INTRODUCTION

The oscillator module handles the clock source selection for 8-bit PIC® microcontrollers. Within this module is an internal high-frequency oscillator (HFINTOSC), an external high-frequency oscillator (EXTOSC), a Secondary Oscillator block (SOSC), a PLL and other low-frequency oscillators. Figure 1 shows how the oscillators are connected in the oscillator module.

This Technical Brief covers the internal oscillator's features, functions and operation. This document discusses the functionality in the oscillator module; how it operates with clock switching and the effects of the 8-bit PIC MCU’s power-saving features.

FIGURE 1: SIMPLIFIED OSCILLATOR MODULE BLOCK DIAGRAM

OSCILLATOR MODULE

The oscillator module provides a diversity of clock sources and features for the CPU, and various peripherals. Though the system clock is initially configured in the Configuration Words, the user may adjust the clock frequency or change the clock source in the user code. Figure 1 shows the module’s block diagram. The diagram includes the clock sources, clock source and postscaler selection, a 4xPLL circuit, Fail-Safe Clock Monitor (FSCM) and Peripheral Module Disable (PMD) support. To learn more on the module, refer to the “Oscillator Module Technical Brief” (TB3148).

Note: The availability of the Peripheral Module Disable (PMD) for the System Clock (SYSCMD) is device-specific. Refer to the device data sheet to check if this feature is present in the device.
High-Frequency Internal Oscillator

The HFINTOSC is the module’s most commonly used internal oscillator with a frequency range up to 64 MHz. The HFINTOSC operating frequency is determined by the combination of the OSCFRQ and N/CDIV settings, with a default value determined by RSTOSC. Figure 2 shows the available oscillator settings in the Configuration Words.

This oscillator is also used as the reference for the constant frequency outputs of the MFINTOSC peripheral clocks. The MFINTOSC frequencies are digital divisors of the HFINTOSC set to provide both 31.25 and 500 kHz.

**FIGURE 2: INITIALIZING HFINTOSC AS THE SYSTEM CLOCK SOURCE THROUGH RSTOSC AND FEXTOSC IN MPLAB® X IDE**

Auto-Enable

Regardless of the system clock source (external or internal), the HFINTOSC is enabled when a peripheral uses the MFINTOSC as a clock source. Since the oscillator frequency is a factor in the device power operation, it takes into account the OSCFRQ setting of HFINTOSC. For lower power consumption with MFINTOSC, the OSCFRQ must have the least value set.

Frequency Selection

At Reset, the HFINTOSC operating frequency is determined by RSTOSC. In the user code, the frequency can be varied in the OSCFRQ register. The operating frequency can also be postscaled further through clock switching. Table 1 provides a list of selectable frequencies under OSCFRQ. At OSCFRQ frequencies above 4 MHz, power consumption is higher and start-up time is longer. These effects are caused by the internal PLL blocks used to achieve the higher frequencies.

**TABLE 1: SELECTABLE HFINTOSC FREQUENCIES (kHz)**

<table>
<thead>
<tr>
<th>OSCFRQ &lt;3:0&gt;</th>
<th>NDIV/CDIV &lt;3:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 &lt;0&gt;</td>
</tr>
<tr>
<td>1 MHz &lt;0&gt;</td>
<td>1000</td>
</tr>
<tr>
<td>2 MHz &lt;1&gt;</td>
<td>2000</td>
</tr>
<tr>
<td>4 MHz &lt;2&gt;</td>
<td>4000</td>
</tr>
<tr>
<td>8 MHz &lt;3&gt;</td>
<td>8000</td>
</tr>
<tr>
<td>12 MHz &lt;4&gt;</td>
<td>12000</td>
</tr>
<tr>
<td>16 MHz &lt;5&gt;</td>
<td>16000</td>
</tr>
<tr>
<td>32 MHz &lt;6&gt;</td>
<td>32000</td>
</tr>
<tr>
<td>48 MHz &lt;7&gt;²</td>
<td>48000</td>
</tr>
<tr>
<td>64 MHz &lt;8&gt;²</td>
<td>64000</td>
</tr>
</tbody>
</table>

**Note 1:** These are default oscillator settings determined by the RSTOCs bits.

**Note 2:** These options are for PIC18 only and are not available in PIC16.
Frequency Tuning

HFINTOSC frequency can be fine-tuned in the user code. This is done by writing to the OSCTUNE register in 6-bit two’s complement (Sign bit + 5 bits). With an OSCTUNE value of zero, the oscillator is tuned to the center frequency. Once OSCTUNE is modified, the frequency begins to shift without affecting code execution. However, there is no indication whether the shift has occurred. The frequency can be shifted up to 3.2% of the calibrated value, in 64 steps, as configured in the OSCTUNE register.

Clock Switching

The clock switching feature of the oscillator module allows the user to change both the system clock source and frequency divider by writing to the New Oscillator Source (NOSC) and New Divider Selection (NDIV) request bits of the Oscillator Control register.

Writing to the NOSC bits initiates a switch between clock sources. Between internal and external clock sources, switching adds a certain delay before the next code execution. The CPU will confirm if the new oscillator source is available and waits for the oscillator to stabilize before it is ready for use.

The operating frequency of HFINTOSC and the other sources can be postscaled further with a write to the NDIV bits. Table 1 shows more options of configurable frequencies, through combinations of the OSCFRQ and N/CDIV bits. For a postscaler-only switch, the clock source is immediately ready, hence the delay is negligible.

Examples of switching sources and postscalers are shown in Figure 3 and Figure 4, respectively. The figures show the behavior of toggling an I/O pin before and after a clock switch. Sample codes for these figures are shown in Example 1 and Example 2. Having NOSC and NDIV combined in the same register avoids having two switching processes. With a single write to the control register, there is less risk to overclocking and underclocking in the intermediate state. To learn more about the switching process, refer to TB3148.

Fail-Safe Clock Monitor

In cases where the external oscillator fails, the oscillator module feature selects the HFINTOSC in the COSC bits to ensure that the device will continue to operate. The device will operate with the frequency set in OSCFRQ and retain the postscaler setting in N/CDIV. An Oscillator Fail Interrupt (OSCIF) is provided to give the user an option to resolve the external clock failure while the system runs on the HFINTOSC. The condition is cleared either upon Reset, after Sleep or a successful switch through NOSC and NDIV.
POWER SAVING FEATURES

Generally, all internal oscillators, including HFINTOSC, are unaffected when the device enters Doze, Idle or Sleep mode. These modes only control or disable the configured system clock. Peripherals which directly use the internal oscillators still continue to operate.

For oscillator modules with PMD support, the PMD feature disables the use of the clock source for any peripheral. Similar to the previous modes, this does not have any effect on the internal oscillators and on the operation of the peripherals they serve.

For a more detailed discussion about these power-saving features, refer to “Doze, Idle and PMD Features of 8-Bit PIC® Microcontrollers” (TB3144).

CONCLUSION

The digitally controlled oscillator of 8-bit PIC microcontrollers offers a wide range of selectable frequencies for the system and peripheral clock sources. The oscillator module provides an easy and smooth process for switching operating frequencies. Power-saving features have no effect on the internal oscillator, which ensures normal operation for peripherals alongside lower power consumption.
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.