Connectivity based Dual V_{dd} Assignment Algorithm for Power Reduction in FPGA

G. Veera Sekhar and Jatindra Kumar Deka
Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam, India

Abstract— *In the past few decades with advancement in VLSI technology, FPGA chip density has increased and FPGA devices now provide a large number of smaller feature size transistors and can support higher clock speeds. While this advancement is beneficial for implementing larger and faster designs within a single chip, it also leads to increased power consumption. With the remarkable growth of FPGA based battery-powered systems, such as personal computing devices, wireless equipment and consumer electronics, low power FPGA design is of increased importance. With this huge continuing increase in FPGA size, its complexity and the new technology emergence has made power estimation and optimization as an ultimate design consideration. In this work, we are supplying dual V_{dd} to the logic blocks, instead of the single V_{dd} used in traditional FPGAs. The assignment of dual V_{dd}(low/high) to a logic block is based on the criticality of the path. We have proposed an alternative algorithm to find out the critical path in the FPGA implementation of the circuit. Depending on the priority of a node in a critical path, we assign low V_{dd} to a node so that the performance of the circuit remains within the specified limit. Due to the application of low V_{dd} to some of the nodes in the FPGA implementation, the power consumption is reduced.*

Keywords: FPGA, Dual Voltage Assignment, Power Reduction, Critical Path

1. Introduction

FPGA, field programmable gate array is the most commonly used programmable device at present. FPGA is an integrated circuit(IC) designed to be configured by the customer or designer hence field programmable. Hardware description language (HDL) is generally used to specify the FPGA configuration similar to that used for ASIC. Using FPGA we can implement any logic function that an ASIC could perform [2]. The main advantages of FPGAs are lower development, lesser time to market and their ability to be reprogrammed several times. Due to these advantages, a lot of research has been done on development environment, architecture and applications for FPGA.

Nowadays usage of mobile devices and portable devices like mobile phones, digital cameras, notebooks, etc. increase rapidly. From end user point of view most often performance, features, size, and weight are the main criteria. This criterion has become the main design constraints in the design process and has an impact on the power consumption [1], [5], [7], [8], [10]. These demand for low power consuming devices and so power becomes most important issue in FPGA architecture design. As the application of FPGA extends to image processing and to other high complexity processes, a multimillion gate FPGAs are become necessary. These consume a lot of energy. Thus in case of mobile devices power optimization or maintenance techniques has become more important to guarantee long battery life. Even in non mobile devices, where power is continuously available the low power design constraint is still important. In many applications we can achieve the desired performance by increasing the operating frequency under given power constraints. It is crucial to implement a power efficient FPGA design without affecting the performance for many FPGA systems. There are a number of techniques proposed already for the reduction of power in ASIC domain. But, they cannot be applied directly to FPGA's, because FPGA and ASIC differ in architecture design. One more disadvantage about consuming high power is that devices generate lot of heat which further affects device performance though we have some techniques to get rid from the heat generated. Low power techniques are much better than in some cooling techniques such as sinks and fans. Thus thermal management and power management has got more importance in the FPGA design architecture [3], [4], [6], [9], [11], [12].

Moreover, programmability of FPGA can be leveraged to develop efficient low power design techniques. There are two possible approaches for the reduction of power consumption by FPGA based systems. (i) Redesign the FPGA device to reduce the static power and its components which contribute to dynamic power such as output gate capacitance and resistance of CMOS gates etc. (ii) take dynamic power into consideration while designing an FPGA circuit. The first option will require modification of FPGA architecture and topological change in its implementation. The second option mainly considers about reducing the dynamic power. Dynamic power consumed by an FPGA is primarily dependent on clock frequency, switching activity, supply voltage and resource utilization. In this work, we consider the reduction of dynamic power by reducing the supply voltage and for that we use two voltage level low V_{DD} and high V_{DD}.

A field-programmable gate array (FPGA) is a user or customer configured integrated circuit, and so is called "field

programmable". A HDL (Hardware Description Language) language can be used to specify the configuration of a FPGA, similar to that an ASIC.

Nowadays FPGA's are used in many applications including many portable devices like mobile applications. Considering both high performance and mobile applications, power consumed by these have become a limiting factor for FPGA's wide applications.

Low power design is important from three different reasons, there are mainly, device temperature, life of the battery and overall energy consumption.

The remaining of the paper is organized as follows: In Section 2, we present the basic architecture of FPGA. Section 3 and Section 4 deal with the power sources of FPGA and low power techniques respectively. The dual V_{dd} assignment algorithm is presented in Section 5. The experiment results is shown in Section 6 and finally the conclusion of the paper is given in Section 7.

2. FPGA Architecture

2.1 Logic Block

Logic blocks are the primary elements of FPGA through which any function can be implemented. The capacity of the logic blocks can be increased by increasing the size, e.g., by increasing the number of inputs, one can augment the possible functions which could be implemented by less number of logic blocks. But the research works showed that the area delay product will increase with the increase of the size. However this could be a waste in some applications where not all inputs are utilized. FPGAs use Look Up Tables (LUTs) for implementation of logic functions. With n input LUT one can implement 2^m possible functions, where m is 2^n and each function requires 2^n bits configuration. Previous work has shown that 4 inputs LUT is optimum in terms of area and power [3]. Fig. 1 shows the basic logic which consists of one 4-input LUT, where a combination function is implemented. A flip-flop that will be needed in sequential circuit design. A 2-to-1 MUX is used to switch between registered and unregistered output.

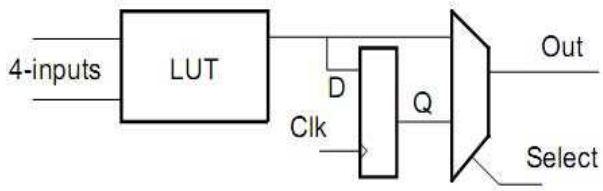

Fig. 1: Basic Logic Block

2.2 The Connection Block

The connection block is responsible of connecting the resources between each other and assures that data can flow to the I/Os. Each connection block consist of a programmable connection block which selects the signals in the given routing channel to be connected to the logic block's terminal, and a programmable switch block that connects between horizontal and vertical routing resources. The structure of Connection Block is shown in Fig. 2.

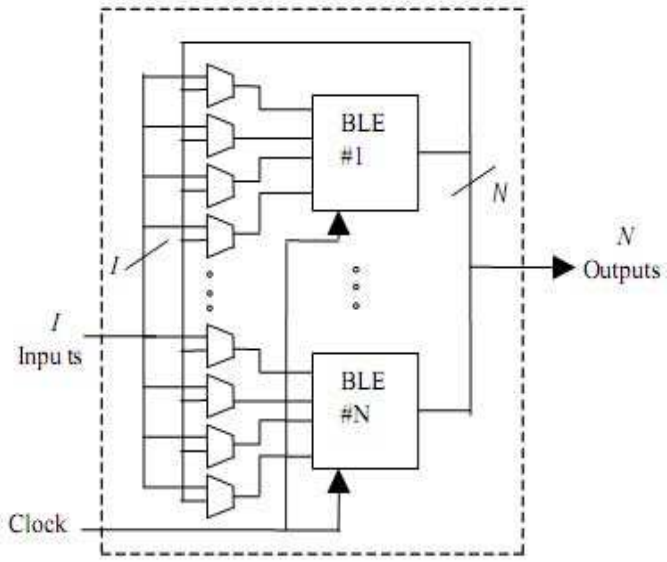

Fig. 2: Clustered Based Logic Block

The programmable interconnect is the core of an FPGA, which connects different LUTs and flip-flops to each other and route signals to and from the input output blocks(IOBs). In addition current generation FPGAs feature a large number of on chip synchronous blocks and a few on chip macro functions such as multiplexers and shift registers. The LUTs in current generation are clustered into a logic element or configurable logic block (CLB).

2.3 Routing Architectures

Since the routing wires consume a major part of the total amount of FPGA area and power, selection of routing architecture is crucial in a FPGA design. Island style routing architecture is used in this work. In Island Style Routing Architecture the logic blocks are connected by a two dimensional, mesh-like interconnect structure with horizontal and vertical routing channels and these are connected by programmable switch boxes. A simplified view of island style routing architecture is shown in Fig. 3. In this routing structure, half of the routing tracks consists of length 1 wires (wire that span for one logic block) and remaining half consists of length 2 wires. Pass transistorize tri-state buffers are used as programmable routing switches. There are also connecting boxes connecting the wire segment to the logic block inputs and outputs.

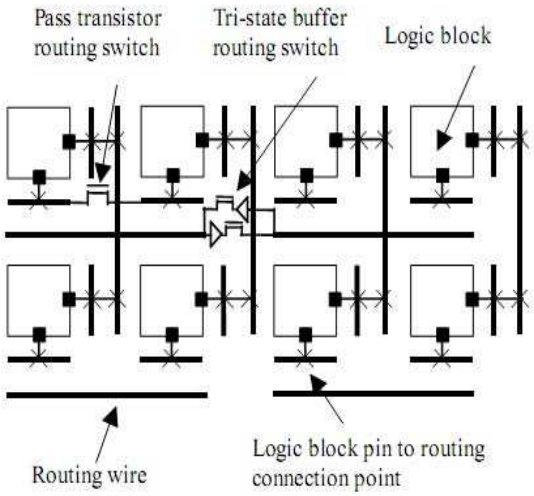

Fig. 3: Island Style Routing Architecture

3. Power Sources in FPGA

In FPGA, power is mainly divided into two categories - static power and dynamic power. Static power is the power consumed by the node when there is no signal transition to that node, that is, the power required to keep it on. Static power dissipation is mainly due to leakage current. Leakage current is caused due to two main reasons - Reverse biased diode leakage current and Sub-threshold conduction of transistors [4]. These leakage currents depend on supply voltage and the technology used. By reducing the supply voltage, the leakage current may be reduced.

Dynamic power is the power consumption caused by dynamic circuit activities which are reflected by dynamic current flowing between two circuit nodes. On the other hand, we can say that dynamic power is proportional to the signal transitions of a design that mapped into an FPGA. This dynamic power is consumed only when there occurs a signal transition at gate output and there are two types of signal transitions.

- *Functional transition:* This is the signal transition necessary to perform the required logic functions between two consecutive ticks.
- *Glitch or spouse transition:* It is an unnecessary signal transition due to the unbalanced path delay to the inputs of a gate. Glitch power consumes most part of the dynamic power. So, a change from supply voltage is transferred to the CMOS gate output capacitance when the output of a CMOS gate changes from level 0 to 1. This transition causes power dissipation in the resistive CMOS network.

The dynamic power consumed by an FPGA is primarily dependent on Clock frequency, Switching activity, Supply voltage and Resource utilization. Unused resources in an FPGA design have a switching activity nil and hence they do not consume any dynamic power. Hence dynamic power consumption in FPGA is proportional to the amount of resource used by a design.

Switching power is the power that is consumed due to the current charging and discharging the capacitance at the gate output. A CMOS inverter is a good example to illustrate this analysis of switching power. When the input to the inverter goes low, the NMOS transistor is cut-off and the PMOS transistor conducts, creating a direct path from the supply voltage to the output capacitance. The equation for switching power is given as,

$$¶_{sw} = f.{V_{dd}}^2.\sum_{i=1}^{n} C_i\, E_i \qquad (1)$$

Where n is the total number of nodes, f is the clock frequency, V_{dd} is the supply voltage, C_i is the load capacitance for node i and E_i is the transition density for node i.

It is obvious from Equation 1, the switching power can be reduced by reducing the supply voltage. But the reduction of supply voltage may degrade the performance of the circuit. In this work, we are exploring the use of dual voltages in FPGA. We will apply high V_{dd} to the nodes which are in critical paths and effect the performance of the circuit. We will apply low V_{dd} to others nodes.

4. Low Power Techniques

In our work, we use dual V_{dd} architecture where, the V_{dd} of a circuit block is selected between V_{ddh} (high V_{dd}) and V_{ddl} (low V_{dd}) by using two transistors connecting the block to the supply voltages. By using our V_{dd} assignment algorithm, we will set a configuration bit which controls the on/off state of each supply voltage transistor. The configuration bit is set in such a way that the block will either connect to one of the power supply or it will disconnect completely from power supply if the block is not in use. That is, we will switch off the block when it is completely idle or unused. This reduces the static power required to a block when it is idle.

We are using cluster based island style FPGA architecture for our proposed dual V_{dd} architecture and the configuration bits are stored in SRAM cells. This facilitates configurable supply voltage for logic blocks and routing multiplexers. The architecture of the FPGA is shown in Fig. 4. The basic logic element consists of one 4-input LUT and one flip-flop. A CLB is formed with the clustering of 8 BLE's together. Dual V_{dd} design requires level conversion when a low V_{dd} block drives a block operating at high V_{dd} and vice versa. In our dual V_{dd} architecture, level conversion only takes place at CLB pins. So, we attach the level converters to the CLB pins.

This placement of level converters (LCs) at the CLB pins reduces the complexity. We have two different architectures

for placing the level converters at CLB pins [16]. One at CLB input pins and another CLB output pins. Fig. 4 shows the second architecture, where LC's are placed at CLB output pins.

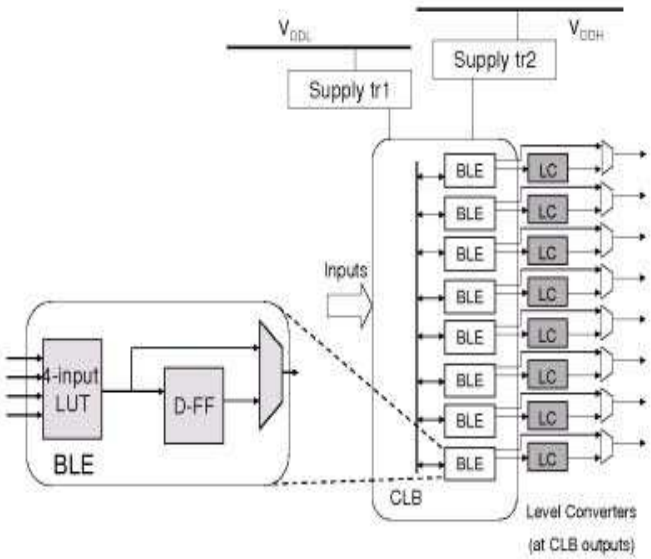

Fig. 4: CLB used in Dual v_{dd} architecture

A routing multiplexer is used in dual V_{dd} architecture as shown in Fig. 4. As per our requirements, the output of previous stage is directly given to the next stage or it may be provided after level conversion. Level converters are used to change the level of a signal voltage from low voltage to high or high to low.

5. Connectivity Based Dual V_{dd} Assignment

This algorithm takes all the nodes and critical paths into consideration. It arranges all the nodes of the critical paths based on their connectivity factor. Connectivity of a node defines the number of paths connected to the node. If a node is connected to more number of the paths than the other nodes then it has the highest priority to get assigned high V_{dd} than the other. Before assigning the high V_{dd} to this nodes, we check whether all the paths passing through this node meet the delay requirement. If some of the paths do not meet the delay specification, we assign high V_{dd} to this selected node. After assigning high V_{dd} to this node, if the performance does not improve or the delay requirement goes below the specified limit, we backtrack and reassign the low V_{dd} to that node. We repeat this procedure for other nodes till we achieve the required performance.

Input: Critical Path nodes and delay factors
Output: Critical path nodes with high V_{dd}
Assign V_{ddh} to all CLBs and routing MUXes;
$P \leftarrow$ List of all paths in the design;
$T \leftarrow$ Longest delay path when all circuit blocks operate at V_{ddh};
$T_d \leftarrow x$*T, where $x >= 1$ is a user defined performance metric;
V_{ddl}Delay(P_i) $\leftarrow$ Delay(P_i) when all blocks in P_i are at V_{ddl};
Path List(P) $\leftarrow$ {All paths $\|$ Criticality(p_i) $>$ Criticality (P_{i+1})};
Critical Path $\leftarrow$ {$P_i \in P \| V_{ddl}$ Delay(P_i) $> T_d$};
N $\leftarrow$ All CLBs in the circuit sorted in descending order based on their priority given to them based on their Connectivity;
while *N is Not Empty* **do**
  Assign V_{ddh} to N_i and to all the routing MUXes driven by N_i;
  Remove N_i from the list;
  if *Delay of any PathList(P_i)* $< T_d$ **then**
    Reset last action;
  else
    Update delay of all paths passing through N_i;
  end
end

Algorithm 1: Priority Based Dual V_{dd} Assignment Algorithm

In this work, we are proposing a low-high priority based algorithm. We first assign V_{ddl} to all CLBs and routing muxes and calculates the delays of each path. After the calculation of delays, we sort the paths such that longest delay path should come first (descending order). Now, define a performance metric T_d, T_d=x$*$t, where x $>$=1 (t is the longest delay) as a threshold voltage applicable to circuit. This should be different for different circuits. Now, consider the CLBs and calculate the number of paths to which each CLB is connected. Assign the number to priority variable. Now, sort them (in descending order) based on their priority value. Now, we consider each path one by one and assign the high V_{ddh} to the first occurring block of that path. Now, check if any path violates this allocation. That is if any path delay that involves this block exceeds than its threshold delay. If there is no path like that, we proceed to next node, otherwise we will reset the block voltage and proceeds further.

This algorithm gives a relatively better V_{ddl} allocation for the nodes in a circuit. We are using low-to-high assignment of V_{dd}. This is preferable than high-low, because in high-low all the blocks will be assigned high voltage at the beginning and will be assigned low voltage afterward. In this each CLB consumes high V_{ddh} at startup and sometimes this may cause

the paths delay to be reduced more than what is necessary in assigning low V_{ddl}.

6. Experimental Results

In this work, we are trying to reduce the power consumed by the FPGA circuits. We proposed one algorithm, which is used to assign the voltages to the nodes such that the total power consumed by the FPGA circuit will be reduced without reducing the specified performance.

We try to decrease the dynamic power consumed by the FPGA circuit. The power consumed by the FPGA circuit is the total power consumed by its each and every individual node. The circuit includes LUTs, switch boxes, input and output pads. So to decrease the power consumed by the complete FPGA circuit means, we have to decrease the voltage (power depends on the voltage applied) applied to the individual node. There are some critical paths in the circuit and we must ensure that the total delay of the critical path shouldn't vary as it is pre programmed to get the expected result. Though we can change the delay metrics of individual nodes of the critical path until we ensure the total path delay is not altered.

To reduce the power consumed by the FPGA, we will assign low voltage (we divide the voltage into two operative voltages depending on the technology used) to the nodes of the circuit that are not on the critical path. We will assign high voltage to some of the nodes on the critical path based on our proposed algorithm. In the process, we must ensure that the circuit performance should not degrade while trying to reduce the power.

In our experimental set up, we have used the ISCAS89 bench mark circuits to show the benefit of our method. The detail flow of our experiment is:

- Logic synthesis and optimization
- LUT mapping (cluster based)
- Packing using t-v pack
- Placement and routing
- V_{dd} assignment
- Power estimation

For our simulation purpose, we have taken the ISCAS89 benchmark circuit in .bench format. We have used the ABC synthesizer for logic synthesis and logic optimization. Next we use the RASP tool for technology mapping, which maps the circuit to LUT based FPGA. It also does the post processing for area reduction. After that we use the placement and routing tool VPR to place and route the FPGA circuit.

We have implemented the algorithm for the assignment of voltage to the different nodes of the FPGA. According to the algorithm, it assign either low voltage or high voltage to a node.

We have used the power evaluation tool FPGAEVA-LP to estimate the power consumed by an FPGA implementation of the benchmark circuits. This tool also estimates the delay of the circuit. For comparison, we have estimated the power consumption by the existing method [14] and by our proposed method.

Circuit	Gates	LUTs	Cri.Nodes	V_{ddh} Nodes(pro)	V_{ddh} Nodes(Exi)
S27	8	10	21	11	16
S400	106	162	32	19	22
S444	119	181	36	19	21
S838a	288	446	50	6	8
S953	311	418	48	26	31
S208	61	96	42	16	21
S298	75	119	30	18	26
S382	99	158	32	20	29
S510	179	211	34	21	26

Table 1: Comparing Both The Solutions

The experiment result is presented in Table 1. The first column of the table indicate the circuit number that we have used. Second and third column provide the number of gates of the circuit and number of LUTs used during FPGA implementation respectively. Column four gives the number of critical nodes of the circuit. Column five and column six indicate the number of nodes to which high V_{dd} have been assigned by our proposed method and the existing algorithm respectively. It has been observed that our proposed method has assigned high V_{dd} to less number of nodes in all test circuits in comparison to the existing method.

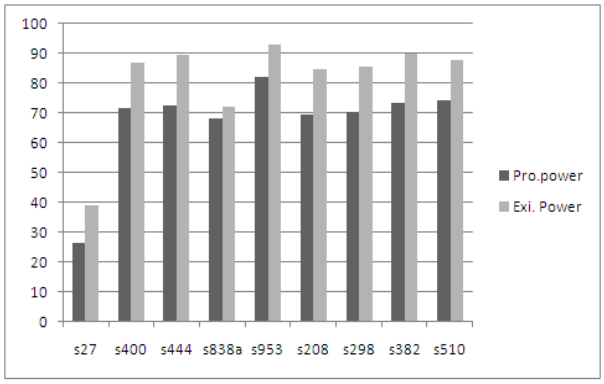

Fig. 5: Graphical Representation of Power consumed by two algorithms

The power consumed by each circuit in both the algorithms are represented graphically in Fig. 5. It is observed that in our proposed method the power consumption is always less. The delay experienced by each circuit in both the algorithms are represented graphically in Fig. 6. Obviously the delay incurred by a circuit in our proposed method is more than the delay incurred by the existing algorithm. It is quite obvious that in our case we are applying high V_{dd} to less number of nodes. But in both the cases, the delay incurred by each circuit is always less than the specified delay of the circuit.

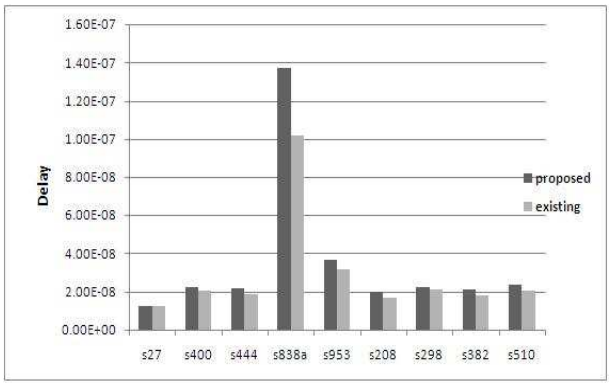

Fig. 6: Graphical Representation of delay by two algorithms

Circuit	PowerSaved(%)
S27	32.436919
S400	17.419271
S444	18.656502
S838a	12.443598
S953	12.013707
S208	17.867090
S298	17.704979
S382	18.536043
S510	15.448448

Table 2: Total Power Saved

The percentage of power saved by our method is shown in Table 2.

7. Conclusion

We have presented a Dual V_{dd} assignment algorithm for low-power FPGA based on priority. The priority of the node is determined by the critical path. If a node is present in several critical paths, then it contributes more for the degradation of performance. In such situation, we apply high V_{dd} to this node. After applying the high V_{dd} to such nodes, we again check for the performance of the circuit. In experiment result, we have observed that our proposed method able to reduce the power consumption without affecting the performance of the circuit.

References

[1] A. P. Chandrakasan and R. W. Brodersen, "Low Power Digital CMOS Design Norwell," *MA,USA: Kluwer Academic Publishers*, 1995.

[2] "Introduction of FPGA," Wikipedia.org.

[3] H. Hassan, "Design Methodologies and CAD Tools for Leakage Power Optimization in FPGAs,"University of Waterlo., 2008.

[4] J.M. Chang and M. Pedram, "Register allocation and binding for low power," in DAC '95: Proceedings of the 32nd annual ACM/IEEE Design Automation Conference. New York, NY, USA: ACM, 1995, pp. 29-35.

[5] N.Weste and K. Eshraghian, "Principles of CMOS VLSI design," AddisonWesley VLSI system series, 2008.

[6] H. Veendrick, "Short-circuit dissipation of static cmos circuitry and its impact on the design of buÂőer circuits, Solid-State Circuits," IEEE Journal of, vol. 19, no. 4, pp. 468-473, Aug 1984.

[7] A. Raghunathan, N. K. Jha, and S. Dey, "High-Level Power Analysis and Optimization," Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[8] Chandrakasan.P and Brodersen.R, "Low Power Digital CMOS Design," Kluwer Academic Publishers, 1995.

[9] J. P. Knight and R. S. Martin, Power-proÂŕler: "Optimizing asics power consumption at the behavioral level," Design Automation Conference, pp. 42-47, 1995.

[10] S. Malik and S. Devadas, "A survey of optimization techniques targeting low power vlsi circuits," Design Automation Conference, pp. 242-247, 1995.

[11] M. E. Rabaey, Jan M.; Pedram, "Low Power Design Methodologies," Kluwer Academic Publishers, 1995.

[12] F. Li, D. Chen, L. He, and J. Cong, "Architecture evaluation for power-efficient FPGAs," in FPGA '03: Proceedings of the 2003 ACM/SIGDA eleventh international symposium on Field programmable gate arrays. New York, NY, USA: ACM, 2003, pp. 175-184.

[13] D. Chen, J. Cong, F. Li, and L. He, "Low-power technology mapping for fpga architectures with dual supply voltages," in FPGA '04: Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. New York, NY, USA: ACM, 2004,pp. 109-117.

[14] Fei Lei, Yan Lin, Lei He, "FPGA Power Reduction using Configurable Dual V_{dd}," Elcetric Engineering Dept, UCLA,2006

[15] Fei Li and Lei He, "Power Modeling and Characteristics of Field Programmable Gate Arrays", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[16] Fei Lei, Yan Lin, Lei He, " Low power FPGA using pre-defined Dual-V_{dd} Dual V_T fabrics", Electrical engineering department, University of California, Losangels.