ProASIC and ProASIC^{PLUS} Timing Constraints

Introduction

Timing constraints are used to generate guidelines for synthesis and place-and-route tools to meet the required timing performance for a design. In the current version of Designer, the timing-driven flow for Flash devices is different than the flow for Antifuse families. In the Flash design flow, Timer does not forward the constraints set in Timer to Layout. Usually, timing constraints set in Timer are saved in the Design Constraint System (DCS) using the SDC format. For Flash devices, the timing-driven place-and-route requires you to apply the constraints in GCF format. Therefore, any constraints set in Timer must be converted to GCF format to be recognized by the place-and-route tool, and not all constraints set in the Timer GUI can be converted to an equivalent GCF format. The following sections describe the Flash timing constraints flow and how the user should use the Timer tool to set timing constraints for these families.

Timing Constraint flow for ProASIC^{PLUS} and ProASIC

After importing design netlist file and constraints in Designer, compile the design. Compile checks for netlist errors (bad connections and fanout problems), removes unused logic, and combines functions to reduce resource utilization and to improve performance. After Compile, the design is ready for Layout (place-and-route). Layout takes the modified netlist and constraint information and maps this information into the selected Actel device.

Actel Designer software for Flash families uses the gfimporter and gfTimer tools during Compile and Layout. In Designer, gfimporter receives the netlist and the GCF file and generates constraints-based inputs (both physical and timing) for place-and-route. When the place-and-route tool receives the timing information, it will try to meet those constraints. At the end of Layout, the place-and-route tool calls the gfTimer to generate a Design-am.sdf file, which in turn is used by Timer to get gate and net delay. The graphical view of this flow is shown in Figure 1 on page 2. As you can see, the Timer read-only Design-am.sdf file does not read the timing constraint from the GCF file and hence does not set the user timing constraints. As a result, no violation is marked in Timer if GCF timing constraints are not met. We recommend that the user open Timer and run timing analysis to verify whether timing requirements have been met.

Timing constraints can be set for ProASIC and ProASIC^{PLUS} families using a GCF file, SDC (Synopsys Design Constraints) file, forward-annotated SDF (Standard Delay Format) file or Timer GUI. The forward-annotated SDF file is generated by synthesis tools. Please see *Designer User's Guide* about using these constraint files in Designer. However, all these different types of constraints have to be converted to the GCF format in order to be used by Compile and Layout. This conversion is done by the Designer tool itself. The place-and-route tool considers these timing constraints and attempts to meet them.

The following sections will describe the basic GCF timing constraints and other Timing constraints, and their equivalent GCF constraints. Also, we will give the recommendation to the user about using these various constraints.

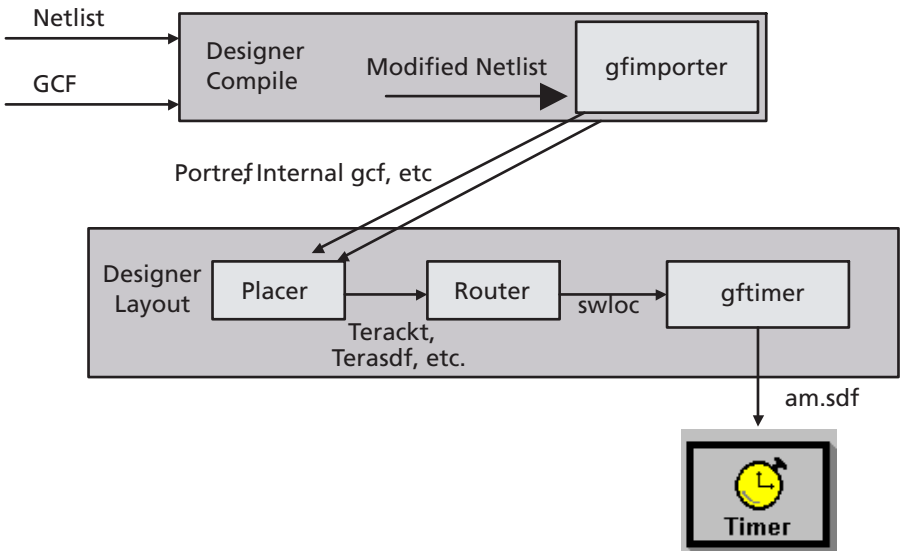

Figure 1 • **Designer Flow for ProASIC and ProASIC^PLUS**

GCF Timing Constraints

The GCF file must conform to the syntax described in the Designer online help. GCF may include various types of constraints:

- Timing constraints
- Global resource constraints
- Netlist optimization constraints
- Placement constraints
- I/O constraints

The following are the descriptions of the timing constraints in the GCF file.

create_clock

Defines clocks of the design. Multiple clocks can be specified for a given design.

```
create_clock -period <period_value> {netname|portname}
```

Where "period_value" is the clock period in nanoseconds and "netname|portname" is the name of the net through which the clocks gets propagated or the name of the external port.

For example, the following statement creates a clock on external port "clk" with a period of 25.0 nanoseconds.

```
create_clock -period 25.0 clk;
```

set_false_path

Defines false paths in the design. These paths will not receive priority during timing-driven place-and-route.

```
set_false_path [-from from_port] [-through any_port] [-to to_port];
```

Where "from_port" must be an input port of the design or output pin of a register or memory instance, "to_port" must be an output port of the design or input pin of a register or memory instance, "any_port" must be any instance pin. Wildcards are permitted.

For example, the following statement sets all paths starting from "resetd" that are going through instance "const2" as false paths:

```
set_false_path -from resetd -through const2/*;
```

set_input_to_register_delay

Defines the timing budget for incoming signals to reach a register:

```
set_input_to_register_delay <delay> [-from inp_port];
```

Where "delay" is the timing budget for this input path, "inp_port" is a register or memory instance output pin. Wildcards are permitted.

For example, the following statement specifies that the timing budget is 22 nanoseconds to the register from all inputs whose names are starting with the letter "I":

```
set_input_to_register_delay 22 -from I*;
```

set_multicycle_path

Defines multi-cycle paths along with the number of required cycles. The budget of these paths will be multiples of the clock period for the ports connected to the "from" port.

```
set_multicycle_path <num_cycles> -from reg_port [-through_ any_port] [-to_port];
```

Where "num_cycles" is the number of clock cycles in which the signal needs to propagate through the path, "reg_port" is the port with register or memory instance (u1/dff1.q), "to_port" must be an output port of the design or an input pin of a register or memory instance, "any_port" must be any instance pin. Wildcards are permitted.

For example, the following statement specifies it takes two clock cycles to reach signals from instance pins /us/u1/dff*.q to instance pins /u4/ mem1/*.D":

```
set_multicycle_path 2 -from /us/u1/dff*.q -to /u4/mem1/*.D";
```

set_register_to_output_delay

Defines the timing budget for outgoing signals to be clocked out.

```
set_resgister_to_output_delay <delay> -to out_port;
```

Where "delay" is the timing budget for this output path. "out_port" must be an output port of the design. Wildcards are permitted.

For example, the following statement specifies the timing budget for clocking out signals on output ports starting with "O" is 22 nanoseconds:

```
set_register_to_output_delay 22 -to O*;
```

set_max_path_delay

Constrains the maximum delay on paths.

```
set_max_path_delay delay_value
hier_inst_name_.inst_port_name
[,hier_inst_name .inst_port_name , … ];
```

Where "delay_value" is a floating integer for delay in nanoseconds, "hier_inst_name" is the hierarchical path to a cell instance, and "inst_port_name" is a port name of a cell instance.

For example:

```
set_max_path_delay 12.5 "mult4/mult/nand2_2".Y, "mult4/mult/
nand3_1".A,"mult4/mult/nand3_1".Y,"mult4/mult/nor2_2".A;
```

Timer GUI and Equivalent GCF Constraints

There is a discontinuity between Timer and the GCF constraints in the design flow for ProASIC and ProASIC^PLUS devices. The timing constraints in the user GCF file do not pass into Timer. However, these constraints are passed to the place-and-route tool, which tries to meet the timing requirements. If you set the constraints in the Timer GUI, Timer does not create the proper equivalent GCF file in certain cases (refer to the "SDC Constraint" section on page 6 for more information). Since the current place-and-route tool for Flash families only uses GCF constraints, some of the constraints in Timer are not passed to the Layout tool. In this case, Layout may not meet the timing requirements. Actel is working to remove the discontinuity in the flow and this flow issue will be fixed in future releases of Designer software.

When Layout is run after setting timing constraints, Designer/Libero requests that the constraints system (DCS) generate the timer.gcf file. The timer.gcf file is not appended; a new file is created each time the Layout is run. Consequently, if timer.gcf is modified manually, it will be overwritten every time that Designer runs Layout. When the constraints are removed from Timer, the previous timer.gcf file is deleted. Use the Summary, Paths or Breaks tab to set constraints in Timer.

The following section describes the constraints that are generated using the Timer.

Summary Tab

create_clock

Set clock constraints in Timer in the Summary tab as shown in Figure 2 on page 5. This adds the following to the timer.gcf. file:

```
create_clock  -period 5.00 "clk";
```

set_input_to_register_delay

The max delay applied to the Input ports to Register default set adds the following to the timer.gcf. file:

```
set_input_to_register_delay 5.00 -from *;
```

set_register_to_output_delay

The max delay applied to the Register to Output ports default set adds the following to the timer.gcf. file:

```
set_register_to_output_delay 3.00 -to *;
```

With the current available GCF constraints, users cannot set max delay constraints on the Input ports to Output ports. No equivalent GCF constraints are generated and they will be ignored (Figure 3 on page 5).

Paths Tab

With the current GCF syntax, max delay constraints on the Register-to-Register default set in Timer cannot be translated into the timer.gcf file. In the GCF format, the start point of a path is defined by the output pin of the source macro (e.g. ff_1.Q), while the end point is identified by the input pin of the sink macro (e.g. ff_2.D). In contrast, the start point of a path in Timer is defined by the valid input pin of the source macro (e.g. ff_1.CLK). This discrepancy in path definition between Timer and GCF files prevents the max delay constraint in Timer from being translated into an appropriate GCF constraint.

This set_max_path_delay constraint is ignored during compile and Layout. We do not recommend you use Paths tab to apply any constraint for timing-driven Layout.

Breaks Tab

False path constraints can be declared in Timer using the "Breaks" tab. This constraint is *generated* if the false path point is a pin. The GCF syntax uses "." to denote the pin separation (Figure 4 on page 6).

```
set_false_path
```

The break applied to a port generates the following in the timer.gcf:

```
set_false_path -through "z_int_16.S";
```

Please note that if there is a broken path in Timer, a flag is raised to write the timer.gcf file. However, there might be no relevant syntax in GCF to generate this constraint. In that particular instance, the file is generated, but is empty (for instance, setting a false path on a port generates an empty GCF file). Also, "set_false_path" commands are not supported for I/O and RAM cells.

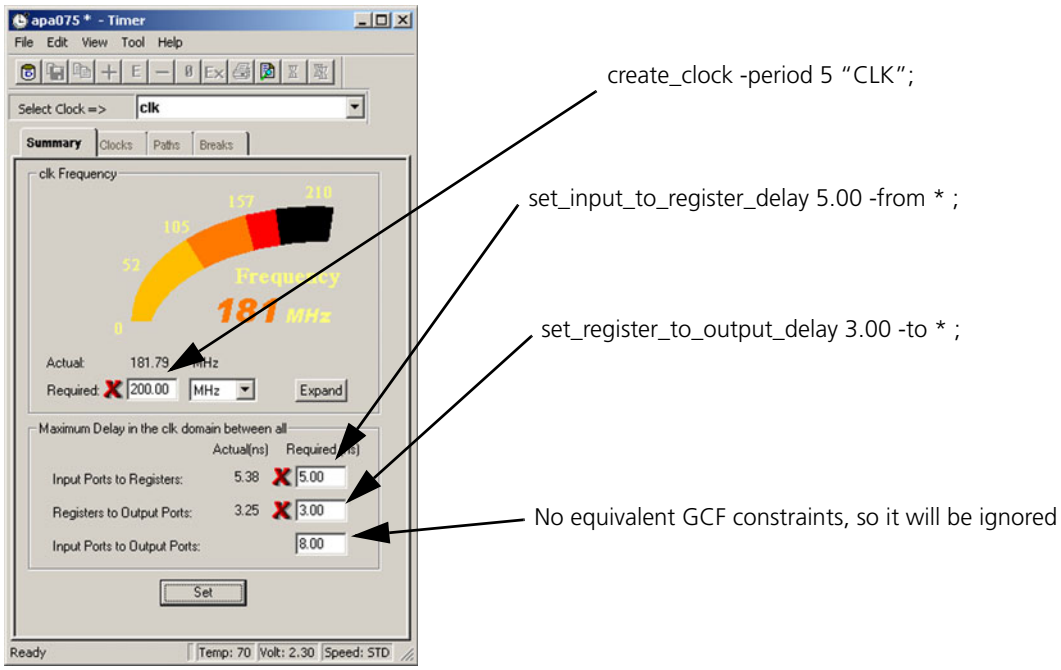

Figure 2 • **Summary Tab in Timer and Corresponding GCF Constraints**

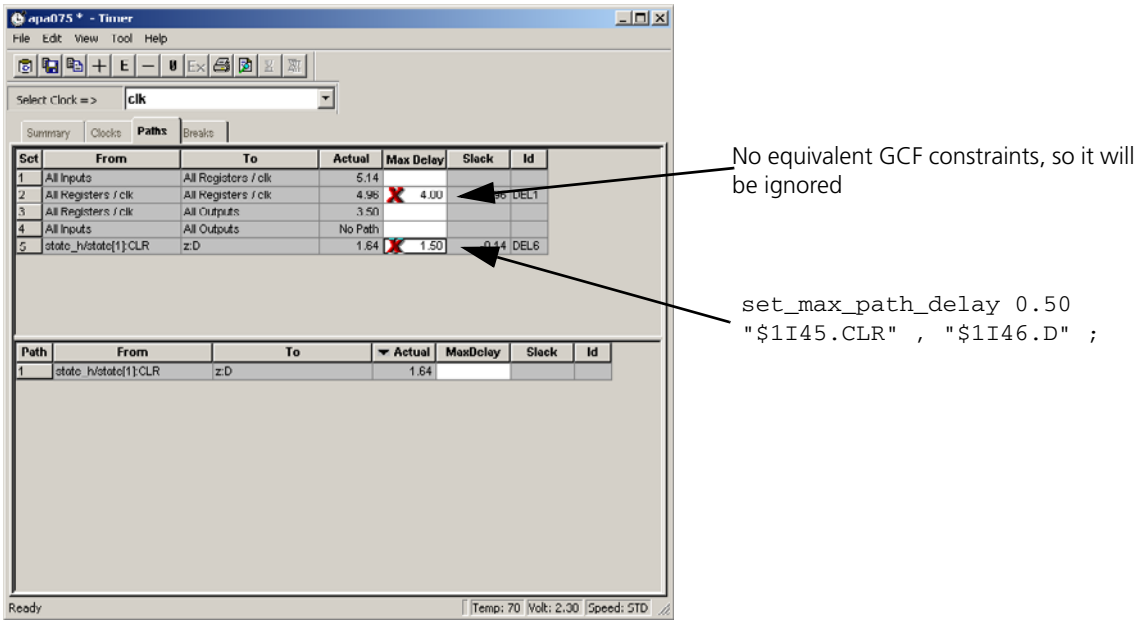

Figure 3 • **Timer Path Tab and Corresponding GCF Constraints**

Clocks Tab

You cannot generate any GCF constraints with the Timer Clocks tab.

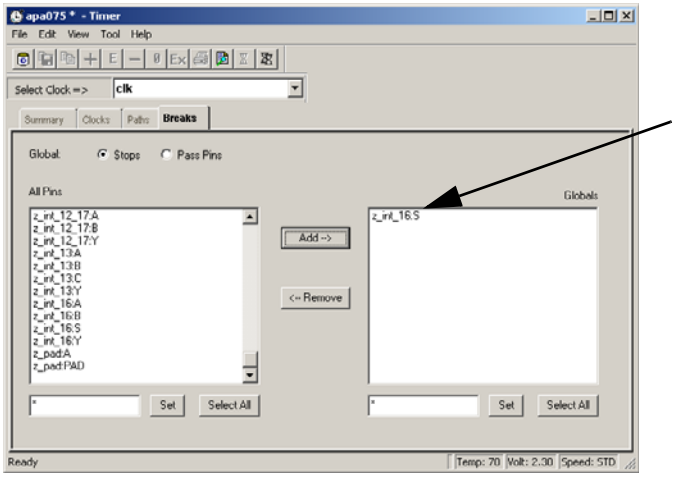

Figure 4 • **Timer Break Tab and Corresponding GCF Constraints**

SDC Constraint

SDC is a Tcl-based constraint file. The commands of an SDC file follow the Tcl syntax rules. Not all object and design constraint commands are supported in Designer. Table 1 lists the SDC timing constraint commands supported in Designer.

Table 1 • **Supported SDC Commands**

Constraint	Command
Clock Constraint	create_clock
Path Constraint	set_max_delay

Please see the Designer online help for details about using SDC.

The user can import SDC constraints during any stage of the design flow. When you import SDC files in Designer, they are passed to Timer and Designer generates the GCF constraints, just as if you had set the constraints in the Timer GUI.

Recommendation for Applying Timing Constraints for ProASIC and ProASIC^PLUS Families

The place-and-route tool considers the timing constraints in the GCF file and attempts to meet them. After routing, Designer issues messages to identify the constraints whether those have been met or not. To understand the complexity of a design and its performance, run place-and-route with no GCF constraints to see if routing can complete without constraints. If routing completes successfully, open Timer to see if the physical design meets timing requirements. If you are using a synthesis tool such as Synopsys Design Compiler, Actel recommends that you use it to generate a forward SDF file containing path constraints only. For other users, please use a combination of GCF and SDC constraints. Timing constraints must be reasonable. Over constraining a design may result in increased run times, while not improving circuit performance. The user must open Timer and run their own timing analysis to verify the timing requirements.

Conclusion

During place-and-route, users expect Timer to use constraints and evaluate them against the post-Layout timing information. When the constraints are entered in the Timer GUI, the timing constraints may not translate into appropriate GCF constraints. In this case, the Layout tool ignores the constraints. Inspect the compile report and check how many constraints have been read by Designer. Check timer.gcf and user.gcf files and compare the number of timing constraints in the GCF files. For the current version of Designer, write the timing constraints in the GCF file and use Timer for static timing analysis only. Both Layout and Timer use the information in the SDC file. Actel is working to remove the discontinuity in the flow and unify Timer and Layout so that all the constraints in Timer pass into Layout. The unified flow will be available in future releases of Designer software.

Related Documents

User's Guides

Designer User's Guide

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

http://www.actel.com

Actel Corporation
2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Tel: (650) 318-4200
Fax: (650) 318-4500

Actel Europe Ltd.
Maxfli Court, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Tel: +44 (0)1276 401450
Fax: +44 (0)1276 401490

Actel Japan
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Tel: +81 03-3445-7671
Fax: +81 03-3445-7668

Actel Hong Kong
39th Floor
One Pacific Place
88 Queensway
Admiralty, Hong Kong
Tel: 852-22735712

51900048-0/08.03