ProASIC and ProASIC$^{\text{PLUS}}$

Software FAQ

Version 1.1
January, 2003

Table of Contents

Actel® Designer Software

1) What variable is available for place-and-route in ProASIC$^{\underline{PLUS}}$ designs?

You can use the following variable in the Designer software. Set the following variable in Designer: PROASIC_QDPLACER_RANDOM_SEED. The value can be from 1 to (2^32)-1.

2) I am running a ProASIC$^{\underline{PLUS}}$ design using both a PC and Solaris with default settings, but I get different performances. Why does the same design with the same settings perform differently?

PC and Solaris have different floating points, which affect the staring points of the place-and-route algorithm.

3) I want to define a minimum frequency for my design. What are my options?

The best approach is to put a constraint on the clock. There are two ways to do this:

1. Open Timer in Designer. Select the required clock from "select clock" drop-down menu and enter the required frequency. Select "Commit" and run the Timing Driven Layout.

2. You can also use "create_clock" command in GCF file and import it along with the netlist into Designer. The syntax for the constraint is as follows:

```
create_clock –period <period value> < netname|portname>;
```

   Where period value is in "ns."

   For example, to impose a 50 MHz constraint on an external clock named CLK, users need to put the following in the GCF file:

```
create_clock –period 20.0 CLK;
```

   Also, if you want to impose a 40 MHz constraint to an internal signal derived_clock, users need to put the following in the GCF file:

```
create_clock –period 25 u4/u2/derived_clock;
```

4) I have a long path in my design, which is a false path (or a multicycle path) and Designer assigns lots of effort and resources to route this path. How can I tell Designer to exclude those paths from criticality assignments or timing calculations?

You can define the false path in two ways:

The “set_false_path” command in a GCF file can be used to define a false path in the design. The syntax for this command is:

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

Where “from_port” must be an input port of the design or a register or memory instance output pin, “to_port” must be an output port of the design or a register or memory instance input pin, “any_port” must be any instance pin. Wildcards are permitted. For example, the following statement sets all paths starting from “resetd” which are going through instance “const2” as false paths:

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

Similarly, multicycle paths can be defined using “set_multicycle_path” in GCF file and importing the file into Designer. The syntax is:

```
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 a register of memory instance, “to_port” must be an output port of the design or a register or memory instance input pin, “any_port” must be any instance pin. Wildcards are permitted. For example, the following statement specifies that 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;
```

You can also set Break in Timer to create a false path. To add break points:
Click the Breaks tab in Timer.
Select Global Stops Pins.
Select Pin(s). The All Pins list box defaults to show all pins. Select the desired pin and click Add.

This will create a GCF file with the following syntax, which will be used by the Designer software:

```
set_false_path -through [any_port];
```

5) I need to specify my input-to-register and register-to-output delays before place-and-route as a constraint because of my board level timing. Can I do that?

Yes, you can do that both in the Timer and also through constraints in the GCF file.

For GCF files, the syntax for those constraints are:

```
set_input_to_register_delay <delay> -from in_port;

set_resgister_to_output_delay <delay> -to out_port;
```

Where "delay" is the timing budget for this input path (ns), "in_port" is a register or memory instance output pin and "out_port" must be an output port of the design. Wildcards are permitted.

In order to set constraints in Timer, open Timer and follow the steps below:

Click the Paths tab. By default, Timer displays four path sets in the set grid. Click All Inputs - All Registers and set delay value to set the input to register delay and click All Register - All Outputs to set the register to output delay. From the File menu, click Commit. Also, you can create a path set in the Timer from input – register and put constraints.

These will create the appropriate GCF file.

6) Designer assigns resources to the nets that are not important. How can I assign the priority in routing the nets?

If no timing constraint is specified, Designer prioritizes the nets or ports based on their fanout. To prioritize the nets or ports, users can assign criticality to them using the following instructions in the GCF file:

```
set_critical criticality_number hier_net_name
[ ,  hier_net_name ];

set_critical_port criticality_number signal_name
[ ,  signal_name ];
```

Where "criticality_number" is from 1 to 5 (1 being the default criticality for every net and 5 being the highest). "hier_net_name" is the full hierarchical net name and "signal_name" is the name of a user-defined signal associated with a specific I/O pin on the part.

For example, the following statement sets all nets associated with device ports IOBus[3] and IOBus[5] to have criticality 3:

```
set_critical_port 3 IOBus[3], IOBus[5];
```

7) Can I set a timing constraint on a specific path?

Yes, to set a timing constraint you need to use the following line in your GCF file:

```
set_max_path_delay delay_value
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;
```

8) Do I have any control on the number of flash switches used to route a specific path?

Yes. Use the following statement to specify the number of switches the router is allowed to route a net through before it has to insert an active repeater while routing the specified net:

```
set_switch_threshold <threshold> <net_name>;
```

Where “threshold” is an integer for the threshold. By default, 8 switches are allowed. The user can set this between 4 and 16. The “net_name” is the name of the net(s) the threshold should be used for. Wildcards are permitted. For example:

```
set_switch_threshold 6 core/fsm/state_*;
```

9) I want to assign the nets that have a fanout higher than a specific number to global paths. Is that possible?

Yes it is possible. Use “set_auto_global_fanout” command in your GCF file. For example, use

```
set_auto_global_fanout 12;
```

to assign the global resources to the nets with fanout more than 12. Please note that there are only 4 global resources on the ProASIC$^{\text{PLUS}}$ devices. If the number of nets with 12 or more fanouts is higher than 4, then only the nets with the highest fanout will be assigned to global paths.

10) I want to use only 3 global paths for high fanout nets promotion and preserve the last one for some other purpose. How can I perform this?

You can set the number of globals to be used for automatic net promotion by using the following constraint in the GCF file:

```
set_auto_global number;
```

Where “number” is the number of global resources to be used for net promotion. It must be a numbr between 1 and 4.

11) What is the GCF file syntax to assign nets to the spines?

Use this statement to guide place-and-route to route the net using the specified global spine from the global network:

```
use_global spine netname;
```

Where “spine” is one of the spines T1 to T<n> or B1 to B<n>, “netname” is the name of the net.

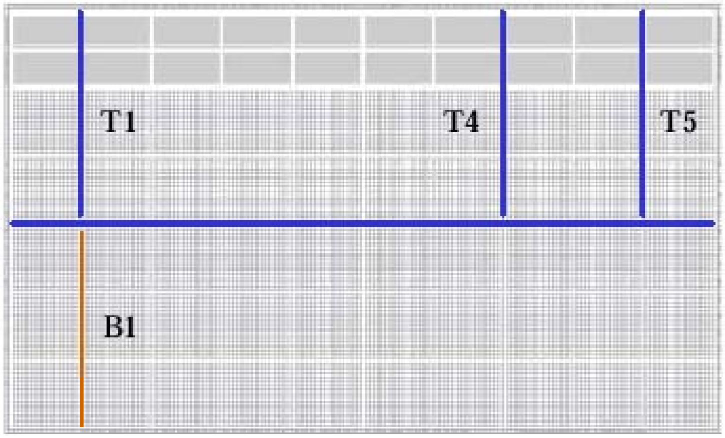

Figure 1: Multiple Spine Distribution

12) If I use B1 spine for a signal net, can I use the remaining three spines at B1 for other nets?

You can use the following command to use multiple spines at the same place:

```
use_global B1 CLKA;

use_global B1 CLKB;
```

13) I need to assign an internal signal T1 and B1 at the same time. Is it possible?

Yes, the internal signal can be assigned to T1 and B1 by using the following constraint in the GCF file. For example:

```
use_global B1,T1 <net_name>;
```

Where “netname” is the name of the internal signal.

If the user gives the spine rectangle as B1, T3, the driven instances of the given net will get a region constraint which will enclose the rectangle including the spine rectangle B1, T1, B2, T2, B3, T3. The user can specify the following type of rectangles:

1. Bn,Bm : n<=m will mean Bn,Bn+1, ... Bm
2. Tn,Tm : n<=m will mean Tn,Tn+1, ... Tm
3. Bn,Tm : n<=m will mean Bn,Tn,Bn+1,Tn+1...Bm,Tm
4. Tn,Bm : n<=m will mean Bn,Tn,Bn+1,Tn+1...Bm,Tm

14) Can I assign a signal to T1 and T3 at the same time?

No, the same signal cannot be assigned directly to T1 and T3 spines. To do so, the signal should be buffered (the net name needs to be changed). Therefore, if the signal is clock signal, a skew equal to a buffer delay will be inserted in the clock network. For example:

```
use_global T1 CLKA;

use_global T3 CLKA_buff;
```

Where CLKA_buff is the buffered signal of CLKA. In this example, the T3 spine carries the same clock as T1 spine, with a timing skew equal to the propagation delay of a buffer.

15) Designer automatically promotes the high fanout signals to global. Is there anyway to turn this off?

Yes, you can turn the automatic promotion off by using the "dont_fix_globals" constraint in the GCF file.

16) How can I turn the netlist optimization on or off?

The netlist optimization can be turned on or off by using "optimize" and "dont_optimize" constraints respectively. If you don't specify the type of optimization, the commands will act globally. However, you can turn on or off a specific optimization procedure by indicating the procedure with the above mentioned constraints. For example,

```
optimize buffer inverter
```

will turn on the inverter and buffer optimization.

17) I need to prevent the optimization from being performed only on a specific net or block. Is it possible?

Yes, it is possible using "dont_touch" constraint. The net name should be hierarchical and wild cards are allowed. Therefore, using wildcards will allow you to apply the "dont_touch" constraint on the whole block. For example:

```
dont_touch /U1/myblock/*;
```

The statement in this example will enable only the buffer and inverter optimization types and optimization will be done on all instances except those contained in the block called /U1/myblock.

18) I want to place specific blocks of my design in a specific part of the device. What is the appropriate approach?

To specify placement of a macro or block, you need to use the "set_location" constraint. To specify the location of a cell (macro) the (x,y) dimensions and the full hierarchical name of the cell should be specified. Rectangular regions can be specified for block placement by defining the bottom-left and top-right corners of the region. Wildcards can be used to identify the block name. The following are the examples for placing a single cell and a block:

```
set_location (1,15) u4/u3/nand3_4;

set_location (1,1 32,32) datapath/*;
```

19) Is it possible to keep a region of the device empty and tell Designer not to place any macros in that region?

Yes, the constraint "set_empty_location" lets you keep parts of the device empty for different purposes. The constraint can be applied to a specific cell or a region. The following examples show the usage of this constraint:

```
set_empty_location ( x ,y);

set_empty_location (x1,y1 x2,y2);
```

where (x1,y1) and (x2,y2) are the dimensions of the bottom-left and top-right corners of the desired region.

20) Due to my board layout, I need to keep one of my I/O pins unused. What is the easiest way to do this?

The easiest way to perform this is by using the constraints in your GCF file. Use the following statement to specify a location in which no I/O pin should be placed. The location can be specified by side and offset, or by name:

```
set_empty_io { package_pin | pad_location};
```

For example, the following statement forces pin B5 to be empty:

```
set_empty_io B5;
```

21) What is the difference between "set_initial_io" and "set_io" constraints?

set_io is similar to fixing a pin in PinEdit. Once the pin is fixed (or set_io is used) then the place-and-route tool will not change its location.

set_initial_io is similar to assigning but not fixing a pin location in PinEdit. The placer will set the I/O location as it is specified in the constraint initially, but if it is required to change the location of that I/O in order to meet the timing constraints, then Designer will take that I/O from its initial location and assign it into an appropriate location.

22) I have instantiated GL33 macros to clock a signal in my design, but after place-and-route the signal is routed as a regular net and the global resources are assigned to some signals that I don't want to be a global. Why does this happen and what is the work around?

By default, Designer software assigns the signals with the highest fanout to the globals regardless of manual GL33 instantiations. To prevent this automatic promotion, users need to use “dont_fix_globals” constraint in the GCF file to turn off the automatic promotion of the nets. By using this constraint, the global resources will be assigned only to the signals that are buffered by global macro in the netlist. As a result, if you turn off the automatic net promotion and you only have used 3 global signals in your design, then one of the global resources will remain unused. To use the extra unused global resources, you can use “set_global” constraint for a particular signal.

23) I have a signal driven by a regular I/O, but I want to assign a global resource to this signal. How can I do this?

You can do it two ways:

Instantiate the GLINT macro and assign your regular I/O to a global network. For example, the following portion of Verilog code will take a reset signal, which is input to a regular I/O, to the global network:

```
IB33 U1 (.PAD(reset) , .Y(reset_int));

GLINT U2 (.A(reset-int), .GL(reset_global));
```

You can use a GCF file and use the constraint set_global in order to promote this signal to global:

```
set_global reset_global;
```

24) Are the ProASIC^{PLUS} global pads bidirectional?

Yes, there are bidirectional buffers such as GLMIOBx for global I/Os. For more information on those macros, please refer to Actel Macro library guide: http://www.actel.com/techdocs/macros/index.html

New Question:

25) When I put a signal on a clock spine, sometimes there is a second signal with a similar name (<netname>_BUF_AUTOnet). What is this, and how do I prevent it from happening again because I want this signal to be in a specific area?

The external I/O pins cannot be connected to the spines directly. If the signal driving the spine is coming from an external input, The Designer software adds a buffer to connect your input to the spine. The output of this buffer, which drives the spine is called "<netname>_BUF_AUTOnet. This will not affect the functionality of your designs, but to prevent this, manually add a buffer after the input.

Design Migration

26) If I want to migrate from a ProASIC to a ProASIC^{PLUS} device, which pins are compatible?

The following tables illustrate the pin difference in ProASIC and ProASIC$^{\text{PLUS}}$ devices.

Pin Differences in FG276

Pin Number	A500K130-FG276 A500K180-FG276 A500K270-FG276	APA150-FG256 APA300-FG256 APA450-FG256 APA600-FG256
H2	I/O	NPECL
H4	I/O	AGND
H14	I/O	NPECL
H15	I/O	AGND
J2	I/O	PPECL
J3	I/O	AVDD
J13	I/O	PPECL
J15	I/O	AGND

Pin Differences in PQ208

Pin Number	A500K50-PQ208 A500K130-PQ208 A500K180-PQ208 A500K270-PQ208	APA150-PQ208 APA300-PQ208 APA450-PQ208 APA600-PQ208 APA750-PQ208 APA1000-PQ208
24	I/O	GL
25	GL	AGND
26	GL	NPECL
27	I/O	AVDD
28	I/O	PPECL
30	I/O	GL
128	I/O	GL
129	I/O	PPECL
131	I/O	AVDD
132	I/O	NPECL
133	GL	AGND

Pin Differences in FG144

Pin Number	A500K50-FG144 A500K130-FG144	APA150-FG144 APA300-FG144 APA450-FG144 APA600-FG144
H2	I/O	NPECL
H4	I/O	AGND
H14	I/O	NPECL
H15	I/O	AGND
J2	I/O	PPECL
J3	I/O	AVDD
J13	I/O	PPECL
J15	I/O	AGND

Pin Differences in FG676

Pin Number	A500K270-FG676	APA600-FG676 APA750-FG676
G10	NC	I/O
G12	NC	I/O
G14	NC	I/O
G16	NC	I/O
G18	NC	I/O
K7	NC	I/O
K20	NC	I/O
N2	I/O	AGND
N5	I/O	NPECL
N24	I/O	NPECL
P2	I/O	AVDD
P5	I/O	PPECL
P20	NC	I/O
P24	I/O	PPECL
P25	I/O	AVDD
P26	I/O	AGND
V7	NC	I/O
V20	NC	I/O
Y10	NC	I/O
Y12	NC	I/O
Y14	NC	I/O

Pin Differences in BG456

Pin Number	*A500K130-BG456 A500K180-BG456 A500K270-BG456	APA150-BG456 APA300-BG456 APA450-BG456 APA600-BG456 APA750-BG456 APA1000-BG456
N3	I/O	AGND
N4	I/O	PPECL
N5	I/O	AVDD
N22	I/O	NPECL
N24	I/O	AVDD
N26	I/O	AGND
P5	I/O	NPECL
P26	I/O	PECL

**The A500K130-BG456 device has lots of NC pins, which are I/O for other devices. Please check the ProASIC^{PLUS} datasheet.*

PLL

27) What type of signals can drive the input of the PLL reference clock?

The input of the PLL reference clock can be:

bidirectional global pads

regular I/O pads

differential LVPECL pair

an internally generated signal

28) What type of signals can the input of the PLL feedback clock be?

The input of the PLL feedback clock can be:

the output of the PLL

regular I/O pads

29) What can the outputs of PLL drive?

The outputs of PLL can drive:

the input of second PLL reference and feedback clock

the global network

the internal nets

regular I/O pads

30) How can I use the Lock function of a PLL to take into account the output of the PLL before it is locked?

Users can employ the "Lock" signal as a soft Enable of the follow-up register driven by GLB and/or GLA. Before it is locked, the output of the PLL will not be propagated to any other logic because the follow up register is not enabled by the Lock signal yet.

Note: If you want to have a follow up register with both Reset and Enable, it has to be implemented in 2 logic tiles.

31) What is the status of the PLL output before the output is locked?

Before the PLL output clock is locked, it still has an output, but the frequency of the output clock is unstable. In order to have a valid clock, you can use the Lock function. Refer to the previous question: "How can I use the Lock function of PLL to take into account the output of the PLL before it is locked?" for details.

32) Can I cascade the PLL?

Yes. Both the primary and secondary output clocks of the first PLL can be used to drive the following up the PLL's reference and feedback input clocks.

33) Do I have to generate multiple PLL macros if the PLLs have same configuration?

No. If you have 2 PLLs that have same configurations like 33 MHz/66 MHz and 44 MHz/88 MHz (Input/Output), you just need to generate 1 PLL macro.

RAM/FIFO

34) What are the functions of DIS and DOS pins of the RAM block?

DIS and DOS pins are used when cascaded RAM blocks are implemented. They are mainly used to control the address of inputs and outputs. RAM blocks generated from ACTgen have these pins connected to corresponding signals automatically. You do not need to worry about these signals.

35) Why can't I find DMUX in the Macro Library Guide?

DMUX is a dedicated high speed MUX for RAM blocks, located in the RAM blocks area. The implementation of DMUX is controlled by the ACTgen tool.

Note: You should not infer DMUX instead of MHX2H for core tile logic.

36) How can I cascade RAMs?

Use ACTgen to generate cascade RAM blocks. ACTgen will handle DIS/DOS pins for cascaded RAMs automatically.

37) How many RAM blocks will be used if I have a RAM256x9 with parity check?

RAM256x9 with parity check will be implemented in 2 RAM blocks.

38) What does M0 or M1 refer to if I have RAM512x9?

M0 refers to the upper half address of the RAM512x9, i.e., address 100000000; M1 refers to the lower half address of the RAM512x9, i.e., address 000000001.

39) How can I have AFULL and AEMPTY flag in the ProASIC^{PLUS} FIFO?

ProASIC^{PLUS} memory blocks don't have AFULL and AEMPTY flags. However, you can use LEVEL signal to generate two internal flags: EQTH and GEQTH. EQTH is true when the FIFO holds the number of words specified by the LEVEL signal. GEQTH is true when the FIFO holds (LEVEL) words or more.

40) What is the limit for the LEVEL signal?

FIFO LEVEL is always 8 bit (both static and dynamic). As a consequence not all values are available, if the FIFO depth is greater than 256. For example:

```
DEPTH = 512, LEVEL can be 0,2,4,6,8 etc.

DEPTH = 1024, LEVEL can be 0,4,8,12,16 etc.
```

Please see "Synchronous/Asynchronous Dual Port FIFO for ProASIC" section on the ACTgen User's Guide for details.

Exemplar Leonardo Spectrum

Library and Settings

41) How do I load the ProASIC<u>PLUS</u> library for Synthesis in LeonardoSpectrum?

From the GUI, select the "Advanced FlowTabs".

Select "Technology Settings"

Select "FPGA/CPLD> Actel > PA".

Click the "Load Library" button.

If you want to use the command line, type "load_library apa."

42) How do I set operating conditions in LeonardoSpectrum?

You can set the operating conditions from the GUI. Select the "Advanced FlowTabs", then the "Technology Settings" item and under "Device," then select operating condition.

Also, you can use command line. For example, you can set worst case conditions for commercial devices using the following commands:

```
Set process COMWCSTD
set temp 70
set voltage 2.3
```

43) How do I change the default fanout setting?

From the GUI, select "Advanced FlowTabs". Then Select the "Advanced Settings" tab, and under "Synthesis Switches" select "Max Fanout" field and enter the Fanout limit.

If you want to use command line, type:

```
set max_fanout_load <fanout_limit>
```

default max fanout load specified in the library.

Select the “Flow” tab. From “Synthesis Switches,” select “Advanced Settings” and unselect “Max Fanout’ and enter the fanout Limit.

The "force_user_load_values" variable, which by default is set to FALSE may be set to TRUE to allow the default setting in the library to be overwritten:

```
set force_user_load_values FALSE
```

44) How do I set the fanout below 10 even though the GUI menu under “Technology => Advanced Settings” gives 10 as the lowest option?

Any number below 10 can be entered directly into the “Max Fanout” field. The corresponding TCL command is:

```
set max_fanout_load <number>; # number can be less than
10
```

45) How do I force LeonardoSpectrum not to optimize a module?

You can set a “dont_touch” or NOOPT attribute for any module.

Example:

```
set_attribute
.work.<entity_name>.<arch_name>.<module_name>
-name NOOPT -value 1

set_attribute .work.ram256x16sa.str_ram256x16sa -name
NOOPT -value 1
```

Or use the “Don’t Touch” Check Box under “Module Constraints” under “Constraints>Module” tab in the GUI.

I/O Insertion

46) How do I insert PADs during synthesis?

There are two ways to infer particular I/O pads during synthesis.

Define the I/O buffer component inside the RTL code and instantiate all the ports with respect to the defined component port.

Use the PAD command:

```
PAD <pad_type> <port_name>
```

Example:

```
PAD OB33PH ready_out
```

47) What are the default pad cells LeonardoSpectrum uses?

Spectrum uses the following cells as default pad cells:

```
IB25U for normal inputs

OB25LL for standard outputs
```

48) How can I change the default pad cell selection?

You can redefine the default I/O pad selection as follows:

```
set default_input_buffer IB33

set default_output_buffer OB33PH

set default_bd_buf IOB33PH

set default_tribuf OTB33PH
```

Exporting Netlist and Forward Constraints

49) How do I export a netlist from LeonardoSpectrum?

You can export a VHDL netlist from the GUI by using the “Output > Output Files” tab and selecting the output file parameters.

Also, you can use the command line. For example, to export the netlist in VHDL 93 format, it requires that the following settings be used in Leonardo to export a VHDL netlist:

```
set vhdl_write_93 TRUE
set vhdl_write_component_package FALSE
set hdl_array_name_style "%s_%d"
auto_write –format vhdl <design_name>.vhd
```

Synplicity® Synplify™

50) How do I set my target library in Synplicity for ProASIC^{PLUS}?

There are two options to set the target device in Synplicity. One is in the GUI and the other is from script (or Synplicity Project file). From the GUI, select “TARGET’ and choose ProASIC^{PLUS} device.

From the Project file, the description is as follows:

```
set_option -technology PROASICPLUS
set_option -part APA750.
```

Note: The Script option is only available in SynplicityPro.

Exporting Netlist and Forward Constraints

51) How can I write forward constraint timing files from Synplicity for timing driven place-and-route in Designer?

By default, Synplicity writes forward constraint SDF files for timing driven place-and-route. By default, Synplicity writes 4000 paths for ProASIC^{PLUS}. You can edit the number of paths in the GUI. Please see Synplicity online help for details.

52) How can I use different I/Os during synthesis in Synplify?

By default, Synplify infers IB33 for Inputs, OB33PH for outputs, and OTB33PH for tristates. If you want to use a different I/O buffer, you have to instantiate the I/O buffer inside the RTL code.

Simulation

53) Where can I find the simulation library file for ProASIC^{PLUS}?

The simulation library file is in the Actel Designer software installation directory:

For Vhdl (vital), it is in:

```
<Designer_install_directory>\lib\vtl\95\apa.vhd
```

For Verilog, it is in:

```
<Designer_install_directory>\lib\vlog\apa.v
```

54) How do I compile the ProASIC^{PLUS} VHDL Vital libraries for ModelTech?

To compile the vital libraries:

Create a library directory with the “vlib” command.

```
vlib apa
```

Map this directory to the library name with the “vmap” command.

```
vmap apa ./apa
```

Compile the three libraries with the “vcom” command.

```
vcom -work apa -explicit apa.vhd
```

55) How do I compile the ProASIC<u>PLUS</u> VHDL Vital libraries for Synopsys VSS?

To compile the libraries:

Create a library directory where you want to compile the libraries.

```
mkdir APA_vss
```

Map this directory to the library name in the .synopsys_vss.setup file.

```
APA: ./APA_vss
```

Compile the three libraries with the vhdlan command.

```
vhdlan -work APA -i apa.vhd
```

56) How do I compile the ProASIC<u>PLUS</u> Verilog library for ModelTech?

To compile the library:

Create a library directory with the vlib command.

```
vlib apa
```

Map this directory to the library name with the vmap command.

```
vmap apa ./apa
```

Compile the library with the vlog command.

```
vlog -work apa ./apa.v
```

57) Why do simulators complain about unbound RAMs and FIFOs in the design when simulating behavioral HDL? What is the limit for the LEVEL signal?

Your design probably contains instantiations of memory netlists generated by Designer. These netlists refer to cells in the simulation libraries. To correct the problem:

Verilog: Refer to the library as you start the simulation.

VHDL: add the following lines to your configuration or to each entity declaration:

```
Library APA;

Use APA.all;
```

Make sure the library is setup properly in the simulator.

58) Why do I get a lot of hold time violations during the behavioral simulation of a design that has ACTgen generated RAMs and FIFOs instantiated?

The memory simulation models of the memories have timing-checks built-in. The simulator will always execute the timing instructions given in the models while checking for setup and hold time violations on the signals in the memory block. The simulator uses unit delays for behavioral code. These unit delays are infinitesimal so they will trigger hold time violations in the memory models. To avoid this, most simulators can be run in a mode where no timing in the library is used. These switches are:

ModelSim:	+notimingchecks
VerilogXL:	+notimingchecks

We are creating new simulation libraries for the ProASIC$^{\underline{\text{PLUS}}}$ device now; the new model should not have this problem.

59) Why do I get hold time violations during the post-layout simulation for a design even when the ProASIC$^{\underline{\text{PLUS}}}$ libraries specify Clock-to-Q delays longer than the hold time of flip-flops and latches?

If the clock signal is not routed on a global network, a hold time violation may occur due to “clock-skew.” In the “import.log” Designer’s report file, check whether all clock signals are routed on global resources. The “import.log” file is in the <design>.dtf directory under the current design project directory.

For more information, call **1.888.99.ACTEL** or visit our website at **http://www.actel.com**

www.actel.com

Actel Corporation
955 East Arques Avenue
Sunnyvale, CA USA 94086
Telephone 408.739.1010
Facsimile 408.739.1540

Actel Europe Ltd.
Maxfli Court, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Telephone +44 0 1276.401450
Facsimile +44 0 1276.401490

Actel Japan
EXOS Ebisu Building 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150, Japan
Telephone +81 0 3.3445.7671
Facsimile +81 0 3.3445.7668