Section 10. Power-Saving Features

HIGHLIGHTS

This section of the manual contains the following major topics:

10.1 INTRODUCTION

All PIC24F devices offer a number of built-in strategies for reducing power consumption. These can be particularly useful in applications which are both power-constrained (such as battery operation), yet require periods of full-power operation for timing-sensitive routines (such as serial communications). This section discusses the four power-saving features implemented in hardware:

- Microcontroller Clock Manipulation
- Instruction-Based Power-Saving Modes (Sleep and Idle)
- Hardware-Based Doze Mode
- Selective Peripheral Control

10.2 MICROCONTROLLER CLOCK MANIPULATION

In general, reducing the microcontroller clock speed for any application will result in a power saving that is roughly proportional to the clock frequency reduction. PIC24F devices allow for a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can switch between low-power operation from an internal RC oscillator, or high-speed and high-precision operation from a crystal oscillator, by simply changing the NOSC Configuration bits. In fact, users can choose between up to four different oscillators at any time, allowing a maximum amount of flexibility in configuring application speed, frequency precision and power consumption.

The process of changing the system clock during operation, as well as restrictions on clock changes, are discussed in more detail in **Section 6. "Oscillator".**

10.3 INSTRUCTION-BASED POWER-SAVING MODES

PIC24F devices have two special Power-Saving modes that can be entered through the execution of a special `PWRSAV` instruction:

- Sleep Mode: The CPU, system clock source and any peripherals that operate on the system clock source are disabled. This is the lowest power mode for the device.
- Idle Mode: The CPU is disabled, but the system clock source continues to operate. Peripherals continue to operate, but can optionally be disabled.

The assembly syntax of the `PWRSAV` instruction is shown in Example 10-1.

Example 10-1: `PWRSAV` Assembly Syntax

```
PWRSAV   #SLEEP_MODE          ; Put the device into Sleep mode
PWRSAV   #IDLE_MODE           ; Put the device into Idle mode
```

Note: `SLEEP_MODE` and `IDLE_MODE` are constants defined in the assembler include file for the selected device.

The Power-Saving modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits one of these two operating modes, it is said to 'wake-up'. The characteristics of the Power-Saving modes are described in subsequent sections.

10.3.1 Sleep Mode

The characteristics of Sleep mode are as follows:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be at a minimum, provided that no I/O pin is sourcing current.
- The Fail-Safe Clock Monitor (FSCM) does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode if the WDT is enabled.
- If the on-chip voltage regulator is enabled, its BOR circuit remains operative during Sleep mode.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some peripherals may continue to operate in Sleep mode. These peripherals include I/O pins that detect a change in the input signal, or peripherals that use an external clock input. Any peripheral that operates from the system clock source will be disabled in Sleep mode.

The processor will exit, or 'wake-up', from Sleep on one of the following events:

- On any interrupt source that is individually enabled
- On any form of device Reset
- On a WDT time-out

10.3.1.1 CLOCK SELECTION ON WAKE-UP FROM SLEEP

The processor will restart the same clock source that was active when Sleep mode was entered.

10.3.1.2 DELAY ON WAKE-UP FROM SLEEP

The restart delay associated with waking up from Sleep mode for different oscillator modes is shown in Table 10-1.

Table 10-1: Delay Times for Exit from Sleep Mode

Clock Source		Sleep Exit Delay	Oscillator Delay	FSCM Delay	Notes
EC		TVREG	—	—	**1**
ECPLL		TVREG	TLOCK	TFSCM	**1, 3, 4**
XT, HS		TVREG	TOST	TFSCM	**1, 2, 4**
XTPLL		TVREG	TOST + TLOCK	TFSCM	**1, 2, 3, 4**
HSPLL		TVREG	TOST + TLOCK	TFSCM	**1, 2, 3, 4, 5**
SOSC	(Off during Sleep)	TVREG	TOST	TFSCM	**1, 2, 4**
	(On during Sleep)	TVREG	—	—	**1**
FRC, FRCDIV, LPRC		TVREG	—	—	**1**
FRCPLL		TVREG	TLOCK	—	**1, 3**

Note 1: TVREG = Start-up delay, only if on-chip regulator is enabled (10 μs nominal).
2: TOST = Oscillator Start-up Timer; a delay of 1024 oscillator periods before the oscillator clock is released to the system.
3: TLOCK = PLL lock time (20 ms nominal).
4: TFSCM = Fail-Safe Clock Monitor delay (100 μs nominal) if FSCM is enabled.
5: HSPLL mode exceeds PIC24F maximum operating frequency.

Note: Please refer to the **"Electrical Characteristics"** section of the product data sheet for maximum operating frequency, TVREG, TFSCM and TLOCK specification values.

10.3.1.3 WAKE-UP FROM SLEEP MODE WITH CRYSTAL OSCILLATOR OR PLL

If the system clock source is derived from a crystal oscillator and/or the PLL, the Oscillator Start-up Timer (OST) and/or PLL lock times must be applied before the system clock source is made available to the device. As an exception to this rule, no oscillator delays are necessary if the system clock source is the secondary oscillator and it was running while in Sleep mode. Note that in spite of the TVREG (if the regulator is enabled) and other delays applied, the crystal oscillator (and PLL) may not be up and running.

10.3.1.4 FSCM DELAY AND SLEEP MODE

When waking from Sleep, a nominal 100 μs delay (TFSCM) is applied (after TVREG, if the regulator is enabled) if both of the following are true:

- The oscillator was shut down while in Sleep mode
- The system clock is derived from a crystal oscillator source and/or the PLL

In most cases, the FSCM delay provides time for the OST to expire and the PLL to stabilize before device execution resumes. If the FSCM is enabled, it will begin to monitor the system clock source after the FSCM delay expires.

10.3.1.5 SLOW OSCILLATOR START-UP

The OST and PLL lock times may not have expired when the power-up delays have expired.

If the FSCM is enabled, then the device will detect this condition as a clock failure and a clock fail trap will occur. The device will switch to the FRC oscillator and the user can re-enable the crystal oscillator source in the clock failure Trap Service Routine.

If FSCM is NOT enabled, the device will not start executing code until the clock is stable. From the user's perspective, the device will appear to be in Sleep until the oscillator clock has started.

10.3.1.6 WAKE-UP FROM SLEEP ON INTERRUPT

User interrupt sources that are assigned to CPU priority level 0 cannot wake the CPU from Sleep mode because the interrupt source is effectively disabled. To use an interrupt as a wake-up source, the CPU priority level for the interrupt must be assigned to CPU priority level 1 or greater.

Any source of interrupt that is individually enabled, using its corresponding IE control bit in the IECx registers, can wake-up the processor from Sleep mode. When the device wakes from Sleep mode, one of two actions may occur:

- If the assigned priority for the interrupt is less than or equal to the current CPU priority, the device will wake-up and continue code execution from the instruction following the `PWRSAV` instruction that initiated Sleep mode.
- If the assigned priority level for the interrupt source is greater than the current CPU priority, the device will wake-up and the CPU exception process will begin. Code execution will continue from the first instruction of the ISR.

The SLEEP status bit (RCON<3>) is set upon wake-up.

10.3.1.7 WAKE-UP FROM SLEEP ON RESET

All sources of device Reset will wake the processor from Sleep mode. Any source of Reset (other than a POR) that wakes the processor will set the SLEEP status bit (RCON<3>) to indicate that the device was previously in Sleep mode.

On a Power-on Reset, the SLEEP bit is cleared.

10.3.1.8 WAKE-UP FROM SLEEP ON WATCHDOG TIME-OUT

If the Watchdog Timer (WDT) is enabled and expires while the device is in Sleep mode, the processor will wake-up. The WDTO and SLEEP status bits (RCON<4:3>) are both set to indicate that the device resumed operation due to the WDT expiration. Note that this event does not reset the device. Operation continues from the instruction following the `PWRSAV` instruction that initiated Sleep mode.

10.3.2 Idle Mode

When the device enters Idle mode, the following events occur:

- The CPU will stop executing instructions.
- The WDT is automatically cleared.
- The system clock source will remain active and peripheral modules, by default, will continue to operate normally from the system clock source. Peripherals can optionally be shut down in Idle mode using their 'Stop in Idle' control bit. (See peripheral descriptions for further details.)
- If the WDT or FSCM is enabled, the LPRC will also remain active.

The processor will wake from Idle mode on the following events:

- On any interrupt that is individually enabled.
- On any source of device Reset.
- On a WDT time-out.

Upon wake-up from Idle, the clock is reapplied to the CPU and instruction execution begins immediately, starting with the instruction following the `PWRSAV` instruction, or the first instruction in the ISR.

10.3.2.1 WAKE-UP FROM IDLE ON INTERRUPT

User interrupt sources that are assigned to CPU priority level 0 cannot wake the CPU from Idle mode because the interrupt source is effectively disabled. To use an interrupt as a wake-up source, the CPU priority level for the interrupt must be assigned to CPU priority level 1 or greater.

Any source of interrupt that is individually enabled, using the corresponding IE control bit in the IECx register and exceeds the current CPU priority level, will be able to wake-up the processor from Idle mode. When the device wakes from Idle mode, one of two options may occur:

- If the assigned priority for the interrupt is less than or equal to the current CPU priority, the device will wake-up and continue code execution from the instruction following the `PWRSAV` instruction that initiated Idle mode.
- If the assigned priority level for the interrupt source is greater than the current CPU priority, the device will wake-up and the CPU exception process will begin. Code execution will continue from the first instruction of the ISR.

The IDLE status bit (RCON<2>) is set upon wake-up.

10.3.2.2 WAKE-UP FROM IDLE ON RESET

Any Reset, other than a POR, will wake the CPU from Idle mode. On any device Reset, except a POR, the IDLE status bit is set (RCON<2>) to indicate that the device was previously in Idle mode. In a Power-on Reset, the IDLE bit is cleared.

10.3.2.3 WAKE-UP FROM IDLE ON WDT TIME-OUT

If the WDT is enabled, then the processor will wake from Idle mode on a WDT time-out and continue code execution with the instruction following the `PWRSAV` instruction that initiated Idle mode. Note that the WDT time-out does not reset the device in this case. The WDTO and IDLE status bits (RCON<4,2>) will both be set.

10.3.2.4 TIME DELAYS ON WAKE FROM IDLE MODE

Unlike a wake-up from Sleep mode, there are no time delays associated with wake-up from Idle mode. The system clock is running during Idle mode, therefore, no start-up times are required at wake-up.

10.3.3 Interrupts Coincident with Power Save Instructions

Any interrupt that coincides with the execution of a `PWRSAV` instruction will be held off until entry into Sleep or Idle mode has completed. The device will then wake-up from Sleep or Idle mode.

10.4 DOZE MODE

Changing clock speed and invoking one of the instruction-based Power-Saving modes are the preferred strategies for reducing power consumption. There may be circumstances, however, where this is not practical. For example, it may be necessary for an application to maintain uninterrupted synchronous communication, even while it is doing nothing else. Reducing system clock speed may introduce communication errors, while using a Power-Saving mode may stop communications completely.

Doze mode provides an alternative method to reduce power consumption while the device is still executing code. In this mode, the system clock continues to operate from the same source and at the same speed. Peripheral modules continue to be clocked at the same speed while the CPU clock speed is reduced. Synchronization between the two clock domains is maintained, allowing the peripherals to access the SFRs while the CPU executes code at a slower rate.

Doze mode is enabled by setting the DOZEN bit (CLKDIV<11>). The ratio between peripheral and core clock speed is determined by the DOZE2:DOZE0 bits (CLKDIV<14:12>). There are eight possible configurations, from 1:1 to 1:256, with 1:1 being the default.

10.4.1 Return from Doze on Interrupt

Doze mode can be configured to automatically return to full-speed CPU execution on an interrupt event. Enabling the automatic return to full-speed CPU operation on interrupts is enabled by setting the ROI bit (CLKDIV<15>). By default, the ROI bit is cleared, and interrupt events have no effect on Doze mode operation. When an interrupt event occurs in Doze mode and ROI is set, the DOZEN bit is automatically cleared.

The return from Doze mode feature allows clock-sensitive functions, such as synchronous communications, to continue without interruption while the CPU executes at a much lower speed, waiting for an event to invoke an interrupt routine and resume processing.

10.5 SELECTIVE PERIPHERAL POWER CONTROL

Sleep, Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked and thus consume some amount of power. There may be cases where the application needs what these modes do not provide: the ability to allocate limited power resources to the CPU while eliminating power consumption from the peripherals. PIC24F devices address this requirement by allowing peripheral modules to be selectively enabled or disabled, reducing or eliminating their power consumption.

10.5.1 Disabling Peripheral Modules

Most of the peripheral modules in the PIC24F family architecture can be selectively disabled, reducing or essentially eliminating their power consumption during all operating modes. Two different options are available to users, each with a slightly different effect.

10.5.1.1 MODULE ENABLE BIT (XXXEN)

Many peripheral modules have a Module Enable bit, generically named, “XXXEN”, usually located in bit position 15 of their control registers (or primary control registers for more complex modules). Here, “XXX” represents the mnemonic form for the module of the module name. For example, the enable bit for an SPI™ module is “SPIEN”, and so on. The bit is provided for all serial and parallel communications modules and the Real-Time Clock. Clearing this bit disables the module’s operation; however, it continues to receive clock signals and draw a minimal amount of current.

As with all earlier PICmicro® devices, timers continue to be under selective operation and are controlled by their own TON bit, also located in position 15. The A/D Converter also has a legacy enable bit, ADON, that has the same function as the XXXEN bits. I/O ports and features associated with them, such as input change notification and input capture, do not have their own Module Enable bits, since their operation is secondary to other modules.

Disabling modules not required for a particular application in this manner allows for the selective and dynamic adjusting power consumption under software control as the application is running.

10.5.1.2 PERIPHERAL MODULE DISABLE BIT (XXXPMD)

All peripheral modules (except for I/O ports) also have a second control bit that can disable their functionality. These bits, known as the Peripheral Module Disable (PMD) bits, are generically named “XXXPMD” (using “XXX” as the mnemonic version of the module’s name, as before). These bits are located in the PMDx Special Function Registers. In contrast to the Module Enable bits, the PMD bit must be set (= 1) to disable the module.

While the PMD and Module Enable bits both disable a peripheral’s functionality, the PMD bit completely shuts down the peripheral, effectively powering down all circuits and removing all clock sources. This has the additional effect of making any of the module’s control and buffer registers, mapped in the SFR space, unavailable for operations. In other words, when the PMD bit is used to disable a module, the peripheral ceases to exist until the PMD bit is cleared. This differs from using the Module Enable bit, which allows the peripheral to be reconfigured and buffer registers preloaded, even when the peripheral’s operations are disabled.

The PMD bit is most useful in highly power-sensitive applications, where even tiny savings in power consumption can determine the ability of an application to function. In these cases, the bits can be set before the main body of the application to remove those peripherals that will not be needed at all.

10.5.2 Selective Disabling of Modules in Idle Mode

To achieve additional power savings, peripheral modules can be selectively disabled whenever the device enters Idle mode. This is done with the Stop in Idle (SIDL) control bit, which is generally located in bit position 13 of the control register of most peripheral modules. The generic name format is “XXXSIDL” (using “XXX” as the mnemonic version of the module’s name, as before). The Stop in Idle feature allows further reduction of power consumption during Idle mode, enhancing power savings for extremely power-critical applications.

Almost all peripheral modules have a Stop in Idle bit, including modules that lack a module enable bit (e.g., input capture and output compare). The Real-Time Clock module is the exception, as it is assumed that an application involving a Real-Time Clock will need to keep the module running continuously.

Table 10-2: Power-Saving Features Register Map

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
RCON	TRAPR	IOPUWR	—	—	—	—	CM	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	xxxx(1)
CLKDIV	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—	—	—	0300
PMDx	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	XXMD	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: RCON register Reset values dependent on type of Reset.

10.6 DESIGN TIPS

Question 1: ***What should my software do before entering Sleep or Idle mode?***

Answer: Make sure that the sources intended to wake the device have their IE bits set. In addition, make sure that the particular source of interrupt has the ability to wake the device. Some sources do not function when the device is in Sleep mode.

If the device is to be placed in Idle mode, make sure that the 'Stop in Idle' control bit for each device peripheral is properly set. These control bits determine whether the peripheral will continue operation in Idle mode. See the individual peripheral sections of this manual for further details.

Question 2: ***How do I tell which peripheral woke the device from Sleep or Idle mode?***

Answer: You can poll the IF bits for each enabled interrupt source to determine the source of wake-up.

10.7 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC24F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Power-Saving Features are:

Title	Application Note #
Low-Power Design using PICmicro® Microcontrollers	AN606

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC24F family of devices.

10.8 REVISION HISTORY

Revision A (August 2006)

This is the initial released revision of this document.

NOTES:

