Section 9. Watchdog Timer (WDT)

HIGHLIGHTS

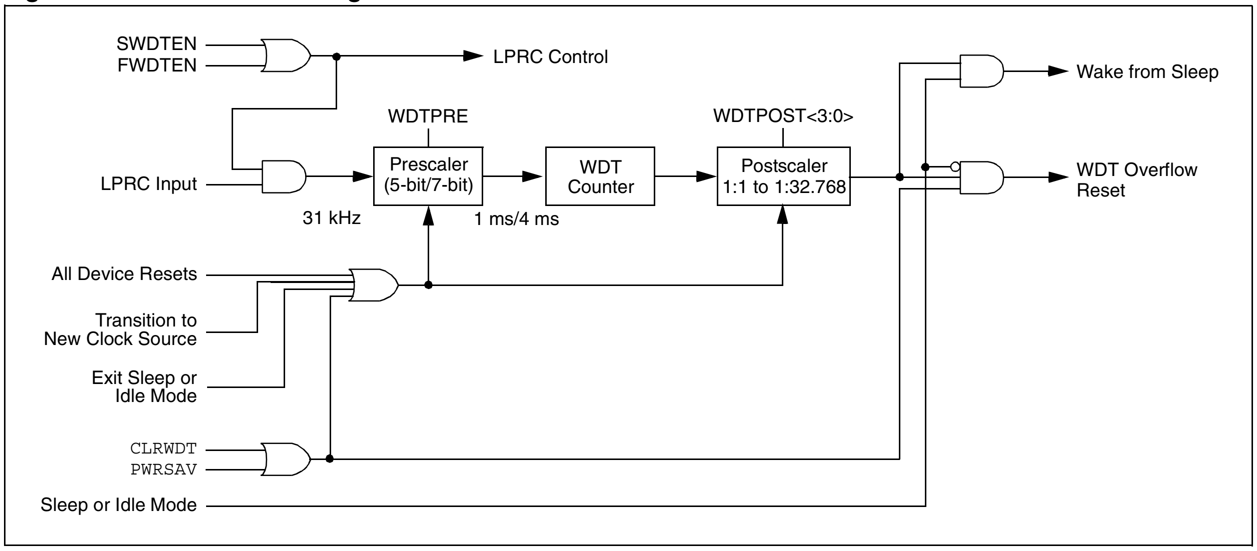
This section of the manual contains the following major topics:

9.1 INTRODUCTION

The primary function of the Watchdog Timer (WDT) is to reset the microcontroller, in the event of a software malfunction, by resetting the device if it has not been cleared in software. It can also be used to wake the device from Sleep or Idle mode. The WDT is a free-running timer which uses the low-power RC oscillator and requires no external components. Therefore, the WDT will continue to operate even if the system's primary clock source (e.g., the crystal oscillator) is stopped under normal operation (e.g., in Sleep mode).

A block diagram of the WDT is shown in Figure 9-1.

Figure 9-1: WDT Block Diagram



9.2 WDT OPERATION

When enabled, the WDT will increment until it overflows or "times out". A WDT time-out will force a device Reset, except during Sleep or Idle modes. To prevent a WDT Time-out Reset, the user must periodically clear the Watchdog Timer using the instructions, `PWRSAV` or `CLRWDT`. If the WDT times out during Sleep or Idle modes, the device will wake-up and continue code execution from where the `PWRSAV` instruction was executed.

In either case, the WDTO bit (RCON<4>) will be set to indicate that the device Reset or wake-up event was due to a WDT time-out. If the WDT wakes the CPU from Sleep or Idle mode, the SLEEP status bit (RCON<3>) or IDLE status bit (RCON<2>) will also be set to indicate that the device was previously in a Power-Saving mode.

9.2.1 Enabling and Disabling the WDT

The WDT is enabled or disabled by the FWDTEN (CW1<7>) Configuration bit. When the FWDTEN Configuration bit is set, the WDT is enabled. This is the default value for an erased device. Refer to the device data sheet for further details on the Flash Configuration Word registers.

9.2.2 Software Controlled WDT

If the FWDTEN Configuration bit is set, the WDT is always enabled. However, the WDT can be optionally controlled in the user software when the FWDTEN Configuration bit has been programmed to ‘0’.

The WDT is enabled in software by setting the SWDTEN control bit (RCON<5>). The SWDTEN control bit is cleared on any device Reset. The software WDT option allows the user to enable the WDT for critical code segments and disable the WDT during non-critical segments for maximum power savings.

9.2.3 WDT Window

The Watchdog Timer has an optional Windowed mode enabled by programming the WINDIS Configuration bit (CW1<6>) to ‘0’. In the Windowed mode, the `CLRWDT` instruction must occur within the last 1/4 of the WDT period. Any `CLRWDT` instruction that occurs within the first 3/4 of the WDT period will cause a WDT Reset, similar to a WDT time-out.

Note: The WDT must be enabled (FWDTEN = 1) to use WDT Windowed mode.

9.2.4 WDT Prescaler and Timer Period

The WDT clock source is the LPRC oscillator, which has a nominal frequency of 31 kHz. This feeds a prescaler that can be configured for either 5-bit (divide-by-32) or 7-bit (divide-by-128) operation. The prescaler is set by the FWPSA Configuration bit (CW1<4>). With a 31 kHz input, the prescaler yields a nominal WDT time-out period (T_{WDT}) of 1 ms when WDTPRE is clear, or 4 ms when WDTPRE is set.

A variable postscaler divides down the WDT prescaler output and allows for a wide range of time-out periods. The postscaler is controlled by the WDTPPOST<3:0> Configuration bits (CW1<3:0>), which allows the selection of a total of 16 settings, from 1:1 to 1:32.768. The WDTPOST bits are configured initially during device programming. Using the prescaler and postscaler, time-out periods ranging from 1 ms to 131 seconds (nominal) can be achieved.

The WDT time-out value can be calculated as shown in Equation 9-1. A complete list of prescaler values and the associated WDT time-out periods is shown in Table 9-1.

Equation 9-1: WDT Time-out Period

$$\text{WDT Period (ms)} = \text{Prescaler Factor x Postscaler Factor}$$

where:

$$\text{Prescaler Factor} = \begin{cases} 1 \text{ for WDTPRE is ‘0’} \\ 4 \text{ for WDTPRE is ‘1’} \end{cases}$$

$$\text{Postscaler Factor} = 1/\text{Postscaler Ratio}$$

Table 9-1: WDT Configuration and Time-out Period

Postscaler Setting (WDTPS3:WDTPS0)	Postscaler Ratio (1/Postscaler Factor)	Time-out Period	
		5-bit Prescaler (FWPSA = `0`)	7-bit Prescaler (FWPSA = `1`)
`0000`	1:1	1 ms	4 ms
`0001`	1:2	2 ms	8 ms
`0010`	1:4	4 ms	16 ms
`0011`	1:8	8 ms	32 ms
`0100`	1:16	16 ms	64 ms
`0101`	1:32	32 ms	128 ms
`0110`	1:64	64 ms	256 ms
`0111`	1:128	128 ms	512 ms
`1000`	1:256	256 ms	1.024s
`1001`	1:512	512 ms	2.048s
`1010`	1:1024	1.024s	4.096s
`1011`	1:2048	2.048s	8.192s
`1100`	1:4096	4.096s	16.384s
`1101`	1:8192	8.192s	32.768s
`1110`	1:16384	16.384s	65.536s
`1111`	1:32768	32.768s	131.072s

Note: The WDT time-out period is directly related to the frequency of the LPRC oscillator, which in turn, may vary as a function of device operating voltage and temperature. Please refer to the appropriate PIC24F device data sheet for LPRC clock frequency specifications.

9.2.5 Resetting the Watchdog Timer

The WDT counter and associated prescalers and postscalers are reset:

- On any device Reset
- When a `PWRSAV` instruction is executed (i.e., Sleep or Idle mode is entered)
- When the WDT is enabled in software
- On the completion of a clock switch, whether invoked by software (i.e., setting the OSWEN bit after changing the NOSC bits) or by hardware (i.e., Fail-Safe Clock Monitor)
- By a `CLRWDT` instruction during normal execution or during the last 25% of the WDT time-out period if WINDIS is ‘`0`’

9.2.6 Operation of WDT in Sleep and Idle Modes

If the WDT is enabled, it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake the device and code execution will continue from where the instruction was executed.

The WDT is useful for low-power system designs, because it can be used to periodically wake the device from Sleep mode to check system status and provide action if necessary. Note that the SWDTEN bit is very useful in this respect. If the WDT is disabled during normal operation (FWDTEN = `0`), then the SWDTEN bit (RCON<5>) can be used to turn on the WDT just before entering Sleep mode.

9.3 REGISTER MAPS

A summary of the Special Function Registers associated with the PIC24F WDT module is provided in Table 9-2.

Table 9-2: Special Function Register Map Associated with the Watchdog Timer

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)

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.

9.4 DESIGN TIPS

Question 1: ***Why does the device reset, even though I have inserted a `CLRWDT` instruction in my main software loop?***

Answer: Make sure that the software loop that contains the instruction meets the minimum specification of the WDT (not the typical value). Also, make sure that interrupt processing time has been accounted for.

Question 2: ***What are good techniques for using the WDT in my application?***

Answer: There are many techniques for using the WDT to prevent applications from locking up or running away. When they are carefully analyzed, most of them depend on three basic principles:

1. Use one, and only one, `CLRWDT` instruction in your application. Placing multiple occurrences throughout the application makes it more difficult to troubleshoot time-out issues.
2. Place the `CLRWDT` instruction inside the main body of the application and not inside a subroutine or an Interrupt Service Routine (ISR). If the instruction is located inside a frequently called routine, there is a good chance that the WDT will constantly be reset and never time-out.
3. Once the application is compiled and sized, place unconditional branch instructions (such as, "`GOTO .`") throughout the unused area of program memory. If something should happen that makes the code "run away" by branching into unused code space, the `GOTO` instructions can bring the microcontroller back to your code, where the WDT can help bring the application back under control.

9.5 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 Watchdog Timer (WDT) module 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.

9.6 REVISION HISTORY

Revision A (May 2006)

This is the initial released revision of this document.