

Static Timing Analysis Using Designer's Timer¹

Introduction

Static timing analysis is an important step in analyzing the performance of a design. Timer is Actel's static timing analysis tool incorporated in Designer software. Timer allows both pre-layout and post-layout static timing analysis and constraints setting.

Generally, static timing analysis is much faster than timing-driven gate-level simulation and does not require stimulus vector generation. Therefore, unlike dynamic analysis, the quality of the static approach is independent of the quality of stimulus vectors. However, proper functionality of the design cannot be checked in static analysis.

An accurate and efficient static timing analysis has many benefits, such as providing quick and efficient information to enhance the design performance and easing the design debugging procedure. This application note presents the most important concepts and techniques of static timing analysis and contains practical examples.

Background

Actel's Designer software includes the static timing analysis tool called Timer. The timing information is accessible after compiling the design netlist in Actel's Designer software. However, the final and accurate timing information will be available after place-and-route. Detailed information on Timer is available in the [Timer User's Guide](#). The Designer software release notes contain Timer updates and are located on Actel's website. This section presents the main features of Timer and their functionality.

Figure 1 represents the main (or summary) window of Timer. In the *Summary* section, Timer reports maximum clock frequency, input to register delay, register to output delay, and input to output port delay for best, worst, and typical cases based on the user preference.

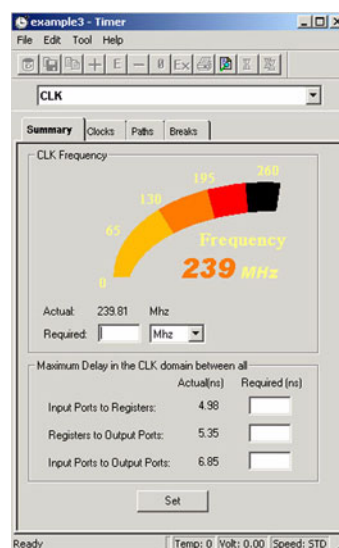


Figure 1 • Timer Summary GUI

1. The contents are based on the Designer release R1-2002 Service pack 2.

Figure 2 shows the *Clock* tab section of the Timer GUI. In this section, the user can apply constraints on the clock or define paths that are not required to meet the timing constraint. For example, users can identify their multi-cycle or false paths in this section.

One of the most useful sections of Timer is the *Path* tab (Figure 3). In this section, Timer reports the detailed path delays for default or custom path sets. The default path sets are input-to-output, input-to-register, register-to-register and register-to-output for the specified clock domain. Additionally, one can define customized paths for a more focused and detailed analysis. Each path in a set can be expanded. In the expanded path report, the delay components in each path (nets or gates) are listed along with a schematic representation of the path.

Breaks is another interactive part of Timer (Figure 4 on page 3). Under this tab, the user can put breaks on a different group of paths to be excluded from timing calculations or define paths to pass into the calculations.

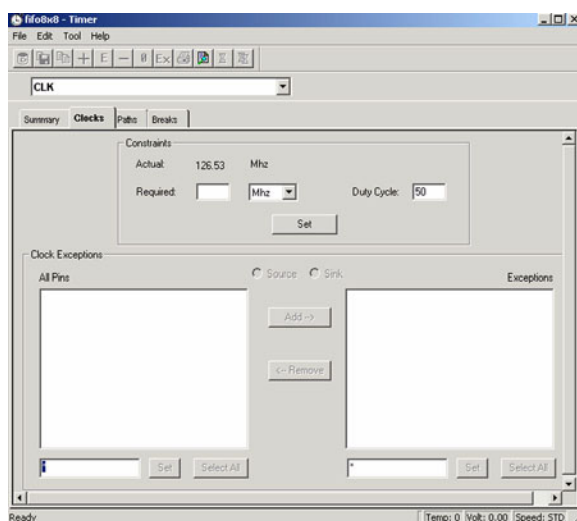


Figure 2 • Clock Tab Section of Timer

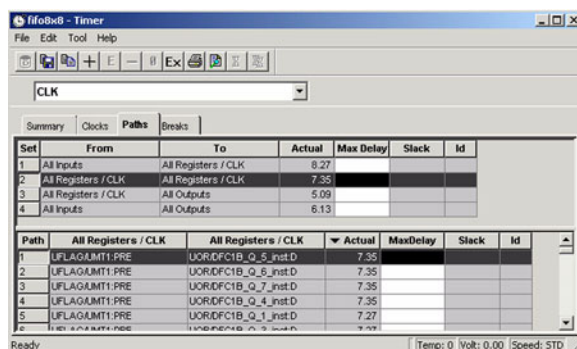


Figure 3 • Paths Tab GUI of Timer

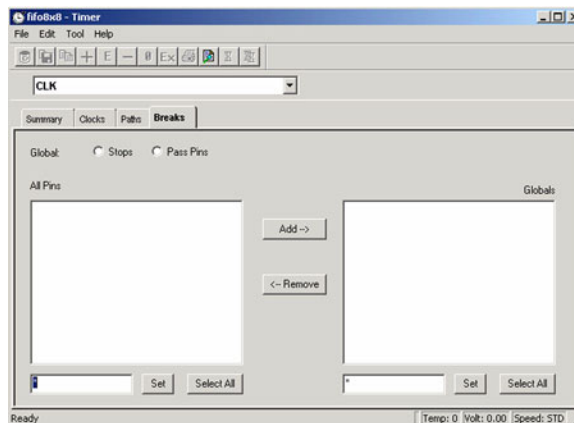


Figure 4 • Breaks Tab GUI of Timer

Path Delay Calculation and Report Methodology

Timing Path Calculation Models

Two popular models for path delay calculation and report are lumped and pin-to-pin models. The main difference between the two models is how the paths and their delays are defined. In the lumped model, all the cell and interconnect delays are lumped together and assigned to a cell (or module) delay. The lumped model has many restrictions, one of which is problems in modeling more complicated modules. For example, SRAM or Full-Adder modules are difficult to model by this approach. Timer uses a pin-to-pin model for the path delay calculation of its FPGA families. The older FPGA families (SX and legacy products) use an input-to-input model where delay elements of a path simply consist of instances where the module, driver, and wire delays are added and assigned to an input pin of that instance. For newer devices (SX-A, RT54SX-S, eX, ProASIC, ProASIC^{PLUS}, and Axcelerator), the path delay is calculated and reported based on the real pin-to-pin model where all the gated and net delays are reported individually. The propagation delay of a module is reported as a single gate delay for that module. The loading of the source (driver) module of the particular net and the routing delay are added together and reported as the net delay for each individual net. The whole path delay is the sum of all the net and gate delays in the path. For example, Figure 5 shows the delays assigned to the input of the register path from PAD input to the INST_2 register.

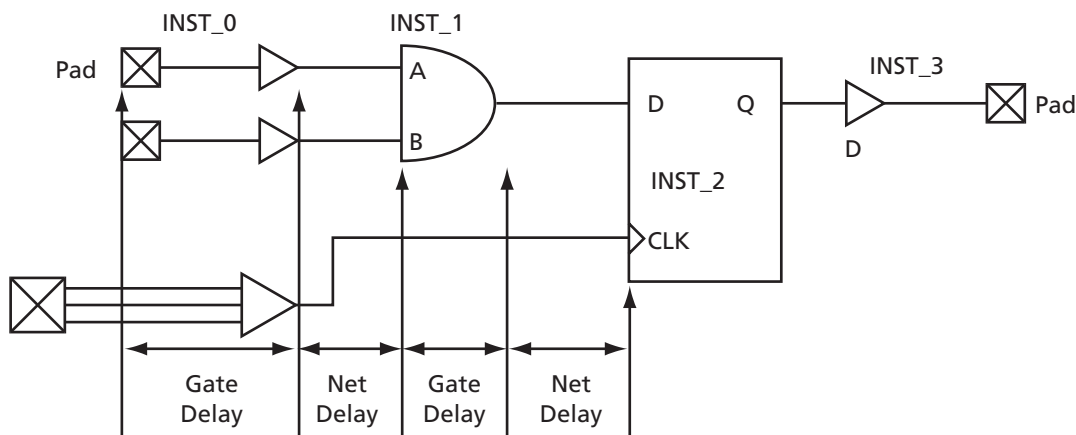


Figure 5 • Example of Pin-to-Pin Delay Calculation

Delay Report for Paths

The number of listed paths for each set is 100 by default. Using timing preferences, this number can be increased up to 200. The user can also set the preferences to have the report sorted by the shortest paths (ascending) instead of the longest (descending). In addition to this, the user can create a custom set of paths to include all the paths that need to be investigated or constrained.

Each path within a path set can be expanded to obtain more detailed delay information about the path. In the expanded path report, all the delay elements in the path (gates or nets) are listed. The expanded path window contains two subsets: a detailed path grid and a graphical representation. Timer employs the user's netlist for the timing report. If Actel's Designer software removes some modules, such as extra buffers during compilation, they will still be present in the timing report (except the designs targeting Axcelerator devices). However, the calculated (and reported) delay value for those macros will be zero. Therefore, any macro (or net) with 0 ns delay in the expanded path delay report represents a macro (or net) that exists in the user's netlist but has been removed (optimized) during compile in the Designer software. In the Axcelerator device design flow, Timer uses the post-compile (AFL) netlist. Therefore, the macros removed from the user's original netlist will not appear in the Timer paths.

If the expanded path is a register-to-register path, the set-up time requirement of the register will be included in the list and accumulated with the other delay components. However, it will not be incorporated in the displayed total path delay value in the main window. Therefore, in order to see the setup time requirement of each path, the user needs to expand the path.

Timer also takes into account the rising and falling edges of the clock signal. If a defined path is expanded, Timer will introduce two paths (if the two paths exist):

- The worst path between the source-pin and the sink-pin (rising edge)
- The worst path between the source-pin and the sink-pin (falling edge)

These paths are parallel and are the worst (longest) paths for each corresponding edge of the clock.

The user can set the preferences to observe the shortest or longest paths in a specific path set. [Figure 6](#) indicates the concept of the shortest and longest paths between two macros. In [Figure 6](#), if $D1(\text{max})$ is larger than $D2(\text{max}) + D3(\text{max}) + D4(\text{max}) + D5(\text{max})$, Path 1 will be reported as the longest path between A and B.

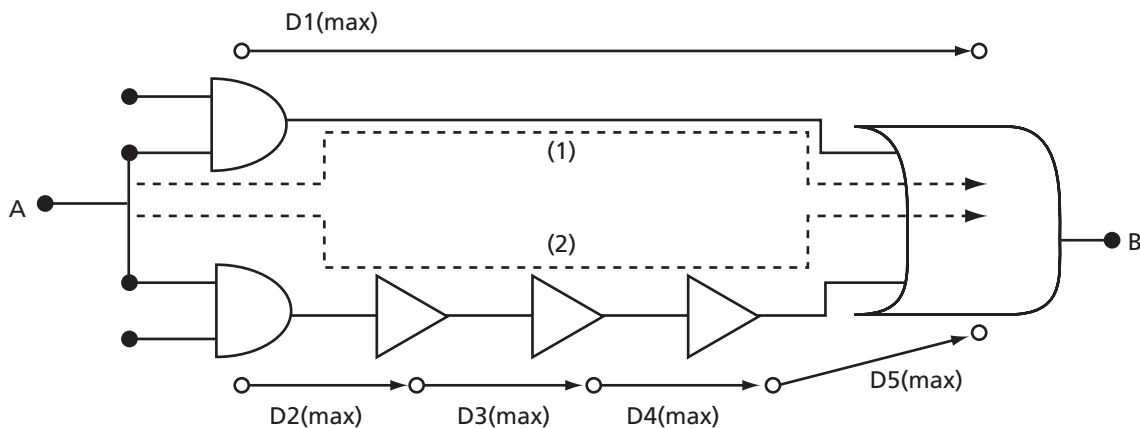


Figure 6 • The Shortest and Longest Paths

Maximum Clock Frequency Calculation

Timer lists the paths for each clock domain that are selected by the user. The maximum running frequency of each clock domain is calculated based on the maximum register-to-register delay of each clock domain. Timer picks the longest register-register path of each clock domain, adds the setup time requirement of the sink (destination) register, and considers it as the minimum clock period (corresponding to the reported maximum clock frequency reported by Timer).

The user can apply constraints on the clock frequency using the summary tab. Based on the user's clock period requirement, Timer calculates the maximum allowed register-to-register path delay based on the following equation:

$$\text{max reg-to-reg path delay} = \text{clock period requirement} - \text{setup time requirement} + \text{clock skew}$$

In the path tab, the calculated max reg-to-reg delay will be shown in the "max delay" column. If the path delay is more than the calculated max delay, the path will be marked as a constraints violation. The user should note that this would happen if there is any constraint on the clock frequency. If there is no constraint, the maximum allowed register-to-register delay will be calculated as the following equation and no violation will be reported:

$$\text{max reg-to-reg path delay} = \text{clock period requirement} - \text{setup time requirement}$$

Setup/Hold Time Calculation and Verification

Register setup and hold time verification is one of the most important steps of static timing analysis. Timer adds the setup time requirement of registers at the top of an expanded path delay if the user chooses to see the longest paths in the Timer preferences (Figure 6 on page 4). The register's internal setup time requirement is included in the maximum clock frequency estimation in Timer. However, it is not included in the cumulative path delay shown in the main path tab GUI unless the path is expanded. For example, the reported path delay under the corresponding set for the path in Figure 7 is 4.1 ns. The 0.8 ns setup requirement is not included in the path delay calculation.

The paths in which the setup time is violated are marked (Figure 8). Therefore, users should be careful when the maximum delay constraint is larger than the actual delay of the path only by a time margin equal to the setup time of a single register.

Actel's dedicated flip-flops have a hold time requirement of zero. The hold time of the registers will be reported in the expanded path delay if the user selects the "Shortest Path" analysis in Timer preferences. The hold time of the register will be added to the expanded path delay, as shown in Figure 9.

GridID	Instance	Net	Macro	Delay	Type	Total	Fanout
1	INST_TOP_TEBUNST_P	INST_TOP_TEBI	DFC1B	0.80 (t)	Tsd	4.90	0
	INST_TOP_TEBUNST_P	INST_TOP_TEBI	MX4	0.00 (f)	Tpd	4.10	1
	INST_TOP_TEBUNST_P	N309	DFC1B	4.50 (r)	Tpd	4.10	1
	INST_TOP_TEBUNST_P	N309	DFC1B	-0.4 (f)	Psk	-0.40	768

Figure 7 • Expanded Path Delay

Path	All Registers / CLK_1	All Registers / CLK_1	Actual	Maxdelay	Slack	Id
1	DF_1_1:CLK	DF_1_3:D	5.8	5.00	-0.95	DEL4
2	DF_1_2:CLK	DF_1_3:D	5.7	5.00	-0.75	DEL4

Figure 8 • Register Setup Time Violation in the Path Delay Report

GridID	Instance	Net	Macro	Delay	Type	Total	Fanout
1	INST_TOP_TEBUNST_P	INST_TOP_TEBI	DFC1B	0.00 (f)	Thd	3.90	0
	INST_TOP_TEBUNST_P	INST_TOP_TEBI	MX4	0.00 (f)	Tpd	3.90	1
	INST_TOP_TEBUNST_P	N309	DFC1B	4.30 (f)	Tpd	3.90	1
	INST_TOP_TEBUNST_P	N309	DFC1B	-0.4 (f)	Psk	-0.40	768

Figure 9 • Register Hold Time in the Short Path Report

If needed, the verification of required hold time can be done manually as follows:

$$\text{minimum path delay} - \text{maximum clock skew} \geq \text{register hold time}$$

For example, in [Figure 10](#) (data path delays are derived from the shortest path of the best case analysis), if the maximum clock skew ($t_{RD_2} - t_{RD_1}$), derived from the longest path in best case analysis, is larger than 2.1 ns ($t_{CO}[\min] + t_{RD1}[\min] + t_{HD}$), then the hold time of the second register is violated. We will discuss the clock skew and its derivation in more detail in the section [“Clock Skew” on page 9](#). During the hold time calculation/verification, the user needs to assume a safety margin, since the minimum delay values are not guaranteed.

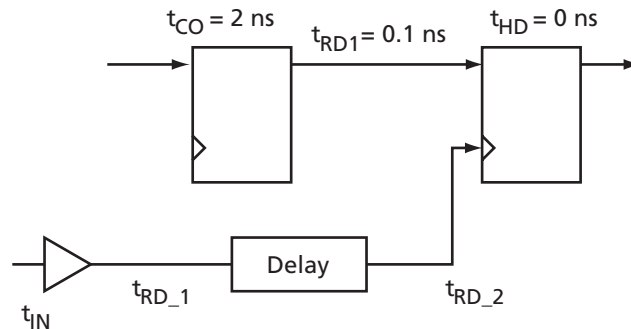


Figure 10 • Example for Hold Time Verification

For special registers or other special macros, such as TMR, CC, or any other macros that use more than one cell to be implemented, larger delays are expected. For these macros, setup, hold, removal, and recovery times become important and need to be checked carefully. Furthermore, the aforementioned timing requirements need to be investigated more carefully in the presence of a noticeable clock skew.

For SX-A, RTSX-S, eX, Axcelerator, RTAX-S, ProASIC, ProASIC^{PLUS}, and any other new Actel devices, Timer can generate a Setup and Hold Time Violation report for register-to-register paths in the same clock domain. Users can generate the report by opening Timer from the Actel Designer software, then going to File > Tool > Report Violation.

The Timing Violation report is only valid in the presence of a user-defined clock constraint. If the design clocks are not constrained, the report context will be empty. The Timing Violation report is formed in four sections, as follows:

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

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

The timing violation report is created based on the operating condition set in the timer preferences. Therefore, to examine the long data paths versus any clock or max delay constraint, the user should export the report while the timer preferences are set to worst case/long paths. On the other hand, to identify all possible hold time violations, the report should be created while the timer preferences are set to best case/short paths. Users should note that after each change in the operating condition in the Timer window, the Calculate Delays option should be selected before exporting the Timing Violation report.

The external setup and hold calculation and report will be discussed in more detail in the [“Analysis of Input and Output Delays” on page 11](#) and the [“Timing Report Interpretation” on page 12](#).

False Paths and Multi-Cycle Paths

False paths are identified as paths that do not propagate a signal. Part A in [Figure 11](#) shows an example of a false path. In this figure, the select inputs of two cascaded MUX cells are driven by the same logic. Therefore, the highlighted path will never propagate a valid signal.

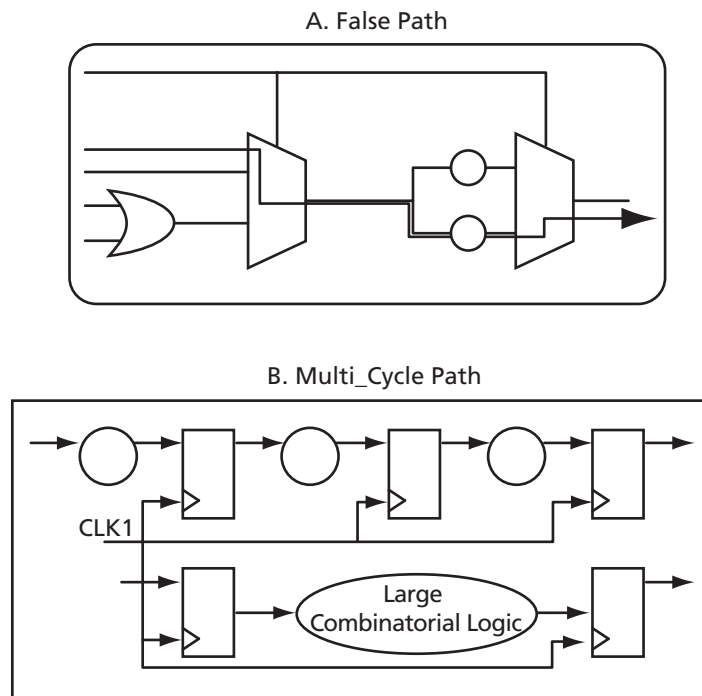


Figure 11 • False and Multi-Cycle Paths

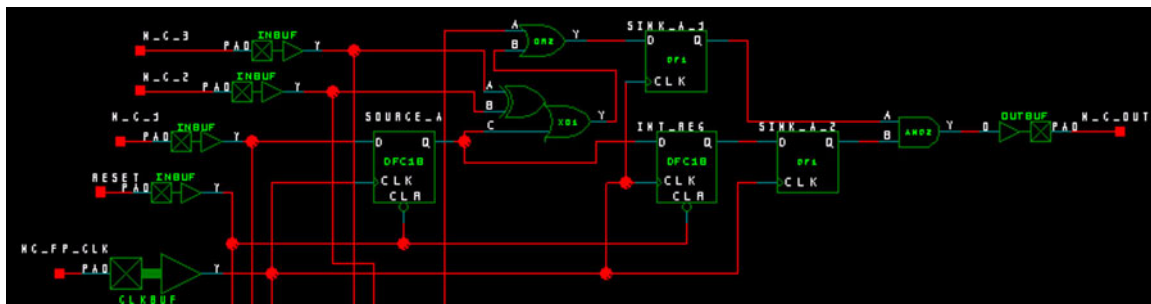
Part B in [Figure 11](#) shows multi-cycle paths. The upper path in part B in [Figure 11](#) contains more than two levels of registers, and therefore, it takes more than one clock cycle for the output to be valid. The lower path in part B in [Figure 11](#) shows another multi-cycle path, which contains only two levels of registers. The large combinatorial logic delay on the data path requires more than one clock cycle to propagate the valid data to the output. If users have these types of multi-cycle paths or false paths in their design, they should be declared so that they will not be constrained by the place-and-route tool, or show up as violations in your timing analysis. This will free routing resources to be used for other critical paths of the design.

[Figure 12 on page 8](#) shows an example of a long path that should be declared in Timer. For instance, in this figure, the path from register SOURCE_A to SINK_A_1 is a multi-cycle path. By default, the Timer GUI for this design will be similar to [Figure 12 on page 8](#).

As shown in [Figure 12 on page 8](#), the very first path in the register-to-register path set is a long path with a 1.73 ns delay. Timer reports the minimum clock period as {1.73 + setup time of SINK_A_1}. [Figure 12 on page 8](#) also shows the Timer GUI if this path is expanded.

The reported minimum clock period for this design will be 2.43 ns ([Figure 12 on page 8](#)). To eliminate this path from the report and obtain the clock period from the remaining single-cycle paths, one needs to put a break in an appropriate location on the path. By asserting a break (using the break tab window of Timer) on pins B or Y of the OR gate in the path, this long path will be broken and not reported in the Timer. [Figure 13 on page 8](#) shows the Timer GUI after inserting a break at the Y pin of the OR gate and committing the change.

As indicated in [Figure 13 on page 8](#), the minimum clock period is reported as 1.63 ns (0.93 ns + 0.7 ns).



Timer GUI
for the Design

File Edit View Tool Help						
Select Clock => MC_FP_CLK						
Summary Clocks Paths Breaks						
Set	From	To	Actual	Max Delay	Slack	Id
1	All Inputs	All Registers / MC_FP_CLK	3.16			
2	All Registers / MC_FP_CLK	All Registers / MC_FP_CLK	1.73			
3	All Registers / MC_FP_CLK	All Outputs	4.57			
4	All Inputs	All Outputs	No Path			
Path	All Registers / MC_FP_CLK	All Registers / MC_FP_CLK	Actual	Max Delay	Slack	Id
1	SOURCE_A:CLK	SINK_A_1:D	1.73			
2	SOURCE_A:CLR	SINK_A_1:D	1.60			
3	INT_REG:CLK	SINK_A_2:D	0.93			
4	SOURCE_A:CLK	INT_REG:D	0.93			
5	INT_REG:CLR	SINK_A_2:D	0.80			
6	SOURCE_A:CLR	INT_REG:D	0.80			

Expanded
Path

File Edit View Window						
Grid1	Instance	Net	Macro	Delay	Type	Total Fanout
1				0.70 (r)	Setup	2.43 0
	SINK_A_1:D	\$1N118	ADLIB:DF1	0.00 (r)	Net	1.73 0
	\$1I119:Y		ADLIB:OR2	0.00 (r)	Cell	1.73 1
	\$1I119:B	\$1N6	ADLIB:OR2	0.00 (r)	Net	1.73 0
	\$1I3:Y		ADLIB:XO1	0.90 (r)	Cell	1.73 1
	\$1I3:C	\$1N4	ADLIB:XO1	0.10 (r)	Net	0.83 0
	SOURCE_A:Q		ADLIB:DFC	0.73 (r)	Cell	0.73 2
	SOURCE_A:CLK	\$1N13	ADLIB:DFC	0 (r)	Net	0.00 0

Figure 12 • Example of a Multi-Cycle Path

File Edit View Tool Help						
Select Clock => MC_FP_CLK						
Summary Clocks Paths Breaks						
Set	From	To	Actual	Max Delay	Slack	Id
1	All Inputs	All Registers / MC_FP_CLK	3.16			
2	All Registers / MC_FP_CLK	All Registers / MC_FP_CLK	0.93			
3	All Registers / MC_FP_CLK	All Outputs	4.57			
4	All Inputs	All Outputs	No Path			
Path	All Registers / MC_FP_CLK	All Registers / MC_FP_CLK	Actual	Max Delay	Slack	Id
1	SOURCE_A:CLK	INT_REG:D	0.93			
2	INT_REG:CLK	SINK_A_2:D	0.93			
3	SOURCE_A:CLR	INT_REG:D	0.80			
4	INT_REG:CLR	SINK_A_2:D	0.80			

Figure 13 • Path Tab GUI after Breaking the Multi-Cycle Path

If the paths are broken, they will not be reported in the path sets. If the user needs to have the false or multi-cycle paths reported but not incorporated in the clock frequency calculations, he/she needs to define them as clock exceptions. The clock exception definition is done under the "Clocks" tab. Any path (identified by its source (driver) or sink (termination)), which is added to the "Exceptions" window, will be excluded from the clock/constraint calculation.

Clock Signal Analysis

Clock signals are among the most critical parts of a design performance. A better understanding of static analysis of clock signals and their interaction in the Timer tool will help users to perform a more accurate and intelligent overall static timing analysis of their design.

Clock Skew

Clock skew is an important factor in design performance; however, clock skew problems may be independent of operating clock frequency. Clock skew is a key factor in setup and hold time verification of the registers. A large clock skew could lead to hold time violations or incorrect register states. The following example shows how a large clock skew might affect the functionality of the design independent of the running clock frequency. Consider the simple circuit shown in part A of [Figure 14 on page 9](#).

If t_{SK} is larger than $t_{CQ} + t_{DLY}$, then the application of the clock and data inputs may result in the wrong register output as shown in [Figure 14 on page 9](#). Note that reducing the clock frequency will not eliminate the problem, indicated in [Figure 14 on page 9](#). There are different solutions, such as adding a delay to the data path or reducing the clock skew by using Actel's global networks to tackle similar problems. However, the first step to approach the problem is estimating the clock skew.

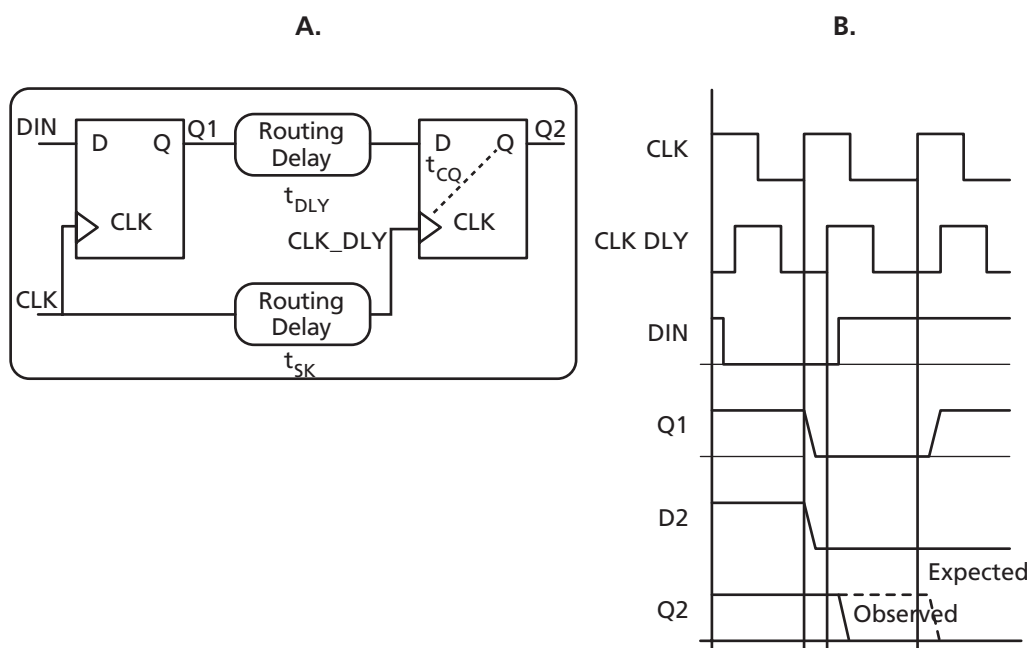


Figure 14 • Clock Skew and Improper Functionality

To calculate the skew for a specific clock domain, the first step is to extract the delay from the clock input (if the source is an external clock) to the clock input of all the registers in that clock domain. To do so, the user needs to define and add an input-to-register path set from the clock input to the CLK port of all the registers. If the clock is generated internally (such as PLL output), the user can define register-to-register paths where the source register output is the generated clock. We will discuss the internal clock resources under "Clock Drivers". Timer keywords can be used to create the appropriate sets. For more information

on the Timer keywords, please refer to the [Timer User's Guide](#). The difference between the maximum and minimum delay of the set is the clock delay of that clock domain. Note that the clock skew is not necessarily equal to the maximum clock delay. The difference between the arrival time of the clock to two registers can be taken as the skew, only if there is a valid data path between those registers.

The following is an example of a simple design with a clock input (CLK) driving two registers. In this example, the clock skew and the shortest data paths between the registers will be extracted to point out the critical paths of the design. The goal is to determine the clock skew between two registers which forms the shortest data path delay (to check for hold time violation). Setting the Timer preferences to show the shortest path in the best case, and using the following keywords in the "Advanced" window of the "Add path set" GUI, will result in the Timer GUI shown in [Figure 15 on page 11](#).

From:

To:

```
$clocks( ) [CLK]$registers(CLK):clockpins( )
```

(This will create a set from CLK input to clock input of all registers driven by CLK)

```
$registers(CLK):clockpins( ) $registers(CLK):$dp( )
```

The maximum clock skew can be obtained by comparing the shortest path delay from CLK input to the source registers and the longest path delay from CLK input to the sink registers of the path set number six in [Figure 15 on page 11](#).

Clock Drivers

In many cases, the clocks are gated for different purposes. For example, users might need to select between different clocks to drive a portion of the design through a MUX architecture. Moreover, the output of the registers might be used as clocks in order to divide the source clock frequency. Regardless of the purpose of gated or internally generated clocks, these clock resources should be identified in Timer for accurate static timing analysis.

Generally, any signal that drives the clock input of registers is considered by Timer as a potential clock signal. Therefore, the name of the clock signals might be different from the user-selected names because Timer may pick a different net (carrying the same logic) than the clock resource. The following example helps users to identify their clock resources for constraint application and analysis. [Figure 16 on page 11](#) shows an example design with multiple clock signals.

If the netlist for the design in [Figure 16 on page 11](#) is compiled in the Designer software, the following ports will be indicated as clocks in Timer:

DIVIDER_CLK: This is a regular I/O driving a regular INBUF and then the CLK port of the register FF1.

CLK_SEL: This input is the selection input of the MUX architecture, which drives the clock input of the FF4 and FF6 registers.

SEC_CLK_MUX: This input is also connected to the clock input of the FF4 and FF6. The user should note that the timing characteristics of this clock are similar to the CLK_SEL input.

FF1:Q: This is the output of the FF1 register. Any signal driving CLKINT (or QCLKINT) buffers will be considered as a clock by Timer. In this particular example, the output of the FF1 register drives the routed clock network (CLKINT buffer) and is connected to the clock input of the FF3, FF4, FF5, and FF6 registers (FF4 and FF6 clock inputs are driven through the MUX).

PLL Outputs

Actel provides embedded PLL macros in some of the device families. In the designs targeting those families, if the PLL macro is used, the output of the PLL is considered as the potential clock signal in the design. The users can apply timing constraints to the output of the PLL as a clock and run timing-driven place-and-route to meet the constraints. However, for ProASIC^{PLUS} devices, Timer does not take the PLL configuration information (such as output frequency and multiplication factors) as a pre-defined constraint. If necessary, the user needs to apply timing constraints on the output of the PLL to be able to achieve better timing performance on the PLL outputs.

Summary Clocks Paths Breaks						
Set	From	To	Actual	Max Delay	Slack	Id
1	All Inputs	All Registers / CLK	0.33			
2	All Registers / CLK	All Registers / CLK	0.00			
3	All Registers / CLK	All Outputs	1.13			
4	All Inputs	All Outputs	1.21			
5	\$clocks () [CLK*]	\$registers(CLK):\$cp()	0.80			
6	\$registers(CLK):clockpins ()	\$registers(CLK):\$dp()	2.15			

Path	From	To	▲ Actual	MaxDelay	Slack	▲
1	PICTLRAMrw/row_2:PRE	PICTLRAMrw/state_h/stateZ0Z_0:D	2.15			
2	PICTLRAMrw/row_6:PRE	PICTLRAMrw/state_h/stateZ0Z_1:D	2.15			
3	PICTLRAMrw/row_7:PRE	PICTLRAMrw/state_h/state_1Z0Z_1:D	2.15			

ready Temp: 0 Volt: 0.00 Speed: -3

Figure 15 • The Shortest Data Path in the Best Case

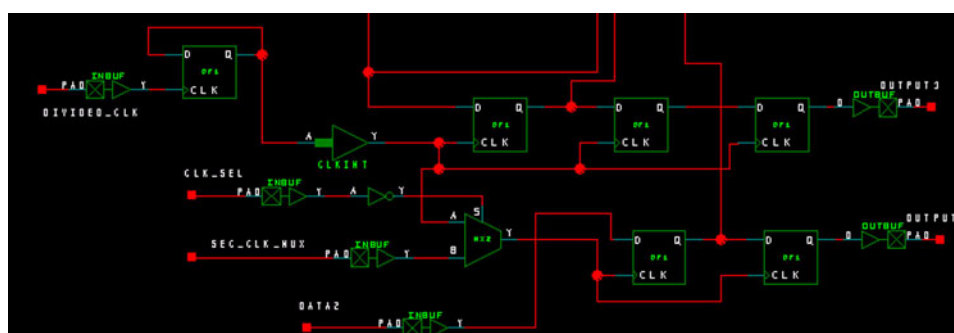


Figure 16 • Example of Multiple Clock Drivers

Analysis of Input and Output Delays

Two of the default path sets in the Timer path tab GUI are the "All Registers-to-All Outputs" and "All Inputs-to-All Registers" sets. In each clock domain, the path delays from the external inputs of the FPGA to the data input of the first level of the registers are reported under the "All Inputs-to-All Registers / <Name of the Clock>" set. The second path set is defined from the clock port of the registers, driving the FPGA outputs to the FPGA output pads.

The input-to-register delays are used to derive the external setup and hold time requirements of the design. The external setup and hold times are useful in both board-level and internal timing analysis. The external setup time is the maximum allowed arrival [to FPGA] time difference between the data and clock input to meet the setup time requirement of the first level internal registers. Similarly, the external hold time is the allowed arrival time difference between the inputs and the clock to meet the hold time requirement of internal registers. The external setup and hold time calculation is basically a comparison between the data propagation and clock propagation. In Figure 17, the external setup time can be obtained using the following:

$$\text{External Setup Time} = \text{External Clock Arrival Time} - \text{External Data Arrival Time} = \text{Delay I} + \text{Register Setup Time} - \text{Delay II}$$

As shown in Figure 17, the external input setup time can be either positive or negative. A negative external setup time means that, due to the large delay in the clock path, the data input can arrive at the FPGA (even after the active edge of the clock) and still meet the setup time requirement of the embedded registers.

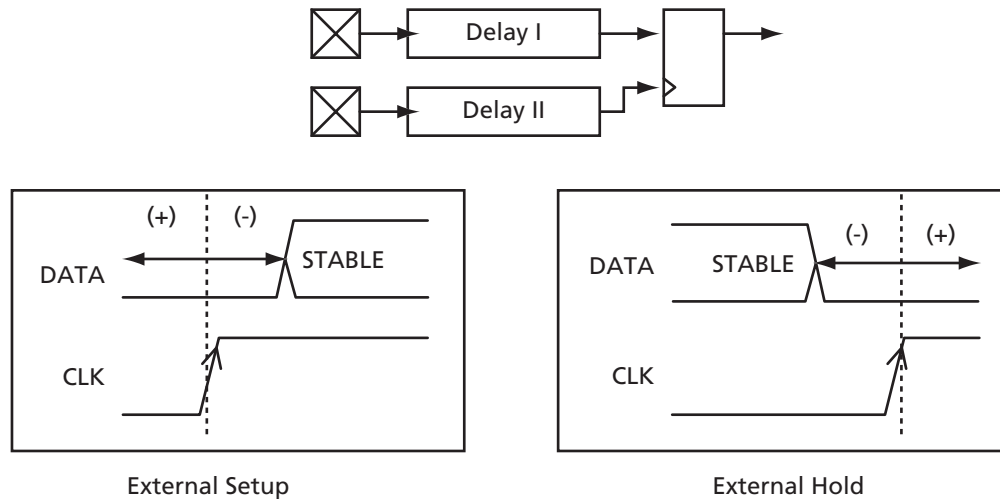


Figure 17 • External Setup and Hold Time

Similarly, the external input hold time is calculated as the following equation:

$$\text{External Hold Time} = \text{Delay II} - \text{Delay I} + \text{Internal Register Hold Time}$$

Again, it is possible to have negative or positive external hold time and meet the timing requirements.

The most convenient approach to obtain the external setup and hold timer requirement of the design is to export the timing report. This will be discussed in more detail in the ["Timing Report Interpretation" section](#).

Register-to-out path delays are used to derive the clock-to-out timing performance of the FPGA. Clock-to-out timing characteristics of a design are vital in order to examine the external setup and hold time requirement of the device derived by the FPGA outputs. In order to obtain the maximum clock-to-out of a specific output, the user can define (add) a path set, sourcing from all registers and sinking into that particular output. Please refer to the [Timer User's Guide](#) for instruction on how to add path sets.

Timing Report Interpretation

A sample timing report generated by the Designer software is presented in ["Appendix A" on page 14](#) of this document. Generally, the timing report is a simple text format interpretation of the Timer GUI and its contained information. The report starts with header expressions illustrating the design and the software setting information. After the header, the report lists all the potential clock domains of the design, their actual frequency, and the required frequency. The required frequency column lists the constraints that the user has implied on each clock. The report format is repeated for each clock domain of the design. In other words, for each clock domain, the report lists specified sets of information. Each set of information is included in a "Section." The definition of the type of information in each set is in the Timer keywords format. For example "Section \$input() to \$Registers (CLK_2):\$DataPins()", indicates that this section includes the timing information of the paths, starting from the design inputs and ending at the D port of the registers, driven by the CLK_2 clock signal. If an input or a register (in that particular clock domain) is not listed in this section, it means that there is no valid path between those points. For some sections, such as the clock-to-out information set, the longest path is also represented in an expanded format, indicating all the delay elements in that path.

The last, and one of the most important sections of the timing report, is the external setup and hold section. In this section, the Designer software lists all the external setup and hold times of the inputs with

respect to the design clock signals. For example, in the report attached in “Appendix A” on page 14, the user can find the following lines in the external setup and hold section:

Setup	Hold	Pins
1.09	0.35	From: INPUT_1 To: CLK_1
-0.03	0.42	From: INPUT_3 To: INPUT_1

These report lines mean that INPUT_1 must arrive to the FPGA at least 1.09 ns before the active edge of CLK_1 and stay valid until at least 0.35 ns after the edge of CLK_1 to meet the setup and hold requirements of the internal registers. Furthermore, INPUT_3 can even arrive up to 0.03 ns after the active edge of INPUT_1 and still meet the setup time requirement of an internal register driven by the INPUT_1 (INPUT_1 is driving the clock port of this register). Figure 18 on page 13 represents the calculated values for this example.

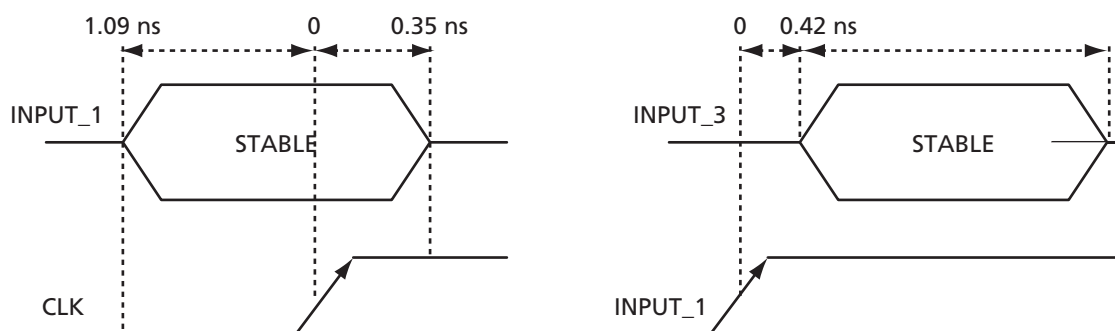


Figure 18 • Graphical Interpretation of External Setup and Hold Time of the Report in Appendix A

Conclusion

Static timing analysis is the most accurate approach to analyze the timing performance of a design before performing dynamic timing analysis through extensive testbenches. Timer, the static timing analysis tool located in the Designer software, is user-interactive and allows users to apply timing constraints on their design. Most of the required timing characteristics of a design are provided in Timer. However, the user needs to combine his/her timing analysis skills with the knowledge of the tool to extract different timing information, such as path delays, clock performance, external setup and hold time, and clock-to-out timing of the design. Different features have been embedded in the Timer tool to help users access the required information faster and more easily. Users need to refer to the Designer software release notes for the latest updates on the Timer tool. In sum, this document, combined with the *Timer User's Guide*, provides reference material for Actel static timing analysis.

Related Documents

User's Guides

Timer User's Guide

http://www.actel.com/documents/timer_UG.pdf

List of Changes

Previous Version	Changes in Current Version (51900002-1/01.04*)	Page
51900002-0/10.02	The "Setup/Hold Time Calculation and Verification" section was updated.	5

* The part number is located on the last page of the document.

Appendix A

Timer Version 01.01.01

Actel Corporation - Designer software Development System Release R2-2002 Development, ©1989-2002

Date: Wed Jun 12 18:07:06 2002

```
Design: F_M
Family: 54SXA
Die: A54SX32A
Package: 208 PQFP
Radiation Exposure: 0 KRad
Temperature: COM
Voltage: COM
Speed Grade: -3
Design State: Post-Layout
Timing: Worst Case
Path Tracing: Longest Paths
Break at Clk/G pins: True
Break at Preset/Clr pins: True
Break at Data pins of Latches: True
Section Clock Frequency
      Actual      Required      ClockName
      323.52MHz      N/A      CLK_1
      721.50MHz      N/A      CLK_2
```

End Section

Section \$Inputs() to \$Outputs()

```
Delay(ns) Slack(ns)      Pins
      3.65      N/A From: INPUT_1
                        To: OUTPUT_1
```

End Section

Section \$Inputs() to \$Registers(CLK_2):\$DataPins()

```
Delay(ns) Slack(ns)      Pins
      0.80      N/A From: INPUT_2
                        To: REG1:D
```

End Section

Section \$Inputs() to \$Registers(CLK_2):\$ClockPins()

```
Delay(ns) Slack(ns)      Pins
      2.39      N/A From: CLK_SEL
                        To: REG2:CLK
```

End Section

Section \$Inputs() to \$Registers(CLK_2):\$AsyncPins()

--- No Paths found ---

End Section

Section \$Registers(CLK_2):\$InputPins() to \$Outputs()

```
Delay(ns) Slack(ns)      Pins
      2.56      N/A From: REG2:CLK
                        To: OUTPUT2
```

End Section

Section \$Registers(CLK_2):\$ClockPins() to
\$Registers(CLK_2):\$InputPins(TmacEX_ASYNC_PINS)

Delay(ns)	Slack(ns)	Pins
0.69	N/A	From: REG1:CLK To: REG2:D

First Path Expansion from REG1:CLK to REG2:D

Total	Delay	Load	Type	Name
1.39	0.70	0	Setup	
0.69	0.15	0	Net	\$1N98
				REG2:D(R)
0.53	0.53	2	Cell	ADLIB:DF1
				REG1:Q(R)
0.00	0.00	0	Net	\$1N90
				REG1:CLK(R)

End Section

Section \$Registers(CLK_2):\$ClockPins() to \$Registers(CLK_2):\$AsyncPins()

--- No Paths found ---

End Section

Section \$Registers(CLK_2):\$AsyncPins() to \$Registers(CLK_2):\$InputPins()

--- No Paths found ---

End Section

The following section is NOT breaking on Clk/G pins of registers

Section CLK_2 to \$Outputs()

Delay(ns)	Slack(ns)	Pins
4.09	N/A	From: CLK_2 To: OUTPUT2

First Path Expansion from CLK_2 to OUTPUT2

Total	Delay	Load	Type	Name
4.09	0.00	0	Net	
				OUTPUT2(F)
4.09	1.96	1	Cell	ADLIB:OUTBUF
				\$1I93:PAD(F)
2.13	0.07	0	Net	\$1N100
				\$1I93:D(F)
2.06	0.53	1	Cell	ADLIB:DF1
				REG2:Q(F)
1.53	0.15	0	Net	\$1N90
				REG2:CLK(R)
1.37	0.69	2	Cell	ADLIB:MX2
				\$1I74:Y(R)
0.69	0.23	0	Net	\$1N77
				\$1I74:B(R)
0.46	0.46	1	Cell	ADLIB:INBUF
				\$1I75:Y(R)
0.00	0.00	0	Net	CLK_2


```

                                $1I75:PAD(R)
0.00      0.00      0      Net
                                CLK_2(R)
End Section
Section $Inputs() to $Registers(CLK_1):$DataPins()
  Delay(ns) Slack(ns)      Pins
      2.67      N/A From: INPUT_1
                        To: REG3:D
      1.85      N/A From: INPUT_3
                        To: $1I2:D
End Section
Section $Inputs() to $Registers(CLK_1):$ClockPins()
  Delay(ns) Slack(ns)      Pins
      2.42      N/A From: CLK_1
                        To: $1I1:CLK
End Section
Section $Inputs() to $Registers(CLK_1):$AsyncPins()
  Delay(ns) Slack(ns)      Pins
      0.87      N/A From: RESET
                        To: $1I1:CLR
End Section
Section $Registers(CLK_1):$InputPins() to $Outputs()
  Delay(ns) Slack(ns)      Pins
      3.34      N/A From: REG3:CLK
                        To: OUTPUT_1
End Section
Section $Registers(CLK_1):$ClockPins() to
$Registers(CLK_1):$InputPins(TmacEX_ASYNC_PINS)
  Delay(ns) Slack(ns)      Pins
      2.39      N/A From: $1I1:CLK
                        To: REG3:D
First Path Expansion from $1I1:CLK to REG3:D
  Total      Delay Load      Type      Name
    3.09      0.70      0      Setup
    2.39      0.00      0      Net      $1N118
                                REG3:D(R)
    2.39      0.69      1      Cell      ADLIB:OR2
                                $1I119:Y(R)
    1.70      0.38      0      Net      $1N6
                                $1I119:B(R)
    1.32      0.69      2      Cell      ADLIB:XO1
                                $1I3:Y(R)
    0.64      0.08      0      Net      $1N4
                                $1I3:C(R)
    0.56      0.56      1      Cell      ADLIB:DFC1B
                                $1I1:Q(R)

```

```

0.00      0.00      0      Net      $1N13
                                $1I1:CLK(R)

End Section
Section $Registers(CLK_1):$ClockPins() to $Registers(CLK_1):$AsyncPins()
    --- No Paths found ---
End Section
Section $Registers(CLK_1):$AsyncPins() to $Registers(CLK_1):$InputPins()
    Delay(ns) Slack(ns)      Pins
        2.29      N/A From: $1I1:CLR
                                To: REG3:D

End Section
The following section is NOT breaking on Clk/G pins of registers
Section CLK_1 to $Outputs()
    Delay(ns) Slack(ns)      Pins
        5.62      N/A From: CLK_1
                                To: OUTPUT_1

First Path Expansion from CLK_1 to OUTPUT_1
    Total      Delay Load      Type      Name
    5.62      0.00      0      Net      $1N13
                                OUTPUT_1(F)
    5.62      1.96      1      Cell      ADLIB:OUTBUF
                                $1I111:PAD(F)
    3.66      0.07      0      Net      $1N115
                                $1I111:D(F)
    3.59      0.68      1      Cell      ADLIB:AND2
                                $1I110:Y(F)
    2.91      0.07      0      Net      $1N113
                                $1I110:B(F)
    2.84      0.56      1      Cell      ADLIB:DF1
                                REG3:Q(F)
    2.27      1.59      0      Net      $1N13
                                REG3:CLK(R)
    0.69      0.00      0      Net      $1N13
                                $1I15:Y(R)
    0.69      0.69      1      Cell      ADLIB:CLKBUFR_0
                                $1I15/U0:Y(R)
    0.00      0.00      0      Net      PAD
                                $1I15/U0:PAD(R)
    0.00      0.00      0      Net      CLK_1
                                $1I15:PAD(R)
    0.00      0.00      0      Net
                                CLK_1(R)

End Section
Section External Setup and Hold
    Setup      Hold      Pins
    1.09      0.35      From: INPUT_1

```

Static Timing Analysis Using Designer's Timer

```

                                To: CLK_1
-0.03    0.42  From: INPUT_3
                                To: INPUT_1
    0.96    0.31  From: INPUT_3
                                To: CLK_1
-0.03    0.46  From: INPUT_2
                                To: CLK_2
End Section
```

Actel and the Actel logo are registered trademarks of Actel Corporation.
All other trademarks are the property of their owners.



<http://www.actel.com>

Actel Corporation

2061 Stierlin Court
Mountain View, CA
94043-4655 USA

Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.

Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom

Phone +44 (0)1276.401450
Fax +44 (0)1276.401490

Actel Japan

EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan

Phone +81.03.3445.7671
Fax +81.03.3445.7668

Actel Hong Kong

39th Floor, One Pacific Place
88 Queensway, Admiralty
Hong Kong

Phone 852.227.35712
Fax 852.227.35999