Section 9. Watchdog Timer and Power-up Timer

HIGHLIGHTS

This section of the manual contains the following topics:

9.1 Introduction 9-2
9.2 Watchdog Timer and Power-up Timer Control Registers 9-3
9.3 Operation 9-9
9.4 Interrupt and Reset Generation 9-13
9.5 I/O Pins 9-14
9.6 Operation in DEBUG and Power-Saving Modes 9-14
9.7 Effects of Various Resets 9-15
9.8 Design Tips 9-15
9.9 Related Application Notes 9-16
9.10 Revision History 9-17

9.1 INTRODUCTION

The PIC32MX Watchdog Timer (WDT) and Power-up Timer (PWRT) modules are described in this section. Refer to Figure 9-1 for a block diagram of the WDT and PWRT.

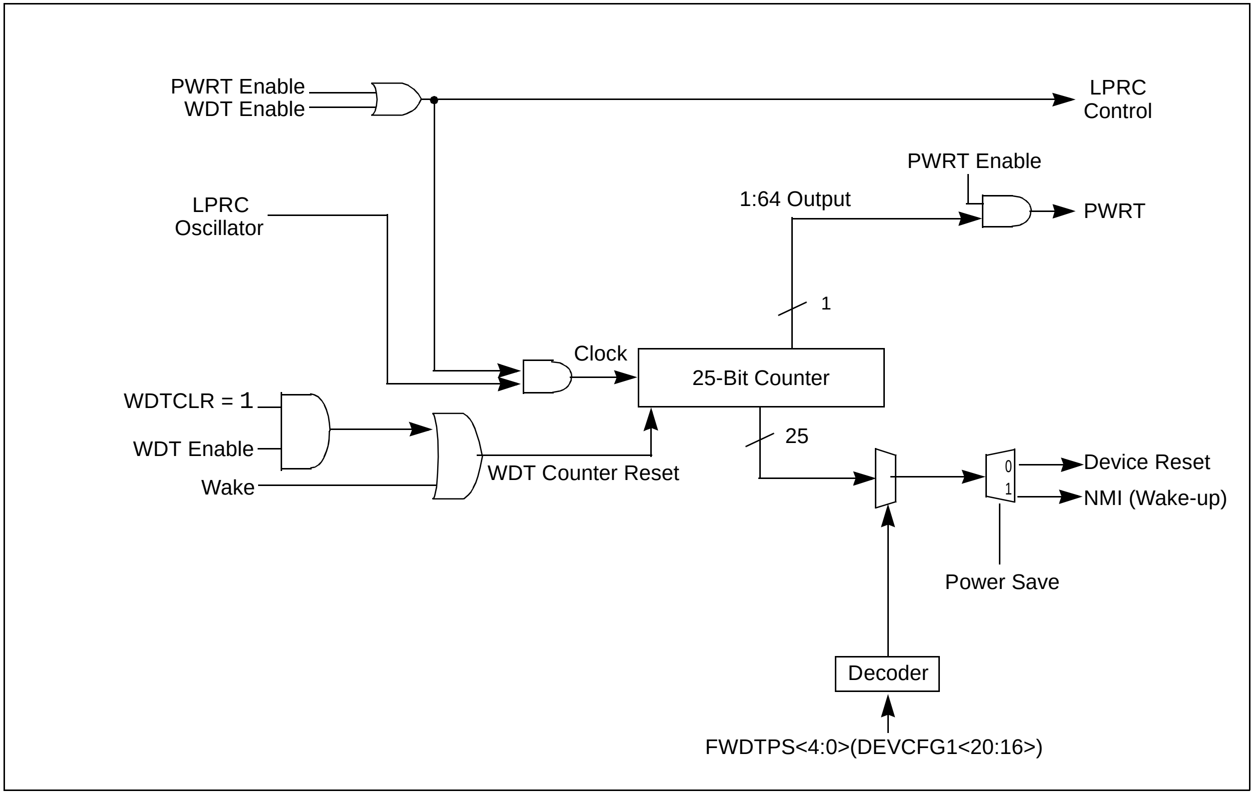
The WDT, when enabled, operates from the internal Low-Power RC (LPRC) oscillator clock source. The WDT can be used to detect system software malfunctions by resetting the device if the WDT is not cleared periodically in software. Various WDT time-out periods can be selected using the WDT postscaler. The WDT can also be used to wake the device from SLEEP or IDLE mode.

The PWRT, when active, holds the device in Reset for a 64 millisecond period after the normal Power-on Reset (POR) start-up period is complete. This allows additional time for the Primary Oscillator (POSC) clock source and the power supply to stabilize. Like the WDT, the PWRT also uses the LPRC as its clock source. Refer to Figure 9-1 for details.

Following are some of the key features of the WDT module:

- Configuration or software controlled
- User configurable time-out period
- Can wake the device from SLEEP or IDLE

Figure 9-1: Watchdog and Power-up Timer Block Diagram



9.2 WATCHDOG TIMER AND POWER-UP TIMER CONTROL REGISTERS

The WDT and PWRT modules consist of the following Special Function Registers (SFRs):

- WDTCON: Watchdog Timer Control Register

  WDTCONCLR, WDTCONSET, WDTCONINV: Atomic Bit Manipulation Registers for WDTCON
- RCON: Resets Control and Status Register

  RCONCLR, RCONSET, RCONINV: Atomic Bit Manipulation Registers for RCON
- DEVCFG1: Device Configuration Register

The following table provides a brief summary of WDT and PWRT-related registers. Corresponding registers appear after the summary, followed by a detailed description of each registers.

Table 9-1: Watchdog Timer and Power-up Timer SFR Summary

Name		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
WDTCON		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	15:8	ON	—	—	—	—	—	—	—
	7:0	—	WDTPS<4:0>					—	WDTCLR
WDTCONCLR	31:0	Write clears selected bits in WDTCON, read yields an undefined value							
WDTCONSET	31:0	Write sets selected bits in WDTCON, read yields an undefined value							
WDTCONINV	31:0	Write inverts selected bits in WDTCON, read yields an undefined value							
RCON		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	15:8	TRAPR	—	—	—	—	—	CM	VREGS
	7:0	EXTR	SWR	—	WDTO	SLEEP	IDLE	BOR	POR
RCONCLR	31:0	Write clears selected bits in RCON, read yields an undefined value							
RCONSET	31:0	Write sets selected bits in RCON, read yields an undefined value							
RCONINV	31:0	Write inverts selected bits in RCON, read yields an undefined value							
DEVCFG1	31:24	—	—	—	—	—	—	—	—
	23:16	FWDTEN	—	—	FWDTPS<4:0>				
	15:8	FCKSM<1:0>		FPBDIV<1:0>		—	OSCIOFNC	POSCMD<1:0>	
	7:0	IESO	—	FSOSCEN	—	—	FNOSC<2:0>		

Register 9-1: WDTCON: Watchdog Timer Control Register

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	r-x	r-x	r-x	r-x	R-1	R-1	R-0
ON	—	—	—	—	—	—	—
bit 15							bit 8

r-x	R-x	R-x	R-x	R-x	R-x	r-0	R/W-0
—	WDTPS<4:0>					—	WDTCLR
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: ('0', '1', x = Unknown)		

bit 31-16 **Reserved:** Write '0'; ignore read

bit 15 **ON:** Watchdog Timer Enable bit

1 = Enables the WDT if it is not enabled by the device configuration
0 = Disable the WDT if it was enabled in software

Note 1: A read of this bit will result in a '1' if the WDT is enabled by the device configuration or by software.

2: When using 1:1 PBCLK divisor, the user's software should not read/write the peripheral's SFRs in the SYSCLK cycle immediately following the instruction that clears the module's ON bit.

bit 14-7 **Reserved:** Write '0'; ignore read

bit 6-2 **WDTPS<4:0>:** Watchdog Timer Postscaler Value.

On Reset these bits are set to the values of the FWTDPS[4:0] of Configuration bits

bit 1 **reserved:** Write '0'; ignore read

bit 0 **WDTCLR:** Watchdog Timer Reset bit
1 = Writing a '1' will clear the WDT.
0 = Software cannot force this bit to a '0'

Register 9-2: WDTCONCLR: Comparator Control Clear Register

Write clears selected bits in WDTCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clear selected bits in WDTCON**

A write of ‘`1`’ in one or more bit positions clears the corresponding bit(s) in WDTCON register and does not affect unimplemented or read-only bits. A write of ‘`0`’ will not affect the register.

Example: `WDTCONCLR = 0x00008001` clears bits 15 and 0 in WDTCON register.

Register 9-3: WDTCONSET: Comparator Control Set Register

Write sets selected bits in WDTCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Set selected bits in WDTCON**

A write of ‘`1`’ in one or more bit positions sets the corresponding bit(s) in WDTCON register and does not affect unimplemented or read-only bits. A write of ‘`0`’ will not affect the register.

Example: `WDTCONSET = 0x00008001` sets bits 15 and 0 in WDTCON register.

Register 9-4: WDTCONINV: Comparator Control Invert Register

Write inverts selected bits in WDTCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in WDTCON**

A write of ‘`1`’ in one or more bit positions inverts the corresponding bit(s) in WDTCON register and does not affect unimplemented or read-only bits. A write of ‘`0`’ will not affect the register.

Example: `WDTCONINV = 0x00008001` inverts bits 15 and 0 in WDTCON register.

Register 9-5: RCON: Resets Control Register

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	r-x	r-x	r-x	r-x	R-0	R/W-0	R/W-0
TRAPR	—	—	—	—	—	CM	VREGS
bit 15							bit 8

R/W-0	R/W-0	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EXTR	SWR	—	WDTO	SLEEP	IDLE	BOR	POR
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 4 **WDTO:** Watchdog Time-out bit

1 = A WDT time out has occurred since either the device was powered up or the WDTO bit was last cleared by software

0 = A WDT time out has not occurred since either the WDTO bit was cleared by software or the device was reset

bit 3 **SLEEP:** SLEEP Event bit

1 = The device was in SLEEP since either the device was powered up or the SLEEP bit was last cleared by software

0 = The device was not in SLEEP since either the SLEEP bit was cleared by software or the device was reset

bit 2 **IDLE:** IDLE Event bit

1 = The device has been in IDLE mode since either the device was powered up or the IDLE bit was last cleared by software

0 = The device has not been in IDLE mode since either the IDLE bit was cleared by software or the device was reset

Register 9-6: RCONCLR: Comparator Control Clear Register

Write clears selected bits in RCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clear selected bits in RCON**

A write of ‘`1`’ in one or more bit positions clears the corresponding bit(s) in RCON register and does not affect unimplemented or read-only bits. A write of ‘`0`’ will not affect the register.

Example: `RCONCLR = 0x00008001` clears bits 15 and 0 in RCON register.

Register 9-7: RCONSET: Comparator Control Set Register

Write sets selected bits in RCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Set selected bits in RCON**

A write of ‘`1`’ in one or more bit positions sets the corresponding bit(s) in RCON register and does not affect unimplemented or read-only bits. A write of ‘`0`’ will not affect the register.

Example: `RCONSET = 0x00008001` sets bits 15 and 0 in RCON register.

Register 9-8: RCONINV: Comparator Control Invert Register

Write inverts selected bits in RCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in RCON**

A write of ‘`1`’ in one or more bit positions inverts the corresponding bit(s) in RCON register and does not affect unimplemented or read-only bits. A write of ‘`0`’ will not affect the register.

Example: `RCONINV = 0x00008001` inverts bits 15 and 0 in RCON register.

Register 9-9: DEVCFG1 Device Configuration Register

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

R/P-1	r-1	r-x	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FWDTEN	—	—	WDTPS<4:0>				
bit 23							bit 16

R/P-1	R/P-1	R/P-1	R/P-1	r-x	R/P-1	R/P-1	R/P-1
FCKSM<1:0>		FPBDIV<1:0>		—	OSCIOFNC	POSCMD<1:0>	
bit 15							bit 8

R/P-1	r-x	R/P-1	r-x	r-x	R/P-1	R/P-1	R/P-1
IESO	—	FSOSCEN	—	—	FNOSC<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 23 **FWDTEN:** Watchdog Timer Enable bit

1 = WDT is enabled and cannot be disabled by software
0 = WDT is not enabled and can be enabled in software

bit 22 **Reserved:** Write ‘1’; ignore read

bit 20-16 **FWDTPS<4:0>:** Watchdog Timer Postscale Select bits. These bits define the WDT period.

10100 = 1:1048576
10011 = 1:524288
10010 = 1:262144
10001 = 1:131072
10000 = 1:65536
01111 = 1:32768
01110 = 1:16384
01101 = 1:8192
01100 = 1:4096
01011 = 1:2048
01010 = 1:1024
01001 = 1:512
01000 = 1:256
00111 = 1:128
00110 = 1:64
00101 = 1:32
00100 = 1:16
00011 = 1:8
00010 = 1:4
00001 = 1:2
00000 = 1:1
All other combinations not shown result in operation = 10100

9.3 OPERATION

If 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 mode. To prevent a WDT time-out Reset, the user must periodically clear the WDT by setting the WDTCLR (WDTCON<0>) bit. To prevent a device Reset the WDTCLR bit must be periodically set within the selected WDT period.

Table 9-2: Results of a WDT Time-out Event for Available Modes of Device Operation

Device Mode	Device Reset Generated	Non-Maskable Interrupt Generated	WDTO[1] Bit Set	SLEEP[1] Bit Set	IDLE[1] Bit Set	Device Registers Reset
Awake	Yes	No	Yes	No	No	Yes
SLEEP	No	Yes	Yes	Yes	No	No
IDLE	No	Yes	Yes	No	Yes	No

Note 1: Status bits are in the RCON register.

.

Note: The LPRC oscillator is automatically enabled whenever the WDT is enabled.

9.3.1 Enabling and Disabling the WDT

The WDT is either enabled or disabled by the device configuration, or controlled via software by writing to the WDTCON register.

9.3.2 Device Configuration Controlled WDT

If the FWDTEN device Configuration bit (DEVCFG1<23>) is set, the WDT is always enabled. The WDT ON control bit (WDTCON<15>) will reflect this by reading a ‘1’. In this mode, the ON bit cannot be cleared in software or any form of Reset. To disable the WDT in this mode, the configuration must be rewritten to the device.

Note: The default state for the WDT on an unprogrammed device is WDT enabled.

9.3.3 Software Controlled WDT

If the FWDTEN device Configuration bit (DEVCFG1<23>) has a value of ‘0’, the WDT can be enabled and disabled by software. In this mode, the ON bit (WDTCON<15>) reflects the status of the WDT under software control. A value of ‘1’ indicates the WDT is enabled and a ‘0’ indicates it is disabled.

The WDT is enabled in software by setting WDT ON control bit. WDT ON control bit is cleared on any device Reset. The bit is not cleared on a wake-up from SLEEP mode or an exit from IDLE mode.

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. The WDT ON control bit can also be used to disable the WDT while the device is awake to eliminate the need for WDT servicing, and then re-enable it before the device is put into IDLE or SLEEP mode to wake-up the device at a later time.

Example 9-1: Sample WDT Initialization and Servicing

```
                       // This code fragment assumes the WDT was not enabled by
                       // the device configuration
                       // The Postscaler value must be set with the device configuration

WDTCONSET = 0x8000;    // Turn on the WDT

main()
{
    WDTCONSET = 0x01;  // Service the WDT

    ... User code goes here ...


}
```

9.3.4 Resetting the WDT Timer

The WDT is cleared by any of the following:

- On any device Reset.
- By a `WDTCONSET = 0x01`, or equivalent instruction, during normal execution. Refer to Example 9-2.
- Exiting from IDLE or SLEEP mode, due to an interrupt.

> **Note:** The WDT timer is not cleared when the device enters a Power-Saving mode. The WDT should be serviced prior to entering a Power-Saving mode.

Example 9-2: Determining Power-Saving Mode After a Reset

```
OSCCONSET = 0x10;                    // set Power-Saving mode to SLEEP
                                     // OSCCONCLR = 0x10;
                                     // set Power-Saving mode to IDLE

WDTCONSET = 0x8000;                  // Enable WDT

while (1)
{
... user code ...

WDTCONSET = 0x01;                    // service the WDT
asm volatile( “wait” );              // put device is selected Power-Saving mode

                                     // code execution will resume here after wake

... user code ...
}


                                     // The following code fragment is at the top of the
                                     // device start-up code

if ( RCON & 0x18 )
{
                                     // The WDT caused a wake-from-SLEEP
asm volatile( “eret” );              // return from interrupt
}

if ( RCON & 0x14 )
{
                                     // The WDT caused a wake-from-IDLE
asm volatile( “eret” );              // return from interrupt
}

if ( RCON & 0x10 )
{
                                     // The WDT timed-out while the device was awake
}
```

9.3.5 WDT Period Selection

The WDT clock source is the internal LPRC oscillator, which has a nominal frequency of 31.25 kHz. This creates a nominal time-out period for the WDT (T_{WDT}) of 1 millisecond when no postscaler is used.

> **Note:** The WDT time-out period is directly related to the frequency of the LPRC oscillator. The frequency of the oscillator will vary as a function of device operating voltage and temperature. Please refer to the specific device data sheet for LPRC oscillator clock frequency specifications.

9.3.5.1 WDT Postscalers

The WDT has a 5-bit postscaler to create a wide variety of time-out periods. This postscaler provides 1:1 through 1:1048576 divider ratios. Time-out periods that range between 1 ms and 1048.576 seconds (nominal) can be achieved using the postscaler.

The postscaler settings are selected using the FWDTPS<4:0> Configuration bits in the DEVCFG1 device configuration register. For more information on the WDT Configuration bits, please refer to **Section 32. “Configuration”**.

Equation 9-1: WDT Time-out Period Calculation

$$\text{WDT Period} = 1\text{ ms} \bullet 2^{\text{Prescaler}}$$

The time-out period of the WDT is calculated as follows:

Table 9-3: WDT Time-out Period vs. Postscaler Settings[1, 2]

FWDTPS<4:0>	Postscaler Ratio	Time-out Period
`00000`	1:1	1 ms
`00001`	1:2	2 ms
`00010`	1:4	4 ms
`00011`	1:8	8 ms
`00100`	1:16	16 ms
`00101`	1:32	32 ms
`00110`	1:64	64 ms
`00111`	1:128	128 ms
`01000`	1:256	256 ms
`01001`	1:512	512 ms
`01010`	1:1024	1.024 s
`01011`	1:2048	2.048 s
`01100`	1:4096	4.096 s
`01101`	1:8192	8.192 s
`01110`	1:16384	16.384 s
`01111`	1:32768	32.768 s
`10000`	1:65536	65.536 s
`10001`	1:131072	131.072 s
`10010`	1:262144	262.144 s
`10011`	1:524288	524.288 s
`10100`	1:1045876	1048.576 s

Note 1: All other combinations will result in operation as if the prescaler was set to `10100`.

2: The periods listed are based on a 32 kHz (nominal) input clock.

9.3.6 PWRT Timer Operation

The PWRT provides an additional delay between the device POR delay and the beginning of code execution to allow the oscillator to stabilize. Devices that do not have an on-board voltage regulator have the PWRT permanently enabled. Devices that incorporate an on-board voltage regulator automatically enable the PWRT only when the on-board voltage regulator is disabled. The PWRT cannot be enabled or disabled by the device configuration or software.

9.4 INTERRUPT AND RESET GENERATION

The WDT will either cause an Non-Maskable Interrupt (NMI) or a device Reset when it expires. The Power-Saving mode of the device determines which event occurs.

The PWRT does not generate interrupts or Resets.

9.4.1 Watchdog Timer Reset

When the WDT expires and the device is not in SLEEP or IDLE mode, a device Reset is generated. The CPU code execution jumps to the Device Reset Vector and the registers and peripherals are forced to their Reset values.

9.4.2 Watchdog Timer NMI

When the WDT expires in SLEEP or IDLE mode, a NMI is generated. The NMI causes the CPU code execution to jump to the Device Reset Vector. While the NMI shares the same Vector as a device Reset, registers and peripherals are not reset.

To cause a WDT time out in SLEEP mode to act like an interrupt, a return-from-interrupt (`RETFIE`) instruction may be used in the start-up code after the event was determined to be a WDT wake-up. This will cause code execution to continue with the opcode, following the `WAIT` instruction that put the device into Power-Saving mode. Refer to Example 9-2.

9.4.3 Determining Device Status When a WDT Event Has Occurred

To detect a WDT Reset, the WDTO (RCON<4>), SLEEP (RCON<3>), and IDLE (WDTCON<2>) bits must be tested. If the WDTO bit is a ‘1’, the event was due to a WDT time out. The SLEEP and IDLE bits can then be tested to determine whether the WDT event occurred while the device was awake or if it was in SLEEP or IDLE mode. The user should clear the WDTO, SLEEP, and IDLE bits in the Interrupt Service Routine (ISR) to allow software to correctly determine the source of a subsequent WDT event.

9.4.4 Wake From Power-Saving Mode By a Non-WDT Event

When the device is awakened from Power-Saving mode by an interrupt, the WDT is cleared. Practically, this extends the time until the next WDT-generated device Reset occurs, so that an unintended WDT event does not occur too soon after the interrupt that woke the device.

9.5 I/O PINS

The PWRT is disabled when the internal voltage regulator is enabled. A device without an internal voltage regulator will always have the PWRT enabled. A device with an internal voltage regulator will enable the PWRT when the VREG pin is tied to ground (to disable the regulator).

9.6 OPERATION IN DEBUG AND POWER-SAVING MODES

> **Note:** In this manual, a distinction is made between a power mode as it is used in a specific module, and a power mode as it is used by the device, e.g., Sleep mode of the Comparator and SLEEP mode of the CPU. To indicate which type of power mode is intended, uppercase and lowercase letters (Sleep, Idle, Debug) signify a module power mode, and all uppercase letters (SLEEP, IDLE, DEBUG) signify a device power mode.

9.6.1 WDT Operation in Power-Saving Modes

The WDT can be used to wake the device from SLEEP or IDLE. The WDT continues to operate in Power-Saving mode. A time out can then be used to wake the device. This allows the device to remain in SLEEP mode until the WDT expires or another interrupt wakes the device.

If the device does not re-enter SLEEP or IDLE mode following a wake-up, the WDT must be disabled or periodically serviced to prevent a device Reset.

9.6.2 WDT Operation in SLEEP Mode

The WDT, if enabled, will continue operation in SLEEP mode. The WDT may be used to wake the device from SLEEP. When the WDT times out in SLEEP, a NMI is generated and the WDTO (RCON<4>) bit is set. The NMI vectors execution to the CPU start-up address, but does not reset registers or peripherals. The SLEEP (RCON<3>) status bit will be set indicating the device was in SLEEP. These bits allow the start-up code to determine the cause of the wake-up.

9.6.3 WDT Operation in IDLE Mode

The WDT, if enabled, will continue operation in IDLE mode. The WDT may be used to wake the device from IDLE. When the WDT times out in IDLE, a NMI is generated and the WDTO (RCON<4>) bit is set. The NMI vectors execution to the CPU start-up address, but does not reset registers or peripherals. The IDLE (RCON<2>) status bit will be set indicating the device was in IDLE. These bits allow the start-up code to determine the cause of the wake-up.

9.6.4 Time Delays During Wake-up

The delay between a WDT time-out and the beginning of code execution depends on the Power-Saving mode.

There will be a time delay between the WDT event in SLEEP mode and the beginning of code execution. The duration of this delay consists of the start-up time for the oscillator in use and the PWRT delay, if it is enabled.

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 delays are required at wake-up.

9.6.5 WDT Operation in DEBUG Mode

The WDT is always frozen and therefore does not time-out in DEBUG mode.

9.7 EFFECTS OF VARIOUS RESETS

Any form of device Reset will clear the WDT. The Reset will return the WDTCON register to the default value and the WDT will be disabled unless it is enabled by the device configuration.

> **Note:** After a device Reset, the WDT ON (WDTCON<15>) bit will reflect the state of the FWDTEN (DEVCFG1<23>) bit.

9.8 DESIGN TIPS

Question 1: ***Why does the device reset even though I reset the WDT in my main software loop?***

Answer: Make sure that the timing of the software loop that clears the WDTCLR (WDTCON<0>) bit meets the minimum time-out specification of the WDT (not the typical value) to ensure operation at different voltage and temperatures. Also, make sure that interrupt processing time has been accounted for.

Question 2: ***What should my software do before entering SLEEP or IDLE mode?***

Answer: Make sure that the sources intended to wake the device have their IEC 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 (SIDL) 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 details.

If the WDT is to be used in SLEEP mode, then the WDT should be serviced before entering sleep to provide a complete WDT interval before the device exits SLEEP mode.

Question 3: ***How do I tell if the WDT or other peripheral woke the device from SLEEP or IDLE mode?***

Answer: Most interrupts have their own unique vector. The vector is determined by the interrupt source. For interrupts that share a vector, the IFS bits for each enabled interrupt source (that shares the vector) can be polled to determine: a.) the source of the interrupt and b.) the source of the wake-up. If the WDT woke the device, the user’s start-up code must check for the WDT time-out event, WDTO (RCON<4>), and branch accordingly.

9.9 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 PIC32MX device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the WDT and PWRT modules are:

Title	Application Note #
No related application notes at this time.	N/A

Note:	Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC32MX family of devices.

9.10 REVISION HISTORY

Revision A (September 2007)

This is the initial released revision of this document.

Revision B (October 2007)

Updated document to remove Confidential status.

Revision C (April 2008)

Revised status to Preliminary; Revised U-0 to r-x;

Revision D (June 2008)

Delete note from Section 9.3; Revise Example 9-2; Change Reserved bits from “Maintain as” to “Write”; Added Note to ON bit (WDTCON Register).

NOTES: