

УДК 621.3.049.771

V.Sh. Melikyan, Sci. D.<sup>1</sup>, A.M. Poghosyan<sup>2</sup>, A.A. Durgaryan<sup>2</sup>, H.P. Petrosyan<sup>2</sup>, M.M. Simonyan<sup>2</sup>

## Method of Parametrical Optimization of Multi-Core Processors

A post-layout design power optimization algorithm is suggested. Both, gate sizing and multi threshold optimization methods are implemented. The main advantages are the improved performance characteristics and intactness of the initial design placement and routing. Free layout spaces due to decrease of optimized cell sizes is suggested to be filled with decoupling capacitors which decreases power supply noises. The algorithm ensures decrease of static and dynamic power by respectably 19% and 11% for eight-core OpenSPARC processor architectures. It demonstrates improved optimization time compared to existing algorithms by about 29%, in expense of decrease of optimized power by 2-5%.

Предлагается алгоритм оптимизации энергопотребления (ОЭ) на пост-топологическом этапе проектирования. Одновременно применяются методы калибровки логических элементов и многопороговой оптимизации. Главными преимуществами предлагаемого метода являются быстрое действие и минимальное воздействие на начальное размещение и трассировку. Свободную площадь, образованную вследствие оптимизации, предлагается заполнить емкостными ячейками, которые уменьшают помехи через цепь питания. Применение алгоритма обеспечивает уменьшение статической и динамической потребляемой энергии соответственно на 19% и 11% для восьмиядерного процессора OpenSPARC. Предложенное решение превосходит существующие алгоритмы по скорости оптимизации примерно на 29%, уступая им по эффективности ОЭ всего на 2-5%.

**Ключевые слова:** оптимизация энергопотребления, калибровка, многопороговое напряжение, временной резерв, многоядерный процессор, ток утечки, ток переключения, помеха через цепь питания, фазовая автоматическая подстройка частоты.

### Introduction

The sizes of integrated circuit (IC) devices are being continuously scaled in order to gain in performance, chip density and price. Together with the transistor sizes working voltages are also scaled, hence the threshold voltages must be decreased

correspondingly [1]. This results in exponential growth of sub-threshold drain-source leakage current. The gate oxide thickness scaling increases the leakage current due to direct gate tunneling exponentially as well. These phenomena dramatically increase total power consumption of a chip. Until recently decreasing the overall power consumption by decreasing the dynamic current while increasing static power was an acceptable approach because the static power due to sub-threshold leakage could be neglected. However, as transistors shrink below 90nm the static power becomes comparable to the dynamic power hence it should be taken into consideration during design process.

The problem is complicated by the IC elements growing susceptibility to process variations. There are studies that show that 30% process variation can result in up to 20-fold leakage power growth [2]. Problem of power consumption exists for all modern very large scale integration (VLSI) systems. It is especially expressed for multi-core processor systems.

In this paper a post-layout power optimization algorithm based on multi threshold ( $V_{th}$ ) and gate sizing is suggested. It can be effectively used for multi-core processors power optimization.

Generally reducing either the  $V_{th}$  voltage or the physical size of a gate leads to the gate delay increase which implies decrease of slack time. From this point of view the dual threshold and/or gate sizing can be effective for the delay-constrained optimization problems only if the given circuit has significant timing slacks available with some or all of its constituent gates.

Techniques that require resizing the channel length and width of transistors [2] are good for custom applications and for planning standard cell library architectures. But they are less suitable for processors design flows that are based on already existing standard library components.

There are numerous power optimization solutions such as the combinatorial algorithm for gate sizing and  $V_t$  assignment introduced in [3]. However, this algorithm is restricted to tree topologies therefore it cannot be used for power optimization of multi core processors. The most optimization solutions like the one presented in [4] are based on sizing the gates for a minimal delay and subsequently optimizing the power. The main drawback is that cell placement and routing is usually affected significantly, which is not always safe in terms of timing.

In this paper a new design strategy is being introduced which can analyse the final post-layout design data to calibrate the non-optimal cells, which were placed because of for example over-constraining of timing.

The main advantages of the proposed algorithm are the improved performance and intactness of the initial placement of gates. This is made possible by timing slack assignment for individual gates. Besides, optimized cells are always replaced by smaller or equal in area ones. It is shown that the presented solution helps decrease static and dynamic power by about respectably 19% and 11% for eight-core OpenSPARC processor architectures. Free layout spaces due to decrease of cell sizes can be filled with decoupling capacitors which decreases power supply noises [5]. This can be especially important for phase locked loop (PLL) jitter decrease in systems with common analog and digital supply.

### 1. Proposed algorithm description

After getting final timing clean design ready for manufacturing, performing timing slack analysis is proposed. If design has significant timing slacks, they can be reduced to gain in power reduction. In Fig.1 a chart of timing slacks for a typical OpenSPARC T1 design is presented.

As we can see from the Fig.1 only approximately 16% of gates are on a critical path. These gates are not supposed to be modified during the optimization. More than 54% of gates have slacks larger than 0,2. These are the cells to be targeted during the optimization process.

The suggested optimization algorithm and its place in a design flow is shown in Fig.2 (a,b).

The algorithm starts by the input of files obtained from post-layout design. These files are StarRC extracted netlist, gate level Verilog netlist, static timing analysis (STA) report files and Liberty (.lib) file for the standard cells. After having all these data the algorithm starts optimization. The most important thing for the algorithm efficiency is the optimal timing slack distribution between logic gates. The simplest strategy of this is assigning equal slack to each gate along the path, however this approach is not the most optimal.

This method does not take into account that some gates can be more efficient in converting extra delay to power reduction, and should be assigned more "extra" delay. This leads to the necessity of introducing efficiency criteria for each individual gate. After assigning these criteria it will be possible to formulate and solve a linear programming problem. The following concept is proposed as such a criterion:

$$Z(i) = \frac{\Delta P(i)}{\Delta D(i)} \left( \frac{t_{\max}}{t(i)} - 1 \right), \quad (1)$$

where  $\Delta P(i)$  is the decrease in power of a cell with  $\Delta D(i)$  delay change,  $t_{\max}$  is the maximal rise/fall transition time specified for a particular design and  $t(i)$  is the output transition of the logic gate. First part of the expression is the gate's power-delay sensitivity and it denotes its power reduction per unit of added delay. Second part of the expression shows amount of slack of the cell transition. The

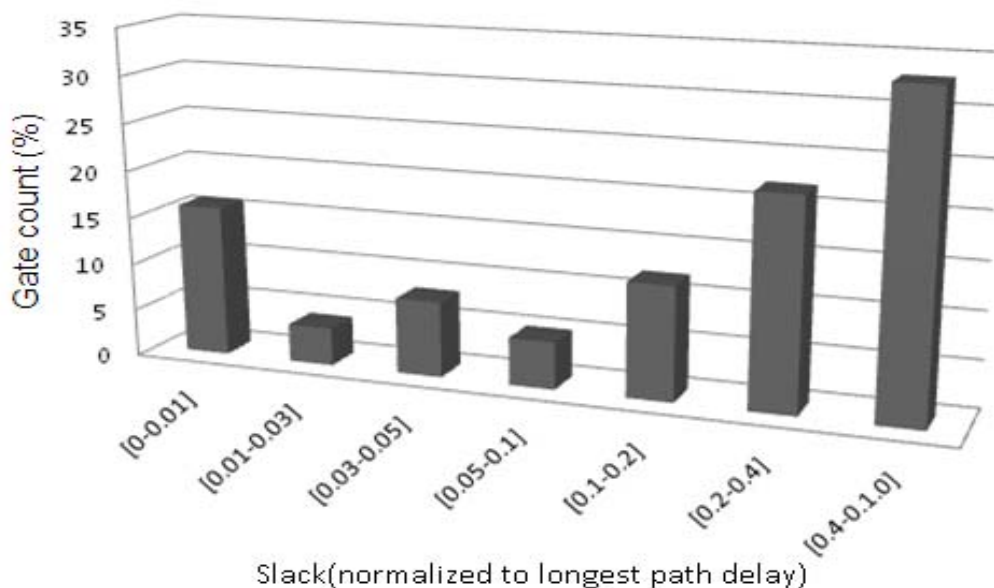


Fig. 1. Distribution of gates with different slack for OpenSPARC T1 processor

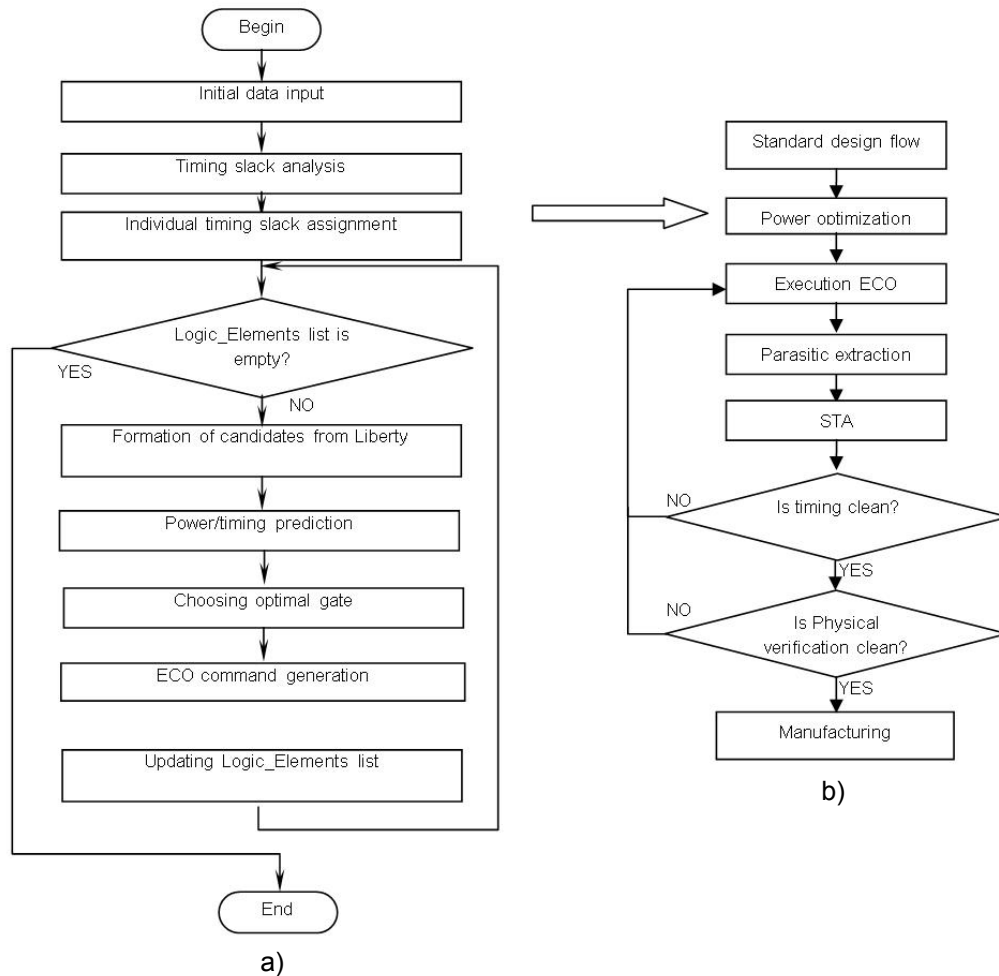


Fig. 2. Optimization algorithm (a) and its place in a design flow (b)

less slack is available for a gate transition the closer is  $t_{\max}/t(i)-1$  to 0. And hence efficiency criteria are lower. Delay distribution between gates is carried out by  $Z_i$  value of the gate starting with the critical path. Extra delay of a gate is defined as follows:

$$d(i) = Z(i) \frac{D_{path}(j)}{\sum_k Z(k)}, \quad (2)$$

where  $D_{path}(j)$  is the slack of the  $j$ -th path,  $k$  is the number of a gate on this path. The next expression shows power reduction of a gate  $i$  with delay change of  $d(i)$ .

$$\Delta P(i) = \frac{Z(i)d(i)}{t_{\max} - 1} \cdot t(i). \quad (3)$$

Thus the mathematical formulation of the problem is as follows:

$$\sum_{i=1}^n \Delta P_i \rightarrow \max \quad (4)$$

$$\begin{cases} d(i)_{effect} \leq d(i) \\ t(i) \leq t_{\max} \end{cases}$$

where  $d(i)_{effect}$  is the delay increase of the replaced gate. After the slacks are assigned to the gates algorithm starts the optimization process with the gate from the Logic\_Elements list having the highest slack. The next step is to find a logically equivalent gate from the .lib file that has lower driving force. Selecting an element with a lower driving force will assure intactness of adjacent cells in terms of placement and routing. Candidate cells can also be from high threshold voltage (hvt) element set. Taking into account that the interconnections are not modified, only input and output capacitances of updated gates are changed for each candidate delay, power or transitions can be obtained from the .lib file. From the candidate cell list a gate is chosen which satisfies to (4) condition. A corresponding ECO command is generated and stored in a list to be used by P&R tool, the gate then is removed from the Logic\_Elements list. The process is continued for the rest of the candidate cells until the list is empty.

After the optimization process is completed the list of ECO commands is fed to P&R tool to make the required changes in the physical design. The

design with updated slacks may undergo optimizations anew if there still remain large slacks.

**2. Experimental results**

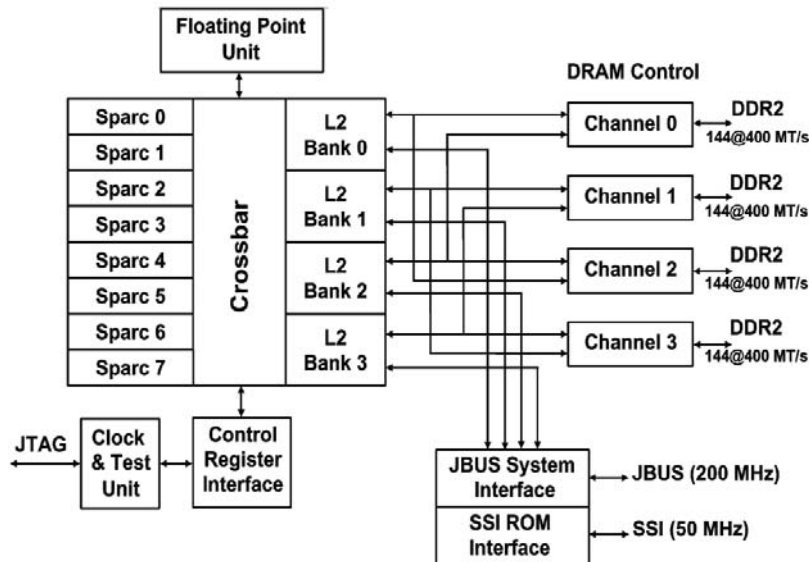
To evaluate the algorithm efficiency an OpenSPARC T1 multi-core processor was designed in SAED90nm educational library (90nm bulk CMOS). The library was scaled into 45nm process. The scaling was done through the simulations for many corners, and the parameters difference between the 90nm and 45nm transistors was calculated. After this the calculation values were averaged and the scaling factors calculated. The logic gates in the library are designed for different driving forces and two  $V_{th}$  options (standard  $V_{th}$  and high  $V_{th}$ ). The block diagram of the eight-core OpenSPARC T1 architecture is shown in Fig. 3.

The optimization was tested for two cases: with only gate sizing allowed and with both sizing and dual- $V_{th}$  option available. For the first case the optimization has shown 7,8% of static and about

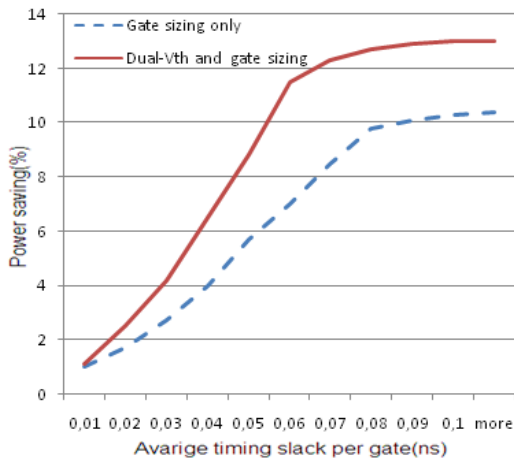
10,3% dynamic power decrease. During the optimization process approximately 27% of logic gates were replaced with their smaller driving force analogs. Total optimization time on a quad-core 3Ghz 8G RAM machine takes less than 5 hours. Optimization with multi-threshold gates available showed 18,8% static and 11,2% dynamic power decrease. The optimization process lasts about 9 hours.

As it was mentioned the efficiency of the algorithm is dependent on available timing slacks in the design. In Fig. 4 dependence of optimized power on the average logic gate slack is presented for two optimization options. It can be seen that the multi-threshold optimization shows better results than size-only optimization. The power saving can be increased by increase of optimization algorithm reiteration.

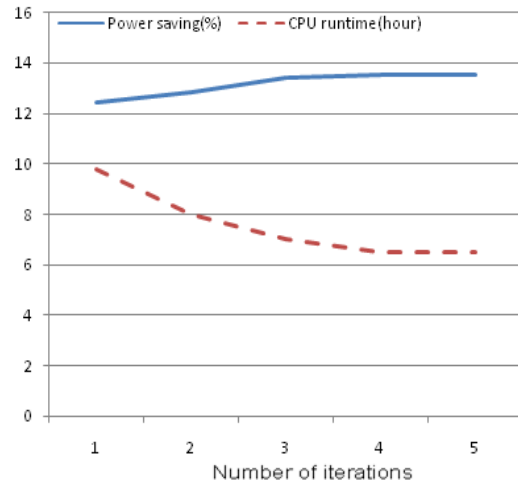
The Fig.5 shows dependence of power saving in percents vs. the number of iterations. Optimization time as well as effectiveness of each iteration decreases. This can be seen from Fig.5. Usually optimal number of iterations is equal to 3.



**Fig.3 OpenSPARC T1 Block Diagram**



**Fig. 4. Optimized power dependence of the average logic gate slack**



**Fig. 5. Power saving and algorithm runtime dependence of the number of iterations**

The method has also been tested on ISCAS85 benchmark circuits mapped into the SAED90 library with Design Compiler tool [6], P&R is done with IC Compiler [7]. The results are presented in the table 1.

The algorithm shows significantly better runtime than that proposed in [8], and less power optimization for about 2%.

**Table 1: Efficient power reduction for the benchmark set using the proposed method.**

Circuit	Number of gates	Initial power (uW)	Optim. with sizing (uW)	Optim. with Vth/sizing (uW)	CPU time (s)
C432	289	213	192	183	1,2
C880	340	255	232	221	1,4
C499	539	432	402	389	2,2
C1355	579	445	418	404	2,3
C1908	722	583	548	525	2,9
C2670	1082	678	644	583	4,2
C3540	1208	1008	877	857	4,8
C5315	2440	1817	1563	1526	9,7
C6288	2310	3099	2727	2665	9,2
C7552	3115	2207	1876	1854	12,8

## Conclusion

A novel algorithm has been suggested for VLSI IC optimization. Tested on an eight core OpenSPARC processor, it has demonstrated approximately 19% static power and 11% dynamic power reduction. Because of using individual gate slack distribution mechanism performance of the proposed algorithm has been significantly improved, reaching 5 hours for sizing-only optimization and 9 hours for multi-threshold optimization. The method was also tested for ISCAS85 benchmark circuits and showed higher performance and comparable power reduction compared to similar algorithms.

## References

1. Michael K., David F. Low Power Methodology Manual for System-on-Chip Design - Springer, 2007 - 300p.
2. Swarup B., Saibal M. Low-Power Variation-Tolerant Design in Nanometer Silicon - Springer, 2011- 456p
3. Kyu-won C., Abhijit C. UDSM (Ultra-Deep Sub-Micron)-Aware Post-Layout Power Optimization for Ultra Low-Power CMOS VLSI // ISLPED'03, Aug. 2003-pp. 72-77
4. Ketkar M., Sapatnekar S. Stand-by power optimization via transistor sizing and dual threshold voltage assignment // Proc. Int. Conf. Comput.-Aided Design, 2002- pp. 375–378.
5. Ken K. Power Supply Noise Reduction // www.designers-guide.org, 2004 - 12p
6. Design Compiler User Guide, Synopsys Inc., 2010- 399p.
7. IC Compiler User Guide, Synopsys Inc., 2010- 100p.
8. Yifang L., Jiang H. A New Algorithm for Simultaneous Gate Sizing and Threshold Voltage Assignment // IEEE transactions on computer-aided design of integrated circuits and systems, vol. 29, no. 2, Feb. 2010 – pp. 223-234

<sup>1</sup> Synopsys CJSC, Yerevan, Armenia

<sup>2</sup> State Engineering University of Armenia, Yerevan, Armenia