Generation of a .vcd-File for PowerPlay on Arria 10 Designs

G. Knittel MPIfR Bonn, Germany gknittel@mpifr-bonn.mpg.de Ver. 1.1 Sep. 11, 2015

1 MOTIVATION

It seems that Quartus II 15.0 (Q15) does not support gate-level simulation for Arria 10 designs, neither functional nor timing. At least, to be precise, Q15 does not support these operations via NativeLink to the ModelSim-Altera Starter Edition. For an accurate power consumption estimation using PowerPlay, a precise description of toggle activities per node is required. These are normally recorded during gate-level simulation, which is not supported. Thus, one has to resort to less accurate default values.

This application note describes how to obtain a .vcd-file (*Value Change Dump*) despite this restriction. However, it is derived from functional simulation only, and might therefore not capture all transitions. The hope is that this is still more aligned with the physical behavior of the chip than a default toggle rate.

Note: a service request has been filed with Altera on Sep. 2, 2015, on this issue. No answer has been received so far, but any information from Altera may void this application note.

2 **PROCEDURE**

The steps are outlined using a small test design. Assumed we have set up a Q15 project for an Arria 10 10AX115U4F45I3SGES part with just one VHDL-file:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
: in std_logic_vector (23 downto 0);
: out std_logic_vector (23 downto 0) := x"000000");
          С
          d
end entity;
architecture rtl of tripleadd is
  signal areg, breg, creg : std logic vector(23 downto 0) := x"000000";
begin
process (clk)
begin
  if (rising_edge(clk)) then
    areg <= a;
breg <= b;</pre>
    creg <= c;
          <= areg + breg + creg;
     d
  end if;
end process;
end rtl;
```

Using Processing \rightarrow Start \rightarrow Start Test Bench Template Writer we generate a test bench skeleton and fill it arbitrarily:

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY tripleadd vhd tst IS
END tripleadd vhd tst;
ARCHITECTURE tripleadd_arch OF tripleadd_vhd_tst IS
SIGNAL clk : STD_LOGIC := '0';
SIGNAL a, b, c : STD_LOGIC_VECTOR(23 DOWNTO 0) := x"0000000";
SIGNAL d : STD_LOGIC_VECTOR(23 DOWNTO 0);
COMPONENT tripleadd
PORT ( clk : IN STD_LOGIC;
                    IN SID_LOGIC;
: IN STD_LOGIC_VECTOR(23 DOWNTO 0);
: IN STD_LOGIC_VECTOR(23 DOWNTO 0);
: IN STD_LOGIC_VECTOR(23 DOWNTO 0);
: OUT STD_LOGIC_VECTOR(23 DOWNTO 0));
              а
              b
              С
              d
END COMPONENT;
BEGIN
clk <= NOT clk AFTER 25 ns;
inst1 : tripleadd PORT MAP ( a => a, b => b, c => c, clk => clk, d => d );
always : PROCESS (clk)
BEGIN
   IF (FALLING EDGE(clk)) THEN
      a <= a + 1;
b <= b + 3;
   c <= c + 5;
END IF;
END PROCESS always;
END tripleadd arch;
```

For NativeLink we need to fill in some forms. Afterwards the "Settings"-form should look like shown below. Note that we are using the built-in Altera-ModelSim Starter Edition.

🖌 Settings - tripleadd	Pastron Longue Agreement, or other	
Category:		Device
General	Simulation	
Files Libraries	Specify options for generating output files for use with other EDA tools.	
IP Settings		
IP Catalog Search Locations	Tool name: ModelSim-Altera	▼
Design Templates Operating Settings and Conditions	Run gate-level simulation automatically after compilation	
Voltage Temperature	EDA Netlist Writer settings	
 Compilation Process Settings 	Eormat for output netlist: VHDL Time scale: 100 us	
Incremental Compilation	Output directory: simulation/modelsim	
Design Entry/Synthesis Simulation	Map illegal HDL characters	
Formal Verification	Options for Power Estimation	
Board-Level Compiler Settings	<u>G</u> enerate Value Change Dump (VCD) file script Script Settings	
VHDL Input Verilog HDL Input	Design instance name:	
Default Parameters		
TimeQuest Timing Analyzer Assembler	More EDA Netlist Writer Settings Click this But	ton
Design Assistant SignalTap II Logic Analyzer	NativeLink settings	
Logic Analyzer Interface	○ N <u>o</u> ne	
PowerPlay Power Analyzer Settings SSN Analyzer	<u>Compile test bench:</u> tripleadd_vhd_tst	Test Benches
	Use script to set up simulation:	
	Script to compile test <u>b</u> ench:	
Check this Radio Button		
	More NativeLink Settings	Reset
	OK Cancel App	ply Help
1		

Figure 1: Settings for Simulation

Clicking on the "Test Benches..."-Button will open the following dialog box (shown after making changes):

isting test bench	settings:				<u>N</u> ew	1
Name	Top Level Module	Design Instance	Run For	Test Bench File(s)		
ripleadd_vhd_tst	tripleadd_vhd_tst	inst1	100 us	simulation/modelsim/tripleadd.vht	Edit	
					Delete	

Figure 2: Test Bench Settings

Clicking on the "New.."-Button will open the final dialog box for this step:

	New Test Bench Settings Create new test bench settings.	
	Test bench name: tripleadd_vhd_tst	
	Top level module in test bench: tripleadd_vhd_tst	
	✓ Use test bench to perform VHDL timing simulation	
	Design instance name in test bench: inst1	Enter path and click "Add"
Check this box and make sure	Simulation period	1
to enter the instance name. Otherwise errors might	 ○ Run simulation until all vector stimuli are used ● End simulation at: 100 us ▼ 	
occur during simulation.	Test bench and simulation files File name: Add	
	File Name Library HDL Version Remove	
	simulation/modelsim/tripleadd.vht	
	Down	
	Properties	
	OK Cancel Help	

Figure 3: New Test Bench Settings

We could now perform an RTL simulation.

However, our goal is to run gate-level simulations so we need to make further changes. These relate to the EDA Netlist Writer. First we need to instruct it to generate a tcl-script which in turn instructs the simulator to record toggle activities of all nodes. Further we instruct it to generate a .do-script for a (non-existing) third-party simulation tool. After all the changes the dialog boxes look like shown in Figure 4 through Figure 6.

🥐 Settings - tripleadd	
Category:	Device
General Files Libraries IP Settings IP Catalog Search Locations Design Templates Operating Settings and Conditions Voltage Temperature Compilation Process Settings Incremental Compilation EDA Tool Settings Design Entry/Synthesis Simulation Formal Verification Board-Level Compiler Settings VHDL Input Verilog HDL Input Default Parameters TimeQuest Timing Analyzer	Simulation Specify options for generating output files for use with other EDA tools. Tool name: ModelSim-Altera Rung gate-level simulation automatically after compilation EDA Netlist Writer settings Format for output netlist: VHDL Time scale: 100 us Map illegal HDL characters Map illegal HDL characters Generate Value Change Dump (VCD) file script Script Settings Design instance name: inst1 Again make sure to enter the proper instance name
Default Parameters	More EDA Netlist Writer Settings NativeLink settings Ngne © Compile test bench: Lyse script to set up simulation: Script to compile test bench: More NativeLink Settings More NativeLink Settings OK OK Close Apply

When clicking on "Script Settings", a dialog box should pop up:

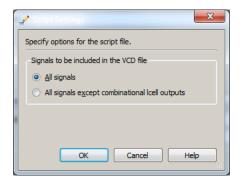


Figure 5: VCD File Script Settings

After clicking on "More EDA Netlist Writer Settings", change the option "Generate third-party EDA tool command script for gate-level simulation" from Off to On.

-	More EDA Netlist Writer Settings	X	
	specify the settings for the EDA Netlist Writer options in your project.		
	<pre></pre>	<u>S</u> how: All ▼	
	Name:	Setting:	Change this Option
	Architecture name in VHDL output netlist	structure	
	Bring out device-wide set/reset signals as ports	Off	
	Disable detection of setup and hold time violations in the input registers of bi-directional pins		
	Do not write top level VHDL entity	Off	
	Flatten buses into individual nodes	Off	
	Generate netlist for functional simulation only	Off	
	Generate third-party EDA tool command script for RTL functional simulation	Off	
	Generate third-party EDA tool command script for gate-level simulation	On A	
	Location of user compiled simulation library	<none></none>	
	Maintain hierarchy	Off	
	Truncate long hierarchy paths	Off	
	Description:		
	Directs the EDA Netlist Writer to generate a command script to run gate-level simulation with tool.	a third-party EDA Reset	
		K Cancel Help	

Figure 6: More EDA Netlist Writer Settings

Now the Q15 tools "Analysis & Synthesis", "Fitter (Place and Route)" and "EDA Netlist Writer" must be executed. Before we do this, we should set up a minimal constraints file defining the clock and a few delays (so that the Timing Analyzer won't complain):

```
# Create a simple 50ns clock
create_clock -period 50 -waveform {0 25} -name clk [get_ports clk]
set_input_delay -clock { clk } 10 [get_ports {a[*]}]
set_input_delay -clock { clk } 10 [get_ports {b[*]}]
set_input_delay -clock { clk } 10 [get_ports {c[*]}]
set_output_delay -clock { clk } 10 [get_ports {d[*]}]
```

Additionally we might then want to make all I/O assignments. Now we can step through the tool flow above.

While running the EDA Netlist Writer will generate the following warning:

Warning (10905): Generated the EDA functional simulation files although EDA timing simulation option is chosen.

This is an indication that no timing-information (.sdo-files) have been generated. Trying to run a gate-level simulation (Tools \rightarrow Run Simulation Tool \rightarrow Gate Level Simulation) results in the following error message:

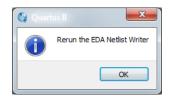


Figure 7: Error Message

The work-around is as follows.

The EDA Netlist Writer should have written (among others) two files:

- simulation/modelsim/tripleadd_dump_all_vcd_nodes.tcl, defining the nodes that should be monitored by the simulator, and
- simulation/modelsim/tripleadd_run_msim_gate_vhdl.do, which is the simulation script for ModelSim.

In this example the simulation script is as follows:

```
transcript on
if {[file exists gate_work]} {
vdel -lib gate_work -all
}
vlib gate_work
vmap work gate_work
vcom -93 -work work {tripleadd.vho}
vcom -93 -work work {<... your path ...>/TripleAdd/simulation/modelsim/tripleadd.vht}
vsim -t lps +transport_int_delays +transport_path_delays -sdftyp /inst1=tripleadd_vhd.sdo -L altera
-L altera_lnsim -L twentynm -L twentynm_hssi -L lpm -L sgate -L twentynm_hip -L gate_work -L work
-voptargs="+acc" tripleadd_vhd_tst
source tripleadd_dump_all_vcd_nodes.tcl
add wave *
view structure
```

view signals run 100 us

We can see that the .tcl-script for writing the .vcd-file will be executed. However, in the absence of any timing information we need to remove all timing-related options from the vsim command line. These are marked red.

After editing the command line we need to save it as a separate file, for example as

simulation/modelsim/tripleadd_run_msim_gate_vhdl_own.do.

In principle we can set the option in Figure 6 to "Off" again.

We can then instruct the system to use this script whenever a simulation run is desired, as shown in Figure 8. The effect of this check box is that the simulation scripts that are generated by the tools now include a line at the end to execute our own simulation script. For example, the script for doing an RTL simulation now looks like the following:

```
transcript on
if {[file exists rtl_work]} {
vdel -lib rtl_work -all
}
vlib rtl_work
vmap work rtl_work
vcom -93 -work work {<... your path ...>/TripleAdd/tripleadd.vhd}
vcom -93 -work work {<... your path ...>/TripleAdd/simulation/modelsim/tripleadd.vht}
vsim -t lps -L altera -L lpm -L sgate -L altera_mf -L altera_lnsim -L twentynm hssi -L twentynm_hip
-L rtl_work -L work -voptargs="+acc" tripleadd_vhd_tst
do <... your path ...>/TripleAdd/simulation/modelsim/tripleadd run msim gate vhdl own.do
```

This means that after performing an RTL simulation, which the system allows us to do, automatically a functional gate-level simulation will follow. We could delete all lines except the last one to save the RTL simulation step, but the system will complain about a modified script before each run, which is not very convenient.

So, after re-executing the Quartus tool flow (Synthesis, Fitter, EDA Netlist Writer) we can then perform an RTL simulation, and implicitly a gate-level simulation, which was not possible before. After quitting ModelSim we will find a .vcd-file at

simulation/modelsim/tripleadd.vcd.

🖌 Settings - tripleadd	and and a second s	
Category:		Device
General Files	Simulation	
Libraries IP Settings	Specify options for generating output files for use with other EDA tools.	
IP Catalog Search Locations Design Templates	Tool name: ModelSim-Altera	_
 Operating Settings and Conditions Voltage 	Run gate-level simulation automatically after compilation	
Temperature Compilation Process Settings	EDA Netist Writer settings Format for output netist: VHDL Time scale: 100 us	
Incremental Compilation EDA Tool Settings	Eormat for output netilist: VHDL Time scale: 100 us Output directory: simulation/modelsim	
Design Entry/Synthesis Simulation	✓ Map illegal HDL characters ✓ Enable glitch filtering	
Formal Verification Board-Level	Options for Power Estimation	
 Compiler Settings VHDL Input 	Generate Value Change Dump (VCD) file script Script Settings	
Verilog HDL Input Default Parameters	Design instance name: inst1	
TimeQuest Timing Analyzer Assembler	More EDA Netlist Writer Settings	
Design Assistant SignalTap II Logic Analyzer	NativeLink settings	
Logic Analyzer Interface PowerPlay Power Analyzer Settings		
SSN Analyzer	<u>Compile test bench</u> : tripleadd_vhd_tst <u>V</u> Use script to set up simulation: simulation/modelsim/tripleadd_run_msim_gate_vhdl_own.do	st <u>B</u> enches
	Script to compile test bench:	
Check this have and	More NativeLink Settings	Reset
Check this box and select own gate-level		Keset
script		
	OK Cancel Apply	Help

Figure 8: Settings with own .do-Script

Note: during gate-level simulation the Altera-ModelSim Starter Edition reports (for this small design already)

** Warning: Design size of 40546 statements exceeds ModelSim Altera Starter recommended capacity. # Expect performance to be adversely affected.

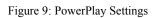
Thus for larger designs no guarantee can be given that this method will work.

Using this .vcd-file we can now configure PowerPlay. The "Settings" dialog box should then look like in Figure 9. While running PowerPlay it will report:

Info (222002): Starting scan of VCD file simulation/modelsim/tripleadd.vcd (0 ns to End of File) for signal
static probabilities and transition densities
Info (222003): Finished scan of VCD file simulation/modelsim/tripleadd.vcd (0 ns to End of File) for signal
static probabilities and transition densities

When PowerPlay has finished execution it will report a high "Power Estimation Confidence", which was the goal of this exercise (see Figure 10a). When using a default toggle rate instead, the confidence is low: "user provided insufficient toggle rate data" (see Figure 10b). Note also the fairly large dynamic power estimate differences.

🥜 Settings - tripleadd		- 0 X
Category:		Device
General	PowerPlay Power Analyzer Settings	
Files Libraries	Select the power analyzer options.	
IP Settings		
IP Catalog Search Locations	Run PowerPlay Power Analyzer during compilation	
Design Templates Operating Settings and Conditions	✓ Use input file(s) to initialize toggle rates and static probabilities during power analysis	
Voltage	Input File(s)	
Temperature Compilation Process Settings	File Name Type Entity VCD Start Time (CD End Time	<u>A</u> dd
Incremental Compilation	simulation/modelsim/tripleadd.vcd Value Change Dump File tripleadd	Edit
 EDA Tool Settings Design Entry/Synthesis 		Edit
Simulation		<u>R</u> emove
Formal Verification		
Board-Level Compiler Settings		
VHDL Input		
Verilog HDL Input Default Parameters		
TimeQuest Timing Analyzer		
Assembler Design Assistant		
SignalTap II Logic Analyzer		
Logic Analyzer Interface		
PowerPlay Power Analyzer Settings SSN Analyzer	Perform glitch filtering on VCD files	
	Write out signal activities used during power analysis	
	Output file name:	
	Write signal activities to report file	
	Write power dissipation by <u>b</u> lock to report file	
	Default toggle rates for unspecified signals	
	Default toggle rate used for input I/O signals: 12.5 %	
	Default toggle rate used for remaining signals	
	Use default value: 12.5 %	
	Use vectorless estimation	
	OK Cancel Apply	Help



PowerPlay Power Analyzer To	ol 🛛			
PowerPlay Power Analyzer Summary				
PowerPlay Power Analyzer Status	Successful - Thu Sep 10 17:01:59 2015			
Quartus II 64-Bit Version	15.0.0 Build 145 04/22/2015 SJ Full Version			
Revision Name	tripleadd			
Top-level Entity Name	tripleadd			
Family	Arria 10			
Device	10AX115U4F45I3SGES			
Power Models	Preliminary			
Total Thermal Power Dissipation	2205.12 mW			
Transceiver Standby Thermal Power Dissipation	0.00 mW			
Transceiver Dynamic Thermal Power Dissipation	0.00 mW			
I/O Standby Thermal Power Dissipation	4.25 mW			
I/O Dynamic Thermal Power Dissipation	1.14 mW			
Core Dynamic Thermal Power Dissipation	1.62 mW			
Device Static Thermal Power Dissipation	2198.11 mW			
Power Estimation Confidence	High: user provided sufficient toggle rate data			

a.) Using the .vcd-file

PowerPlay Power Analyzer To	ol 🛛		
PowerPlay Power Analyzer Summary			
PowerPlay Power Analyzer Status	Successful - Thu Sep 10 17:29:58 2015		
Quartus II 64-Bit Version	15.0.0 Build 145 04/22/2015 SJ Full Version		
Revision Name	tripleadd		
Top-level Entity Name	tripleadd		
Family	Arria 10		
Device	10AX115U4F45I3SGES		
Power Models	Preliminary		
Total Thermal Power Dissipation	2206.32 mW		
Transceiver Standby Thermal Power Dissipation	0.00 mW		
Transceiver Dynamic Thermal Power Dissipation	0.00 mW		
I/O Standby Thermal Power Dissipation	4.21 mW		
I/O Dynamic Thermal Power Dissipation	1.97 mW		
Core Dynamic Thermal Power Dissipation	1.98 mW		
Device Static Thermal Power Dissipation	2198.15 mW		
Power Estimation Confidence	Low: user provided insufficient toggle rate data		

b.) Using a default toggle rate of 12.5%

Figure 10: PowerPlay Power Analyzer Summary