# Synplify Achieving Optimal Synthesis Results

The technical development team of Synplicity, Inc. has been cooperating with Xilinx to deliver Synplify, a fast, easy-to-use synthesis tool that produces extremely high-quality results. The latest version of Synplify supports the Xilinx XC3X00A/L, XC4000E/X, XC5200, and XC9500 families, as well as the XC40125XV device.

Synplify accepts timing constraints placed in a constraint file. In this article you will learn how to apply design constraints for synthesis and optimization.

## Design Constraints

Synplify supports user-defined timing constraints to help improve your synthesis results. Timing constraints, for timing-driven synthesis, should be specified in a synthesis timing constraint file, *<design_name>.sdc*, and added into the list of source files. Synplify encourages you to set constraints to closely match your design goal by ±5%.

The following design constraints are used for synthesis and are passed to the implementation tools for timing-driven place and route.

- **Clock Constraint** - Allows you to specify a specific frequency goal for synthesis.

```
define_clock CLK1 -freq 33.0
```

- **Input Delay Constraint** - Allows you to model the interface of the inputs of your FPGA with the outside environment, such as the delay before the signal arrives at the input pin.

```
# Set the input delay on input
  port A to 10.0 ns.
define_input_delay A 10.0
```

- **Output Delay Constraint** - Allows you to model the interface of the outputs of your FPGA with the outside environment, such as the delay of the logic outside your FPGA that is driven by your outputs.

```
# Set the default output delay
  for all outputs to 10.0 ns
define_output_delay -default 10.0
```

## Optimization Constraints

Synplify provides optimization constraints, which focus the synthesis engine on critical timing paths within a design; these constraints are not passed to the place and route tools. They control synthesis optimization without over-constraining the place and route engine.

- **-improve <ns>** - Using this option forces Synplify to restructure your design during optimization to try and meet the clock frequency goal. In the example, the input delay for input_a is 1 ns. The "-improve" option forces Synplify to try harder and reduce the clock period by 2.0 ns.

```
define_input_delay input_a
  1.0 -improve 2.0
```

- **-route <ns>** - Using this option forces Synplify to use additional route delay in its calculations to try and meet the clock frequency goal. In the example, the routing delay reported by the place and route engine for input_a was 1.8 ns more than predicted by Synplify. Also, input_a has a 1.0 ns input delay. The "-route" option forces Synplify to re-run and try to improve results by 1.8 ns, by adding 1.8 ns of additional route delay to the timing calculations.

```
define_input_delay input_a
  1.0 -route 1.8
```

## Summary

Synplicity is focused on delivering the highest quality results for Xilinx FPGAs and CPLDs. The Synplify 3.0b release introduces a new Xilinx mapper that has shown 5-20 percent performance and area improvements. The Synplify 3.0c release will support features such as:

- Passing timing constraints to the Alliance Series implementation tools.
- Speed and area improvements to the Xilinx mapper.
- Synthesis support for Spartan, Virtex, and XC4000XV. ◆



XILINX®