

## Introduction

The timer circuits within the various Power Manager devices are designed to provide an approximate delay between power sequencing events. Given the wide range in power supply startups (hundreds of milliseconds to sub milliseconds) the timers are programmable from tens of microseconds to hundreds of milliseconds and up to two seconds in the second-generation devices. The timer circuits are digital counters that are driven from either the internal clock or external clock source. All digital counters have an inherent uncertainty window that is as wide as its input clock period. Thus all digital timers that are triggered by an asynchronous event have a timing inaccuracy of up to one clock period. In this application note we will examine how the clocks for the timers are generated and how timing accuracy can be optimized. The worst case timing inaccuracy from a first generation Power Manager device (ispPAC<sup>®</sup>-POWR604, ispPAC-POWR1208, and ispPAC-POWR1208P1) can be as much as 50% at lower delay settings. For the second-generation Power Manager devices, the worst-case inaccuracy is 12.5% at lower delay settings. Unless otherwise specified in this application note all timer accuracy descriptions are based on the timer circuits independently of the clock reference. Clock reference inaccuracy should be added to the timer inaccuracy to compute the overall timer inaccuracy.

## First Generation Timer Circuits

The ispPAC-POWR604, ispPAC-POWR1208, and ispPAC-POWR1208P1 belong to the first generation Power Manager devices from Lattice Semiconductor Corporation. These three devices share the same timer architecture and have similar behavior. The timer values for these devices range from 32  $\mu$ s to 524.3 ms. All the timers use a 250kHz clock reference. This reference clock is connected to a prescaler that divides the 250kHz by 4, 8, 16, 32, 64, 128, 256 or 512. The output clock period from the prescaler determines unit delay of all timers on-chip. The timer architecture is shown in the sub-schematic screen of PAC-Designer<sup>®</sup> in Figure 1. The prescaler block is shown as "Timeout Range." Figure 2 shows the drop-down list in the "Clock & Timer Settings" dialog box from PAC-Designer for the various prescale values. Figure 2 also shows the associated clock periods." Figure 3 shows the "Timeout" list from the same dialog box as in Figure 2 along with the associated count values and the maximum timing inaccuracy as a percentage.

**Figure 1. Clock and Timer Circuits Sub-schematic from PAC-Designer**

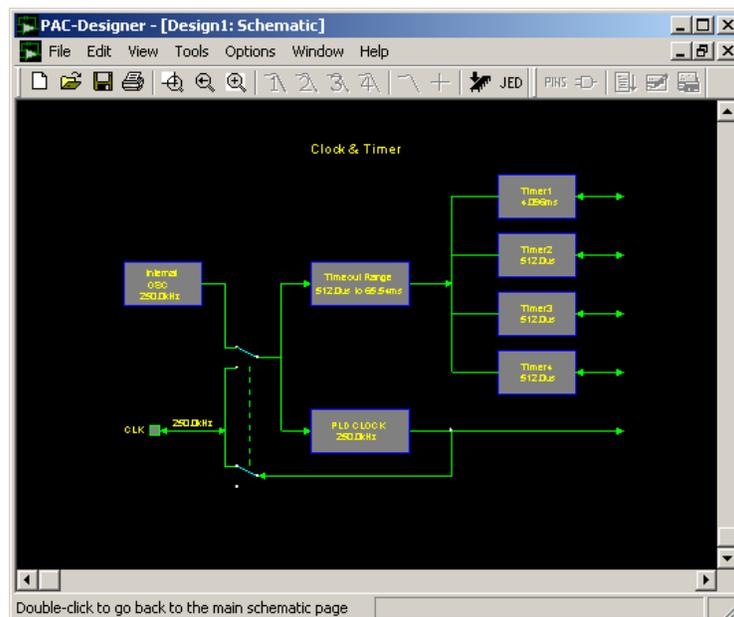


Figure 2. PAC-Designer Dialog Box Selection of Timer Clock

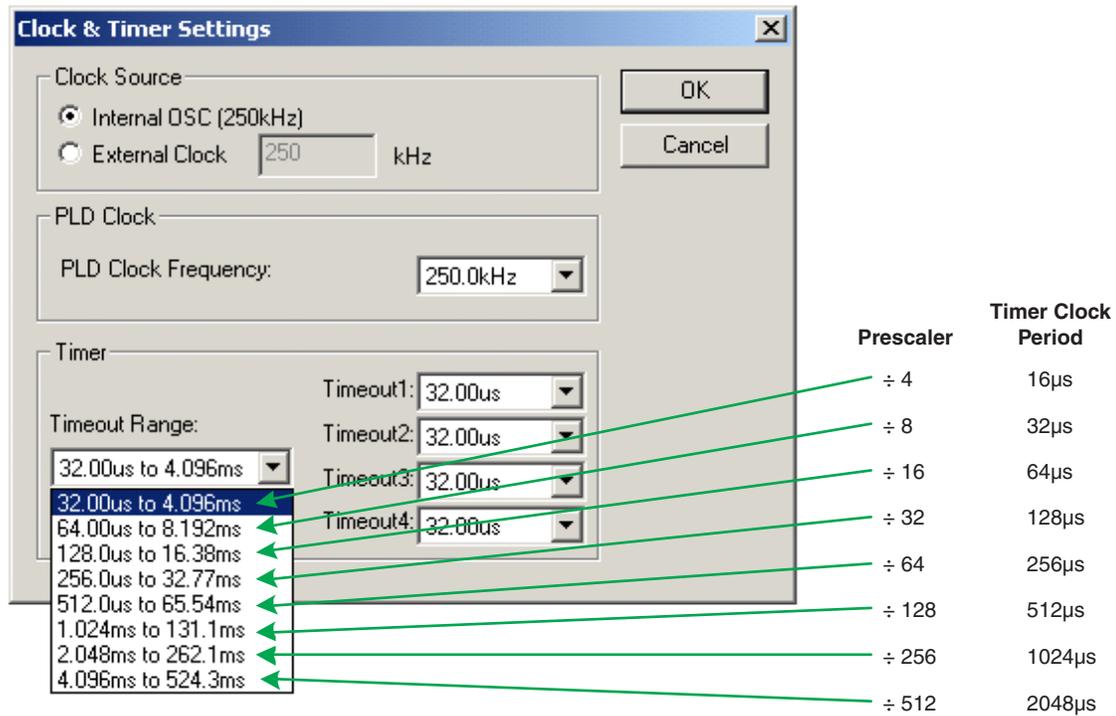


Figure 3. PAC-Designer Dialog Box Selection of Timeout Values

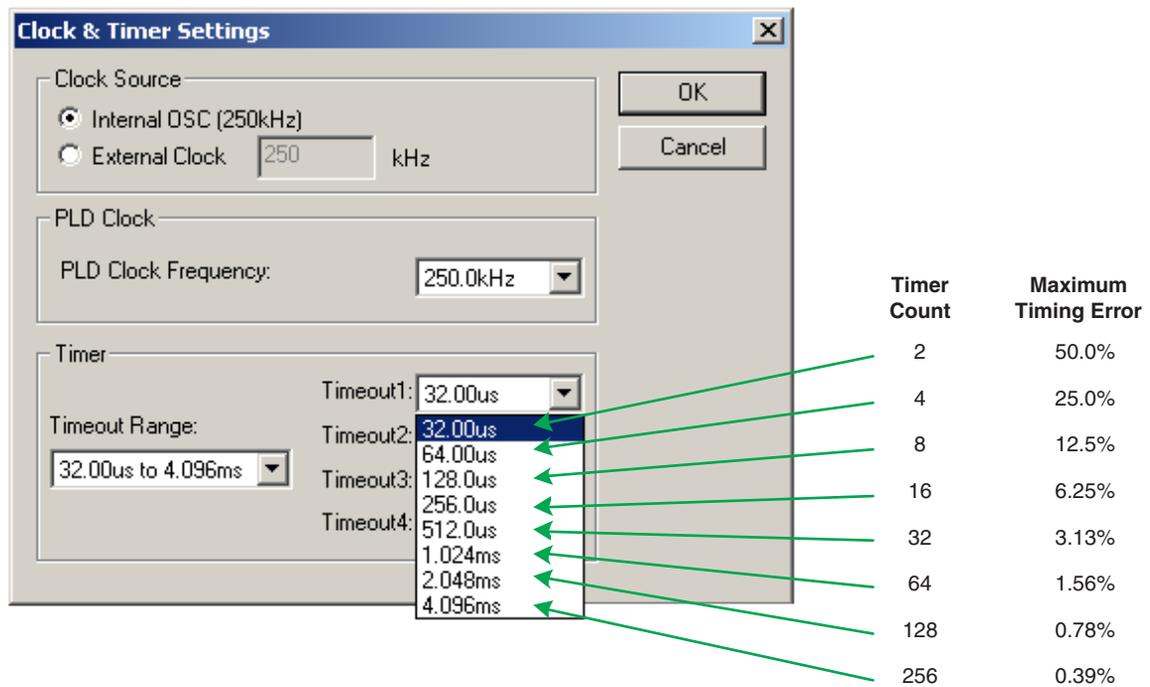


Figure 4(a) provides a more detailed block diagram of a first generation timer. The Timer Clock is generated by dividing either the internal 250kHz clock source or an external clock source (not shown). Using PAC-Designer, the divider value N can be set to any of the following values: 4, 8, 16, 32, 64, 128, 256, or 512 (Figure 2). Note that the resulting Timer Clock provides the time base for all timers. Each timer is controlled by its individual Timer Gate signal. When this signal is Low the counter is reset, the Timer Clock is blocked, and the Terminal Count output is set Low. When the Timer Gate signal goes High the Timer Clocks are passed to the counter. Using PAC-Designer the count value M can be set to any of the following values: 2, 4, 8, 16, 32, 64, 128, or 256 (Figure 3). A Macrocell in the PLD generates the Timer Gate signal in response to its input conditions. In this case it is the change in status of the VMON input. While the Timer Gate signal is at logic '1' the counter counts down from its preloaded value. When the counter value reaches zero the Terminal Count signal becomes logic '1'. Each Terminal Count signal is routed to the input side of the AND array to support the programmable logic control. The expected timeout value can be calculated by multiplying the divider value N times the counter value M and dividing by the clock source frequency.

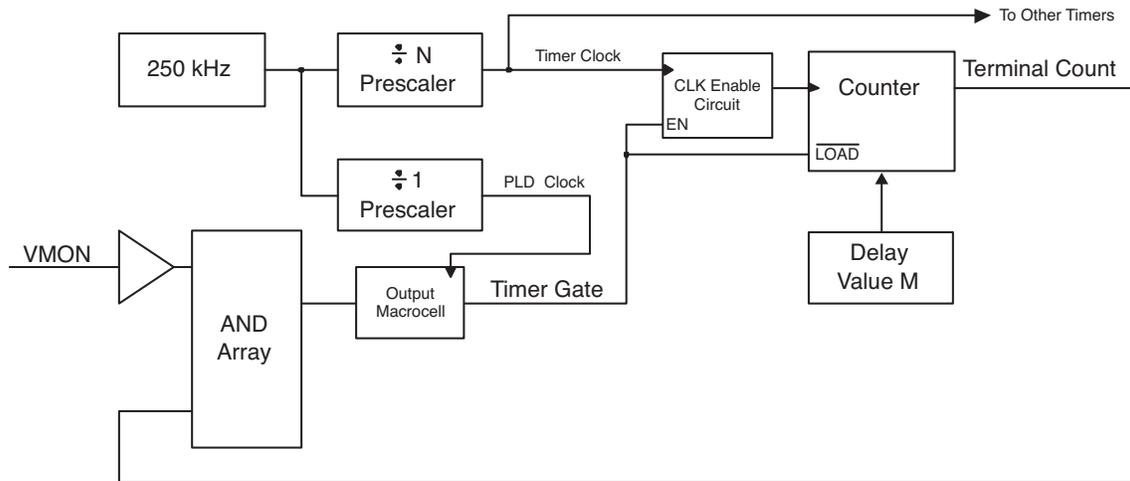
Figure 4(b) shows a short sequence. At Step 1 the sequence waits on a VMON input to transition to logical '1' and jumps to Step 2 when the VMON signal is true. Step 2 enables the Timer Gate and waits for the Terminal Count signal to be true. When the Terminal Count signal is logical '1' the sequence continues to Step 3 and beyond.

The waveforms in Figure 4(c) correspond to both the block diagram and the example sequence. The Timer Gate signal is active one PLD clock cycle after the VMON input is true. As seen, the start of a timer may occur anytime within the timer clock period, which results in reducing the first count period. This reduced period is designated in the diagram as the timing error  $t_{\text{error}}$ . After counting M Timer Clock edges the Terminal Count output is true. The Actual Time-out when measured from this edge back the VMON edge falls short by the amount shown as  $t_{\text{error}}$ . The expected Time-Out calculation is shown at the bottom of the waveforms (for the internal 250kHz clock source). The actual Time-Out can be somewhat less than the expected value by up to one whole Timer Clock period based on the random timing relationship between the VMON input and the Timer Clock edge.

To calculate the maximum inaccuracy a timer may have as a percentage, use the following equation where M is the count value. This equation holds true for all Power Manager timers.

$$\% \text{ Error} = 1/M \times 100 \quad (1)$$

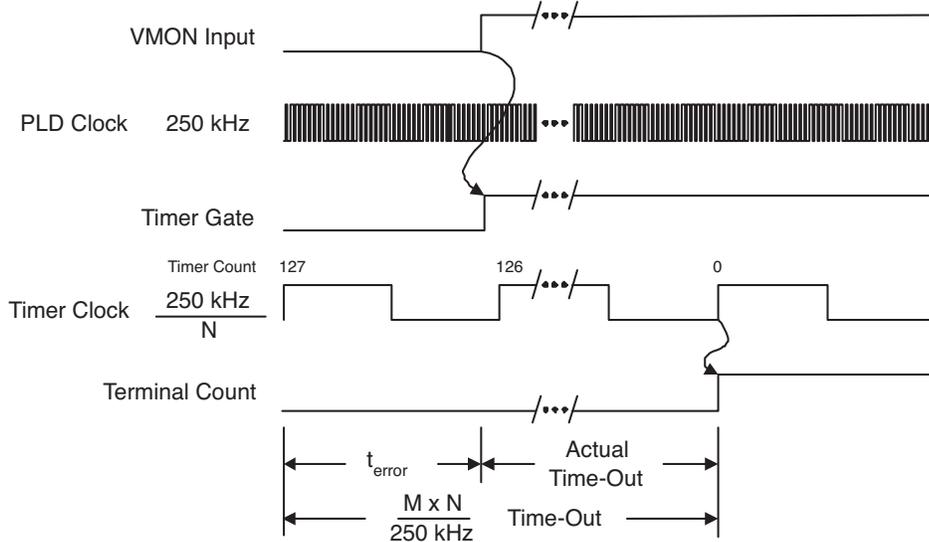
Figure 4. First Generation Timer Block Diagram and Waveforms



(a)

- Step 1 Wait for VMON
- Step 2 Wait for 4.096 ms using timer
- Step 3 ...

(b)



(c)

### Timer Delay Inaccuracy

For most of the timeout settings, there are multiple combinations of prescaler and timeout counter settings that will provide the same timeout value. However, the combinations that use the larger timeout counts (128 and 256) will provide the most accurate timers while the smaller timeout counts (2 and 4) will result in timers with the most inaccuracy. For example a 2 ms delay can be programmed using a prescale value of 4 (16  $\mu$ s clock period) and a timeout count of 128, which results in a maximum timer inaccuracy of 0.8% (100/128). Alternatively, a prescale value of 256 (1 ms clock period) and a timeout count of 2 results in a maximum inaccuracy of 50% (100/2). By careful selection of "Timeout Range" and "Timeout" values, one can optimize timer accuracy.

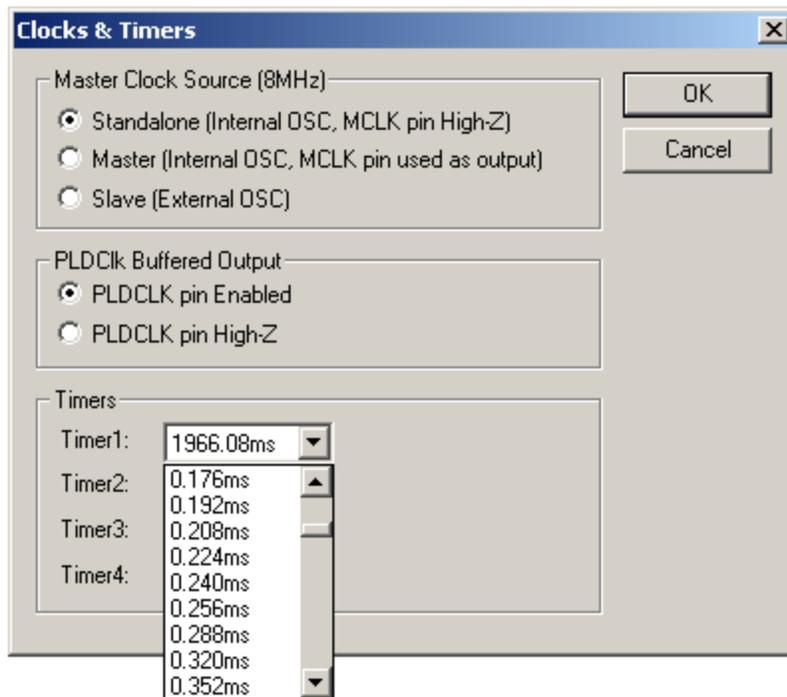
## Second Generation Timer Circuits

The ispPAC-POWR1220AT8, ispPAC-POWR1014/A, and ispPAC-POWR607 devices belong to the second-generation Power Manager family (PMII). The differences between the timer architectures are as follows:

Parameter	Power Manager II	Power Manager
Maximum Time Delay	2 seconds	512 ms
Maximum Error	12%	50%
Prescaler Association	Individually set on a per timer basis	Common prescaler for all timers

Because each timer has a built-in prescaler, the timeout range for each timer can be independently set. To simplify timeout selection, PAC-Designer presents a single long list of timeout values (see Figure 5). Each entry in the list combines both a prescaler setting and a counter value. There are 16 possible settings for the prescaler and eight possible settings for the timer. Figure 6 shows the architecture of clock and timer circuits in Power Manager II devices and also lists the possible settings for the prescaler and counter.

**Figure 5. Second Generation Clock and Timer Dialog Box from PAC-Designer**



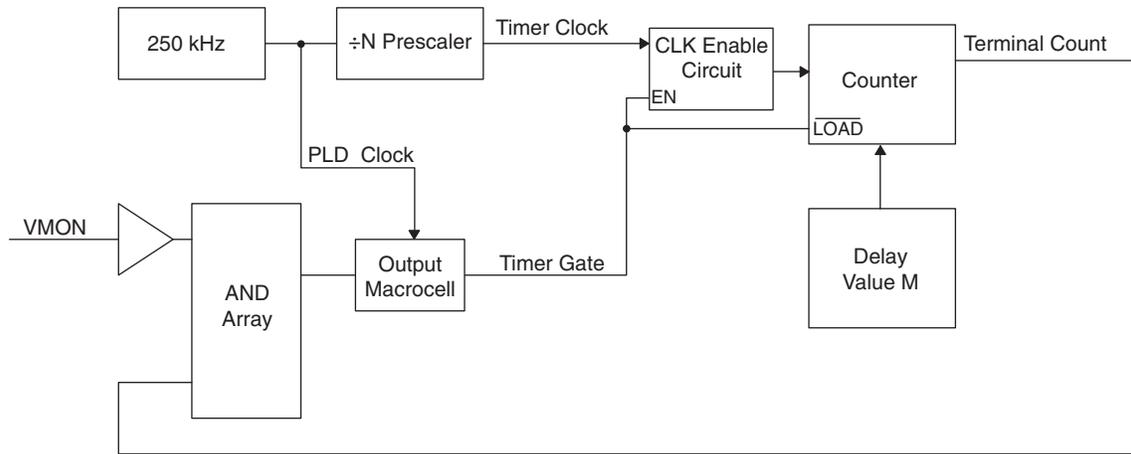
Because the counter values are limited to selections between 8 and 15 the maximum timing inaccuracy are bound between one cycle out of 8 and one cycle out of 15. Thus, the worst case maximum inaccuracy is 12.5% instead of 50% as was the case in the first generation Power Managers. As with all digital timers the maximum timing inaccuracy that can be observed corresponds to a single period of the timer clock.

From the LogiBuilder window inside PAC-Designer one can use the menu item View -> ABEL source to see the prescaler and timer settings after the design has been compiled. The ABEL source is used both by the fitter to package the design within the PLD core and by the simulator for debugging purposes. The timers are external to the PLD core and do not require any ABEL code to support their settings. However, the functional simulator does need to know the timer settings and therefore ABEL code is needed to drive the simulator. Figure 7 show the "Clocks and Timers" dialog from PAC-Designer and the corresponding section of the ABEL code that supports the simulation of the timers. Two macro keywords are used in the ABEL code exclusively for the Power Manager timer support and they are: XLAT\_PRESCALER and XLAT\_STIMER. The XLAT\_PRESCALER describes the connec-

tions and value of the prescaler, while XLAT\_STIMER describes the connections and value of the counter. The prescaler value is the second to last parameter in the list (in Figure 7 Timer 4 is using a divide of 32768). The counter value is the last parameter in the list for the XLAT\_STIMER macro (in Figure 7 Timer 4 is using a count of 7). Note the simulator software timers use one less count than the actual hardware. Therefore, the actual timeout for Timer 4 is  $4\mu\text{s} \times 32768 \times 8 = 1.048$  seconds.

To minimize potential timing inaccuracy one can restrict timeout selections to those listed in Table 1. These values listed only use a count value of 15 and thus the worst-case inaccuracy is 6.67%.

Figure 6. Second Generation Prescale and Timer Block Diagram



Prescaler Settings				Counter Settings	
N	Timer Clock Period	N	Timer Clock Period	Count M	Maximum Timing Error
÷1	4µs	÷256	1.024ms	8	12.5%
÷2	8µs	÷512	2.048ms	9	11.1%
÷4	16µs	÷1024	4.096ms	10	10.0%
÷8	32µs	÷2048	8.192ms	11	9.09%
÷16	64µs	÷4096	16.384ms	12	8.33%
÷32	128µs	÷8192	32.768ms	13	7.69%
÷64	256µs	÷16384	65.536ms	14	7.14%
÷128	512µs	÷32768	131.072ms	15	6.67%

Figure 7. Timer Settings and Corresponding ABEL Code

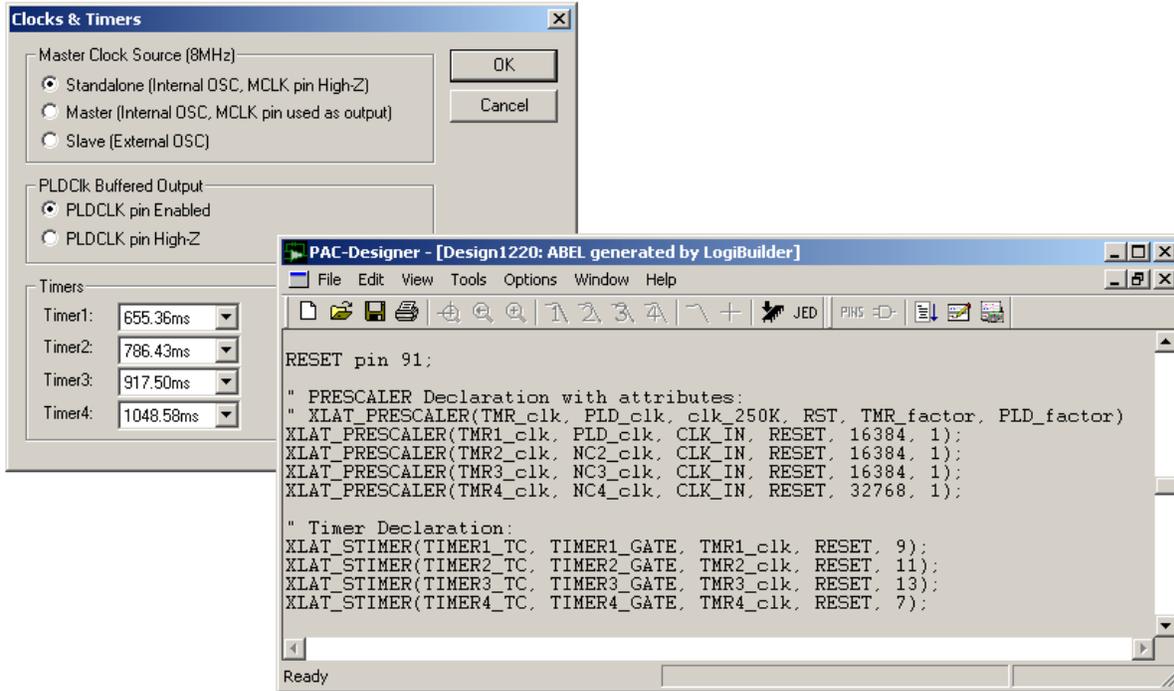


Table 1. Exact Timer Settings with Worst Case Timing Errors of 6.67%

60µs	120µs	240µs	480µs	960µs	1.92ms	3.84ms	7.68ms
15.36ms	30.72ms	61.44ms	122.88ms	245.76ms	491.52ms	983.04ms	1966ms

### Clock Oscillator Accuracy

All of the timing accuracies discussed thus far are based on the asynchronous nature of gating a digital timer and missing a portion of a single clock cycle. Any inaccuracy in the clock reference can essentially be added to the digital timing inaccuracy. Thus, if a clock is running 10% fast and a timeout from Table 1 is used the resulting worst-case timing inaccuracy is about 15.99%. The reason it is not 16.67% (simply adding the two inaccuracies) is that only a portion of the timeout is clocked with the fast clock (up to one clock cycle may be missed). On the other hand, if the clock source is running 10% slow, then the timeout can be 10% long for the complete count. Thus the range of inaccuracies for a timer using a setting from Table 1 is -16% to +10% when the clock accuracy is included.

### Summary

In both first and second generation Power Manager devices some inherent lack of timer accuracy will be observed due to the asynchronous timer gating signal with respect to the counter clock. In both generations of timer circuits, higher counter settings will have better performance over lower counter settings. It was also discussed how the clock oscillator accuracy influences timer accuracy.

### Related Literature

- ispPAC-POWR604 Data Sheet
- ispPAC-POWR1208 Data Sheet
- ispPAC-POWR1208P1 Data Sheet
- ispPAC-POWR1220AT8 Data Sheet

- ispPAC-POWR1014 Data Sheet
- ispPAC-POWR1014A Data Sheet
- ispPAC-POWR607 Data Sheet

### **Technical Support Assistance**

Hotline: 1-800-LATTICE (North America)  
          +1-503-268-8001 (Outside North America)

e-mail: [isppacs@latticesemi.com](mailto:isppacs@latticesemi.com)

Internet: [www.latticesemi.com](http://www.latticesemi.com)

### **Revision History**

<b>Date</b>	<b>Version</b>	<b>Change Summary</b>
December 2007	01.0	Initial release.