
		00045	if (P17CXX)
		00046	INCLUDE <p17Cxx.inc>
		00001	LIST
		00002	; P17CXX.INC Standard Header File, Version 2.01 Microchip Technology, Inc.
		00298	LIST
FE00 FFE2		00047	__CONFIG (_MC_MODE & _WDT_NORM & _XT_OSC)
		00048	endif
		00049	;
		00050	if (P16C5X + P16CXX + P17CXX != 1)
		00051	MESSG "WARNING - USER DEFINED: One and only one device family can be selected"
		00052	endif
		00053	;
		00054	INCLUDE <BO_RAMT.inc>
		00029	list
		00055	INCLUDE <PMEM_END.inc>
		00116	list
		00056	;
		00057	;
		00058	*****
		00059	***** Start program here.
		00060	*****
		00061	;
0000		00062	org Reset_Address
		00063	; ; in the LIST directive
		00064	if (P16C5X)
		00065	org 0h ; Override the start of this code.
		00066	CLRF STATUS ; Force program memory to Page 0
		00067	CLRF FSR ; Force Data Memory to Bank 0
		00068	endif

```

00069 ;
00070     if ( P16CXX )
00071         CLRF   PCLATH      ; Force program memory to Page 0
00072         CLRF   STATUS      ; Force Data Memory to Bank 0
00073     endif
00074 ;
00075     if ( P17CXX )
00076         CLRF   PCLATH, F    ; Force program memory to Page 0
0000 2903     00077         CLRF   BSR, F      ; Force Peripheral / GP Data Memory to Bank 0
0001 290F     00078     endif
00079 ;
0002 C100     00080         GOTO    RAM_TEST      ; At any reset,
00081                                     ; test the RAM
00082 ;
00083 ; In RAM_TEST, program execution is held-off until a valid "warm" reset
00084 ; occurs. That is, the contents of some RAM locations retain the
00085 ; values that were written to them. The probability that the RAM would power-up
00086 ; in that state is dependent on the number of bytes of RAM used. The
00087 ; more RAM, the less the probability (probability = 1 / ( 2 ** 8(N+1) ).
00088 ;
00089 ;
0100         00090     org    MAIN                ; In Program Memory Page 0
0100         00091     RAM_TEST
0100 B0A5     00092         MOVLW   BYTE_0
0101 0520     00093         SUBWF   RAM0, F
0102 9204     00094         BTFSS   STATUS, Z      ; Result = 0?
0103 C110     00095         GOTO    LD_RAM        ; NO, Load Ram
00096 ;
0104 B00F     00097         MOVLW   BYTE_1      ; YES, Check next
0105 0521     00098         SUBWF   RAM1, F      ; location
0106 9204     00099         BTFSS   STATUS, Z      ; Result = 0?
0107 C110     00100         GOTO    LD_RAM        ; NO, Load RAM
00101 ;
00102 ;         :                ; YES, Do Again
00103 ;         :                ;
00104 ;
0108 B05A     00105         MOVLW   BYTE_n      ; YES, Check nth
0109 0522     00106         SUBWF   RAMn, F      ; location
010A 9204     00107         BTFSS   STATUS, Z      ; Result = 0?
010B C110     00108         GOTO    LD_RAM        ; NO, Load RAM
00109 ;
00110     if ( P16C5X || P16CXX )
00111         CLRF   RAM0      ; YES, Time-out
00112         CLRF   RAM1      ; occurred, clear
00113 ;         :                ; RAM locations
00114 ;         :
00115         CLRF   RAMn      ;
00116     endif
00117 ;
00118 ;
00119     if ( P17CXX )
010C 2920     00120         CLRF   RAM0, F      ; YES, Time-out
010D 2921     00121         CLRF   RAM1, F      ; occurred, clear
00122 ;         :                ; RAM locations
00123 ;         :
00124         CLRF   RAMn, F      ;
010E 2922     00125     endif
00126 ;
010F C117     00127         GOTO    Time_Out      ; Initialize Device
00128 ;
0110         00129     LD_RAM
0110 B0A5     00130         MOVLW   BYTE_0      ; Load RAM
0111 0120     00131         MOVWF   RAM0      ; locations to
0112 B00F     00132         MOVLW   BYTE_1      ; compare against
0113 0121     00133         MOVWF   RAM1      ;
00134 ;         :
0114 B05A     00135         MOVLW   BYTE_n      ;
0115 0122     00136         MOVWF   RAMn      ;
00137 ;
0116 C116     00138     HERE    GOTO    HERE      ; Wait for WDT TO
0117         00139     Time_Out ; YES, Good Reset
0117 0004     00140         CLRWDT      ; Start here
00141 ;         :                ; Initialize Device
00142 ;         :                ; Application Code
00143 ;
00144     if ( Debug )
00145         if ( P16C5X )
00146             CLRF   PORTB      ; PORTB output latch is cleared
00147             MOVLW   0x00      ;
00148             TRIS   PORTB      ; Port B is output
00149             BCF     PORTB, 0 ;
00150             BSF     PORTB, 0 ; Toggle pin B0
00151         endif

```

```

00152 ;
00153     if ( P16CXX )
00154         CLRF    PORTB        ; PORTB output latch is cleared
00155         BSF     STATUS, RP0   ; Bank 1
00156         CLRF    TRISB        ; Port B is output
00157         BCF     STATUS, RP0   ; Bank 0
00158         BCF     PORTB, 0      ;
00159         BSF     PORTB, 0      ; Toggle pin B0
00160     endif
00161 ;
00162     if ( P17CXX )
00163         CLRF    PORTB, F      ; PORTB output latch is cleared
00164         CLRF    DDRB, F      ; Port B is output
00165         BCF     PORTB, 0      ;
00166         BSF     PORTB, 0      ; Toggle pin B0
00167     endif
00168 endif
00169 ;
00170     GOTO    Time_Out         ; Return to start of Program
00171 ;
00172     org     PROG_MEM_END     ; End of Program Memory
00173 ERR_LP_1
00174     GOTO    ERR_LP_1         ; If you get here your program was lost
00175 ;
00176     if ( P16C5X )
00177         NOP
00178         ; This will cause the Program memory rollover
00179         ; for PIC16C5x devices
00179     endif
00180 ;
00181 ;
00182     end

```

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```

0000 : XXX-----
0040 : -----
0100 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX---
0140 : -----
1F80 : -----
1FC0 : -----X
FE00 : X-----
FE40 : -----

```

All other memory blocks unused.

```

Errors   :    0
Warnings :    0
Messages :    0

```