INTRODUCTION

Phase Locked Loop (PLL) circuits are increasingly used in microcontrollers to achieve higher internal clock frequencies. This allows better performance while reducing overall noise. Several of Microchip’s PIC18 microcontrollers feature 4x PLLs in their clock generation circuits. This makes it possible to generate an internal 40 MHz clock from an external 10 MHz crystal.

One drawback in the use of PLL circuits is that they create a small, but still measurable level of transient phase shifts, or jitter. This Technical Brief shows the influence of PLL jitter on Microchip’s PIC18 microcontrollers, how it affects the overall clock of the microcontroller and how the combined effects of jitter and crystal drift are well below the CAN 2.0 specification.

EXTERNAL CLOCK, INTERNAL CLOCK AND MEASURABLE JITTER

The microcontroller clock frequency generated from a PLL circuit is subject to a jitter, also defined as Phase Jitter or Phase Skew. For its PIC18 Enhanced microcontrollers, Microchip specifies Phase Jitter ($P_{jitter}$) as being 2% (Gaussian distribution, within 3 standard deviations) and Total Jitter ($T_{jitter}$) as being $2 \times P_{jitter}$.

The CAN protocol uses a bit stuffing technique that inserts a bit of a given polarity, following five bits with the opposite polarity. This gives a total of 10 bits transmitted without resynchronization (compensation for jitter or phase error).

Given the random nature of the jitter error added, it can be shown that the total error caused by the jitter tends to cancel itself over time. For a period of 10 bits, it is necessary to add only two jitter intervals to correct for jitter induced error: one interval in the beginning of the 10-bit period and another at the end. The overall effect is shown in Figure 1.

FIGURE 1: EFFECTS OF PHASE JITTER ON THE MICROCONTROLLER CLOCK AND CAN BIT TIME

![Diagram showing nominal clock, clock with jitter, and CAN bit time with jitter](image-url)
Once these considerations are taken into account, it is possible to show that the relation between the jitter and the total frequency error can be defined as:

**EQUATION 1:**

\[
\Delta f = \frac{T_{\text{jitter}}}{10 \times \text{NBT}} = \frac{2 \times P_{\text{jitter}}}{10 \times \text{NBT}}
\]

where jitter is expressed in terms of time and NBT is the Nominal Bit Time.

For example, assume a CAN bit rate of 125 Kb/s, which gives an NBT of 8 µs. For a 16 MHz clock generated from a 4x PLL, the jitter at this clock frequency is:

**EQUATION 2:**

\[
2\% \times \frac{1}{16 \text{ MHz}} = \frac{0.02}{16 \times 10^6} = 1.25 \text{ ns}
\]

and resultant frequency error is:

**EQUATION 3:**

\[
\frac{2 \times (1.25 \times 10^{-9})}{10 \times (8 \times 10^{-6})} = 3.125 \times 10^{-5} = 0.0031\%
\]

Table 1 shows the relation between the clock generated by the PLL and the frequency error from jitter (measured jitter induced error of 2%, Gaussian distribution, within 3 standard deviations) as a percentage of the nominal clock frequency.

This is clearly smaller than the expected drift of a crystal oscillator, typically specified at 100 ppm or 0.01%. If we add jitter to oscillator drift, we have a total frequency drift of 0.0132%. The total oscillator frequency errors for common clock frequencies and bit rates, including both drift and jitter, are shown in Table 2.

**CONCLUSION**

For Microchip’s PIC18 microcontrollers, the errors introduced by the crystal drift, plus the errors introduced by the PLL jitter, are well below the ones required by the CAN 2.0 bus specification. This makes PIC18 microcontrollers using the internal PLL well-suited for high-speed CAN bus-based communications.

### TABLE 1: FREQUENCY ERROR FROM JITTER AT VARIOUS PLL GENERATED CLOCK SPEEDS

<table>
<thead>
<tr>
<th>PLL Output</th>
<th>(P_{\text{jitter}})</th>
<th>(T_{\text{jitter}})</th>
<th>Frequency Error at Various Nominal Bit Times (Bit Rates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 MHz</td>
<td>0.5 ns</td>
<td>1 ns</td>
<td>8 µs (125 Kb/s) 0.00125% 0.00250% 0.005% 0.01%</td>
</tr>
<tr>
<td>24 MHz</td>
<td>0.83 ns</td>
<td>1.67 ns</td>
<td>4 µs (250 Kb/s) 0.00209% 0.00418% 0.008% 0.017%</td>
</tr>
<tr>
<td>16 MHz</td>
<td>1.25 ns</td>
<td>2.5 ns</td>
<td>2 µs (500 Kb/s) 0.00313% 0.00625% 0.013% 0.025%</td>
</tr>
</tbody>
</table>

### TABLE 2: TOTAL FREQUENCY ERROR AT VARIOUS PLL GENERATED CLOCK SPEEDS (100 ppm OSCILLATOR DRIFT INCLUDING ERROR FROM JITTER)

<table>
<thead>
<tr>
<th>Nominal PLL Output</th>
<th>Frequency Error at Various Nominal Bit Times (Bit Rates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 MHz</td>
<td>8 µs (125 Kb/s) 0.01125% 0.01250% 0.015% 0.02%</td>
</tr>
<tr>
<td>24 MHz</td>
<td>4 µs (250 Kb/s) 0.01209% 0.01418% 0.018% 0.027%</td>
</tr>
<tr>
<td>16 MHz</td>
<td>2 µs (500 Kb/s) 0.01313% 0.01625% 0.023% 0.035%</td>
</tr>
<tr>
<td></td>
<td>1 µs (1 Mb/s) 0.01313% 0.01625% 0.023% 0.035%</td>
</tr>
</tbody>
</table>
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.