Clock Skew and Short Paths Timing

Table of Contents

Clock Skew

Differences in clock signal arrival times across the chip are called clock skew. It is a fundamental design principle that timing must satisfy register setup and hold-time requirements. Both data propagation delay and clock skew are parts of these calculations. Clocking sequentially-adjacent registers on the same edge of a high-skew clock can potentially cause timing violations or even functional failures. Figure 1 shows an example of sequentially-adjacent registers, where a local routing resource has been used to route the clock signal. In this situation, a noticeable clock skew is likely.

In Figure 1, all registers are clocked at the same edge, but the arrival time of the edge is different at each register. Figure 2 on page 2 indicates an example of the clock skew for the circuit shown in Figure 1.
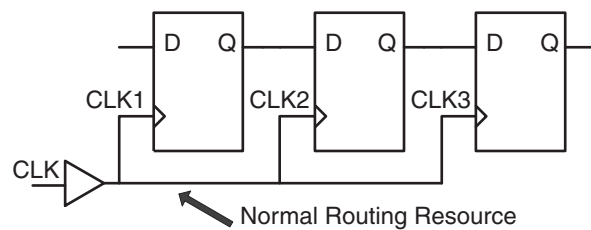
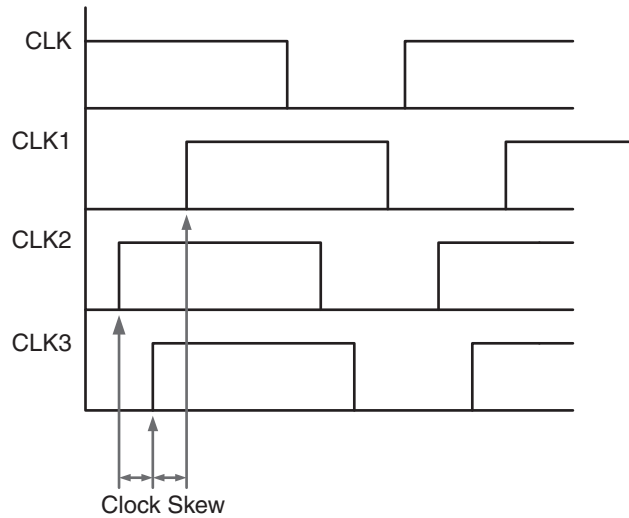*Figure 1* • **Sequentially Adjacent Registers with Clock Skew**

Figure 2 • **Clock Arrival Time Fluctuations in the Circuit of Figure 1**

Clock Skew and the Short Path Problem

The problem of short data paths in the presence of clock skew is very similar to hold-time violations in flip-flops. The problem arises when the data propagation delay between two adjacent flip-flops is less than the clock skew. Figure 3 shows an example of a simple circuit with waveforms to illustrate a short-path problem.

In Figure 3, since the same clock edge arrives at the second flip-flop later than the new data, the second flip-flop output switches at the same edge as the first flip-flop and with the same data as the first flip-flop. This causes U2 to shift the same data on the same edge as U1, resulting in a functional error.
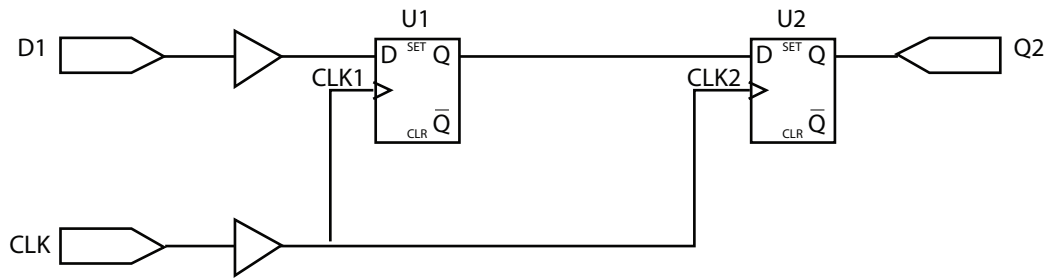
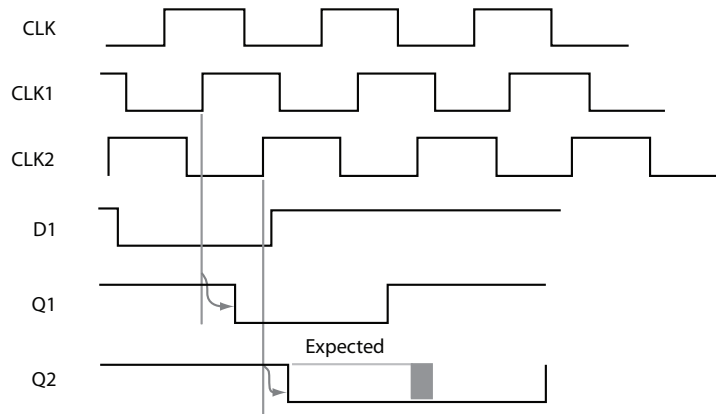

Figure 3 • **Clock Skew Circuit**

Figure 4 • **Clock Skew Sample Waveforms**

Figure 4 provides pictorial representation of the clock skew sample waveforms. The following sections discuss on how to analyze clock skew, short paths, and identify potential problems.

Measuring Clock Skew using the Timer

The first step in coping with clock skew problems is to measure the clock skew. You must perform a static timing analysis of the design after place-and-route to determine the amount of the clock skew.

For SX-A, RTSX-S, eX, Axcelerator®, RTAX-S, ProASIC, and ProASIC^{PLUS} the timer can generate a setup and hold-time violation report for register-to-register paths in the same clock domain. You can generate the report by opening Timer from Microsemi Designer software and going to **File** > **Tool** > **Report Violation**.

The timing violation report is only valid if you have specified one or more clock constraints. If the design clocks are not constrained, the report will be empty. The timing violation report has the following four sections:

- Header: This section contains software version, design name, operating condition, device type, speed grade, and Timer preferences.
- Clock Constraint Violation: This section reports the critical paths limiting any clock frequency constraint set in the **General** tab window.
- Max Delay Constraint Violation: This section reports the critical paths limiting any **Max Delay** constraint set in the **Timer Path** tab window.
- Min Delay Constraint Violation: This section lists short data paths that are susceptible to hold-time violations.

In the timing violation report, the skew of the clock network is taken into account in calculating the slack. The report is sorted by slack for each section; a negative slack indicates a violation.

The timing violation report is created based on the operating conditions set in the timer preferences. Therefore, to examine the long data paths versus any clock or Max Delay Constraint, you must export the report while the Timer preferences are set to worst case/long paths. On the other hand, to identify all the possible hold-time violations, the report should be created while the Timer preferences are set to best case/short paths. After each change in the operating conditions in the Timer window, the "calculate delays" option should be selected before exporting the timing violation report.

Timer, the static timing analysis tool in Microsemi's Designer software, can add specific path sets, which include the paths from the clock source (external or internal) to the CLK input of the registers. Note that the maximum difference of the path delays reflects the worst-case skew of the clock network. However, this is not necessarily the worst-case clock-skew/short-path relationship since all the registers on the clock network may not be sequentially adjacent.

In Designer versions beginning with R1-2003 SP1, the Timer tool calculates and reports the clock skew of each register-to-register path (the skew between the source and sink registers). This is done in the expanded path window for each path. In the previous Designer software releases, one basic set of paths needed to be added to the Timer path sets to extract the clock skew. The set should include all the paths starting (source) from the clock signal driver and ending (sink) in all the registers driven by that clock. The path can be defined through the GUI using filter keywords or through the Timer keywords using the **Advanced** tab of the Add Path Set GUI. For more information on the Timer Keywords and GUI, refer to the *Timer User's Guide* and online help.

As an example, consider an external clock, named "CLK_IN," driving a set of registers in a design. The following keywords will define the source and sink points of the desired paths to identify the clock skew:

- From: $inputs(CLK_IN)   To: $clocks(CLK_IN)[*CLK]

The above keywords will set up a path set from the CLK_IN input to the CLK ports of all the relevant flip-flops. Since this document discusses the short-path problems, you must set the Timer preferences (**File > Preferences**) to Best Case in order to view the shortest paths (Figure 5).

Clock Skew and Short Path Analysis

As mentioned earlier, clock skew and short-path problems emerge when the data propagation path delay between two sequentially adjacent flip-flops is less than the clock skew between the two. Figure 6 is a general diagram of the delay blocks in a sample circuit.
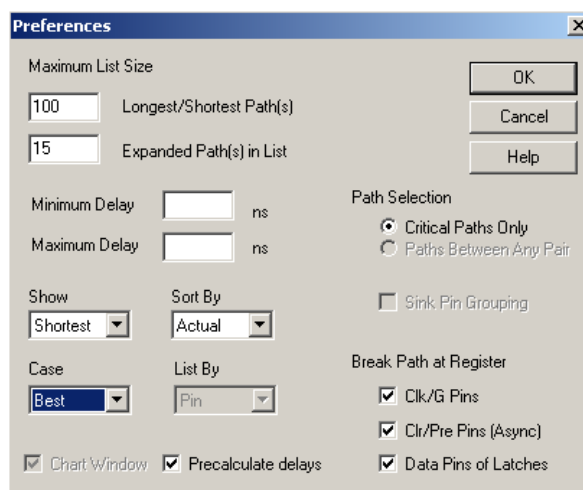

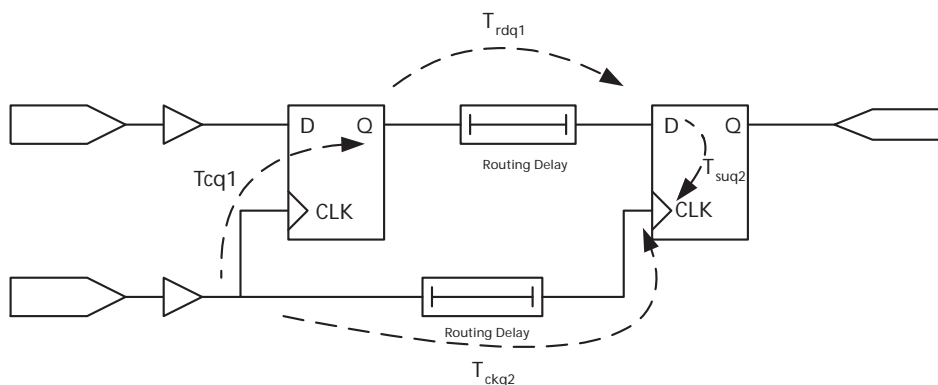*Figure 5 •* **Setting Shortest Paths and Best Case in Timer**


Figure 6 • **General Delay Blocks in a Simple Circuit**

The delays in Figure 6 on page 4 are as follows:

- tCQ1: The clock to out delay of the first flip-flop
- tRDQ1: The propagation delay from the output of the first flip-flop to the input of the second one
- tCK2: The clock arrival time at the second flip-flop minus the clock arrival time at the first flip-flop
- The short-path problem will definitely emerge in this circuit if

$$tCK2 > tCQ1 + tRDQ1 - tHOLD2$$

EQ 1

Where tHOLD2 is the hold-time requirement of the sink flip-flop. The regions are illustrated in Figure 7.
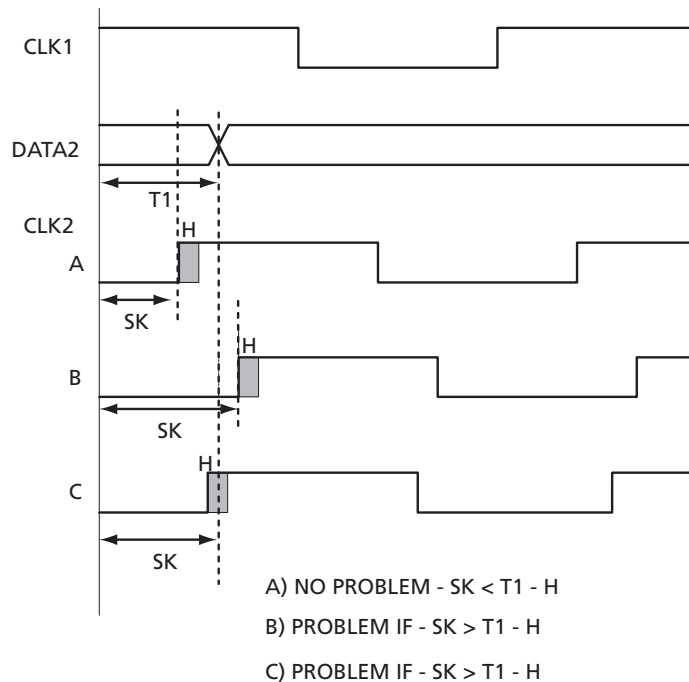
Figure 7 • **Illustration of the Short-Path Problem**

In practice, it is best to use EQ 1 to identify potential problems and violations.

Therefore, in order to identify the paths with the problem, you need to extract the clock skew (for example tCK2) and the short-path delays (for example tCQ1 + tRDQ1 - tHOLD2).

The clock skew extraction method was discussed in the "Measuring Clock Skew using the Timer" section on page 3.

Minimizing the Clock Skew

The short-path problem is created by the existence of an unacceptably large clock skew. Therefore, minimizing (which means nearly removing) the clock skew is the best approach to reduce the risk of short-path problems. Microsemi devices offer global routing resources, which reduce skew.

If there are any free global resources available on the device, you must assign their clock signals to these resources. Maintaining the clock skew at a value less than the smallest register-to-register delay in the design by using low-skew global resources improves the robustness of the design against any short-path problems. The hardwired H-clocks and quadrant clock resources (HCLK, QCLKA, QCLKB, QCLKC, and QCLKD) in A54SX-A, RTSX-S as well as global resources (GL) in Microsemi's Flash FPGAs are designed to prevent design errors due to clock skew. The skew will always be less than the shortest possible data path. Routed global networks R-clocks such as CLKA and CLKB offer reduced clock skew.

However, Microsemi recommends adding design margin since all possible configurations of routing and clock loading cannot be accurately characterized.

When designing for RTSX-S/SU, you must follow the steps below:

1. Use H-clocks (hard-wired clocks) at the maximum extent.

2. R-clocks (routed clocks) driving only sequential loads (R-cells) may be used with RTSX-SU (UMC). Our experimental data on minimum hold-time does not include MEC devices, so RTSX-S (MEC), R-clocks driving only sequential loads (R-cells) should be used only if functional testing is performed at all operating condition corners.

   – If R-clocks must be used, the use of combinatorial loads (C-cells) should be avoided. The timing model does not capture higher loading effect on clock skew in min-delay verification (hold- check).

   – If R-clocks are used with combinatorial logic loads, designers must plan for adequate on-silicon testing for hold violations.

3. Clock networks must not be created using regular routing resources.

To verify that R-clocks do not drive C-cells, you can export the CC flip-flop report from the Designer software tool. These reports list all the flip-flops that are formed by C-cells.

Note: The reports do not show the C-cells that do not form flip-flops such as regular buffer. Therefore, you would have to look into Netlist Viewer for any C-cell buffer on R-clock path.

The summary report shows the actual CC flip-flop count at the bottom of the report. The extended report shows the names of all the CC flip-flops (with macro name, instance name, and clock net name) in the design. You can generate these reports by going to:

Designer >Tools Menu > Reports > Resources > Flip Flop > Summary/Extended

For more information on the available global resources on each device family, please refer to the following documents provided on the *Microsemi SoC Products Grup Website*:

• *Global Clock Networks in Actel Antifuse Devices*

• *Using Global Resources in Axcelerator Family*

• *Efficient Use of ProASIC Clock Trees*

Microsemi's Designer software version R1-2003 with Service Pack 1 and later releases support clock skew analysis for RTSX-S, eX, ProASIC, ProASICPLUS, and Axcelerator. For future software updates, please refer to the software or service pack release notes.

Design Techniques for Limited Global Resources

In cases where designs include multiple clock domains, there may not be enough low-skew global resources in the targeted FPGA for all the external/internal clock signals. Therefore, regular routing resources and buffers are used to build clock trees for the clock network. Since using the regular nets for clock signals may cause noticeable clock skew, you must employ specific design techniques to reduce or eliminate the risk of a short data path problem. The following sections describe well-known design techniques used to make designs more robust against clock skew.

Add Delay in Data Path

Figure 6 on page 4 suggests that increasing the tRDQ1 (routing delay in the data path), and consequently, increasing the total delay of the data path to a value greater than the clock skew, will eliminate the short path problem.

To insert delay in the data path, you can employ BUFD or INVD macros from Microsemi's library as delay elements. For more information on usage of these macros, refer to the *Using the BUFD and INVD Delay Macros* application note.

Figure 8 on page 7 shows a simple example of BUFD insertion in the data path.

The amount of the inserted delay (number of BUFD or INVD macros) in the data path should be large enough so that the data path delay becomes sufficiently greater than the clock skew. Manual placement of the buffer can significantly improve the added delay. Microsemi recommends positioning the buffer so that it is not adjacent to either the source or receiving register. This prevents using fast routing tracks.

Clock Reversing

Clock reversing is another approach to get around the problem of short data paths and clock skew. In this method, the clock signal arrives at the clock port of the sink (receiving) register sooner than the source (transmitting) register. Therefore, the receiving (sink) register clocks in the transmitting (source) value before the transmitting register receives its clock edge. Figure 9 shows a simple example of implementing the clock reversing approach.

The BUFD macro, shown in Figure 9, adds delay to the clock driving the source register. When sufficient delay is inserted, the receiving register receives the active-clock edge before the source register.

The clock reversing method is not effective in circular structures such as Johnson counters and linear feedback shift registers (LFSRs). This is because it is not possible to define the sink register explicitly. Figure 10 on page 8 shows an example of a circular structure with clock reversing interconnection and shows that the short-path problem exists between flip-flops U1 and U3.
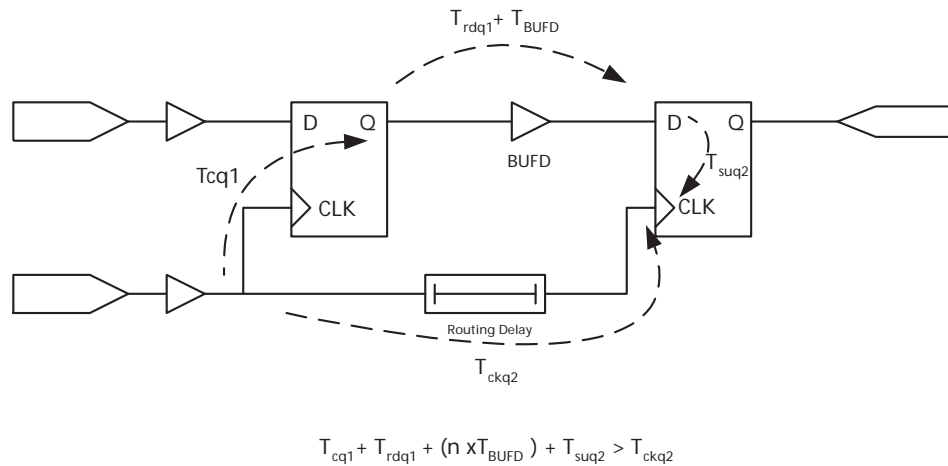

$$T_{cq1} + T_{rdq1} + (n \times T_{BUFD}) + T_{suq2} > T_{ckq2}$$

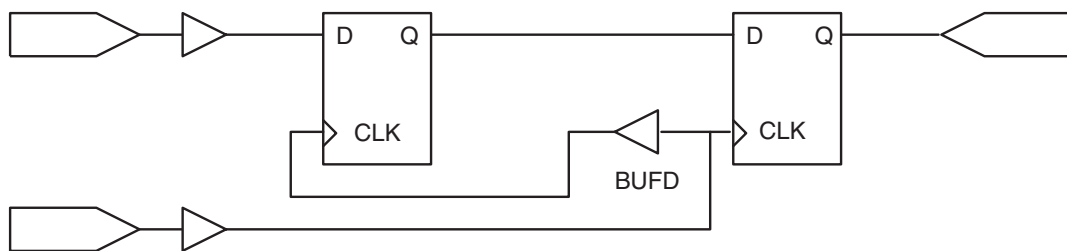*Figure 8* • **BUFD Delay Insertion in the Data Path**

Figure 9 • **Clock Reversing Methodology**
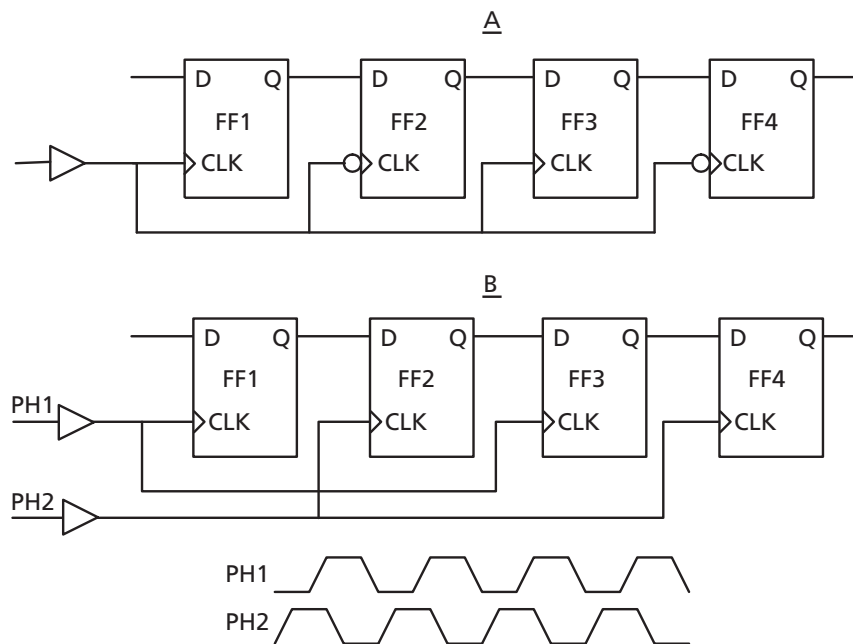
Alternate Phase Clocking

One of the known methodologies to avoid clock skew issues is alternate-phase clocking. The following are the most common methods of alternate phase clocking:

- Clocking on alternate edges
- Clocking with two phases

In the first method, sequentially adjacent registers are clocked on the opposite edges of the clock. Part A of Figure 11 shows an example of alternate sequentially adjacent registers being clocked on opposite edges. As can be seen in part A of Figure 12 on page 9, this method provides a short path-clock skew margin of about one-half clock cycle for clock skew.
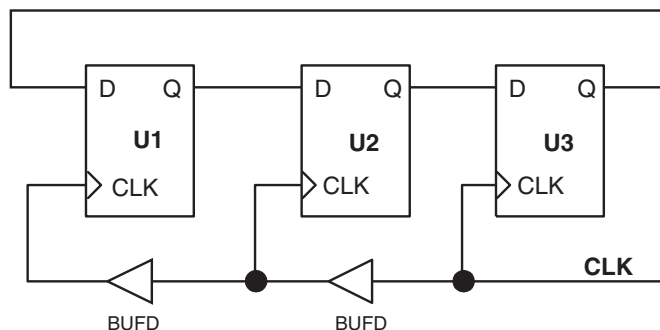
*Figure 10* • **Clock Reversing in a Circular Structure**

Figure 11 • **Alternative Edge/Phase Clocking**

Part B of Figure 11 on page 8 shows a set of adjacent registers that are alternately clocked on two different phases of the same clock. In this case, between the two adjacent registers, there is a safety margin approximately equal to the phase difference of the two phases.

The risk of the short-path problem is extremely low in these cases since in each data path there is at least one register (clocking on the alternate edge/phase) in the data path delay. Figure 12 shows an example of the signal waveforms in part A of the Figure 11 on page 8 circuit.
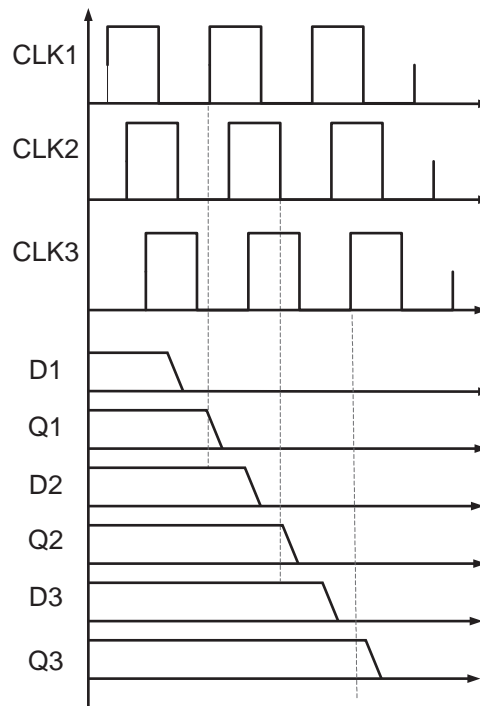
Figure 12 • **Signal Propagation for the Circuit in Figure 11**

You must note that the usage of alternate-phase clocking may require completely different clock constraints on the original clock signal. For example, in case of clocking on alternate edges, the new constraint on the clock frequency will be half the original frequency since the adjacent registers are clocked on opposite edges of the same clock cycle. Figure 10 on page 8 shows another example of opposite-edge clocking. In this example, the up-counter increments its output at the falling edge of the clock while the updated value is registered to the next level of logic on the rising edge of the clock.

Ripple Structures

In a ripple structure, each register output drives the next register clock port. Therefore, the sink register will not clock unless the source (driver) register is toggled. Figure 11 on page 8 shows an example of a three-bit ripple-down and ripple-up counter.

The output of each counter register drives the clock port of the next register instead of its data input port. This eliminates the clock skew as the registers do not toggle on the same clock.

The first register is clocked on the positive edge of the CLK signal and the second- and third-stage registers are clocked on the positive edge of the output of the previous register. Figure 11 on page 8 shows sample waveforms of the three-bit ripple counter output.
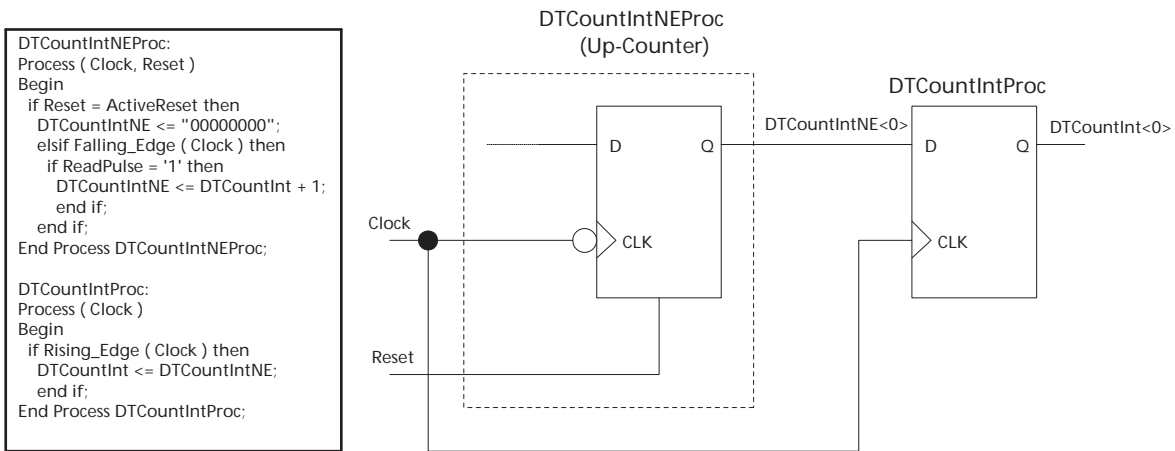
```
DTCountIntNEProc:
Process ( Clock, Reset )
Begin
 if Reset = ActiveReset then
   DTCountIntNE <= "00000000";
   elsif Falling_Edge ( Clock ) then
     if ReadPulse = '1' then
       DTCountIntNE <= DTCountInt + 1;
      end if;
   end if;
End Process DTCountIntNEProc;

DTCountIntProc:
Process ( Clock )
Begin
 if Rising_Edge ( Clock ) then
   DTCountInt <= DTCountIntNE;
    end if;
End Process DTCountIntProc;
```


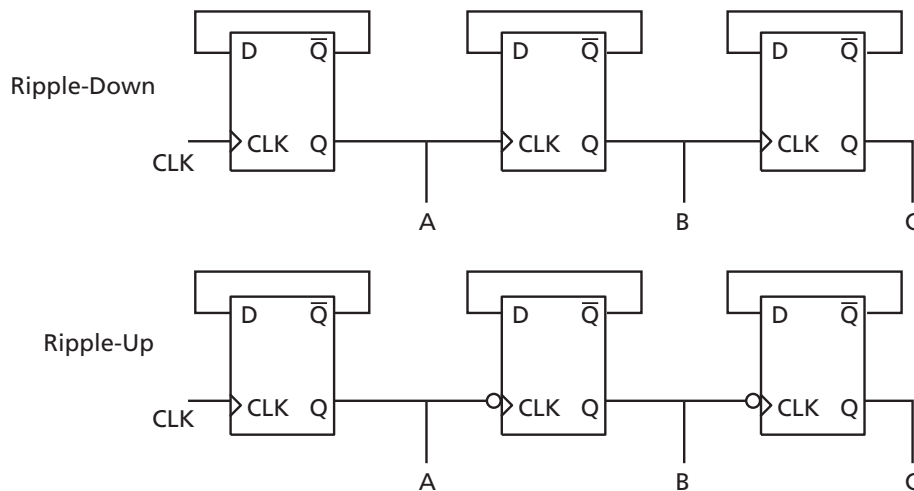*Figure 13 •* **Counter Design With Opposite-Edge Clocking**

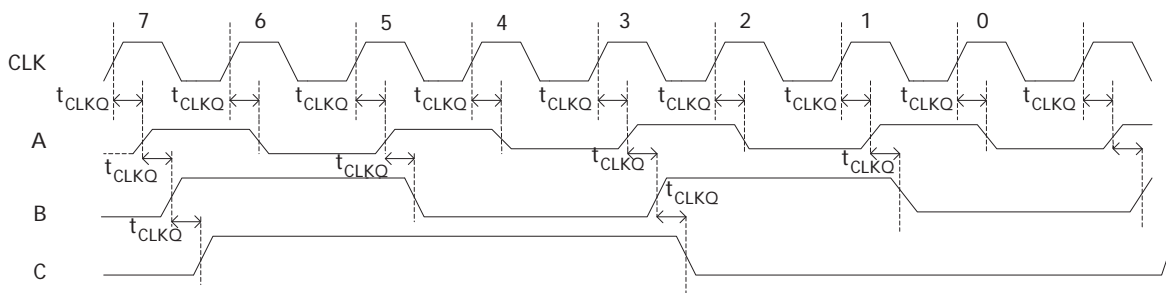*Figure 14 •* **Three-bit Ripple Counter**

Figure 15 • **Three-Bit Ripple Counter Input and Outputs**

Insight Into Minimum Delay

Design engineers tend to push the speed of the devices IC manufacturers provide, and emphasis is placed on improving the performance of devices through innovative design techniques and careful timing analysis. Maximum delays in the worst-case operating conditions are used to calculate the fastest speed at which a device can operate. Therefore, all device manufacturers screen devices to ensure the maximum delays are equal to or better than the posted performance. Manufacturers also tend to provide devices that are guaranteed to exceed the standard timing. Microsemi gives devices a speed grade rating such as -1, -2, or in some cases -3 for the fastest possible device. Since great emphasis is placed on satisfying the maximum delay values, manufacturers do not typically specify the minimum delay values. In other words, manufacturers guarantee devices to be faster than X, but do not guarantee that devices will be slower than Y.

This poses an interesting problem for designers who must satisfy timing parameters for both maximum and minimum delays. Since maximum delays are guaranteed, normal delay extraction and timing analysis highlights any potential timing issues for maximum delays. Some designers address the minimum delay issue by using the minimum delays from the fastest possible speed grade. Some multiply a derating factor to the extracted data to ensure added margin. While both of the above techniques improve the reliability of the design, the best ways to avoid functional errors are to use design techniques that do not depend on the minimum delay values for proper functionality OR use dedicated hardwired resources such as the HCLK network in SX-A and RTSX-S devices. Several of the design techniques were outlined in the previous section, while the HCLK, QCLK and GL networks are designed to be free of errors due to clock skew.

Conclusion

The difference in the arrival times of the clock signals between two sequentially-adjacent registers (clock skew) may cause the design to malfunction when the data paths are short. The simplest method to help prevent the short data path problem is to minimize the clock skew by using the low-skew global routing resources for clock signals. Microsemi devices provide various types of global routing resources that significantly reduce skew.

For designs that have a large number of clock domains, you may run out of global routing resources and be forced to build a clock tree using regular nets to distribute a clock signal. This introduces clock skew and possibly short data path problems. Various design techniques, such as inserting data path delays or clock reversing, can be used to reduce the risk of the design malfunction caused by clock skew. However, due to the limitations and shortcomings of each design technique, Microsemi recommends that you must fully exploit the available global routing resources of the selected device.

Microsemi's Designer software version R1-2003 with service pack 1 (and later releases) supports clock skew analysis for supported FPGA families. For future software updates, please refer to the *Designer software or service pack release notes*.

Related Documents

Application Notes

Global Clock Networks in Actel Antifuse Devices

www.microsemi.com/soc/documents/GlobalClk_AN.pdf

Using Global Resources in Axcelerator Family

www.microsemi.com/soc/documents/AX_Global_Resources_AN.pdf

Efficient Use of ProASIC Clock Trees

www.microsemi.com/soc/documents/A500K_Clocktree_AN.pdf

Using BUFD and INVD Delay Macros

www.microsemi.com/soc/documents/BUFD_INVD_AN.pdf

User's Guide

Timer User's Guide

www.microsemi.com/soc/documents/smarttime_ug.pdf

List of Changes

The following table lists critical changes that were made in each revision of the document.

Revision*	Changes	Page
Revision 1 (June 2011)	Updated "Clock Skew" section.	1
	Updated "Clock Skew and Short Path Analysis" section.	4
	Updated "Measuring Clock Skew using the Timer" section.	3
	Updated Figure 7.	5
	Included "Insight Into Minimum Delay" section.	11

Note: *The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.