New Approach to Efficiently Assess the Power Consumption of Field Programmable Gate Array Devices

Esteve Hassan
Electrical and Computer Engineering Technology
Mohawk College of Applied Arts and Technology
Hamilton (ON), Canada
e-mail: esteve.hassan@mohawkcollege.ca

Bilal Al Momani
Electrical and Computer Engineering Technology
Mohawk College of Applied Arts and Technology
Hamilton (ON), Canada
e-mail: bilal.al-momani@mohawkcollege.ca

Abstract— In this paper, the design application is a telemetry system intended for health monitoring applications. The Field Programmable Gate Array (FPGA) is used as the brain control unit at both transmitter and receiver sides. The transmitter side is recording data packets through external interfaced sensors. Verilog Hardware Description Language (Verilog-HDL) has been used to implement the various functionalities required by the FPGA device. The power performance of the FPGA-based design will be assessed using the XILINX Xpower tool. A Model sim Code coverage feature has been incorporated to make sure that the test bench will cover all the nets branch statements of the design and create the most accurate Value Change Dump (VCD) file for the power consumption assessment process.

Keywords- FPGA; Telemetry system; power assessment.

I. INTRODUCTION

Electronic systems development is becoming more and more complex, fast, powerful, and power-consuming. Indeed, the transistor miniaturization dramatically increases the power consumed by a whole chip [1]. The main consequences of this trend are the addition of elaborated cooling circuits and the reduction of battery life for the embedded systems. As for timing and die area, power consumption becomes a critical constraint for electronic system design. A previous study [2] has demonstrated the beneficial effect of power optimization at high-level; it is then necessary to develop high-level estimation tools which use power models for all kind of components (Application-Specific Application Circuit (ASIC), FPGA) in a system. Playing many important roles in recent applications, FPGAs devices are used in a wide scale of designs ranging from small glue logic replacement to System-on-Chip. The main advantage of FPGA compared to ASIC chips is the flexibility: a design can be reprogrammed partially or totally in-situ. This functionality is realized by a configuration plan and requires a large number of transistors for Static Random Access Memory (SRAM) FPGA; therefore, the drawback is important static power consumption. Moreover, FPGA builders are currently improving this circuit characteristic to facilitate their integration in System on Chip (SoC). The health care field became one of the most recent applications of the FPGA designers [3].

The challenge is to raise or at least maintain the present level of health care provision without ending up in an uncontrolled cost explosion. The increasing number of researchers and manufacturers who are working on a new generation of wireless technology applications for the medical field has led to improved quality and reduced cost of patient care. One of the areas in healthcare that best lend itself to wireless technology is patient monitoring, also known as wireless telemetry.

FPGA vendors are facing the difficult task to accurately specify the energy consumption information of their products on the device data sheets because the energy consumption of FPGAs is strongly dependent on the target circuit including resource utilization, logic partitioning, mapping, placement, and route. While major Computer Aided Design (CAD) tools have started to report average power consumption under given transition activities, energy optimal FPGA design demands more detailed energy estimation. This work aims to present a useful methodology for estimating the power consumption of an FPGA-based system designed for medical applications. Modelsim code coverage capability will be used to investigate the different styles of test bench coding on the overall power consumption estimation of the FPGA device.

In Section 2, a system overview is presented. Section 3 is outlining the power estimation methodology for FPGA devices. Modelsim code coverage is explained in Section 4. Section 5 is giving the merits of the new method to perform an accurate power consumption assessment. Finally, conclusions are drawn in Section 6

II. SYSTEM OVERVIEW

The main blocks of the transmitter side FPGA are shown in Figure 1. The different units of the system were coded with Verilog HDL simulated with ModelSim SE V6.0a and implemented with ISE14.7. The final implementation was targeting the Spartan-3 device since it provides the various features that solve the designer's challenge throughout the entire system. The transmitter FPGA consists mainly of an SPI (Serial Peripheral Interface), RLE (Run Length Encoding) compressor, and framer units. The operation of

 ISBN: 978-1-61208-838-9

the system units and the flow of data through the system are controlled by the main FSM (Finite State Machine) controller.

At the receiver side, a data recovery unit is needed to extract the clock from the received bitstream. The de-framer and the RLE decompresser blocks are designed to reconstruct the original data bytes sent by the transmitter.

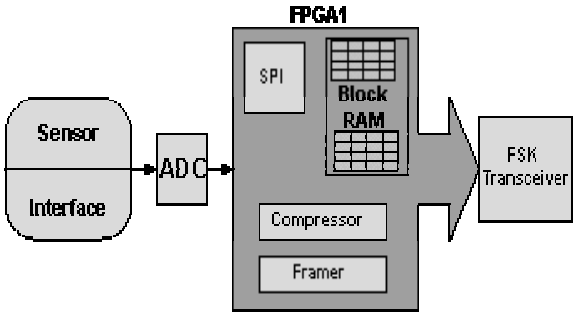

Figure 1. Building blocks of the transmitter FPGA

A. SPI main units

An efficient SPI unit has been modeled, as shown in Figure 2. The Master out Slave in (MOSI) signal has been omitted from the design based on the hardware requirements where data only needed to be transferred from the ADC to the FPGA system. The main units of the SPI are functioning as follows:

1. Clock Divider Unit: Divides the system clock by a certain factor to generate the required SPI clock frequency.
2. Data out clock synchronizer: used to generate both the rising edge (dout7) and the falling edge (dout16) of the ADC clock.
3. ADC Enable unit: triggered on when the start_conv signal is asserted to generates the following signals:
   a. Capture signal to capture data transfer from the ADC to the SPI register after each byte transfer.
   b. Increment signal used to change the address inside the Block RAM unit.
4. Slave Chip Select (CS).
5. SPI Register Unit: contains the SPI serial in/parallel out register, which is enabled when the capture signal is asserted and receives input serial data through ADC_Din signal. Spiout (output) signal carries the information data bytes to the Block RAM unit.
6. Distributed Block RAM: stores the data bytes in locations determined by the increment signal.

Typically, test benches have become the standard method to verify HDL designs. Test benches invoke the functional design, and then simulate it. Accordingly, an efficient test bench has been written to mimic the behaviour of the ADC and verify the operation of the SPI system units.

III. FPGA POWER ESTIMATION

Power consumption is mandatory information in modern digital system design.

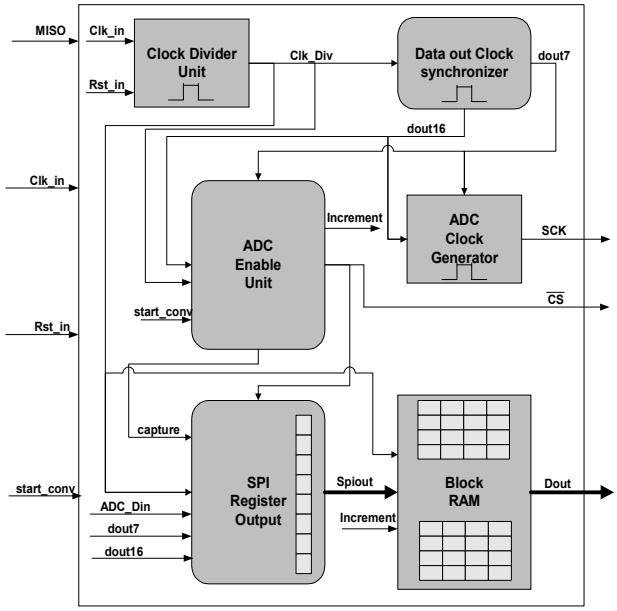

Figure 2. Block units of the developed SPI inside the FPGA

Chip vendors are naturally in charge of supplying energy consumption information of their products on the device data sheets. However, it is not possible for vendors to specify power consumption information of SRAM-based FPGAs because it is not only dependent on the target device and operating frequency, but is highly dependent on the design and operating conditions. Power consumption is strongly dependent on the target circuit including resource utilization, low-level features such as logic partition, mapping, placement and route.

A. Related Work

For FPGA, some methodologies and models have already been developed to estimate the power consumed specifically by the logic elements. For example, a probabilistic model is proposed by [4]; developed for a CAD tool, this model estimates, at the transistor level, the 0.18 μm Complementary Metal Oxide Semiconductor (CMOS) FPGA power consumption based on place and route. The switching activity used to calculate the dynamic power is determined by the transition density of the signal. The static power is evaluated by a sub-threshold current estimation. The resulting absolute error of this model is 23% compared to measurements. Some techniques are proposed in [5] to reduce both static and dynamic power consumption like drowsy mode, clock gating, guarded evaluation, counter and state machine encoding, but no estimation model is proposed. For a design in Virtex-II, [6] proposes an estimate of the dynamic power consumed by logical elements after routing. Lastly, [7] has presented a Register Transfer level power estimator based on determination of wire length and

switching activity with an average error of 16.2%. The first parameter is calculated by applying Rent's rule during high-level synthesis. The second parameter is evaluated by a fast switching activity calculation algorithm. The model developed by [8] allows estimating the power consumption of distributed memory (using logic elements) in past FPGA families with technical parameters. Another model proposed in [9] uses technical parameters such as effective capacitance of each resource which is hardly obtained. All these methodologies and models use low-level parameters and technical characteristics which are not available before place and route. More precise approaches to estimate FPGA power consumption were described in [11] and [12].

B. *XPower XILINX Tool*

XPower is a commercial-off-the-shelf tool to estimate power consumption of Xilinx SRAM-based FPGAs. The design flow of the XPower is shown in Figure 3. In this paper the implementation of the Xilinx XPower will be investigated. XPower reads in either pre-routed or post-routed design data, and then makes a power model either for a unit or for the overall design based on the power equation: $P=CVf$ where P is average power consumption, C is equivalent switching capacitance, V is supply voltage and f is operating clock frequency or toggle rate. It considers resource usage, toggle rates, input/output power, and many other factors in estimation. Because XPower is an estimation tool, results may not precisely match actual power consumption. The frequency, f, is determined by users or provided by simulation data from the ModelSim family of HDL simulators.

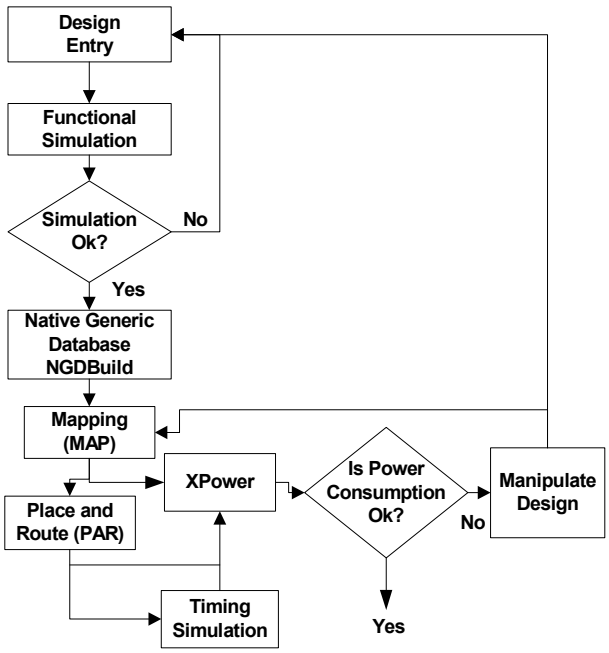

Figure 3. XPower tool design flow

XPower provides two types of information called data view and report view. The data view shows the power consumption of individual parts of a design such as signals, clocks, logic and outputs. The report viewer represents the total power consumed by a given design, which is again classified into power consumption of clocks, logic and outputs, and static (leakage) power. The power consumption of clocks, logic and outputs are calculated by equivalent switching capacitance models. The static power is based on constant value quoted in a data book or calculated by an equation associated with temperature, device utilization and supply voltage.

IV. MODELSIM CODE COVERAGE

ModelSim code coverage can display a graph and report file feedback on which statements, branches, conditions, and expressions in the source code have been executed. It also measure bits of logic that have been toggled during the execution. Therefore, it can be considered as trace tool and probe at the same time that provides history of the software execution.

As code execution is almost invisible without an accurate trace tool, it is common for the entire blocks or modules of code to go unexecuted during test routines or redundant user-case suites. Coverage metrics showing which functions are not executed are useful for writing new, additional tests or identifying unused "dead" code. In applications where code size is critical, removing dead code reduces both waste as well as risk in the targeted design. In many cases, code coverage can also be used to analyze errant behavior and the unexpected execution of specific branches or paths. The FSM can be extracted with code coverage as in Figure 4 for the compressor unit.

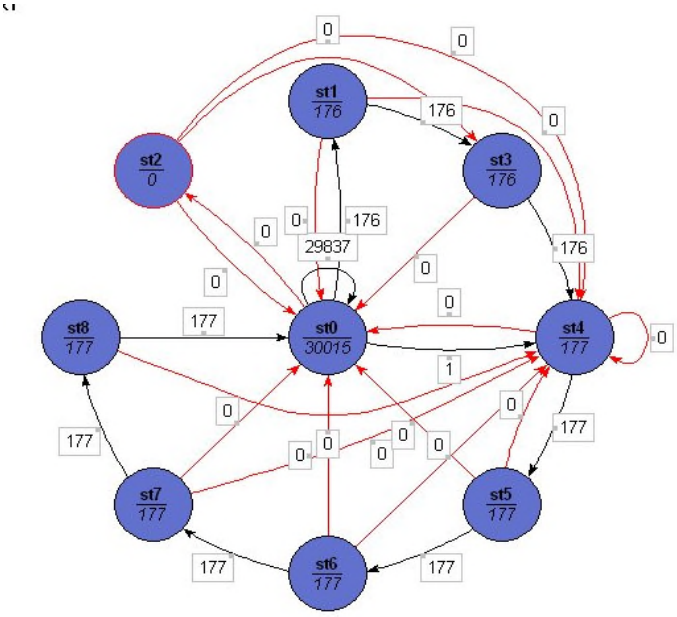

Figure 4. Transmitter FPGA-Compressor unit FSM

The FSM figure clarifies the number of the states involved in the design and the interaction between these states. A test bench has been written to examine the behavior of the HDL design. The code coverage is enabled to check the covered and uncovered parts of the design code by the test function which can lead to alter the design and consequently change the power consumption.

Figure 5 shows some uncovered statements indicated by red X mark. In addition, an example of missed branches is shown in Figure 6 where the X_T mark indicates that the true branch of the conditional statement was not covered.

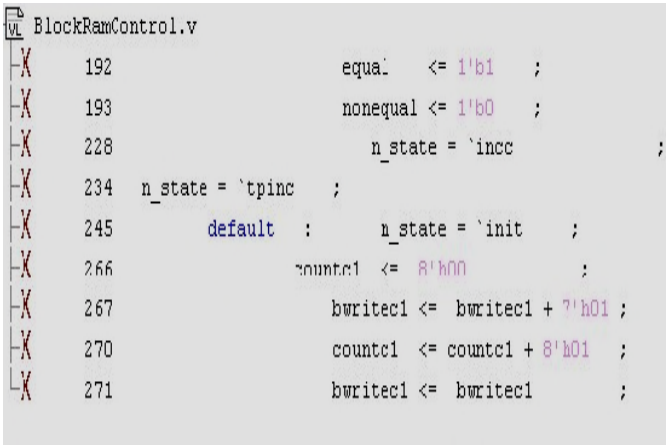

Figure 5. Not covered design statements

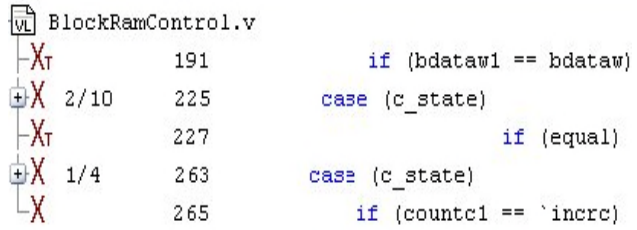

Figure 6. Not covered design branches

V. ACCURATE FPGA POWER ESTIMATION

The power characterization using XPower tool is done and based on the mapped/placed/routed design. In general, the total power consumption of a CMOS component is given by Equation.1. The dynamic power is due to the component activity while static power represents the power consumed by the leakage current.

$$P_{total}=P_{dynamic}+P_{static} \quad (1)$$

The static power given by the XPower is constant, and calculated by the multiplication of maximum leakage current absorbed by the FPGA core and its supply voltage. On the other hand, dynamic power is varying according to the switching activity of the design. Therefore, two factors determine the accuracy of the XPower analyzer estimation: the accuracy of the data within Xpower analyzer and the stimulus provided by the user. It is necessary to mention that XPower relies upon stimulus data to estimate the power consumption for internal components. Valid input frequencies and toggle rates are necessary parameters to generate a proper power estimate. The main four important files that need to be invoked by the tool is the design (*.ncd), simulation (*.vcd), physical constraints (*.pcf) and setting (*.xml) files.

There are few strategies that can be implemented to reduce the power consumption of the FPGA device; these are:

1. Turn off clocks when they are not in use.
2. Make Block RAMs to operate in “no read on write” mode. This reduces toggling of the output of the BRAM.
3. Use clock enables to reduce switching activity on the output of Flip Flops (FFs).
4. Partition logic driven by global clocks into clock regions and reduce their number to which each global clock is routed.
5. Reduce the total number of columns to which a clock is routed.
6. Reduce the total length of heavy loaded signals.

Mainly, we followed the recommendations in 3 &4 to reduce the power consumption of our design. Therefore, the global design has been partitioned into a lot of small blocks.

In order to investigate the impact of test bench writing style on the accuracy of the power estimate, two methods have been exercised. In the first one, the ADC_Din (line carrying input data coming from the ADC) has been stimulated by variable 8-bits data samples, as expected in the practical case. In the second method, only 0 data value is stimulating the ADC_Din. As an example, the control logic of the compressor BRAM has been considered to see the difference in the code coverage represented by the summary reports given in Figures 7 and 8.

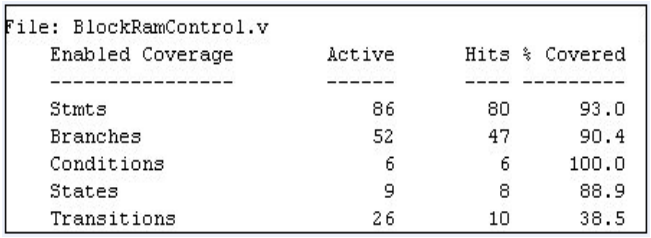

Figure 7. BRAM control coverage report without adc_in

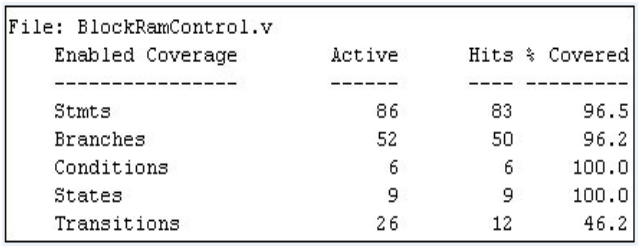

Figure 8. BRAM control coverage report with adc_in

The hits count shows the number of times the indicated code part has been reached or executed. It is obvious that this count has been increased for all the design parts except the conditions. The states of the design have been fully covered in Figure 8 because the system is dealing with all the possible design options that are required in the verification stage. Sample ModelSim waveforms for the two adc_in condition are shown in Figures 9 and 10.

To investigate the effect of adc_in on the power consumption of the FPGA device, XPower tool has been used for this purpose. Table.1 summarizes the total power and current estimates for both configurations.

TABLE 1. TOTAL CURRENT AND POWER ESTIMATES

Total Power and Current estimates	I(mA) With adc_in	P(mW) With adc_in	I(mA) Without adc_in	P(mW) Without adc_in
Device		59		51
Vccint 1.20V	12	16	11	13
Vccaux 2.50V	16	41	15	38
Vcco25 2.50V	1	2	0	0
Quiescent Vccint 1.20V	10	12	10	12
Quiescent Vccaux 2.50V	15	38	15	38

As given above, more power is needed for the design when ADC_Din is clocking with different serial data. This leads to the conclusion that the FPGA device will consume higher power if the analog input signal to the ADC is rapidly changing. IN this case, the compressor unit of the design will be fully functioning with all the possible transition state.

The second case is assuming that the analog input has steady value which is rare in practical, but useful to have a power estimates for different working conditions. Thus total dynamic power is more depending on the states of the input signals. In comparison, the power values in the two cases are different due to the code coverage analysis that has been discussed earlier. As more code has been covered with ADC_Din is varying, then we can consider that the obtained power estimate with such case has more credibility.

Quiescent power is the same for both configurations since it depends on the device itself using default conditions in moderate environments. To reduce the power consumption of the FPGA without loosing the accuracy, more work need to be done outside than inside the device. For examples, reduce the ADC resolution or the analog input can be good options. The test bench should be written in a very optimum way to provide stimulus for all the inputs and read efficiently all the outputs. This is an important issue in the XPower tool since it provides one of the main files for the tool to estimate the power. The code coverage sometimes can help to extract the unnecessary parts from the design which has a great benefit for the power consumption. As a final comment, obtaining the high code coverage can lead to higher but more accurate power estimate using XPower.

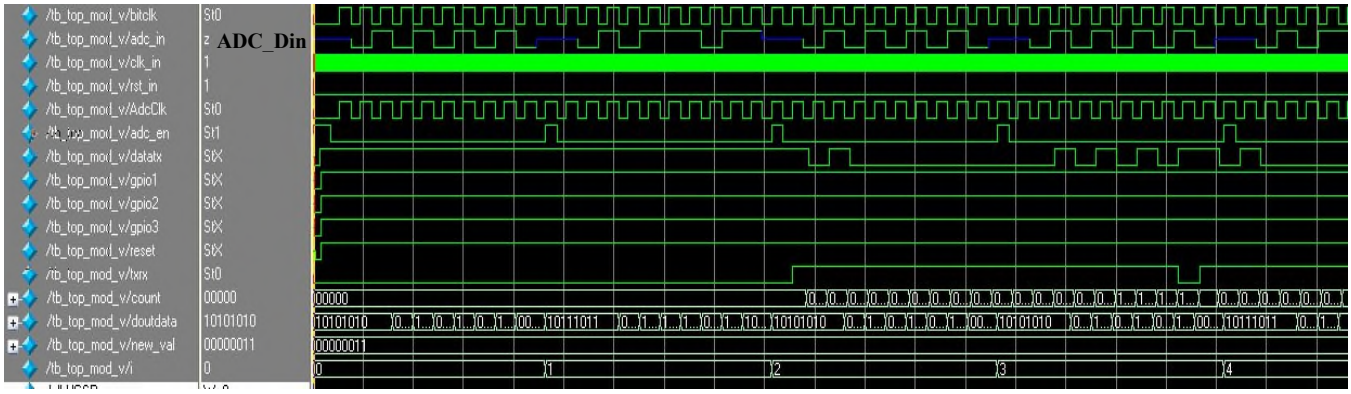

Figure 9. Simulation output waveform of the design with for variable adc_in

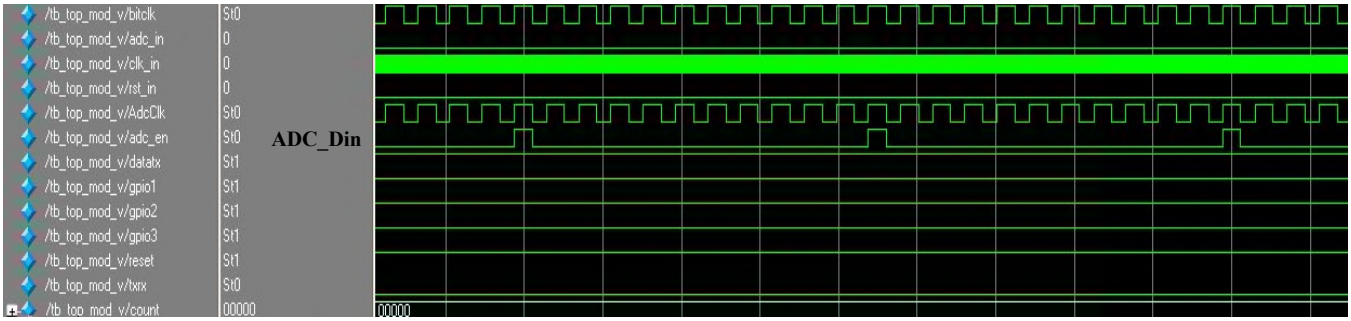

Figure 10. Simulation output waveform of the design with 0 data on adc_in

VI. CONCLUSIONS

In this paper, the power consumption of an FPGA-based system designed for medical applications has been investigated. The use of XPower from XILINX was the main focus of this work as an efficient tool to get good power estimates for the target FPGA device. The relation between the test bench coverage and power estimate accuracy was studied under different design conditions. It has been found that a good test bench with higher design code coverage capability can achieve more accurate power estimates. The presented results reflected clearly the efficiency of this method which can be applied with similar performance on any other Xilinx FPGA devices.

REFERENCES

[1] N. S. Sung et al., "Leakage Current: Moore's Law Meets Static Power", IEEE Computer Magazine, December 2003, pp 68 – 75.

[2] J. M. Rabaey,M. Pedram, Low Power Design Methodologies, Kluwer Academic Publisher, 1996, ISBN 0-7923-9630-8.

[3] P. Dillinger, J. F.Vogelbruch, J. Leinen, S. Suslov, R. Patzak, H. Winkler, and K .Schwan, "FPGA based real-time image segmentation for medical systems and data processing", IEEE 14th NPSS Real Time Conf Proc., June 4-10 Sweden, 2005.

[4] K. W. Poon, Power Estimation For Field Programmable Gate Arrays, Ph.D Thesis, department of Electrical and Computer Engineering, University of British Colombia, Vancouver BC, Canada 1999.

[5] X. Guo-Lin, "Power-sensitive design techniques on FPGA devices," Electronics Quality Journal, 2002.

[6] L. Shang, A. S. Kaviani, K. Bathala, " Dynamic Power Consumption in VIRTEX-II FPGA Family", in FPGA'02, February 24-26,Feb 2002, Monterey, California, USA.

[7] D. Chen, J. Cong, Y. Fan, "Low-Power High-Level Synthesis for FPGA Architecture", ISPLED'03, August 25-27 Seoul Korea, 2003.

[8] A. D. Garcia, Estimation et optimisation de la consommation de puissance des circuits logiques programmables du type FPGA [Estimation and Optimization of the Power Consumptionlogic circuits of the FPGA type], Ph.D Thesis, Ecole nationale suprieure des télécommunications de Paris, 2000.

[9] D. Elleouet, N. Julien, D. Houzet, J. Cousin, and M. Martin, "Power consumption characterization and modeling of embedded memories in XILINX", IEEE Digital System Design Conf Proc., 2004, pp.394-401.

[10] H. Gyu Lee, S. Nam, and N. Chang, "Cycle-accurate Energy Measurement and High-Level Energy Characterization of FPGAs", IEEE Quality Electronic Design Symp Proc. 2003, pp.267-272.

[11] T. Fryza and M. Waldecker, "Precise Measurement of Power Consumption and Execution Time for Embedded Platforms," 25th International Conference on Systems, Signals and Image Processing (IWSSIP), Maribor, Slovenia, 2018, pp. 1-4.

[12] N. Lawal, F. Lateef and M. Usman, "Power consumption measurement & configuration time of FPGA," Power Generation System and Renewable Energy Technologies (PGSRET), Islamabad, Pakistan, 2015, pp. 1-5.

 ISBN: 978-1-61208-838-9