

Timing Margin Analysis for Clock Buffers in High Speed Synchronous Networking Systems

By Cameron Katrai

Abstract

High performance systems need low skew clock generation and distribution. Clock skew is defined as the difference in time between simultaneous clock transitions within a system. This Application Brief describes how skew has become a major constraint that forms the upper boundary for the system clock's frequency. By avoiding complicated architecture or faster logic, reducing the system clock skew also reduces costs. Recommendations and supporting analysis are given for designing near optimal clock networks.

Clock networks must be designed to minimize skew or the differences in delay throughout a clocking network. The ideal is that every component, such as sequential elements (flip-flops and latches) that need clocking, should receive the edge of the clock at the same time within each clock period. Fully synchronous designs require this methodology and are highly recommended since they can tolerate higher clock rates and make it easier to perform timing analysis. Synchronous operation means that all of the flip-flops or devices that need clocks in the system use the same clock signal.

To ensure that the network operates as closely to the ideal as possible, the skew must be minimized along the entire clocking network. This ensures that all sequential elements see a common clock edge. This application note shows how to calculate timing margin and the minimum clock output skew.

Design Example

Figure 1 shows an example of a typical synchronous design. Both the delay in the data path and the clock skew affect data transfer from register to register. The data path delay is defined as the delay from a register (F1) in a section to the D input of another register (F2). As shown, this delay also includes the delay through Logic L1. To make this synchronous design complete the output of F2 is fed back to Logic L2 and then to the input of F1 register. The example shows that as the clock frequency increases, the system will eventually fail. The failure is because the data will not meet the setup time at the input of F1 or F2. Figure 2 shows the timing diagram for such a synchronous design. Timing Margin is defined as the excess time in a clock cycle for a synchronous design to perform properly.

Note that this example does not consider the delay through the clock paths A and B. It is understood that the system designer has already matched the trace lengths for both clock A and clock B by equalizing the trace lengths. The measure of clock skew between clock A and clock B results solely from output-to-output skew inherent in the clock buffer. In other words, Path1 = Path2.

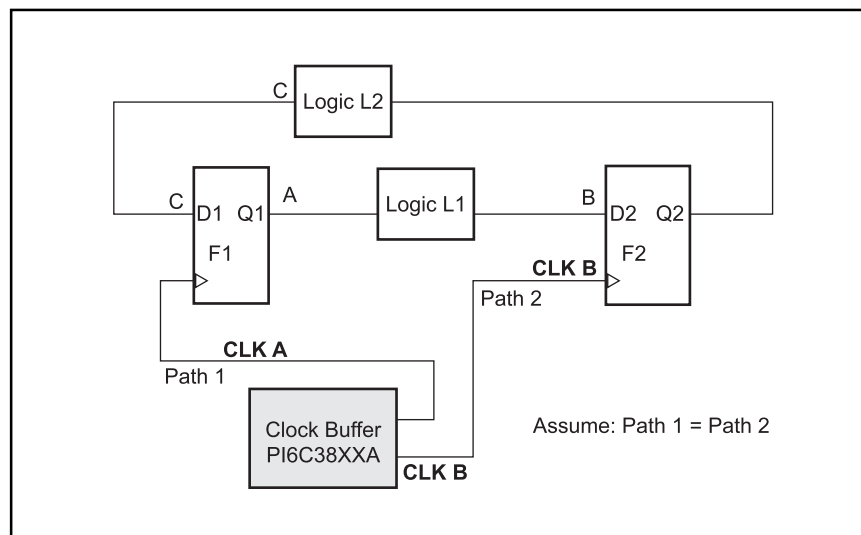


Figure 1. Block Diagram of a Synchronous Design

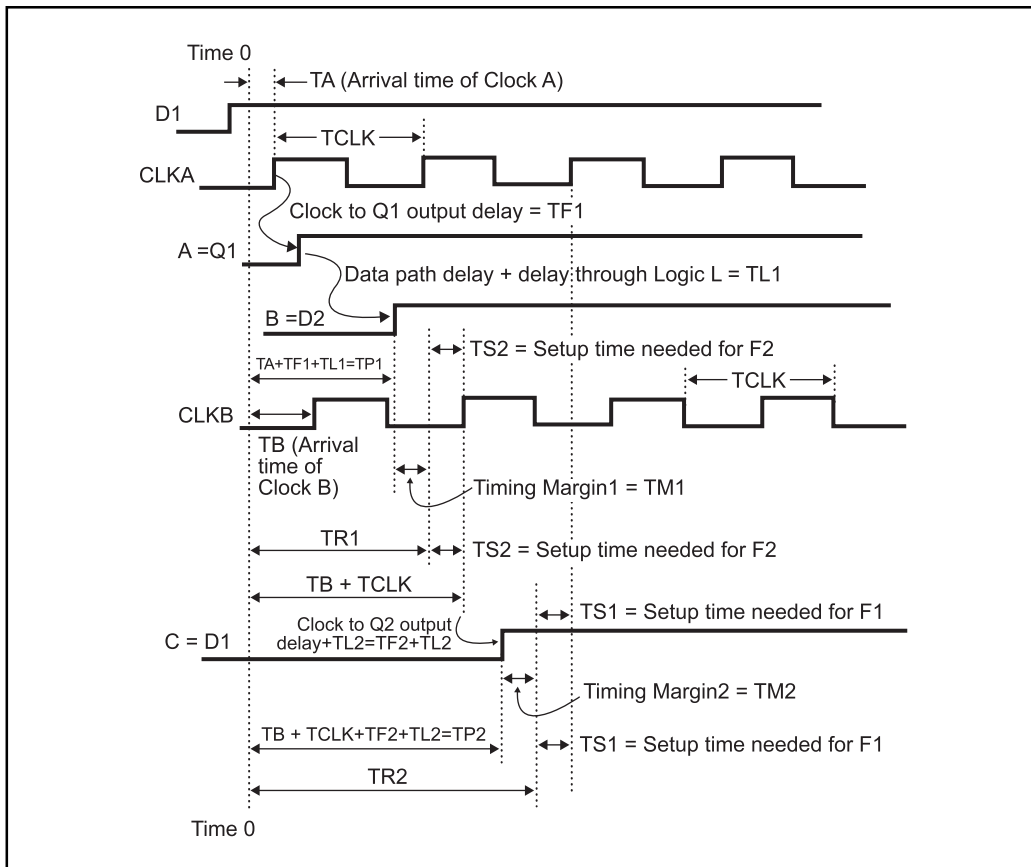


Figure 2. Timing Analysis of a Synchronous Design

The clock Skew or Tskew is the difference between a rising edge on CLK A in F1 and CLK B in F2.

$$T_{skew} = TB - TA$$

If the data path from F1 to F2 is too long and the delay through Logic L is too much, a setup time violation is realized on F1. It may make sense to delay the arrival of clock B to resolve this condition. However, delaying clock B is not the solution in synchronous designs. Delaying clock B more than a certain period will make it impossible for the output of F2 to make the setup time at the input of F1.

In-depth Analysis

As shown in Figure 2, if the data delay through Logic L1 and the data path is more than it should be, the data may not meet the setup time for F2. Figure 2 also shows that the output of F2 is fed back to F1. A major break down in the system happens when the data is not available on time on the input of F1.

We can calculate the required maximum time for data to be late (TR1) to meet the setup time at F2. From Figure 2, we have:

$$TR1(\text{Timing Required}) + TS2(\text{setup for F2}) = TCLK + TB(\text{Arrival time of Clock B})$$

$$TR1 = TCLK + TB(\text{Arrival time of Clock B}) - TS2(\text{setup for F2})$$

However, we can also calculate the predictable arrival time (TP1) of data at F2:

$$TP1 = TA(\text{Arrival time of Clock A}) + TF1(\text{clock to Q1 for F1}) + TL1(\text{delay of L1 and data path})$$

To have any Margin we must have the following:

$$TM1(\text{Margin1}) = TR1 - TP1 = TCLK + TB - TS2 - TA - TF1 - TL1 > 0$$

$$TM1 = TCLK - TF1 - TL - TS2 + (TB - TA) > 0$$

As noted before, as TB is increased, the margin gets better, however, as discussed, simply increasing TB will not help since the data needs to be fed back to F1 and get clocked in again with CLK A. As you increase TB the data may not be ready to be loaded back into F1.

Let us analyze this further:

$$TR2 + TS1 = TA + 2TCLK$$

The equation above holds since CLKA loads the feedback data from F2 to F1 after two clock periods.

$$TR2 = TA + 2TCLK - TS1$$

As shown previously, to have any margin meet the setup time for F1:

$$TM2 = TR2 - TP2$$

$$TP2 = TB + TCLK + TF2 + TL2$$

$$TM2 = TR2 - TP2 = TA + 2TCLK - TS1 - TB - TCLK - TF2 - TL2$$

$$TM2 = TA - TB + TCLK - TS1 - TF2 - TL2$$

$$TM2 = TCLK - TS1 - TF2 - TL2 - (TB - TA)$$

$$Tskew = TB - TA$$

$$TM2 = TCLK - TS1 - TF2 - TL2 - Tskew$$

The last equation shows that as Tskew increases, TM2 may not be met. This proves that as clock skew increases, timing margin may not be met. It is, however, important to note that design engineer's usually give a value for Timing Margin as a percentage of Clock period. Usually this value is 10 percent of total clock period. This gives a better value for Tskew as function of clock period.

$$TM2 = .1 \times TCLK$$

$$Tskew = TCLK - .1TCLK - TS1 - TF2 - TL2$$

$$Tskew = 0.9 TCLK - TS1 - TF2 - TL2$$

Note, that as Clock period TCLK shortens, Tskew must also decrease to meet the requirement.

Design example

A synchronous system is running at 140 MHz. Flip Flop propagation delay is 700ps. The required Flip Flop Setup time is 150ps. Logic delay and data path delay is 5000ps. If the required Margin is 10 percent of the Clock period, calculate the maximum skew between clock outputs.

$$Tskew = TCLK - TF2 - TL2 - TS1 - Margin = 7142.85 - 700 - 5000 - 150 - (.1 \times 7142.85) = 578.57ps$$

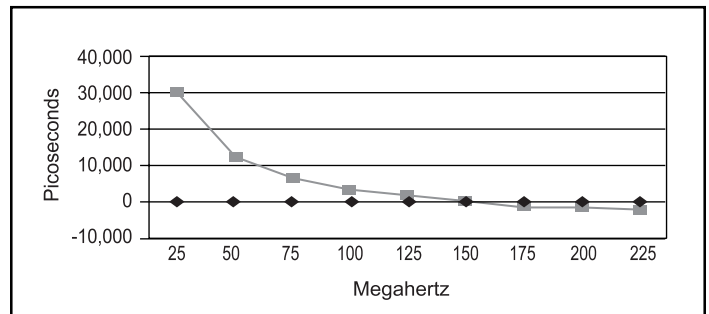
Sensitivity Analysis

Today's fast logic families provide TS as low as 150ps, TF as low as 700ps, and the logic delay is in the order of 5000ps (three Mux sections and a trace delay of 4 inches). Design Engineers use margins as low as 10 percent of clock period. Table 1 depicts the maximum Tskew needed at several different frequencies.

Notice how fast Tskew moves toward zero. As clock frequency moves from 150 MHz to 175 MHz, the Tskew takes on a negative value. The negative value of Tskew indicates that a faster logic should replace the existing logic. As shown in Table 1, at 141 MHz the maximum Tskew needed to meet the specification in the design mentioned above is 500ps.

Table 1. Frequency and Tskew relationship

Frequency (MHz)	TCLK (ns)	Tskew (ps)
25	40	30150
50	20	12150
75	13.333	6149.7
100	10	3150
125	8	1350
141	7.05	500
150	6.666	150
153.84	6.5	0
175	5.7142	-707.142
200	5	-1350
225	4.4444	-1850



Graph 1. Tskew Versus Frequency

Conclusion

Clock Skew has an important impact on overall performance of the system. Pericom provides clock drivers that meets 500ps output-to-output skew. Meeting up to 141 MHz for the design mentioned in this ApNote, Pericom's collection of Clock Drivers have found applications in high-performance ATM systems to provide clocking for devices such as FPGAs, controllers, and switching elements. Using advanced 0.5 micron CMOS technology to achieve an industry-leading speed grade, Pericom's PI6C38XX series provide the best characteristics in terms of tight skew, jitter, and phase error specifications. Pericom also provides a series of zero-delay clock buffers from one output (PI6C2501), to ten outputs (PI6C2510). Zero-Delay clock buffer allow zero-delay between the input and the output of the clock buffer. They are excellent choices for delivering the clock to different boards such as peripherals, memory, I/O, and transmission boards.