Jitter in PLL-Based Systems: Causes, Effects, and Solutions

Jitter is extremely important in systems using PLL-based clock drivers. The effects of jitter range from not having any effect on system operation to rendering the system completely non-functional. This application note provides the reader with a clear understanding of jitter in high-speed systems. It introduces the reader to various kinds of jitter in high-speed systems, their causes and their effects, and methods of reducing jitter. This application note will concentrate on jitter in PLL-based frequency synthesizers.

What is a PLL-Based Frequency Synthesizer ?

Frequency Synthesizers use one or more *Phase-Locked Loops (PLL)* to generate one to many different frequencies on their outputs, from one or more reference sources. The reference frequency is usually generated by a crystal attached to the synthesizer. It is rarely generated from an external oscillator. The design goal of frequency synthesizers is to replace multiple oscillators in a system, and hence reduce board space and cost. *Figure 1* shows a block diagram of a Phase-Locked Loop (PLL).

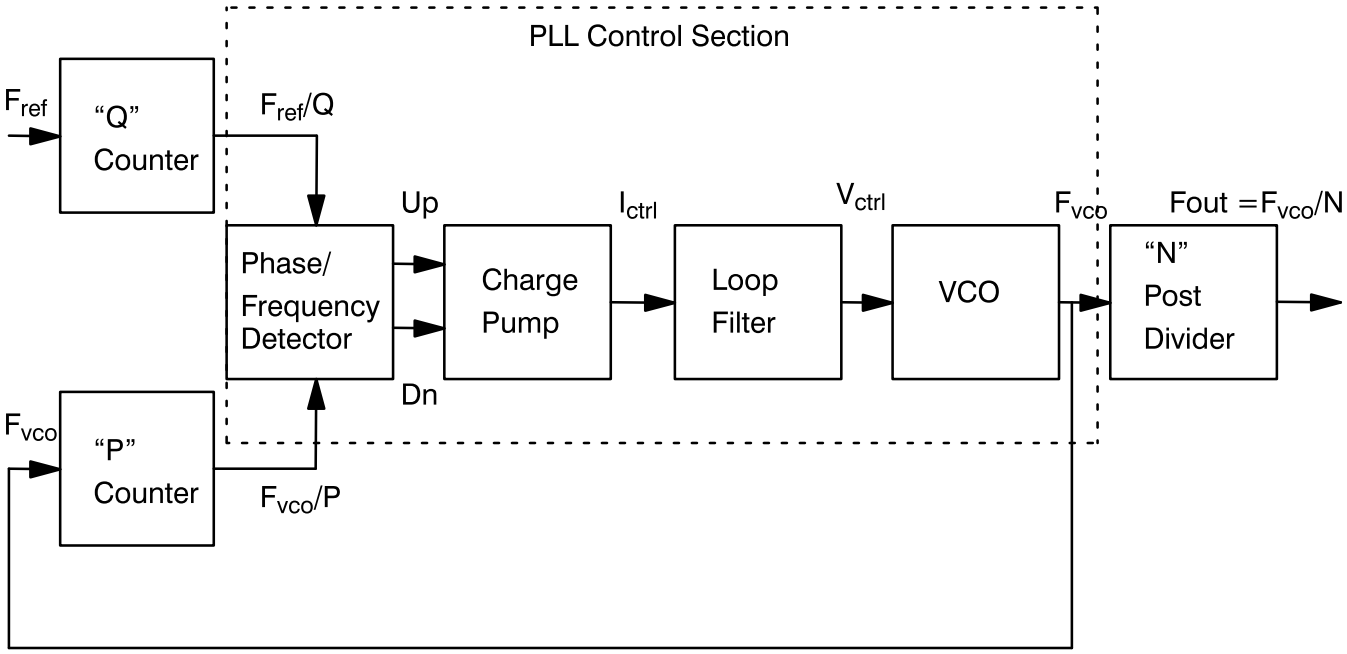


Figure 1. Block Diagram of a Phase-Locked Loop

A PLL has two inputs: a reference input, and a feedback input. A PLL corrects frequency in two ways. The first, frequency correction, corrects large differences in frequency between the reference input and the feedback input. Frequency correction is activated when the input frequency is changing significantly, or when the device is powered up. Frequency correction is the "rough" tuning of the PLL. "Fine" tuning occurs when phase correction is activated.

The Phase/Frequency Detector detects differences in phase and frequency between the reference and feedback inputs and generates compensating "Up" and "Down" signals. The pulsewidth of the "Up" signal is greater than the "Down" signal, if the feedback input frequency is less than the reference frequency, and vice versa. These control signals are then passed through a charge pump and a loop filter, to generate a control voltage, which feeds into a Voltage-Controlled Oscillator (VCO). The frequency of this oscillator is dependent on the V_{ctrl} input. At steady state, the VCO frequency is:

$F_{vco} = F_{ref} * P/Q$

The output frequency of the PLL can be expressed as

$F_{out} = (F_{ref} * P)/(Q * N)$

where

- F_{vco} = VCO Frequency
- F_{ref} = Reference Frequency
- P = Multiplier, lies in feedback path
- Q = Divider, lies in reference path
- N = Post Divider

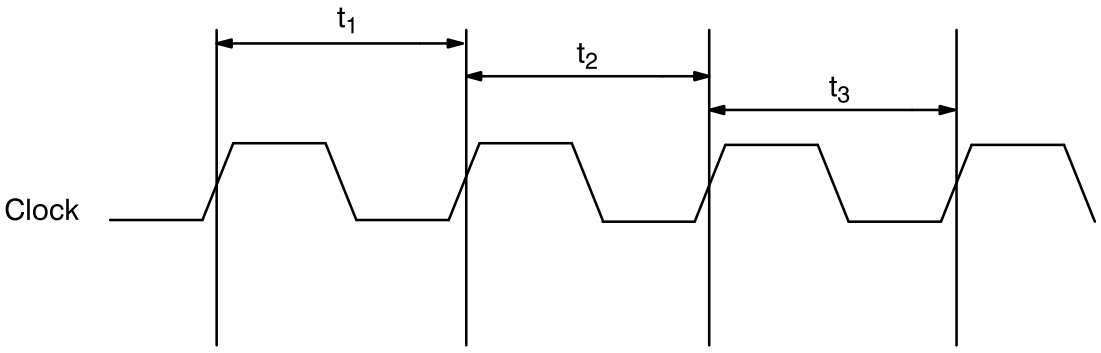
Clock Jitter

Jitter can be defined as the deviations in a clock's output transitions from their ideal positions. The deviation can either be leading or lagging the ideal position. Hence, jitter is sometimes specified in ±ps. Jitter is also specified in other units, like a percentage of frequency, or absolute value, in ns. Jitter measurements can be classified into three categories: cycle-cycle jitter, period jitter, and long-term jitter. Additionally, all jitter measurements are made at a specified voltage.

Cycle-Cycle Jitter

Cycle-cycle jitter is the change in a clock's output transition from its corresponding position in the previous cycle. This kind of jitter is the most difficult to measure and usually requires a Timing Interval Analyzer. *Figure 2* shows a graphical representation of cycle-cycle jitter. J_1 and J_2 are the jitter values measured. The maximum of such values measured over multiple cycles is the maximum cycle-cycle jitter.

Until recently, cycle-cycle jitter was not particularly meaningful in most cases. However, like the incorporation of PLLs in CPUs (e.g., the 486 and the Pentium™ processors), cycle-to-cycle jitter has taken on new significance. Consider the case shown in *Figure 3* where the output of one PLL_1 is the reference of



Jitter $J_1 = t_2 - t_1$
Jitter $J_2 = t_3 - t_2$

Figure 2. Cycle-Cycle Jitter

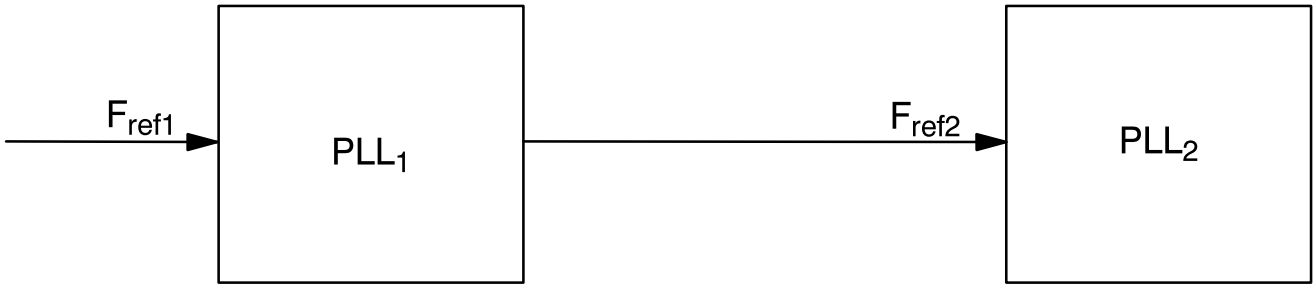


Figure 3. Application for Cycle-Cycle Jitter Measurement

PLL_2. In this case, if PLL_2 cannot lock to the reference frequency, the cycle-cycle jitter of the output of PLL_1 will have exceeded the maximum jitter allowable for PLL_2 to lock. If PLL_1 is the clock generator for PLL_2 embedded in the CPU, the output jitter of PLL_1 must be sufficiently low to successfully time the inputs to PLL_2.

Period Jitter

Period jitter measures the maximum change in a clock's output transition from its ideal position. *Figure 4* shows period jitter.

Period jitter measurements are used to calculate timing margins in systems. Consider, for example, a microprocessor-based system in which the processor requires 2 ns of data set-up time. Assume that the clock driving the microprocessor has a *maximum* of 2.5 ns period jitter. In this case, the rising edge of clock *can* occur before data is valid on the data bus. Hence, the processor will be presented with incorrect data, and the system will not operate. This example is illustrated in *Figure 5*. The system designer needs to take period jitter into account while designing the system.

Long-Term Jitter

Long-term jitter measures the maximum change in a clock's output transition from its ideal position, over many cycles. The term "many" depends on the application and the frequency. For PC motherboard and graphics applications, this term "many" usually refers to 10−20 microseconds. For other applications, it may be different. *Figure 6* shows a graphical representation of long-term jitter.

A classic example of a system affected by long-term jitter is a graphics card driving a CRT. Assume that a pixel of data is meant for the pixel at co-ordinates (10,24) on the CRT. Because of a jittery clock, this data may drive a pixel at location (11,28) on the CRT. Over an extended period of time, the data meant for pixel (10,24) may be driving a pixel far away from its ideal (10,24) location. Since this effect of a jittery clock is usually consistent over all pixels, the overall effect of a jittery clock is to cause an image to shift from its ideal display position on screen. This effect is sometimes called "running" of the screen.

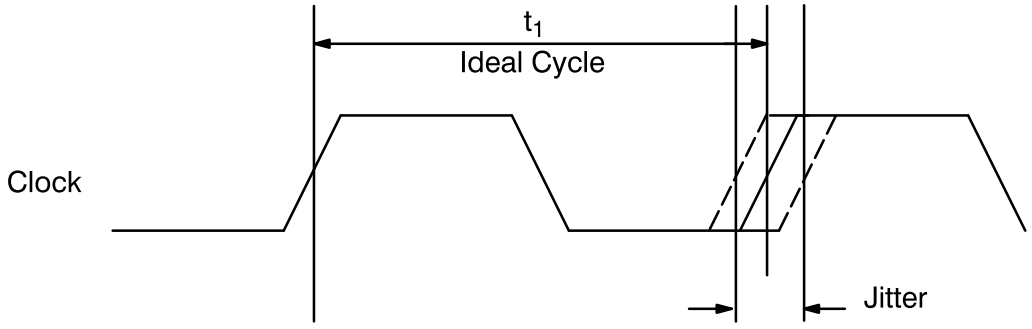


Figure 4. Period Jitter

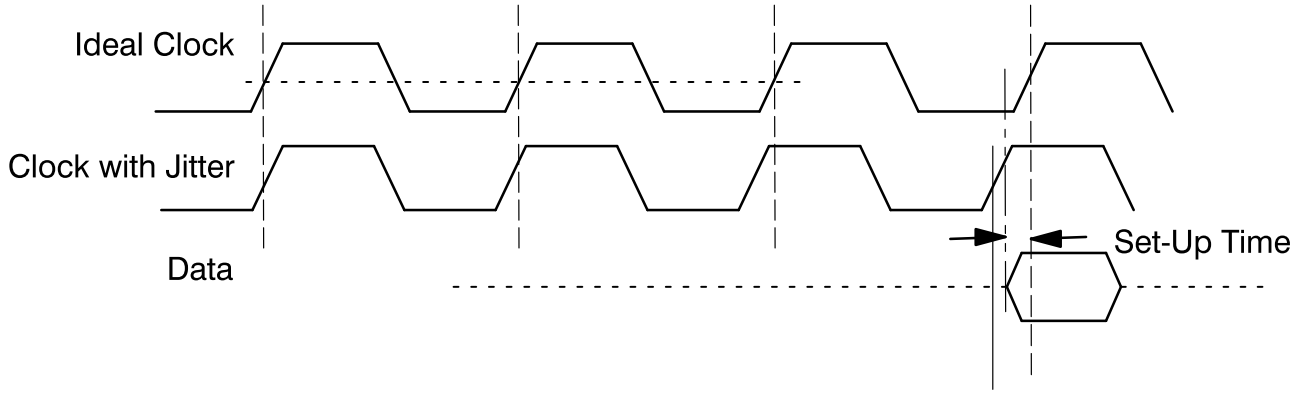


Figure 5. Application for Period Jitter Measurement

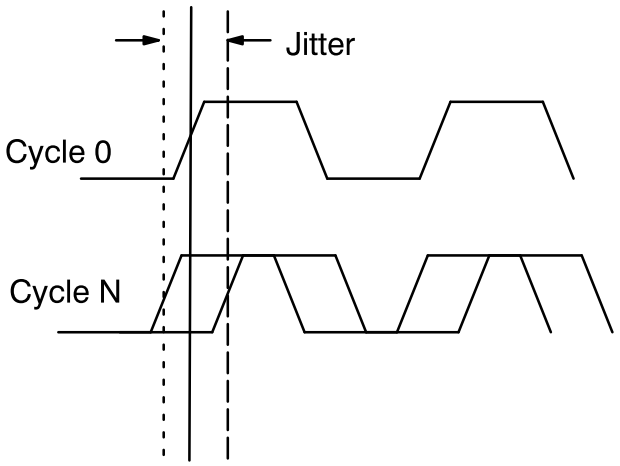


Figure 6. Long-Term Jitter

Causes of Jitter

There are four primary causes of jitter as indicated below in decreasing order of importance.

- Power supply noise on a PLL's supply inputs, which appears on the output as jitter. This is the largest, though not always constant, contributor to jitter. Power supply noise manifests itself through various ways, some of which are:
  - — *Ground Bounce:* When there is a surge of current through the output drivers, the inductance of the leads to the supply planes (V_{cc} and GND) have a voltage drop across it (value = L.di/dt). This raises or lowers the effective ground potential of the device. Hence, if the output frequency is dependent on the effective supply voltage, this frequency will change because of ground bounce. Second, the threshold voltage of transistors within the oscillator changes, which causes a change in frequency. This has a twofold effect. First, the output frequency changes. Second, if the oscillator feeds a PLL, this PLL tries to correct the change in frequency. Both of these effects appear on the outputs as jitter.
  - — *V_{dd} Noise: Figure 7* shows an inverter in the internal counter of the PLL. The threshold voltage of the input is half the V_{dd} potential. Assume for example, that the V_{dd} signal has a 100-mV p-p noise ripple associated with it. This noise will cause a shift in the threshold voltage at the input of the inverter. The change in the triggering level of this inverter will cause jitter. If this noise signal has a rise time of 1 V/ns, then 100 ps of peak-peak jitter will appear on the outputs of the inverter, due to the 100-mV p-p ripple voltage.

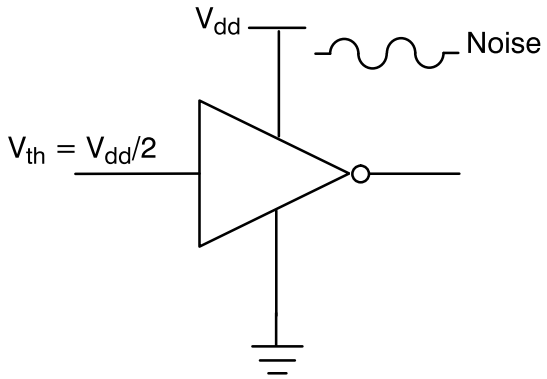


Figure 7. Effect of V_{dd} Noise on Jitter

- The PLL in a frequency synthesizer has a dead-band associated with it, during which the phase and frequency detector does not detect small changes in the input phase. Since these changes are not detected, they do not get corrected and appear on the outputs in the form of jitter.
- Random thermal noise from the crystal reference, or any other resonating device.
- Random mechanical noise from vibrations of the crystal reference.

Measuring Jitter

Since we have defined three kinds of jitter, we will propose three methods of measuring them.

Cycle-Cycle Jitter

Measuring cycle-cycle jitter is extremely difficult. A Timing Interval Analyzer (TIA) is required to perform this measurement. In this case, the output of the jittery clock is connected to a TIA, and the measurement to be specified is the difference of time periods of consecutive clock cycles. The maximum of this difference over multiple cycles is the cycle-cycle jitter.

Period Jitter

A simple method of measuring period jitter requires a storage oscilloscope. Set the trigger for the rising edge of clock. Then scroll the display to the next rising edge of the clock and turn on the persistence. If the scope is set up correctly, the width of the blurring on the displayed transition will indicate the amount of period jitter in the clock. An example of period jitter measurement is shown in *Figure 2*. The peak in the horizontal histogram indicates the fundamental frequency, while the spreading around this frequency shows the jitter.

Long-Term Jitter

Long-term jitter is probably the easiest to measure. It uses a measuring technique called *differential phase measurement*. The jittery clock is connected to an oscilloscope with a delayed time-base feature. The scope is set to trigger on the rising edge of clock. Then, using the delayed time-base feature, the same clock waveform is displayed on the screen.

To make sure that the scope calibration and characteristics can perform the jitter measurement, measure the output of a stable clock source, like a crystal oscillator. If the waveform has no blurs or bands, the scope can correctly measure long-term jitter.

Methods of Reducing Jitter

As discussed before, two major causes of jitter are power supply noise and ground bounce. Reducing the power supply noise and eliminating ground bounce will reduce most of the jitter in a system.

Reducing Power Supply Noise

Power supply noise can be reduced by bypassing and filtering the power supply appropriately.

Bypassing, by using a large tantalum capacitor (10–1000 μF) *attached to the board power supply*, will prevent a fall in voltage caused by current surges, as well as reduce power supply ripple. Attach this capacitor as close as possible to where the V_{dd} and GND signals enter the PCB.

This large capacitor will, however, be ineffective at very high frequencies. Hence, a small capacitor, 0.1 μF, will be required to *filter* high-frequency noise. Cypress recommends attaching a 0.1-μF ceramic capacitor *on every V_{dd} pin* of the frequency synthesizer. These capacitors must be attached as close to the pins as is physically possible. Surface mount capacitors are preferred because of their low lead inductance.

If the part has separate analog and digital power supply pins, use a 22Ω resistor in series with a 22-μF capacitor to ground to filter low-frequency noise. Using a smaller capacitor in parallel with the 22-μF will ensure better attenuation.

Finally, using a regulated power supply (such as from a 3-pin regulator, or a Zener diode), with the above bypassing and filtering techniques, will ensure better power supply rejection.

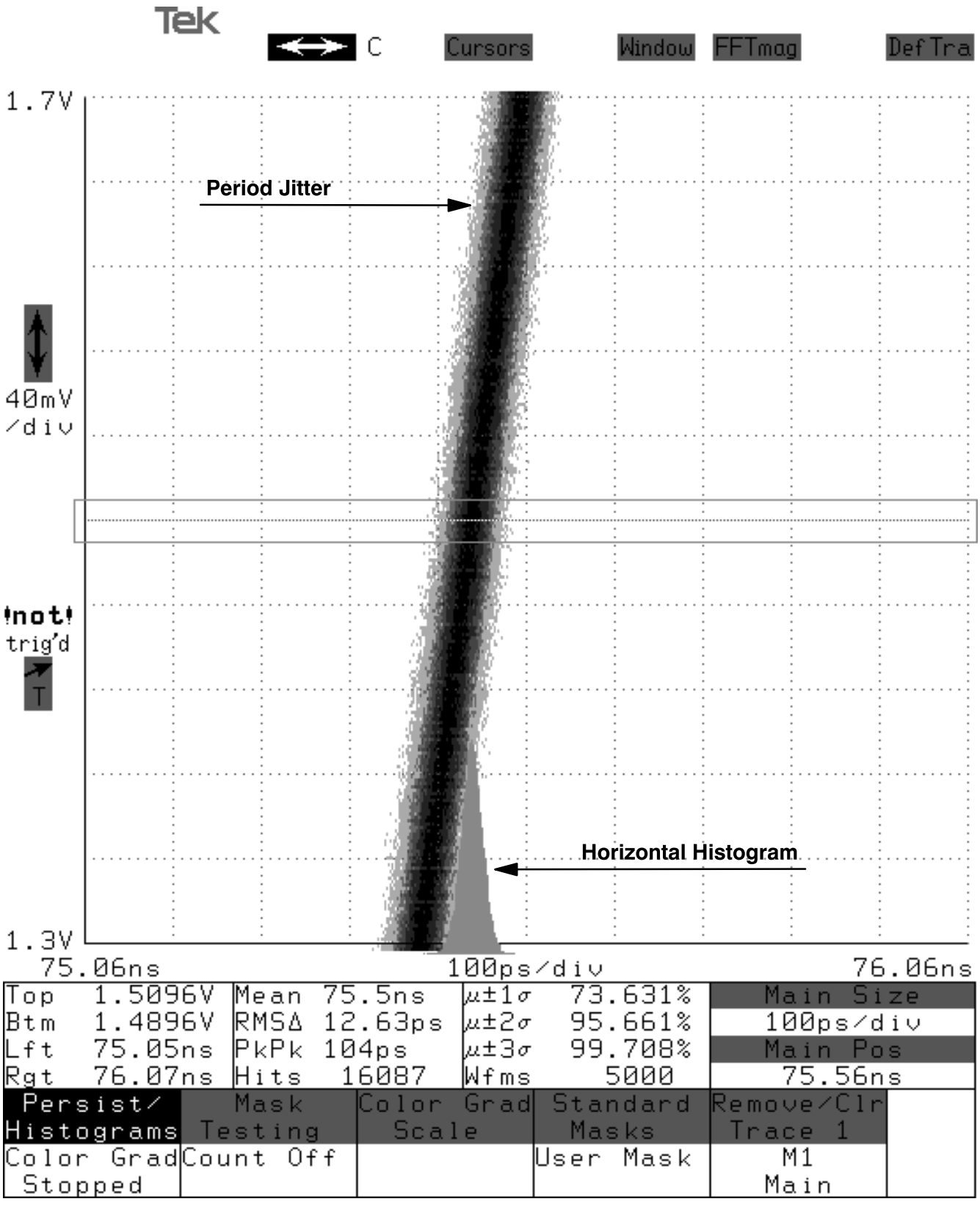


Figure 8. An Example of Period Jitter Measurement

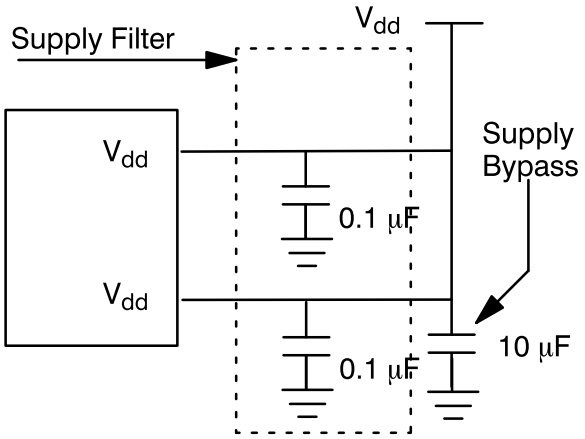


Figure 9. Power Supply Noise Filter Circuit

Figure 9 shows a circuit which can be used to reduce power supply noise for clock generators with multiple digital power supplies, such as the CY2254. In case the clock generator has separate analog and digital power supplies, such as the ICD2028, use the circuit shown in *Figure 10*.

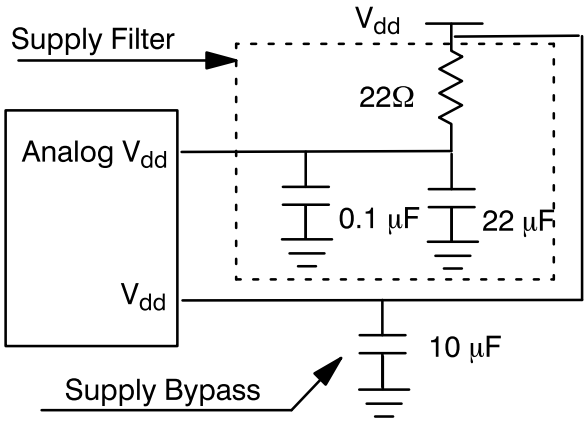


Figure 10. Power Supply Noise Filter Circuit

In addition to using power supply bypass and filtering techniques, avoid routing any high-frequency signals below the clock generator. This will minimize noise-coupling effects, and will result in reduced jitter on the outputs of the clock generator.

Eliminating Ground Bounce

Ground bounce can be eliminated in three ways. The first is to reduce the number of loads on the output of the device. A second method of reducing ground bounce is to provide large ground planes on your PCB. Finally, if you have two or more ground pins, connect them *individually* to the ground plane, instead of shorting them together. The third way is to install a series resistance on the output pins. This will limit the output current and reduce ground bounce.

Conclusion

This application note has discussed the various jitter measurements which can be made on a system. It also discussed the causes and effects of jitter, and presented techniques for reducing jitter in PLL-based systems. Using this information the reader should be able to design more reliable high-speed systems.

References

1. High Speed Digital Design, A Handbook of Black Magic, *Howard Johnson & Martin Graham,* 1993 Prentice-Hall, Inc.

Pentium is a trademark of Intel Corporation.