

Passing Hierarchical Timing Constraints from Synopsys Tools to MAX+PLUS II Version 9.0

Technical Brief 48

August 1998, ver. 1

SYNOPSYS®

Synopsys
700 East Middlefield Road
Mountain View, CA 94043
(650) 962-5000
<http://www.synopsys.com>

Introduction

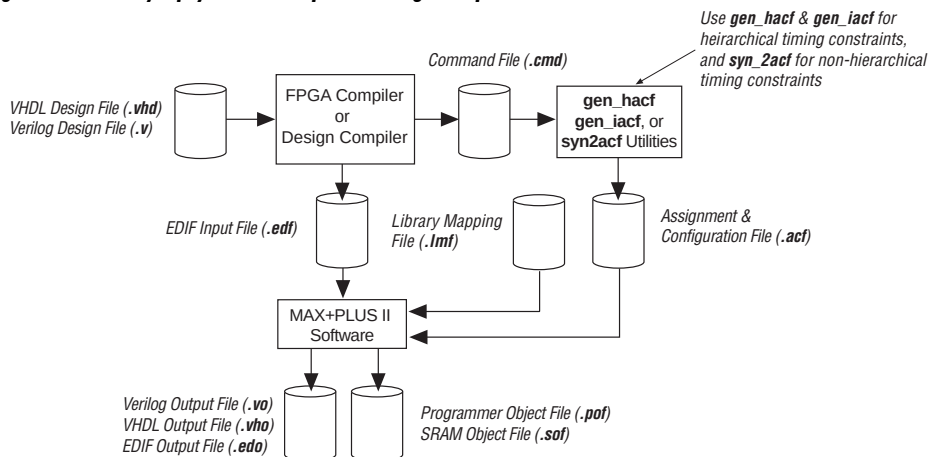
The interface between the Altera® MAX+PLUS® II software and the Synopsys Design Compiler and FPGA Compiler provides designers with an easy path for synthesizing designs and targeting Altera devices. Altera has made significant improvements to the Altera/Synopsys design flow, including:

- Support for importing hierarchical assignments from a Synopsys Design Compiler or FPGA Compiler design
- Support for the latest speed grades in Synopsys tools
- More efficient synthesis results
- Improved multiplexer support

The MAX+PLUS II version 9.0 software contains two new utilities—**gen_hacf** and **gen_iacf**—that convert timing assignments from hierarchical Synopsys designs to a MAX+PLUS II compatible format; you do not need to re-enter timing constraints for the lower-level design files in the MAX+PLUS II software.

Figure 1 shows a flow diagram of the Altera/Synopsys interface for UNIX workstations.

Figure 1. Altera/Synopsys FPGA Compiler & Design Compiler Interface for UNIX Workstations



Using the gen_hacf & gen_iacf Utilities

You can use the **gen_hacf** and **gen_iacf** utilities to convert hierarchical timing constraints from a Synopsys Command File (.cmd) to a MAX+PLUS II Assignment & Configuration File (.acf). When passing either hierarchical or non-hierarchical timing constraints from Synopsys tools to the MAX+PLUS II software, data is converted from the Synopsys Command File format to the ACF format. Table 1 on page 2 provides commands and descriptions for passing timing constraints from Synopsys to the MAX+PLUS II software.



For complete instructions on passing timing constraints, see MAX+PLUS II Help, or the MAX+PLUS II Altera Commitment to Cooperative Engineering Solutions (ACCESSSM) Key Guidelines, which are available in the Altera Technical Support (AtlasTM) section of the Altera web site at <http://www.altera.com>. These guidelines are also available on the MAX+PLUS II Programmable Logic Development Software CD-ROM and can be installed with the MAX+PLUS II software.

Table 1. Commands for Passing Hierarchical Timing Constraints Using the gen_hacf & gen_iacf Utilities	
Command	Description
<code>read -f vhdl/verilog <design name></code>	Reads the design files. Repeat this command for each design file in the hierarchy, starting with the lower-level files and ending with the top-level file.
<code>elaborate <design name></code>	Creates a design from the intermediate hardware description language (HDL) formats.
<code>current_design=<design name></code>	Defines the current design.
<code>include <timing file name></code>	Specifies the Synopsys Command File that contains the timing constraint for the current design.
<code>compile</code>	Compiles the design.
<code>design_name=<design name></code>	Sets the <code>design_name</code> variable used in the gen_iacf utility to <code><design name></code> .
<code>include /<path name>/gen_iacf.cmd</code>	Generates the files needed to pass timing constraints.
<code>/<path name>/gen_iacf <design name></code>	Generates intermediate constraint files for designs that have timing constraints. Repeat this command for all designs that have timing constraints.
<code>write -f edif -o <top-level design name>.edf -h</code>	Generates a hierarchical EDIF Input File (.edf) for processing in the MAX+PLUS II software.
<code>/<path name>/gen_hacf <top-level design name> <sub-design file list></code>	Merges the intermediate constraint files into a single hierarchical ACF.

Below is a **sample dc_shell script** for a design that only has timing constraints on lower-level designs. It utilizes the new hierarchical **gen_iacf** and **gen_hacf** utilities using the commands in Table 1. For a more detailed sample script, see the MAX+PLUS II ACCESS Key Guidelines.

```

/* dc_shell script example to interface with new hierarchical
   gen_iacf and gen_hacf utilities
   This example includes timing constraints on lower-level
   designs only */
Repeat Read Statement {read -f vhdl LOWER1.vhd
For Each Lower-Level File {read -f vhdl LOWER2.vhd
                           read -f vhdl TOP.vhd

Repeat For Each {elaborate LOWER1
Lower-Level File {current_design=LOWER1
                  {include timing1.cmd
                  {compile
                  {design_name=LOWER1
                  {include /usr/maxplus2/synopsys/bin/gen_iacf.cmd
                  {sh /usr/maxplus2/synopsys/bin/gen_iacf LOWER1

elaborate LOWER2
current_design=LOWER2
include timing2.cmd
compile
design_name=LOWER2
include /usr/maxplus2/synopsys/bin/gen_iacf.cmd
sh /usr/maxplus2/synopsys/bin/gen_iacf LOWER2

current_design=TOP
write -f edif -o TOP.edf -h
sh /usr/maxplus2/synopsys/bin/gen_hacf TOP subdesign.list

quit

```

Although you can use the commands in Table 1 to pass non-hierarchical timing constraints, Altera recommends that you use the commands listed in Table 2 with the latest **syn2acf** utility. Table 2 provides a sample of the commands used with the **syn2acf** utility and their descriptions.

Table 2. Commands for Passing Non-Hierarchical Timing Constraints Using the **syn2acf Utility**

Command	Description
<code>read -f vhd1/verilog <design name></code>	Reads the design files.
<code>include timing.cmd</code>	Specifies the CMD file that contains the timing constraint.
<code>compile</code>	Compiles the design.
<code>current_design=<design name></code>	Defines the current design.
<code>include /<path>/syn2acf.cmd</code>	Generates the files needed to pass timing constraints.
<code>/<path>/syn2acf <design name></code>	Starts the syn2acf utility.

Below is a sample **dc_shell** script for running the **syn2acf** utility using the commands in Table 2.

```

/* dc_shell script example to interface with the syn2acf utility */
Repeat Read Statement { read -f vhd1 LOWER1.vhd
For Each Lower-Level File { read -f vhd1 LOWER2.vhd
                           read -f vhd1 TOP.vhd
                           include timing.cmd
                           compile
                           current_design=TOP
                           include /usr/maxplus2/synopsys/bin/syn2acf.cmd
                           sh /usr/maxplus2/synopsys/bin/syn2acf TOP
                           quit

```

Additional Information

Altera provides extensive support documentation to help you successfully use the MAX+PLUS II software with third-party EDA tools. For technical support, contact Altera Applications at (800) 800-EPLD or go to the MAX+PLUS II ACCESS Key Guidelines on the Altera web site. For answers to common questions regarding the Altera/EDA software tool interface, search Atlas Solutions on the Altera web site.

The following documents provide more detailed information:

- MAX+PLUS II ACCESS Key Guidelines
- *MAX+PLUS II Programmable Logic Development System & Software Data Sheet*
- *EDA Software Support*
- *Technical Brief 39 (Using Synopsys Design Compiler & FPGA Compiler to Synthesize Designs for MAX+PLUS II Software)*

These documents are available by contacting Altera Literature Services at (888) 3-ALTERA; you can also download them from the Altera web site at <http://www.altera.com>.



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>

Copyright © 1998 Altera Corporation. Altera, ACCESS, Atlas, MAX, MAX+PLUS, and MAX+PLUS II are trademarks and/or service marks of Altera Corporation in the United States and other countries. Other brands or products are trademarks of their respective holders. The specifications contained herein are subject to change without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services. All rights reserved.