Technical Brief

Using Synopsys Design Constraints (SDC) with Designer

This technical brief describes the commands and provides usage examples of Synopsys Design Constraints (SDC) format with Actel's Designer Series software. SDC is a widely used format that allows designers to utilize the same sets of constraints to drive synthesis, timing analysis, and place-and-route. This document includes information about SDC design objects, timing constraints, and timing exceptions.

System Setup

- Designer versions: R1-2001 and later
- Both PC and UNIX Designer versions support the SDC constraint format
- Supported devices: SX-A, eX

Understanding and Using SDC with Designer

SDC is a Tcl-based format-constraining file. The commands of an SDC file follow the Tcl syntax rules. Designer will accept an SDC constraint file generated by a third-party tool. This file is used to communicate design intent between tools and provide clock and delay constraints. The Synopsys Design Compiler, Prime Time, and Synplicity tools can generate SDC descriptions, or the user can generate the SDC file manually.

Generated SDC File

There can be slight differences between a user generated SDC file, and SDC files generated by other tools.

For example, suppose you write the following constraint:

```
create_clock -period 100 clk
```

The SDC file from Design Compiler generates the same constraint in a different format:

```
create_clock -period 100 -waveform {0 50}
[get_ports {clk}]
```

The SDC file from Prime Time generates this constraint in yet another format:

```
create_clock -period 100.000000 -waveform
{0.000000\ 50.000000}[get_ports {clk}]
```

As long as constraint syntax and arguments conform to the Tcl syntax rules that SDC follows, Designer will accept the SDC file.

Importing an SDC File

Importing an SDC file into Designer and compiling the design will allow Timer to build the timing graph and map the constraints. To import SDC into Designer, the procedure is as follows (See Figure 1 on page 2):

1. Invoke Designer.
2. Import the EDIF netlist. From the menu, select **File -> Import Netlist**.
3. Import the SDC file. From the menu, select **File -> Import**. Select **File Type** with extension **.sdc**, choose the **.sdc** file to import.

4. Compile and perform Timing-Driven Layout.

When Compile and Layout are complete and Timer is invoked, the constraints from the SDC file will be incorporated in the timing of the design and will be reflected in Timer. For more details on using Timer, refer to the *Timer User's Guide*.

Supported Commands and Syntax

Design goals, such as performance, determine the types of constraints that can be set in your design. This section lists the SDC access commands and restraints as well as giving usage examples. For the latest supported commands and objects, please see the *Designer Release Notes* or *Designer User's Guide*.

Design Object Access Commands

Most constraint commands require a command argument. Designer supports the SDC access commands shown in Table 1.

Table 1* • *Supported SDC Access Commands

Design Object	Access Command
Clock	get_clocks
Port	get_ports

get_clocks returns the named clock with the argument.

Example:

```
create_clock -period 10 [get_clocks CK1]
```

get_ports returns the named ports with the argument.

Example:

```
set_max_delay -from [get_ports data1] -to
[get_ports out1]
```

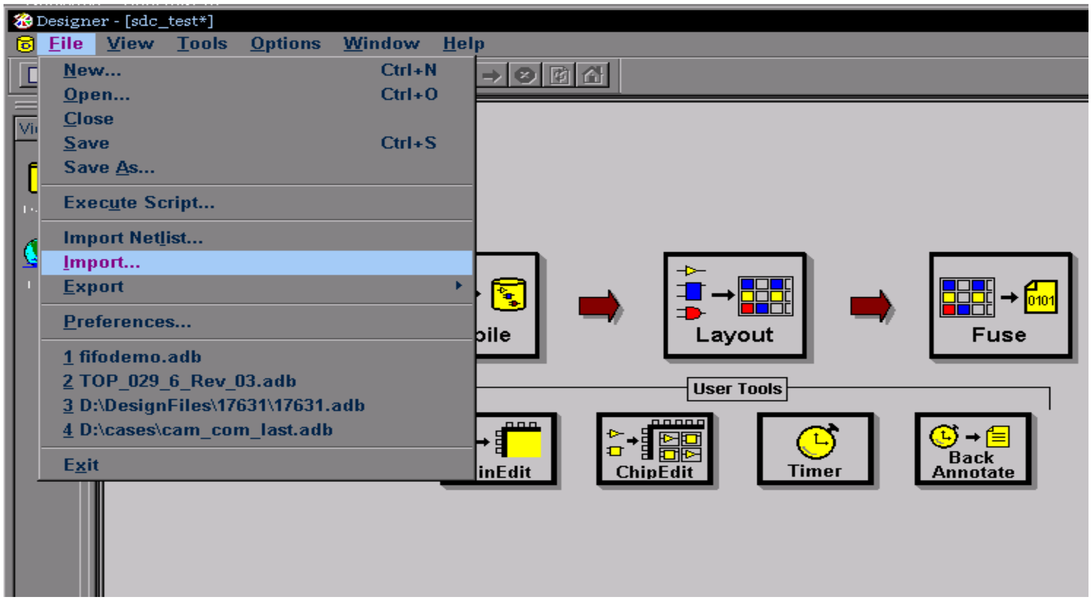


Figure 1 • Importing an SDC File in Designer

Timing Constraint Commands

Design Constraint command examples are listed in Table 2.

Table 2 • Supported SDC Constraint Command

Constraint	Command
Clock Constraint	create_clock
Path Constraint	set_max_delay

Clock Constraint

The **create_clock** constraint is associated with a specific clock in a sequential design and determines the maximum register-to-register delay in the design. The following is a description of command syntax for specifying a clock:

```
create_clock -period period_value [-name
clock_name] [-waveform edge_list]
[port_pin_list]
```

`period_value` (specified in ns) is mandatory. There will be no clock created if the period is not supplied.

`clock_name` is optional. It is unnecessary if port_pin_list contains one name.

`edge_list` is optional and not supported in the current version of Designer. If supplied, it must contain exactly 2 edges. The duty cycle info will then be added to the clock constraint.

`port_pin_list` may contain either zero names or one name.

The following are valid and invalid examples of the **create_clock** command:

Valid Command Examples

`create_clock -period 5 -name CK1`

`create_clock -period 4 -name CK1 -waveform 0 2`

`create_clock -period 6 [get_ports CK1]`

`create_clock -period 11 -name CK1 -waveform 0 2 5 7` (valid, but the waveform will be ignored)

`create_clock -period 2 -name CLOCK [get_ports CK1]` (valid, but the name of the clock will be CK1 and not CLOCK)

Invalid Command Examples

`create_clock -period 10` (no name is supplied)

`create_clock -period 3 [get_ports {CK1 CK2}]` (more than one name in the port_pin_list)

`create_clock -period 7 -name CK [get_ports {CLK11 CLK2}]`

`create_clock clk -name CLK1 -period 20 -waveform 3 13`

Effects of the create_clock Constraint

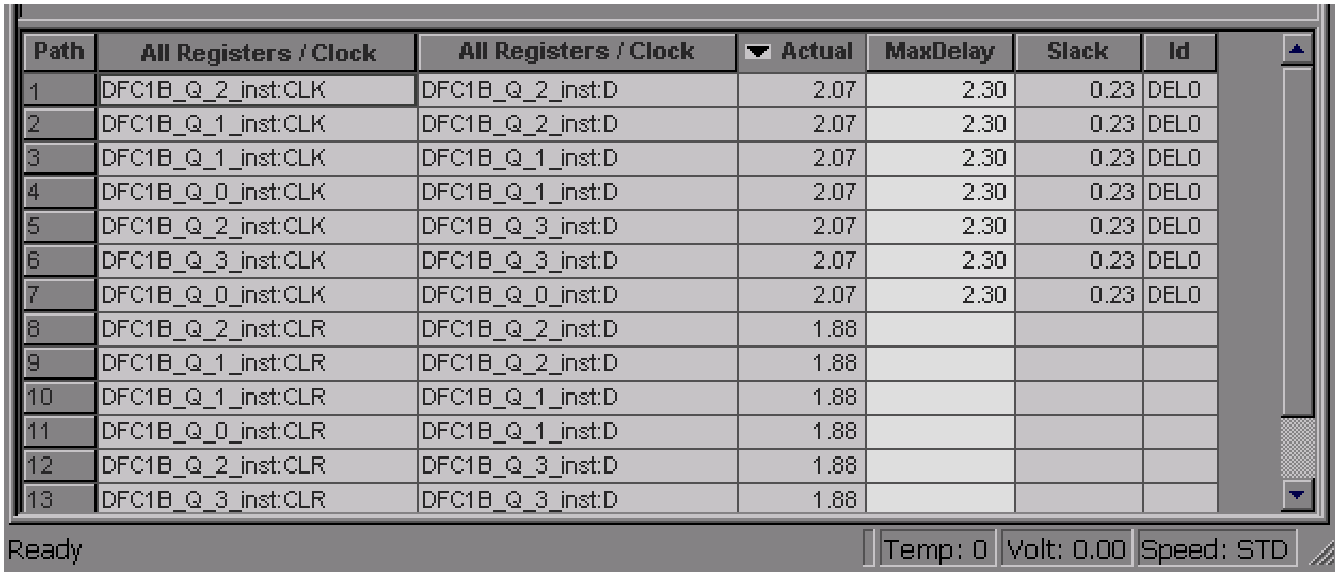
Example:

The original design has a clock period of 3.13 ns. The revised design is to run with a 2.30 ns period.

The following clock constraint is given to Designer:

`create_clock -period 3 [get_port Clock]`

Figure 2 on page 3 shows the results obtained after compiling and performing Timing-Driven Layout on the constraint, showing the improvement of register-to-register delay. This design has 0.23 ns of slack on the constrained registers.



Path	All Registers / Clock	All Registers / Clock	Actual	MaxDelay	Slack	Id
1	DFC1B_Q_2_inst:CLK	DFC1B_Q_2_inst:D	2.07	2.30	0.23	DEL0
2	DFC1B_Q_1_inst:CLK	DFC1B_Q_2_inst:D	2.07	2.30	0.23	DEL0
3	DFC1B_Q_1_inst:CLK	DFC1B_Q_1_inst:D	2.07	2.30	0.23	DEL0
4	DFC1B_Q_0_inst:CLK	DFC1B_Q_1_inst:D	2.07	2.30	0.23	DEL0
5	DFC1B_Q_2_inst:CLK	DFC1B_Q_3_inst:D	2.07	2.30	0.23	DEL0
6	DFC1B_Q_3_inst:CLK	DFC1B_Q_3_inst:D	2.07	2.30	0.23	DEL0
7	DFC1B_Q_0_inst:CLK	DFC1B_Q_0_inst:D	2.07	2.30	0.23	DEL0
8	DFC1B_Q_2_inst:CLR	DFC1B_Q_2_inst:D	1.88			
9	DFC1B_Q_1_inst:CLR	DFC1B_Q_2_inst:D	1.88			
10	DFC1B_Q_1_inst:CLR	DFC1B_Q_1_inst:D	1.88			
11	DFC1B_Q_0_inst:CLR	DFC1B_Q_1_inst:D	1.88			
12	DFC1B_Q_2_inst:CLR	DFC1B_Q_3_inst:D	1.88			
13	DFC1B_Q_3_inst:CLR	DFC1B_Q_3_inst:D	1.88			

Ready | Temp: 0 | Volt: 0.00 | Speed: STD

Figure 2* • *Post-Layout Timing Examination Showing Slack after Timing-Driven Layout on the Clock Constraint

Max Delay Constraint

The **set_max_delay** constraint sets the path delay of the specified ports to a restricted value. The syntax is as follows:

```
set_max_delay [-from from_list] [-to to_list]
delay_value
```

- `from_list` is mandatory
- `to_list` is mandatory
- `delay_value` is specified in ns

The following examples give sample command syntax for specifying maximum path delay:

Valid commands

```
set_max_delay -from [get_ports data2] -to
[get_ports {out1 out2}] 9
```

Invalid commands

```
set_max_delay -from [get_ports {IN10 IN11}] 5
```
(the to_list is not supplied)

Effects of the set_max_delay Constraint

Example 1:

Feedthrough signals are hard to constrain because they do not belong to any clock domain. However, the **set_max_delay** constraint is one way to properly do it. Sample paths are illustrated in Figure 3. The commands to constrain these paths are given below.

```
set_max_delay -from [get_ports A] -to
[get_ports Y] 12
```

```
set_max_delay -from [get_ports A] -to
[get_ports X] 8
```

```
set_max_delay -from [get_ports B] -to
[get_ports X] 3
```

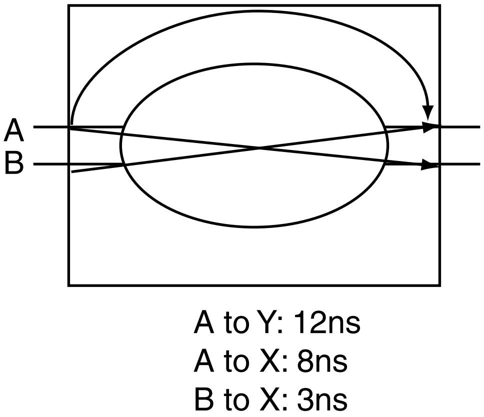


Figure 3* • *Feedthrough Signal Paths

Example 2:

Design originally has an In-to-Out delay path, Clock->Q<1> = 5.97 ns after standard layout. The designer hopes to set the maximum delay to 5.80 ns. The following **set_max_delay** constraint was given to Designer:

```
set_max_delay -from [get_ports Clock] -to
[get_ports Q<1>] 5.80
```

The results obtained after compiling and performing a Timing-Driven Layout on the constraint are shown in Figure 4 on page 4.

Designer could account for this constraint and set the max delay to 5.80 ns. The actual delay with this constraint for this Clock->Q<1> path is 5.63 ns, with a positive slack of 0.17 ns.

Additional Notes

Synplify's Synplicity also has provided the SDC forward constraint option, where the synthesis tool writes out this file as well as the synthesized netlist. Actel's Designer can import this SDC file and timing driven place and route can be performed on the design.

Limitations

Not all object and design constraint commands are supported in Designer. There are limitations on SDC support. Refer to the latest Designer series Release notes for latest supported Object Access, Design Constraints, and Supported Features.

Naming Conventions

- There is no wildcarding; the * and ? characters cannot be used in the object names.

Example:

```
set_max_delay -from [get_ports data1] -to
[get_ports {out1 out2}] 9 (valid)
```

```
set_max_delay -from [get_ports data*] -to
[get_ports out*] 9 (not supported)
```

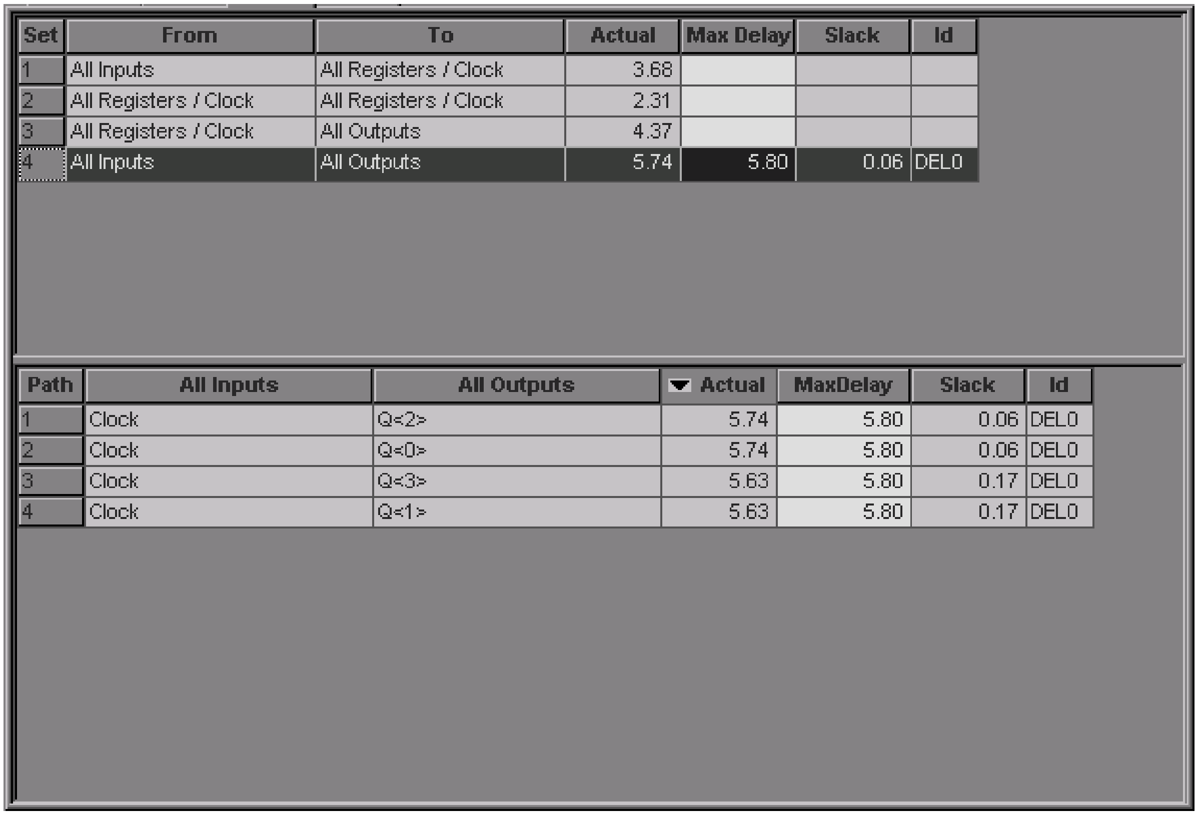
- Object names – The timing graphical interface will display the internal Actel port names. While the internal Actel netlist prevents special characters from being used, in the case where the internal name is different from the "user" netlist, there may be discrepancies in the GUI. These could also be different from the names in the SDC files.

No Multi-file Support

Constraints imported from one SDC file will be discarded if a second SDC file is imported. All constraints must be imported from a single SDC file.

Conclusion

With the widely used Synopsys Design Constraint (SDC) format integrated into Designer, users have another option to set timing constraints in their design. Actel Designer will take into account these constraints, therefore allowing the same sets of constraints to drive synthesis, timing analysis, and place and route. Results of these constraints and net delays can then be observed by using Timer.



Set	From	To	Actual	Max Delay	Slack	Id
1	All Inputs	All Registers / Clock	3.68			
2	All Registers / Clock	All Registers / Clock	2.31			
3	All Registers / Clock	All Outputs	4.37			
4	All Inputs	All Outputs	5.74	5.80	0.06	DEL0

Path	All Inputs	All Outputs	▼ Actual	MaxDelay	Slack	Id
1	Clock	Q<2>	5.74	5.80	0.06	DEL0
2	Clock	Q<0>	5.74	5.80	0.06	DEL0
3	Clock	Q<3>	5.63	5.80	0.17	DEL0
4	Clock	Q<1>	5.63	5.80	0.17	DEL0

Figure 4* • *In-to-Out delays with set_max_delay Constraints Set

http://www.actel.com

Actel Europe Ltd.
Maxfli Court, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Tel: +44 (0)1276 401450
Fax: +44 (0)1276 401590

Actel Corporation
955 East Arques Avenue
Sunnyvale, California 94086
USA
Tel: (408) 739-1010
Fax: (408) 739-1540

Actel Asia-Pacific
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Tel: +81-(0)3-3445-7671
Fax: +81-(0)3-3445-7668