APPLICATION NOTE

Xilinx has acquired the entire Philips CoolRunner Low Power CPLD Product Family.
For more technical or sales information, please see: www.xilinx.com

XAPP324

Ambit - XPLA Designer-XL Design Flow for Philips CPLDs

1999 Jan 29

Philips Semiconductors

PHILIPS

Ambit - XPLA Designer-XL Design Flow for Philips CPLDs

XAPP324

Xilinx has acquired the entire Philips CoolRunner Low Power CPLD Product Family. For more technical or sales information, please see: www.xilinx.com

INTRODUCTION

Philips Semiconductors provides XPLA Designer-XL for use with Ambit's BuildGates[1] at no charge. This allows Cadence users to target Philips CPLDs as large as 960 macrocells. This note discusses the use of Philips XPLA Designer-XL with Ambit's BuildGates.

Philips Semiconductors has developed a family of advanced 3-volt and 5-volt complex programmable logic devices (CPLDs). The XPLA 1 series, designated as the PZ5000 - (5-volt) and PZ3000 (3-volt) series devices, is footprint compatible with the Altera 7000 series devices. The XPLA 2 series consists of the PZ3960 and PZ3320. The principle advantage of Philips CPLDs over all existing CPLDs is that they do not consume static power. The other advantages are 25% higher combinatorial logic capacity and an increased ability to fit designs with fixed pinouts. The XPLA 1 family is in-system programmable and programmable on Data I/O and BP Microsystems programmers. The XPLA 2 family uses SRAM to store its configuration memory, and is configured similarly to existing FPGAs.

Philips offers XPLA Designer-XL and XPLA Professional. XPLA Designer-XL is used to interface to third party tools, including BuildGates. XPLA Designer-XL is available on Sun and HP workstations. XPLA Designer-XL can be obtained from the Philips website http://www.coolpld.com, ftp site ftp://www.coolpld.com, and is available on cdrom.

The following documentation is related to this note.

BuildGates User Guide

Complex Programmable Logic Devices Data Book IC27 (1998 edition)

XPLA Designer User's Guide

Technical support is available from

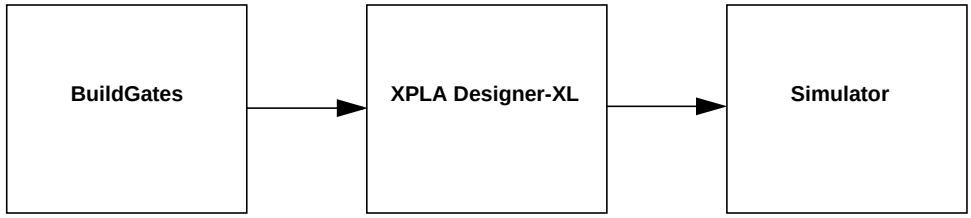
coolpld@abq.sc.philips.com

(888) coolpld

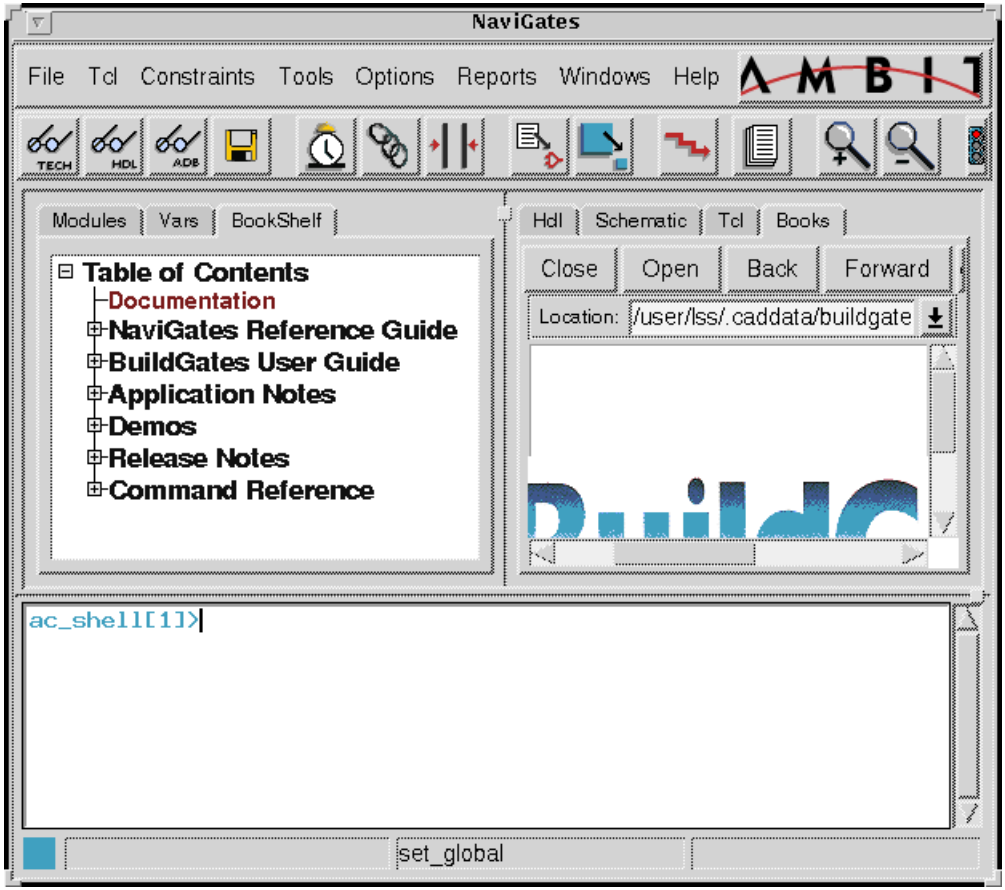
http://www.coolpld.com or ftp://www.coolpld.com

DESIGN FLOW

The following sections describe how to use XPLA Designer-XL with BuildGates. BuildGates is invoked first and an gate-level verilog netlist is created. This netlist can be directly input into XPLA Designer. Alternately, a Philips tool dino can be used to translate the gate-level Verilog to edif. Then XPLA Designer-XL accepts ether the edif or gate-level Verilog input file to create the jedec, report, and simulation files.



(1) Philips recognizes the copyrights and trademarks of Cadence, Inc., including BuildGates.



BuildGates can be used from its GUI as shown above for the ppg design or from the unix prompt. This application note uses the following script for the ppg design to synthesize Verilog to a gate level Verilog netlist.

```
read_ver ppg.v
do_build_generic
set_clock_propagation ideal
set_clock clock -period 10.0 -waveform {0.0 5.0}
set_global echo_commands true
set_global clock_gating_to_be_checked false
set_global buscomp_generator “%s_%d”
set_global buscomp_generator “%s_%d”
set_global fix_multiport_nets true
# set_global no_buffer_at_integraton_level true
```

```
set_clock_arrival_time -rise 0.1 -fall 5.1 -clock clock CLK50
set_data_arrival_time 0.0 -clock clock [find -port -no_clock -input *]
set_data_required_time 8.4 -clock clock [find -port -output *]
read_alf ps.alf
set_global target_technology ps
set_current_module PPG
set_top_timing_module PPG

do_optimize -no_design_rule
do_xform_optimize_slack -effort high

# slowly fix slew
set i 1
while { $i <= 3} {
set_global depth_for_fast_slew_prop $i
do_xform_optimize_slack -effort medium -dont_reclaim_area -incremental
incr i
}
# fix hold without disturbing setup slack
do_xform_timing_correction -preserve_slack -fix_hold -dont_reclaim_area -incremental

do_dissolve_hierarchy -hier
report_hierarchy
report_area -cells
report_timing -summary > ppg.rpt
write_verilog ppg.v.net
exit
```

This script is run at the unix prompt with

```
> ac_shell -f ppg.cmd | tee log
```

The most important architectural considerations to keep in mind when writing code to target Philips CPLDs are

1. The flip flop can have an asynchronous reset or an asynchronous preset. Both cannot be asynchronous. If both are needed, make one synchronous.
2. Latches are implemented in combinatorial logic in the XPLA 1 and XPLA 2 families.
3. The first generation devices in the XPLA 1 family had a limited number of clocks. If a large number of clocks are needed, select an enhanced clocking device.
4. If the design uses state machines and is a very tight fit, use binary encoding in the FSM.

The files ps.alf and ps.dino are provided on the ftp site ftp://www.coolpld.com/software/ambit/. These files are also distributed with XPLA Designer-XL in <xpla_install_path>/lib/ambit. The ps.alf library should be installed as a BuildGates library.

Edif flow

The edif flow is currently available only to users who have access to the **dino** tool. After generating the gate level netlist, copy $XPLA_PATH/lib/ambit/ps.dino to the project directory and enter

dino ppg.v.net -lib ps.dino -edif ppg.edf

Verilog flow

The Verilog flow requires the user to do the following:

1. Copy $XPLA_PATH/lib/ambit/ps.v to the project directory.

2. Add ‘include “ps.v” at the beginning of the Verilog design file.

3. Use “-it verilog” as the argument in the control file (discussed later)

Sun and HP Workstation Design Flows

XPLA Designer-XL is provided for the Sun an HP workstations in the xpla2_solaris.tar.Z and xpla2_hpux.tar.Z files.

Installing XPLA Designer-XL on a Sun or HP workstation

(1) Create an install directory for XPLA Designer-XL, such as /export/home1/xpla.

(2) Uncompress the xpla2_solaris.tar.Z (xpla2_hpux.tar.Z) file to generate an xpla2_solaris.tar (xpla2_hpux.tar) archive file. Extract (tar xvf xpla2_solaris.tar) the files and subdirectories. The install directory should contain the following directories and files: README.TXT, examples, xpla2_solaris.tar, bin, and lib.

(3) Set the environment variables XPLA_PATH to the XPLA Designer-XL directory in your .cshrc or .kshrc file.

C Shell :

% setenv XPLA_PATH /export/home1/xpla

Korn Shell :

% export XPLA_PATH=/export/home1/xpla

Add the $XPLA_PATH/bin in the PATH environment variable

(4) To test the XPLA Designer-XL installation, copy the Manchester Encoder (me) design files to a test directory.

cd $XPLA_PATH/examples

mkdir install_test

cp me.* install_test

cd install_test

xsh me (this should produce jedec,fit,tim files)

cd ..

rm -r install_test

Using XPLA Designer-XL on a workstation

XPLA Designer-XL runs at the Unix prompt. Unlike the PC edition, there is no GUI.

1. Using BuildGates and targeting Philips, create either <design>.edf or <design>.v.
2. Copy an existing control file to the project directory, such as

   cp $XPLA_PATH/examples/template.ctl <design>.ctl

3. Edit <design>.ctl, replacing the existing design name “foo” with the name of the design being compiled.
4. In <design>.ctl, edit -dev <device> to the target appropriate device. Enter xsh (xsh2) with no argument to obtain the XPLA 1 (XPLA 2) device list. The 1998 Complex Programmable Logic Devices IC27 provides device nomenclature. Several generations of XPLA 1 have been produced. It is recommended that enhanced clocking devices, designated by the letter C or A following the macrocell count (e.g. PZ3032CS10BC), be used when possible.
5. At the command prompt, enter xsh <design> (without suffix) if targeting an XPLA 1 device. Enter xsh2 <design> to target an XPLA 2 device. This produces the jedec, timing, and results files.
6. If needed, vary the maximum Product Terms per equation parameter -**th** <number> in <design>.ctl, where number varies between 5 and 37. This can have a significant effect on speed/area results. Running three iterations, with **th** set at 10, 20, and 30, and reviewing xplaopt results for utilization and number of delay levels, is usually sufficient. This is illustrated in the results provided by XPLA Designer-XL below for three values of the **th** parameter. This shows that using PLA sharing can reduce the macrocell count and number of delay levels. It should be noted that setting **th** to a high value is not always constructive.

   th 10
   %% Network final ======> 101 Mcells, 36 PLApts, 242 PALpts, 4 Levels
   th 20
   %% Network final ======> 94 Mcells, 218 PLApts, 231 PALpts, 4 Levels
   th 30
   %% Network final ======> 85 Mcells, 170 PLApts, 190 PALpts, 2 Levels

Format of the control file

A control file must be named the same as its design name, with the extension of “.ctl”

The ‘#’ character at the beginning of a line is used to indicate a comment.

The control file contains up to 3 sections: command, property, and pin_assignment. The only required lines to edit in a control file template.ctl are -i <file> and -dev <device> in the [command] section. The **th** parameter is commonly varied, and the **pre** parameter and the [pin_assignment] section are generally used. For hard to fit designs, increase **bfi**. The grouping of signals in the [property] section are seldom needed to meet fit requirements.

COMMAND SECTION ([command])

-i <design>.[edf | v | phd] Required field used to specify input filename

-it <edif | verilog | phdl> Required field used to specify input type

-th <number> Specify max pterm for each equations (default is 11 - range is 5 - 37)

-fi <number> Specify max fanin for each equation (default for 22v10 is 22; default forlarger devices is 36 - range is 5 - 37)

-bfi <number> Specify max fanin for each logic block (default is 36 - range is 36 - 40)

-vho Directs fitter to generate delay-annotated VHDL simulation model (default is to not generate VHDL model)

-vo Directs fitter to generate delay-annotated Verilog model (default is to not generate Verilog model)

-reg Apply register synthesis to the design

-co <best | none> Specify collapsing method (default is best)

-effort <exhaust | fast> Specify synthesis effort (default is fast)

-xor <all | exp | none> Specify xor synthesis type (default is none) - XPLA 2 devices only

-dev <device> Required field used to specify device to target

-pre <keep | try | ignore> Specify pin assignment effort by fitter: (default is try). If keep is specified, the fitter will not complete a fit if the pin assignments cannot be kept.

PROPERTY SECTION ([property])

-maxpt <signal> Specify maximum pterm for specified pin/node

-keep <signal> Directs optimizer to keep (do not optimize out) specified signal

-retain <signal> Directs optimizer to retain (do not optimize out) specified signal

-dut on Activates the global tristate pin GTS(N) (default is disabled, allowing the pin to be used as a user input)

-isp off Disables the 4 JTAG pins use for in system programming. When off, the device can be programmed through the JTAG pins once. This allows the JTAG pins to be used for user I/O. (default is on)

-tri-state all Tristates all pins

-fm_group <signals> Directs fitter to group specified signals within a fast module (XPLA 2 only)

-lb_group <signals> Directs fitter to group specified signals in the same logic block.

-slow_slew_rate <outputs> Enables the slow slew rate output buffer on specified outputs . (default is fast slew rate) XPLA 2 only

PIN ASSIGNMENT SECTION ([pin_assignment])

[pin_assignment]

<signal1>:<pin_number>

<signal2>:<pin_number>

To keep pin assignments, enter

-pre keep in the [command] section.

If the fitter report produces the following message(s),XPLA Designer has changed the signal names slightly.

> WARNING 3271: 'UVI2' found in pin file but not in design file, ignored.

This occurs when busses which are originally named din[4] are renamed renamed by XPLA Designer-XL to din_4_. In some cases, the case of a signal name may change. To work around this, run the design using -**pre** ignore, look at the signal names in the fit (or spf) report, and revise the signal names in the .ctl file to match those in the .fit file.

Control File template

[command]
-it edif
-i foo.edf

-effort exhaust
-dev pz3032cS10a44
-reg
-th 20
-bfi 36
-pre keep
[property]
#dut on
#isp off
[pin_assignment]
bit0:4
bit1:5
bit2:6

Simulation

The design.vo and design.vho files are delay-annotated Verilog and VHDL models intended to be used in timing simulation. In these simulation models, busses are broken into discrete signals, so the testbench for the behavioral code may require revision.

Using the Manchester encoder in the $XPLA_PATH/example directory, run a timing simulatiion in Verilog-XL by

verilog me.vo me_tb.v

Run a VHDL timing simulation in QuickVHDL by

qhlib work

qhmap work “./work”

qvhcom me.vho

qvhcom me_tb.vhd

qhsim testbench v1

NOTES

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088–3409
Telephone 800-234-7381

Let's make things better.

Philips Semiconductors

PHILIPS