Automated FPGA Power Characterization Methodology

Dalia A. El-Dib, Yousra Alkabani, Magdy El-Moursy, and Heba A. Shawkey

***Abstract*— Power characterization methodology for Field Programmable Gate Arrays (FPGAs) is described. The methodology includes putting an FPGA through a pre-determined set of measurements upon which a few calculations are needed to reach real leakage power and average dynamic power consumption of all FPGA units. Look Up Tables (LUTs), wires, and Flip Flops (FFs) are included in the presented methodology. The characterized power can be used to estimate the total power consumption of any design based on its total FPGA usage at a very early stage of the design. Since the characterized power is determined using direct measurements, these are claimed to represent a more accurate measure than those obtained by currently available simulation tools which give power values up to ten times higher than actual measured ones.**

I. Introduction

At the beginning of the design cycle the system architecture is chosen including whether to use FPGAs or ASICs for the design. It is necessary to quantify the differences between different technologies in terms of cost, performance and power consumption [1]. Simulation tools for ASICs are highly mature knowing the technology parameters of the desired chip. However, power/energy models for FPGA devices lack the needed accuracy for efficient power optimization [2]. In current FPGA CAD tools, power optimized compilation, is not even an option [1]. This work aims to provide an automated tool for FPGA power characterization to give more accurate power estimation models at a high level of abstraction. This in turn can help in optimizing the power/energy efficiency of FPGA designs.

Power consumption in digital CMOS circuits has many sources, but is measured in two states, switching and static. Static power is measured when the circuit is not in the process of switching and is essentially called leakage power. The dynamic power is the primary power dissipated during the process of switching. The distribution of power dissipation among the components of FPGA varies by design density and FPGA type. However, the main contributors to power dissipation are the routing resources, logic and clock distribution network. Some detailed estimates for power shares of different components of many FPGA families were attempted [3] showing that the routing power share is usually the biggest among all power dissipation components.

Accurate power modeling is necessary for power optimization in a design. In general, power characterization techniques of FPGA devices could be classified into technology dependent and technology independent techniques. Technology independent techniques are divided into measurement based techniques [2][4][5][6][7], non-measurement based techniques [8][9], and mixed techniques [3]. Jevtic and Carreras presented a methodology for estimating dynamic power of Virtex II with no knowledge of the proprietary information about the chip layout [2][7]. Many different designs are downloaded to the FPGA and simulation vectors are generated and fed to the design then the power is directly measured from the chip. Afterwards, some analysis is conducted to extract the capacitance values for all on chip resources. These values can then be used to optimize future designs for the same FPGA chip. In order to do that, two FPGA boards are needed, one to generate the simulation vectors and the other to download the designs and to measure power dissipation. However, leakage power measurements showed that individual FPGA boards might differ in their leakage power characteristics as shown in the paper. Thus, it is highly desirable to use only one FPGA to do the whole characterization procedure. Since multipliers are complex and power hungry they are used as good measure for power dissipation. Oliver et al. generated the inputs to the multipliers on the FPGA using an auxiliary circuit implemented on the same FPGA. Their input generator is based on a linear feedback shift register [4]. In this case, the only external input to the FPGA is the onboard clock signal. The power consumption of the shift register is negligible if compared to the benchmark circuit. However, the work presented here aims to completely avoid the generation of input simulation vectors. Most importantly, this work aims to outline a new methodology capable of defining the power dissipation of different components of any FPGA.

Section II outlines the proposed FPGA power characterization considerations and setup. Then, the proposed characterization procedure for FPGA is outlined in section III followed by measurement results in section IV. Finally, some conclusions are presented in section V.

II. FPGA Power Characterization Considerations

The proposed technique for FPGA power characterization is automated and is design independent which makes it efficient to characterize any FPGA. The proposed methodology is based on downloading designs with a variable number of

Dalia El-Dib is Visiting Scholar with Dalhousie University, Department of Electrical and Computer Engineering, Halifax, NS, Canada. She is also with Electronic Research Institute, email: dafeldib@alumni.uwaterloo.ca.
Yousra Alkabani is with the Deptartment of Computer and Systems Engineering, Ain Shams University, Cairo, Egypt, email: yousra.alkabani@eng.asu.edu.eg.
Magdy El-Moursy is with the Design Creation Devision, Mentor Graphics Corporation, Cairo, Egypt. He is also with Electronics Research Institute, email: magdyaelmoursy@gmail.com.
Heba Shawkey is with Electronics Research Institute, email: heba shawkey@eri.sci.eg.

connections, Configurable Logic Blocks (CLBs), LUTs, Registers and/or IOBs, etc.. . The actual power consumption is measured and the power consumption of each circuit component is determined. Based on the power of each component, the power consumption of any design is determined. Neither extra board nor internal custom circuit are needed to generate the simulation vectors. Onboard clock generator is used as sole input for different testing designs.

Benchmarks to estimate power consumption of FPGAs are hardly available. Due to their complexity, different types of multipliers with and without pipelining and/or targeting both embedded multipliers blocks or entirely in LUT implementation [4] were previously used as testbenches. When estimating the power using simulation tools, it is best to apply realistic input vectors to the testbench and measure the average power consumption. When input vectors are not available, which is the case with most testbenches, one can assume that all nets in the design toggle at the same rate and have the same static probability [1]. Oliver and Boemo claimed that this gives power measurements on par with those measured using proper testbenches. Jevtic et al. used two FPGA boards, one to generate the input simulation vectors and one for the circuit implementation and power measurement [2].

The work presented in this paper adopts actual measurements instead of simulation results. The suggested work avoids the generation of input simulation vectors. Instead, circuits that are composed of either multiple counters or multiple simple logic circuit units are used. Counters eliminate the need for any input vector generation circuits except for the clock. Simple logic circuit units are placed in a successive arrangement, where each unit receives inputs from the previous stage and generates output to the next stage. The first unit receives input only from the onboard clock and its divisions and multiples. If designed carefully these logic units toggle at the same rate, have the same number of inputs, and have the same number of outputs. Both settings, multiple counters and multiple simple logic circuit units, serve the need for the repetition of identical power consuming modules to reach accurate power consumption estimates. The power characterization procedure which is described in this work is implemented on Xilinx Spartan3AN FPGA, because it provides easy means of measuring onboard power consumption.

III. FPAG Power Characterization Methodology

The variation in static/dynamic power among many FPGA boards is studied by implementing one design on many identical FPGA boards, measuring the dynamic and static power and comparing them among the described different FPGA boards. The power characterization of various FPGA components are also studied following the procedure described. Power characterization for LUTs, FFs and direct wires are considered. The required testbench designs along with the relevant applied equations and calculations are described.

A. CLBs as LUT-FF pairs

Two testbench circuits are designed with each configured CLB to utilize only one LUT and one FF. Then, the CLBs are placed as shown in Fig.1 and Fig.2.

The circuits satisfy the following conditions:

- The CLBs should be placed next to each other and connected using direct lines only.
- Different sizes of the circuit are tested by changing the number of the employed units.
- All direct lines must toggle at a specified rate. (for example, one half of the direct lines toggles at the onboard oscillator frequency and the other half toggles at half of this frequency)
- Both, horizontal and vertical, orientations of the CLBs must be tested since horizontal direct lines are expected to consume less power (at least for Spartan 3AN for example).

In Circuit1, the total power of the circuit, $P1$, can be represented as follows:

$$P1 = P_{dyn\_LUT} * n + P_{dyn\,FF} * n + P_{dyn\_direct} * (m + 0.5m) + P_{other} + P_{leak}, \quad (1)$$

where P_{dyn_LUT}, P_{dyn_FF}, and P_{dyn_direct} represent the dynamic power of one LUT, one FF and one direct wire, respectively. P_{other} represents the extra power of additional direct lines connecting the feedback path from the output of CLBs to their input and the global lines connecting the reset, clk and other power components. Also, P_{leak} represents the total leakage power of the design at hand (since leakage happens during both static and switching states). m is the number of direct wires toggling at the onboard oscillator frequency and n is the number of LUT-FF pairs.

Similarly, in Circuit2, the total power of the circuit, $P2$, is represented as follows:

$$P2 = P_{dyn_{LUT}} * n + P_{dyn_{FF}} * n + 2 * P_{dyn\_direct} * (m + 0.5m) + P_{other} + P_{leak}. \quad (2)$$

Note that P_{other} along with P_{leak} are identical across both designs in Circuit1 and Circuit2 and are considered constants. The only difference between the equations for both $P1$ and $P2$ [(1)&(2)] is that the third term is multiplied by two. It is imposed by the duplication of interconnecting direct wires in Circuit2 with respect to those in Circuit1. Both $P1$ and $P2$ are measured for increasing value of m. Once the proper size is reached at which the power value per unit stabilizes, the direct wire power, P_{dyn_direct}, is calculated as follows:

$$P_{dyn\_direct} = (P2 - P1)/(m + 0.5m). \quad (3)$$

B. CLBs as FFs and LUT-FF pairs

Two circuits are designed. One of them has its CLBs configured as FFs and the other has its CLBs configured as

LUT-FF pairs. The circuits satisfy most conditions described in the last subsection and are shown in Fig.3 and Fig.4. In Circuit3, the total power $P3$ can be represented as follows:

Two circuits are designed, one of them has its CLBs configured as LUT-FF pairs (as shown in Fig.4 of the previous subsection) and the other has its CLBs configured as LUTs as depicted in Fig.5. The circuits satisfy most conditions described in the past subsections. The total power $P5$ can be represented as follows:

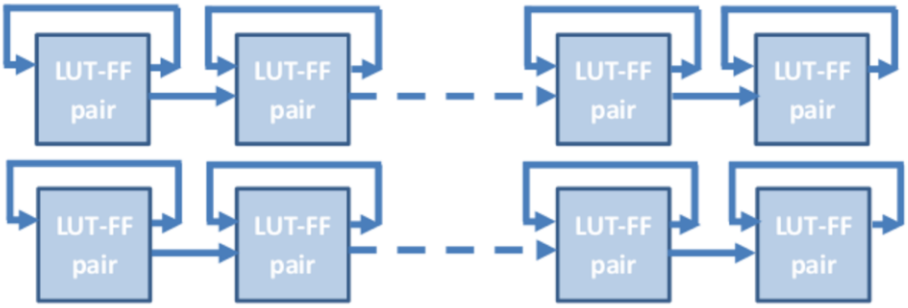

Fig. 1. Circuit1: Each CLB is configured as an LUT-FF pair and only one direct line connects two successive CLBs.

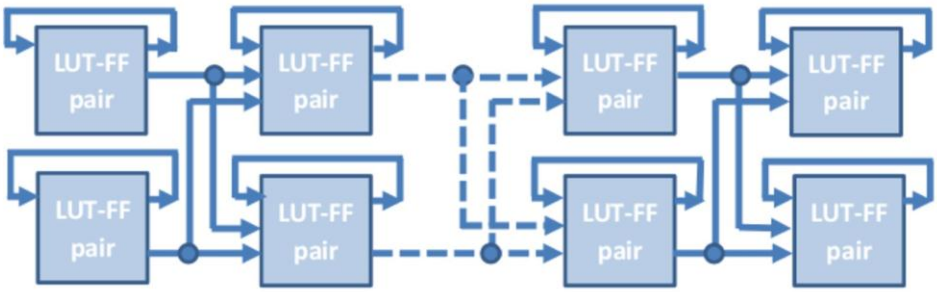

Fig. 2. Circuit2: Each CLB is configured as an LUT-FF pair and only two direct lines connect two successive CLBs.

$$P3 = P_{dyn\_FF} * n + P_{dyn\_direct} * m + P_{other} + P_{leak}. \quad (4)$$

Whereas, the total power $P4$ of Circuit4 can be represented as follows:

$$P4 = P_{dyn\_FF} * n + P_{dyn\_LUT} * n + P_{dyn\_direct} * m + P_{other} + P_{leak}. \quad (5)$$

P_{other} and P_{leak} are identical across both designs and can be considered constants. Both $P3$ and $P4$ are measured for increasing value of m. Once the proper size is reached at which the power values per unit stabilize, P_{dyn_LUT} is calculated as follows:

$$P_{dyn\_LUT} = (P4 - P3)/(n). \quad (6)$$

C. CLBs as LUT-FF pairs and LUTs

$$P5 = P_{dyn\_LUT} * n + P_{dyn\_direct} * m + P_{other} + P_{leak}. \quad (7)$$

Both $P4$ and $P5$ are measured for increasing value of m. Once the proper size is reached at which the power values per unit stabilize, P_{dyn_FF} is calculated as follows:

$$P_{dyn\_FF} = \frac{P4 - P5}{n}. \quad (8)$$

Measurements and results of applying the mentioned setup and equations are detailed in the next section.

IV. Measurements and Results

Many units of the described Spartan3AN starter kit are considered. Power measurements are done by measuring *ICCINT* through the available pins on the board using a

multimeter as described by Chapman [10]. Current is measured in *mA* and to get the power in *mW* the current value is multiplied by the value of *VCCINT* (1.2*V*). Note that, *VCCINT* is the main power supply for the FPGA's internal logic. To measure the power for I/Os and DCMs, similar measurements need to be done using *ICCAUX* or *ICCO* instead of *ICCINT*

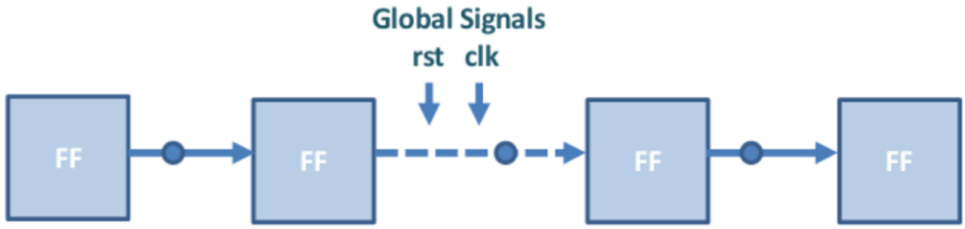

Fig.3. Circuit3: CLBs are configured as FFs and only one direct line is connecting two successive CLBs

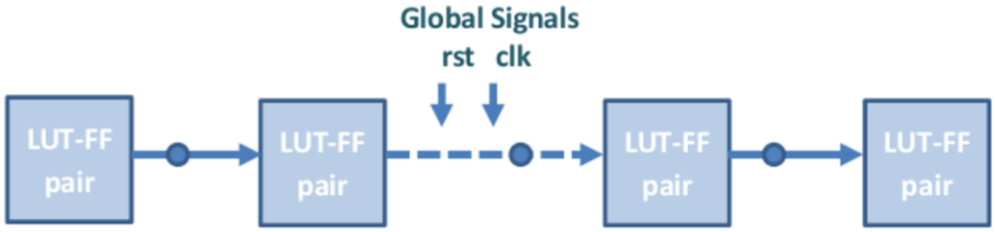

Fig.4. Circuit4: CLBs are configured as LUT-FF pairs and only one direct line is connecting two successive CLBs

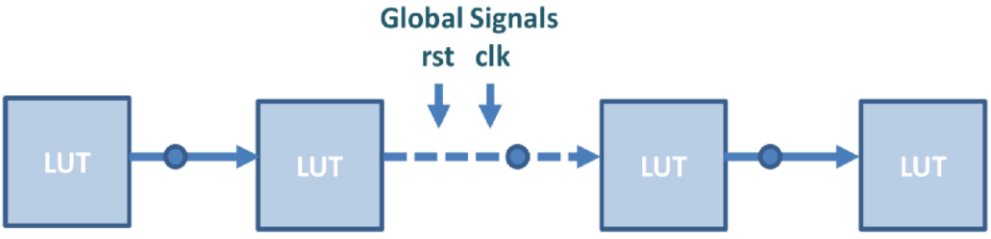

Fig. 5.Circuit5: CLBs are configured as LUTs and only one direct line is connecting two successive CLBs

[11]. The resources powered by *VCCINT* in XC3S700A are:

1) CLB
   -1472 CLBs (5888 Slices)
   -11776 LUTs/FFs
   -5888 SRAM16/SRL16
   -94208 Distributed RAM bits
2) Interconnect
   -Long lines: spans 6 tiles (24 lines to span the FPGA horizontally and vertically)
   -Hex lines: spans 3 tiles (horizontally and vertically)
   -Double lines: spans 2 tiles (every other tile horizontally and vertically)
   -Direct connection: connects neighboring tiles

(horizontally, vertically, and diagonally)

3) Multipliers

-20 multipliers in 2 columns

4) Block RAM

-20 block RAMs in 2 columns with a total of 360K bits

The power characterization of various FPGA components is examined in the next subsection. This is done by using a few successive measurement/calculation procedures as described in section III. Then, the importance of repetition and the variation in static/dynamic power are presented in the following subsections.

A. FPGA components power characterization

Appropriate VHDL code for the desired circuits are designed to ensure the respective CLB configuration and identical toggling rates of units under test. In addition, user constraint files (.ucf) are written to place the CLB in suitable locations to enforce only intermittent direct line connections in both horizontal and vertical orientations for direct line power testing. Also, tool control language (.tcl) files are needed to generate the bit files for successive values of n for all needed orientations. The written code is described in the appendix. The various successive designs (bit files) are then downloaded to the FPGA and the current *ICCINT* is measured by a millimeter. Many measurements are done for n ranging from 20 to 700. Then $P2$ and $P1$ are used to calculate the power for a single direct line on average as described in equation 3. Similarly, $P3$ and $P4$ are employed to calculate the power for a single LUT on average as in equation 6. Average values attained are shown in Table I

TABLE I
AVERAGE POWER VALUES DRAWN FOR COMPONENTS OPERATING AT 1.2V AND TOGGLING AT THE GIVEN OPERATING FREQUENCIES

Component	Power	Operating Frequency
Vertical direct Line	1.68 μW	133 *MHz*
Horizontal direct Line	2.64 μW	133 *MHz*
LUT	3.84 μW	66.5 *MHz*

B. Importance of Repetition

The counters are replicated multiple times on the FPGA to reduce the effect of measurement errors. Experiments are conducted using 1, 10, 20, 40, 60, 80, 100, 150, and 200 counters on one boards. The dynamic power per counter as implemented using multiple counters on one FPGA (with static power=10.61mW) is shown in Table II. It can be seen that the dynamic power per counter settles at 0.18μW (0.15 mA) as the number of implemented counters increases.

TABLE II
POWER ($ICCINT \times 1.2V$) VALUES FOR ONE BOARD IMPLEMENTING MULTIPLE COUNTERS (mW)

Number of Counters	Total Power	Dyn. Power (Total Power-Static Power)	Dyn. Power per Unit
1	11.0868	0.4776	0.48
10	13.9776	3.368	0.34
20	16.9872	6.378	0.32
40	20.8116	10.2024	0.25
60	24.2868	13.68	0.23
80	27.1104	16.5	0.21
100	28.8216	18.212	0.18
150	37.158	26.548	0.18
200	46.2	35.59	0.18

C. Static and Dynamic Power for one counter for different FPGA boards

In order to validate the proposed methodology, different FPGA boards are used for power measurements. Six Xilinx Spartan 3A starter kits are employed to get the results. Only one 20-bit counter with synchronous reset is implemented and the power consumed in various cases is measured. The results are shown in Table III. In this table, the static power of the device is measured when the reset is active, whereas the total power is measured when the reset is inactive (the counters are switching freely). The dynamic power is the difference between the total power and the leakage power. It can be seen that the static power is different from a board to the other where the maximum is 15.696mW, the minimum is 10.61mW, and the average is 13.68mW. The static current value reported by Xilinx Power Estimator (XPE) is 13.15mA (15.78mW) and by Xilinx Power Analyzer (XPA) is 13.17mA (15.8mW). The values reported by XPE and XPA have an error range of 0.5% to 49% if compared to actual power measurement values. It is noted, that actual measurements give more accurate results than power estimation tools.

It is noted, that the dynamic power values of different boards are not as different as in the case of static power (in Table II). Thus, characterizing an FPGA and reporting average dynamic power values for all its units then using these values for power estimation of different designs is a reliable technique. The average dynamic current is 0.35mA (0.42mW). However, the measured dynamic current is much less than the one reported by XPE 3.81mA (4.57mW) and XPA 2.32 mA (2.78μW). The values reported by XPE and XPA are five to ten times the actual values making XPA and XPF results unacceptable. It is expected that this error decreases with larger designs however the need for more accurate power estimation tools is obvious. The proposed approach is much more accurate than the provided tools with the FPGA since it characterizes the real power consumed on the FPGA.

TABLE III
STATIC AND DYNAMIC POWER VALUES (*ICCINT**1.2*V*) FOR DIFFERENT BOARDS USING A 20-BIT COUNTER (*mW*).

Board Number	Static Power	Total Power	Dyn. Power (Total Power - Static Power)
1	12.73	11.009	0.4
2	15.697	16.166	0.469
3	13.034	13.432	0.398
4	14.147	14.448	0.301
5	14.363	14.662	0.299
6	14.232	14.894	0.662
AVG	13.68	14.1	0.42

V. CONCLUSIONS

Available power consumption estimation tools can report inaccurate values with up to ten times the actual measured dynamic power and up to 50% error in leakage power. An automated FPGA power consumption estimation methodology is proposed. It can be applied at a very early stage of the design cycle using a sequence of real time power consumption measurements. These are applied to determine the total leakage power of the FPGA chip and the dynamic power consumption of the FPGA building blocks. The methodology is used to determine the unit power consumption of the direct line (horizontal/vertical) and the LUT.

APPENDIX

The VHDL code to implement the building block in Fig.1, where there is only one direct line connecting two successive LUTs is as follows:

```
entity LUTOneDirectLineFF is
Port ( reset, clk: in STD LOGIC;
  a :in STD LOGIC vector (1 downto 0) ;
  cout : out STD LOGIC vector (1 downto 0));
end LUTOneDirectLineFF;

architecture Behavioral of LUTOneDirectLineFF is
   signal cout_internal: std logic vector(1 downto 0);
begin
cout(0)<=cout_internal(0);
cout(1)<=cout_internal(1);

process(a,reset,clk)
begin if (clk'event and clk='1')
then if (reset ='1') then
    cout_internal(0)=a(0);
  else
   cout_internal(0)=NOT(cout_internal(0));
   end if;
end if; end
process;
process(a,reset,clk) begin if
(clk'event and clk='1') then if
(reset ='1') then
    cout_internal(1)=a(1);
  else
   cout_internal(1)=NOT(cout_internal(1));
   end if;
end if; end
process; end
Behavioral;
```

We also need a block to half the onboard clock, *clk*, as some CLBs are operated at *clk* and the others are operated at *clk* divided by two.

```
entity clk_divide is
Port ( clk,reset: in STD LOGIC;
  clk_divide: out STD LOGIC);
end clk_divide;

architecture Behavioral of clk_divide is
   signal clk_internal:std logic;
begin
process(clk, reset, clk_internal)
begin
   if (clk ='1' and clk'event)
then
   if (reset='1') then
    clk_internal<='0';
   else
    clk_internal<=not(clk_internal);
   end if;
end if;
clk_divide<=clk_internal;
end process;
end Behavioral;
```

Then, we need the following generic code to connect a given number of these building blocks in groups of 25 units, for example, along with one clock divider. *entity*

```
LUTOneDirectLine generic FF is generic
   (count: integer:=25);
Port ( clk: in STD LOGIC; reset : in STD
   LOGIC; cout : out STD _LOGIC vector(1
   downto 0));
end LUTOneDirectLine generic FF ;


architecture Behavioral of LUTOneDirectLine generic FF is
component clk_divide
 Port ( clk,reset: in STD LOGIC;
   clk_divide: out STD LOGIC);
end component;
component LUTOneDirectLineFF
```

```
 Port ( clk,reset: in STD LOGIC; a :in STD
   LOGIC vector (1 downto 0) ; cout : out STD _
   LOGIC vector(1 downto 0));
 end component;
 signal cout_internal0,cout_internal1: std logic vector(count
 downto 0);
 signal clk _divided: std logic;
 begin
 clk_divide inst: clk_divide port map(
 reset=reset,clk=clk,clk_divide=clk_divided);
 GEN: for I in count downto 1 generate
 LUTOneDirectLineInst :LUTOneDirectLineFF
 port map (
   a(1)=>cout_internal1(I-1),
   a(0)=>cout_internal0(I-1),
   reset=>reset, clk=>clk,
   cout(1)=>cout_internal1(I),
   cout(0)=>cout_internal0(I)
 );
 end generate GEN;
 LUTOneDirectLineInstFirst :LUTOneDirectLineFF
 port map ( a(0)=>clk, a(1)=>clk_divided,
 reset=>reset, clk=>clk,
 cout(1)=>cout_internal1(0),
 cout(0)=>cout_internal0(0)
 );
 cout(0)<=cout_internal0(count);
 cout(1)<=cout_internal1(count);
 end Behavioral;
```

When done, we connect a generic number of these groups of 25 building blocks in one top module as follows:

```
 entity LUTOneTop Generic is generic
 (TopCount: integer:=8);
 Port ( clk : in STD LOGIC; reset : in STD LOGIC; cout0,
 cout1 : out STD LOGIC vector(TopCount downto 1));
 end LUTOneTop Generic;
architecture Behavioral of LUTOneTop Generic is
component LUTOneDirectLine generic FF is
 Port ( clk: in STD LOGIC; reset : in STD
   LOGIC; cout : out STD LOGIC vector(1
   downto 0));
 end component ; begin
 GEN: for I in TopCount downto 1 generate
 LUTOneInst :LUTOneDirectLine generic FF
 port map ( clk=>clk, reset=>reset,
 cout(1)=>cout1(I), cout(0)=>cout0(I)
 );
 end generate; end
 Behavioral;
```

Note that TopCount has an initial value of 8, but is then assigned successive different values in the tcl file. In the following snapshot you will find the first few lines of the user constraint file (.ucf) needed to assign the components in a horizontal row and assigning the input output signals to specific pins.

```
 #The net constraints are for One line LUT interconnection
 # on Spartan 3A XC3S700AN NET
 clk PERIOD = 50ns ;
 NET "reset" LOC = V8;
 #(SW0) for reset
 # on board oscilator for clock; NET
 "cout1[1]" LOC=R20;
 NET "cout0[1]" LOC=T19;
 INST "Gen[1].LUTOneInstclk_divide _inst" LOC=SLICE
X30Y51;
 INST "Gen[1].LUTOneInstLUTOneDirectLineInstFirst"
LOC=SLICE X30Y52;
 INST "Gen[1].LUTOneInstGen[1].LUTOneDirectLineInst"
LOC=SLICE X30Y53;
 INST "Gen[1].LUTOneInstGen[2].LUTOneDirectLineInst"
LOC=SLICE X30Y54;
 .....
 ##################
 INST "Gen[2].LUTOneInstclk_divide _inst" LOC=SLICE
X32Y51;
 INST "Gen[2].LUTOneInstLUTOneDirectLineInstFirst"
LOC=SLICE X32Y52;
 INST "Gen[2].LUTOneInstGen[1].LUTOneDirectLineInst"
LOC=SLICE X32Y53;
```

The VHDL code for the building block in Fig.2 in which there are only two direct lines connecting two successive LUTs is as follows. Similarly, these building blocks are aligned in blocks of 25 then repeated multiple times horizontally and vertically.

```
 entity LUTTwoDirectLinesFF is
 Port ( reset,clk: in STD LOGIC;
  a :in STD LOGIC vector (1 downto 0) ;
  cout : out STD _LOGIC vector (1 downto 0));
 end LUTTwoDirectLinesFF;

 architecture Behavioral of LUTTwoDirectLinesFF is
   signal cout_internal: std logic_vector(1 downto 0);
 begin
   cout(0)<=cout_internal(0);
   cout(1)<=cout_internal(1);
 process(a,reset,clk) begin if
 (clk'event and clk='1') then
   if (reset ='1') then
    cout_internal(0)<=a(0) AND a(1);
   else
   cout_internal(0)¡=NOT(cout_internal(0));
   end if;
```

```
end if; end
process;

process(a,reset,clk)
begin
if (clk'event and clk='1') then
  if (reset ='1') then
   cout_internal(1)¡=a(0) OR a(1); else
   cout_internal(1)¡=NOT(cout_internal(1));
  end if;
end if;
 end process;
end Behavioral;
```

REFERENCES

[1] I. Kuon and J. Rose, "Measuring the gap between fpgas and asics," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 26, no. 2, pp. 203–215, 2007.

[2] R. Jevtic and C. Carreras, "Power measurement methodology for fpga devices," Instrumentation and Measurement, IEEE Transactions on, vol. 60, no. 1, pp. 237–247, 2011.

[3] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic power consumption in virtex ii fpga family," in Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays, ser. FPGA '02. ACM, 2002, pp. 157–164.

[4] J. Oliver and E. Boemo, "Power estimations vs. power measurements in cyclone iii devices," pp. 87–90, 2011.

[5] R. Jevtic, C. Carreras, and V. Pejovic, "Floorplan-based fpga interconnect power estimation in dsp circuits," in Proceedings of the 11th international workshop on System level interconnect prediction. ACM, 2009, pp. 53–60.

[6] G. C. Cardarilli, A. Del Re, A. Nannarelli, and A. D. Re, "Power characterization of digital filters implemented on fpga," in Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on, vol. 5. IEEE, 2002, pp. V–801.

[7] R. Jevtic and C. Carreras, "A complete dynamic power estimation model for data-paths in fpga dsp designs," INTEGRATION, the VLSI journal, vol. 45, no. 2, pp. 172–185, 2012.

[8] R. Jevtic, C. Carreras, and G. Caffarena, "Fast and accurate power estimation of fpga dsp components based on high-level switching activity models," International Journal of Electronics, vol. 95, no. 7, pp. 653– 668, 2008.

[9] K. K. Poon, A. Yan, and S. J. Wilton, "A flexible power model for fpgas," in Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream. Springer, 2002, pp. 312–321.

[10] K. Chapman, "Practical power testing, a reference design for the spartan™ fpga starter kit," 2008.

[11] X. Inc., "Xilinx spartan-3a/3an starter kit board user guide," UG334 (v1.1), 2008.