ISSN No:-2456-2165

Altera FPGA'S for Assessment of Low Power and Energy Consumption

K. Baby Sai Srija
Dept. of ECE
VVIT, Nambur
Guntur, India, A.P

Abstract:- Designing of low power was a main issue in manufacturing the mobile systems including its performance. Mobile electronics having high features usually have difficult calculations and tough processing that lead to a short battery life, mainly when low energy design is not taken into consideration. To prevent overheating of components along with VLSI technology, thermal design requires low-power techniques. In this research, many low power designs for ASICs, processors and FPGAs are examined in which ASICs are more flexible for using techniques that are oriented towards less power consumption. Three different designs used a cutting edge FPGA device to compare and analyze consumption of power and energy, and matrix multiplication is to be done on the Altera platform.

Keywords:- *FPGA, NIOS, Matrix Multiplication, ASIC, Energy-Efficient.*

I. INTRODUCTION

The performance, characteristics, size and weight of final users are the main quality criteria. Low power technologies continue to be the key in order to avoid additional costs for some cooling procedures, like fans and heat sinks. In the first section, we examined and discovered various low-energy design technologies in ASICs, processors and FPGAs. The power and energy cost is compared and analyzed in the three designs, multiplying the matrices of A and B having n to 32-bits and leading to a third matrix C having n to 64-bits. The designs of matrix multiplication are presented to analysis and comparison of power and energy efficiency designs.

To improve the energy efficiency, various metrics and techniques are used to assess the merits of the different solutions. Based on physical design and logic synthesis, the impact of design automation tools is analyzed [1]. To cover the gap between FPGAs and ASICs a new category of devices known as structured ASICs is introduced to reduce the problems in cost and capabilities providing challenges to the device manufactures [2]. An architecturally based scaling strategy is presented with low power operation technique which indicates that the optimum voltage is much lower than that determined by the other scaling considerations such as increased silicon areas for reduced power consumption [3]. Without a costly redesign process, accurate and efficient power estimation is required to meet the power specifications of VLSI circuits [4]. A new design tool for FPGAs has total control over placement at each level of design and object interface was chosen to reuse the code and simply code-sharing between designers [5]. Within the three different platforms the matrix multiplication area has been largely increased [6]. The design of matrix multiplication in Xilinx FPGA uses the architecture of linear array with 15 processing elements and the algorithm presented in [7]. To estimate the average power dissipated, different power estimation methods are used such as statistics, simulation or probabilistic models [8]. When compared to a bigger design, the smaller design will dissipate more power, if it has a more total effective capacitance [9]. Most cases saving energy is from 10% to 30%. Due to input value order the swapping operands are given to ALUs and floating point operations to reduce the switching activity [10]. Some DSPs like Fujitsu offer the function of loading two memory operands into one cycle registers. This technique is useful, for the benefit of saving energy. Previous study has shown that the saving by increasing double loads reached 47 percent [11]. Leakage Power Optimization in FPGAs for CAD tools and Design Methodologies is discussed in [12]. Altera Stratix IV FPGAs deliver a breakthrough level of system bandwidth and power efficiency for high-end applications; Stratix IV FPGAs surpass all other high-end FPGAs, with the highest logic density, most transceivers, and lowest power requirements shown in [13, 14]. A first step towards software power minimization i.e. Power analysis of embedded software [15]. Protocols and architectures for wireless sensor networks have been referred [16]. Power-play Power Analysis is explained in [17]. Design Techniques for Low Power and Modeling of Energy for Processors, ASICs and FPGAs are shown [18].

II. POWER ESTIMATIONTECHNIQUES

The power estimate was one of the biggest problems for FPGA suppliers recently. Designers can use efficient and precision power estimation tools to control the efficiency of a variety of implementations.The dissipation of power depends heavily on the switching operation, which means the highest active circuit consumes high power. The power dissipation of short circuit consists of power dissipated during each transition's up and down times because of the flow of current directly from the ground. The easiest way to consider power dissipation due to short circuit is to set it at the 10 percent dynamic power dissipation. We therefore consider only the average dynamic power of the power estimation techniques as:

ISSN No:-2456-2165

$$P_{Dynamic} = \frac{1}{2} V_{DD}^2 f_{clk} \Sigma_{j=1}^{n} C_j \alpha_j \quad (1)$$

Where f_{clk} is clock frequency operated, V_{DD} is the voltage operated, C_j and α_j are the capacitance of load and the number of transitions per clock cycle of the j^{th} node respectively. The power estimation is based on the above equation to determine the total number of transitions in each node of a circuit. The power estimation is very simple in regular switching. For example a node with C_L response capacitance and a clock signal is generated with frequency f, the power on an average dissipated is $V^2_{DD}fC_L$, in which V_{DD} is the voltage that is operated. The switching activity generally depends on the pattern of input. This makes difficult in the power assessment. For such difficult systems, it is mostly not possible to simulate the circuit of all achievable inputs. However, the primary inputs should be separate. The main reason why individual power is needed is to find the circuit components that consume more power. The fraction of cycles in which transitions and average power of an input signal can be calculated using probability techniques in a single simulation.

Power-play Power Analyzer CAD tool is provided by Altera to FPGA designers, through which the power can be estimated after completing the design. Altera also gives an early power estimator file called the *Quartus II Early Power Estimator*. The user usually needs the design to run using any specific test vectors or test tools to analyze the CAD output. The power analyzer Power-play estimate the system power and fits the target device for finding out how the output design is routed and positioned. Four types of signal sources can be activated with the Power-play power analyzer. Fig. 1 shows the possibility of dividing total dissipated capacity into two parts: one due to IO and VCCIO pin currents, linked to a 3.3V power supply. See attachment1 for additional information. After the design is synthesized, the power analyzer Power-play estimates system power and adapts the goal for knowing about the design which is placed and routed. The ammeter is measured directly by placing it on the tool that is added to this resistor.The Power-play power analyzer makes four types of signal sources possible.

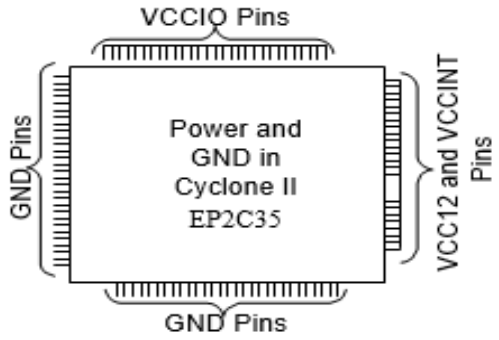

Fig.1:- Pins for Supply of Power in Cyclone II EP2C35

In the FPGA device the currents drawn by the VCC12 and VCCINT pins form the second element. Directly connected to the 1.2-V controller output are the VCC12 and VCCINT pins. We need to put a small 1.25Ω shunt to the given input of the controller circuit in order to evaluate the overall power drawn by pins of VCC12 and VCCINT, since the difficulty of the circuit does not facilitate a shunt in the response. Evaluating the input power does not change the response power, as the power flow through the regulator's ground that to only a few mA. The below Fig.2 shows the currents drawn from VCC12 and VCCINT pins through which measuring is done in a schematic way. The power drawn by VCC12 and VCCINT pins in the center of the FPGA represents the voltage.

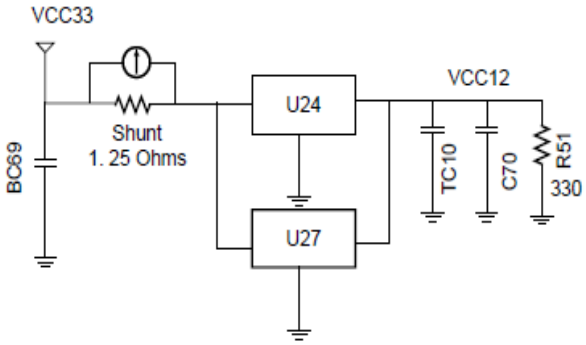

Fig. 2:- Power measurement with VCCINT and VCC12 pins.

We measured many power ratings for the similar design and estimate them with the expected power for several frequencies. The following Fig.3 explains that the Quartus II Power-play Analyzer was an instrument of exact power analysis. It was reported that it provides an average accuracy of ±10 percent with accurate design information. Compared with the Power-play estimate in an on chip memory system and a NIOS II, this accuracy result is in line with our conclusions.

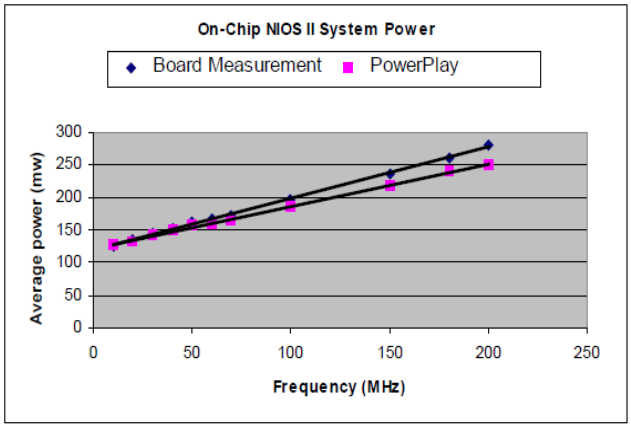

Fig.3:- Power-play Estimation and Real Power Measurement

The accuracy of the Power-play Power Analyzer's results has been evaluated for actual cyclone II power measurement, in which we have used several HW-Designs to multiply the matrix of n×n & NIOS II\e of basic computer system. The following Fig.4 shows that the Power-play Power Analyzer Quartus II is a tool that provides accurate power estimates of 7 percent on average, compared with actual measurements.

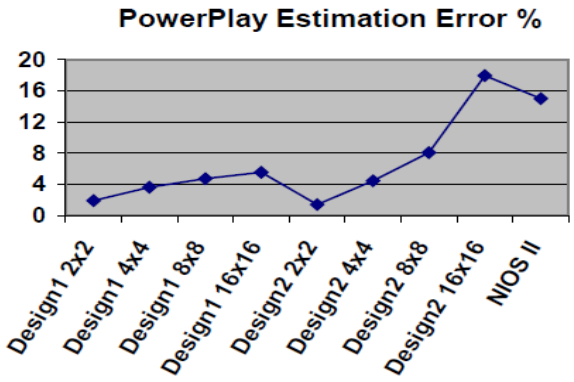

Fig. 4:- Power-play real measurement compared with expected error.

A. Multiplication of Matrices in three forms

In three different designs, we analyzed and compared energy and power use, multiplying two A and B matrices of n to 32-bit items and producing n to 64-bit C matrices. The two models utilize the Hardware system of FPGA with varies 2n and $2n^2$ registers for storage and the last one uses the NIOS II\e system. We have also shown that NIOS II\e is not an alternative for the energy efficiency of multiplying n×n matrices in comparison with the HW multiplier in FPGA. Here we actually represent a highly standard algorithm, architecture and resources which are used in 3 fully various matrix solutions. Two classifications of design include our solutions:

- General Purpose Processor based design.
- Single Purpose Processor based design.

A solution for the design of *Altera NIOS* that actually utilizes an easy and basic converter in the chip storage for multiplying the array. *First design VHDL program* and *Hardware second program design VHDL*; neither solution uses a new program memory to multiply the matrix but uses several various equipments. *Physical first design VHDL* acquires 2m registers for volatile stored, m multipliers and m-1 continuous adders, *Physical second design VHDL* and $2m^2$ registers that temporarily elements are stored m-1 cascaded adders are used.We had used a computer network which is actually based on a fundamental core processor that uses a user-friendly multiplication results in accordance with a NIOS II\e on-die storage and an algorithm and also a summation convention also in the Fig.6 below.

```
for (l = 0; l < p; l++) {
for (m = 0; m < p; m++) {
for (n = 0; n < p; n++) {
C[l][m] = C[l][m] +A[l][n]*B[n][m];
} } }
```

Fig.5:- Convention algorithm with matrix multiplication

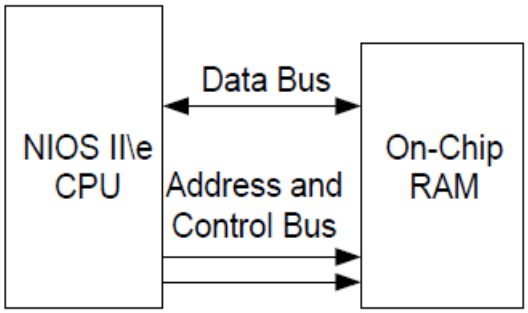

Fig.6:- On-die memory based on basic computer system

There is no HW multiplier in NIOS II\e and NIOS II utilizes the Shift Add multiplication algorithm that causes a huge throughput difference from HW multiplication solutions. Matrix multiplication for first hardware problem uses temporary 2m registers for holding components with m-1 cascaded adders and m multipliers. In design architectures wherein separate memory blocks with matrices A, B and C are used as illustrated in Fig. 8

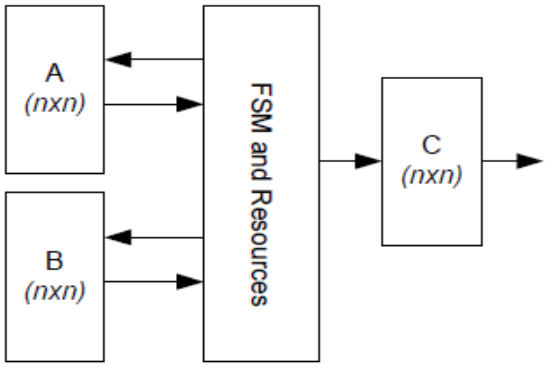

Fig. 7:- Matrix multiplication for hardware solution in high-level architecture

For calculating a line in the C matrix, we want A readings and also the B matrix (n-1) n readings because the first column of the B matrix is almost actually read in conjunction with any matrix of A row. The latency for m rows is given by:

$$L = [m + (m-1)m]m = m^3 \qquad (2)$$

Because the memory access gap remains 3, the set of first m registers set is devoted for rows A and the next set of m logs dedicated for columns B. Each line in A is only to be found once in order to calculate the entire C matrix, while the B columns should be found m times. This parallel by removing the exponentiation product m^2, reduces latency only through the equation of latency and allows matrices to be separated by the various memory blocks, which allow computed elements to be written similarly and the first column of the B matrix read in parallel to any row of matrix A. The Fig. below explains the diagram of timing in which the matrix C items are available in the adder response and at C block memory locations.

ISSN No:-2456-2165

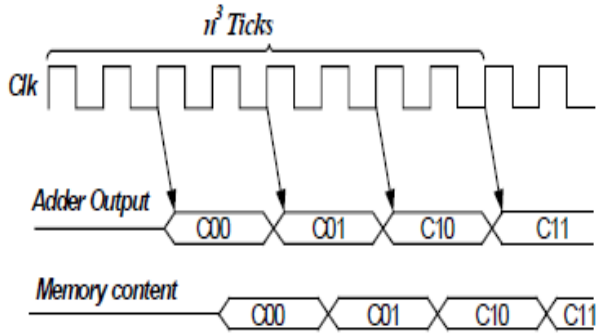

Fig. 8:- Diagram for timing using 2x2 matrix multiplication.

B. Hardware design 2 – VHDL program

Here we maintained m-1 cascaded adders and m multipliers with same architecture even with the previous solution in the HW Matrix solution. However, considering temporary storage registries rather than 2m registries, we can be even used $2m^2$ to save memory full occupied gaps. The following Fig. shows the high-level algorithm that is used for C matrix calculation. The above algorithm and the $2n^2$ registers have reduced the order of such complexity to 2. We preserve A lines along with B columns during first m^2 cycles while calculating the starting C lines, the data for calculation is in availability. During this whole period, storage takes place after each n cycle of such one line of A matrix as well as a column of B matrix and single item in C row starting will be calculated. Next m^2 ticks of clock, B and A are previously placed and no additional memory occupation is required for getting B and A terms, at this time remaining C matrix terms are produced at each and every clock cycle. Latency may be written as follows:

$$L = m^2 + m^2 - m = 2m^2 - m \quad (3)$$

Fig. 9 diagram for timing at the response of adder and at memory of block C is placed which are in availability.

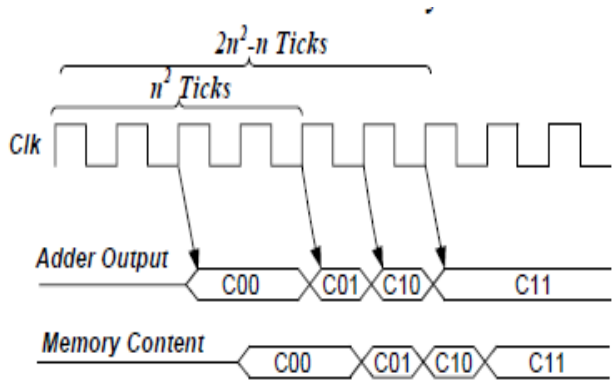

Fig: 9:- Diagram for timing using 2x2 matrix multiplications.

III. RESULTS

Here, we have discussed energy as well as power resources. We shall also examine the massive variation in between energy and power because in designing of low-power both the field terms were freely interchanged.

A. Power& Energy Analysis

When comparing, all the 3 designs have the same multiplication in matrix at 50 Mega Hz. No matter how large the issue and the activity of switching explained in the Figure 1, NIOS II processor appears to be dissipating almost the same power amount. In comparison again with dispersed static power with the entire core of FPGA along with consumed dynamic power through function with increase of dynamic activity is not really dominant as a result of data changes. We will surmise that NIOS II isn't an extreme case design for internal switching observation.

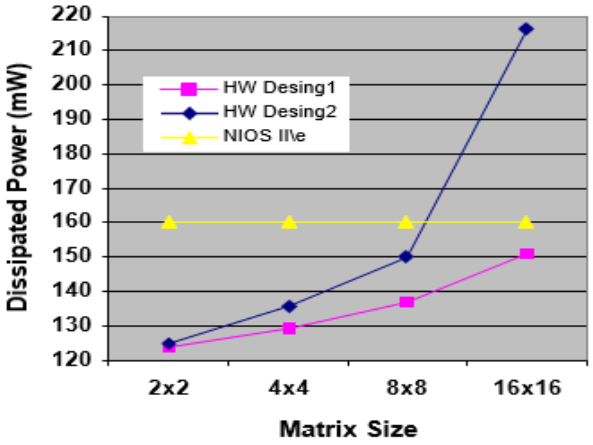

Fig. 10:- Different designs with matrix size versus power dissipated.

This also demonstrates that Hardware Design1 utilizes low power than HW Design2 and NIOS II results. The dissipation is from 6% to 23% low power than NIOS II and also 16% low on average for every size of problems. Increasing the size of problem also increases the total number of signals logically and thus the power dissipation goes high as the total number of switching activities per one second increases. Even the area of NIOS II is lower than the Hardware Design1 to the 8x8 and 16x16 sizes of matrix, the power consumed is higher, which also tells that the NIOS II results in larger efficient capacity, the final conclusion is that bigger design did not every time mean larger dissipated capacity. For suppose we use 2 models with same voltage as well as to run at equal frequency, then larger design of dynamic power was any of larger total efficiency that, the total load capacity is increased by the total number of clock cycle transitions seen in the following equation:

$$P_{Dynamic} = \frac{1}{2} V_{DD}^2 f_{clk} \Sigma_{j=1}^{n} C_j \alpha_j \quad (4)$$

ISSN No:-2456-2165

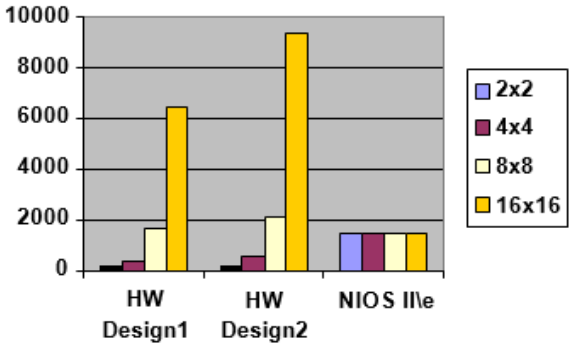

Fig. 11:- Different designs are occupied by number of logic elements.

Where f_{Clk} is frequency of clock operated, V_{DD} is the voltage operated; $C_j$and α_j are the capacity of load and the total number of change of state of the j^{th} node per clock cycle. The total effective capacity is larger when distinguished to the larger design; the smaller design dissipates higher power [9].

For the multiplication of small matrix sizes, like as 4x4 and 2x2, Design2 gives nearly the same current as first design, which gives 0.8 percent and 6 percent high power in 4x4 and 2x2. The effect of $2m^2$ not permanent storage registers, on the other hand, only appears in higher sizes such as 16x16 and 8x8, where second design gives 10 percent and 44 percent extra power than first design for 16x16 and 8x8. For matrix multiplications of different sizes like 4x4 and 2x2, second design gets 23 percent and 16 percent low current than NIOS II. Whatever, for bigger sizes, the power loss in second design2 maximum reaches the NIOS II results the size of 8x8 at 8 percent low dissipation in power and also exceeds NIOS II power taken by 36 percent with the size of 16x16, because of the large total number of registers that are clocked from $2m^2$ temporary elements stored.

Even if the designs were designed for mobile devices often driven by batteries with a separate computer time, the mean power dissipation isn't the measurement for battery consumption, and naturally current is almost every time better for heat dissipated. The below figure gives how maximum energy of dissipation to each size of matrix multiplication.

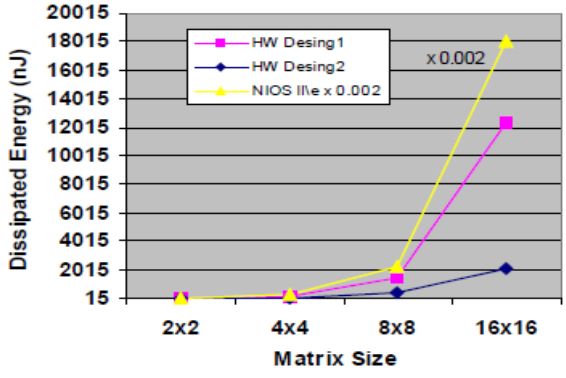

Fig. 12:- Matrix size versus dissipated energy compared with first, second designs and NIOS II\e solution.

The impact of waiting time on energy efficiency is significant. For multiplying matrices, Design2, which brings higher current as well as occupies more space than Design1, is more economical. The energy that is saved is as follows and the difference in waiting time is $m^3 - 2m^2 - m$ and it is going high constantly with m. We also can notice that the 2nd design saves energy significantly, mainly for larger matrix multiplication dimension like 8x8 and 16x16. Fig. 12 shows that, because of the absence of such a Hardware embedded multiplier in which the processor and the difference was enormous; NIOS II\e was very much away from competing along with the Hardware designs to conservation of energy. The following graph gives us how much the energy is preserved when transferring from first design to second design.

When multiplying by the shift add algorithm, energy is mainly wasteful. Between 1271 and 4208, NIO S II\e dissipates more than Design2 which is more than Design 1 between the 961 and 729. These energy differences were valid to the matrix that are utilizes in all our attempts, because the variation depends only on the total number of 1s as well as the maximum possible load for to left operands b [j] [k] a reservation, even if the gaps still are large, if we look at top case of latency when A is a zero.

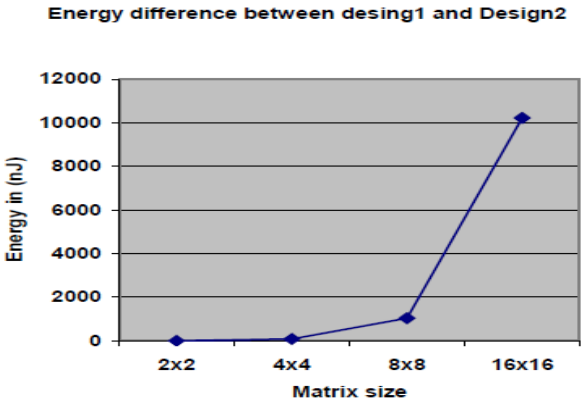

Fig. 13:- Second design using saved energy versus matrix size.

IV. CONCLUSIONS &DISCUSSIONS

In low power design, ASICs enable a high degree of flexibility and all HW technology could also be used for ASICs and not for FPGAs. There are many techniques in programmable processors for minimizing capacity, but most of those are not helpful for on-die memory, like we saw in NIOS II\e. Altera's power estimate tool is a reliable power estimation tool with an average 7% absolute error, but true measurement is always the best way to accurately sensitize the energy consumption. Design2 is a fast as well as energy-efficient design both for n×n matrix propagation using $2n^2$ temporary storage elements as matched with first design that employs 2m temporary elements stored until current, surfaces, waiting time and energy are combined. NIOS II\e converter, instead, it is not a better option for multiplied array, specifically to built-in storage power systems, which underlines installer-flow design

inefficiency. A possible factor in reducing energy costs is migration into a non-installation based configuration.

The NIOS C2H acceleration compiler is also very worth using for future operations to watch up-to what extend it will affect NIOS II\e energy loss. It is important, however, that other processors which have a built-in HW multiplier as well as other architecture instead of a simple RISC processor be involved in order to enrich our study with more information. Clock gating is a way of reducing current as well as in further work it could have been very much interested to read. The energy efficiency of the matrix multiplication component library also can be evaluated by several providers like Xilinx and Altera.

REFERENCES

[1]. Wild Andreas. And Gary Yeap. (1996). *International Journal of System and High Speed Electronics.* For Introduction To Low-Power VLSI Design. DOI: 10.1242/S0129156496000098
[2]. ZahiriBehrooz. (2003). *IEEEComputer Society.* Structured ASICs: Opportunities and Challenges. ISBN: 0-7695-2025-1.
[3]. Sheng, S.; Brodersen, R.W; Chandrakasan, A.P. (April 1992). CMOS digital design for Low-power. *Solid-State Circuits, IEEE Journal of Volume 27*, Issue 4, Page(s):473– 484.
[4]. Farid N. Najm.(1994). In VLSI Circuits a Survey of Power Estimation Techniques. *IEE*Tanrs. ISSN: 1063-8210.
[5]. Martin Morf, Michael J. Flynn, Oskar Mencer.(1998). PAM-Blox: High Performance FPGA Design for Adaptive Computing. *IEEE* Computer *Society*. ISSN: 1082-3409.
[6]. Viktor K. Prasanna and Choi Seonil. (2002). Energy Efficiency of FPGAs and Programmable Processors for Matrix Multiplication. *IEEE International Conference*. ISBN: 0-7803-7574-2.
[7]. Seonil Choi , Viktor K. Prasanna and Ju-Wook Jang. (2005). Energy- and Time-Efficient Matrix Multiplication on FPGAs. *IEEE International Conference.* IISSN: 1063-8210.
[8]. Farid N. Najm. (1994). In VLSI Circuits a Survey of Power Estimation Techniques. *IEE Tanrs*. ISSN: 1063-8210.
[9]. Prasad Sharat C., Kaushik Roy. (1999). Low-Power CMOS VLSI Circuit Design. A Wiley-Interscience publication. ISBN-0-471-11488-X. Pages 143-200, 321-348.
[10]. Prasad Sharat C, Kaushik Roy. (1999). Low-Power CMOS VLSI Circuit Design. A Wiley-Interscience publication. ISBN-0-471-11488-X. Pages 143-200, 321-348.
[11]. SHARAD.M, WOLFE.A, LEE and VIVEK.T. (1996). Instruction Level Power Analysis and Optimization of Software. *Springer Netherlands. ISSN: 0922-5773.*Pages:223-238.
[12]. Hassan Hassan. (2008). Phd thesis: In FPGAs Design Methodologies and CAD Tools for Leakage Power Optimization. *University of Waterlo.* http://hdl.handle.net/10012/3856.
[13]. Altera Corpora. (2008). Stratix IV Device Handbook. [Online]. Available: http://www.altera.com/literature/hb/stratix-iv/stx4_5v1.pdf
[14]. Altera Corpora. (2008). Stratix IV Device Handbook.[Online].Available: http://www.altera.com/literature/cp/fpgas-optimal-results-396.pdf
[15]. Malik.S, Wolfe.A, and Tiwari.V. (2002). Power analysis of embedded software: a first step towards softwarepower minimization. *IEEE Trans.* ISSN: 1063-8210 On. page(s): 437-445.
[16]. Andreas Willig. Holger Karl. (2005). PROTOCOLS AND ARCHITECTURES FOR WIRELESS SENSOR NETWORKS. ISBN: 978-0-470-09510-2. Pages 32-34.
[17]. Altera Corp. (2008). Power Analysis Power Play. [Online]. Available: http://www.altera.com/literature/hb/qts/qts_qii53013.pdf?GSA_pos=2&WT.oss_r=1&WT.oss=powerplay%20power%20analysis
[18]. RenbiAbdelghani. (2009). Low Power Design Techniques and Energy Modeling for ASICs, FPGAs and Processors. Master's thesis, Jönköping University.