Switching-Activity Driven Gate Sizing and Vth Assignment for Low Power Design

Yu-Hui Huang
Department of Computer Science
National Tsing Hua University
HsinChu, Taiwan 300
yhhuang@nthucad.cs.nthu.edu.tw

Po-Yuan Chen
Department of Computer Science
National Tsing Hua University
HsinChu, Taiwan 300
pychen@cs.nthu.edu.tw

TingTing Hwang
Department of Computer Science
National Tsing Hua University
HsinChu, Taiwan 300
tingting@cs.nthu.edu.tw

Abstract

Power consumption has gained much saliency in circuit design recently. One design problem is modelled as "Under a timing constraint, to minimize power as much as possible". Previous research regarding this problem focused on either minimizing dynamic power by gate sizing, or reducing leakage power by dual threshold voltage assignment on non-critical path. However, given a timing constraint, an optimization algorithm must be able to utilize gate sizing and threshold-voltage assignment interchangeably, in order to minimize total power consumption including dynamic and leakage power in active mode and leakage power in idle mode. We find that switching-activity of a gate plays an important role in making decision as to choosing gate sizing or threshold-voltage assignment for performance improvement. For high switching-activity gates, threshold-voltage assignment should be used while for low switching-activity gates, gate sizing should be utilized. We develop an algorithm to perform gate sizing and threshold-voltage assignment simultaneously taking switching activity into consideration. The results show that under the same timing constraint, our circuits have 16.26%, and 18.53%, improvement of total power as compared to the original circuits for the cases where the percentage of active time are 100%, and 50%, respectively.

1 Introduction

Power consumption has gained much saliency in circuit design recently. In general, power sources are classified as dynamic and leakage power. Literature on low power design is abundant [3, 4, 5, 6] in which various techniques have been proposed to minimize either dynamic power or leakage power.

One design problem is modelled as "Under a timing constraint, to minimize power as much as possible". To solve this problem, one direction of research is focused on minimizing dynamic power. The minimization is based on the dynamic power consumption model where to maintain the performance, sizes of gates on critical paths remain unchanged while sizes of gates on non-critical paths are sized down to utilize their timing slack. By doing so, the gate capacitance is reduced, so is the dynamic power consumption.

The other direction is focused on reducing leakage power consumption by dual threshold assignment. It uses low-Vth transistors for gates on critical-path and high-Vth transistors on non-critical-path. This strategy is used in [3, 4, 5, 6, 8] and has shown that it has significant saving of leakage power and dose not mitigate the performance of circuit.

Gate sizing-up will increase gate capacitance hence large dynamic power and minor leakage. On the other hand, low threshold-voltage assignment will cause large leakage power increase. However, given a timing constraint, in order to minimize total power consumption including dynamic and leakage power in active mode and leakage power in idle mode, an optimization algorithm must be able to utilize sizing and threshold-voltage assignment interchangeably.

We find that switching activity of a gate plays an important role in making decision as to choosing gate sizing or threshold-voltage assignment. For high switching activity gates, re-assigning Vth should be used while for low switching activity gates, gate sizing should be utilized. Moreover, gates on both critical path and non-critical path should be taken into consideration. By utilizing these two techniques on both critical path and non-critical path, we can minimize the total power of circuit.

The rest of the paper is organized as follows. Section 2 will present our motivation. An algorithm taking switching activity into consideration for sizing and threshold-voltage assignment will be shown in Section 3. Benchmark results are presented in Section 4. Finally, conclusion remarks are drawn in Section 5.

2 Motivation

To enhance the performance of a circuit, we can size-up gate or change the Vth of gate from high to low. The former method will increase dynamic power and small leakage power while the latter method increase leakage power.

We observe that the switching activity of a gate will determine whether to use gate sizing or threshold-voltage assignment technique, in order to improve the timing of a circuit and to achieve lower power. If a gate has high switching activity, low Vth-assignment technique should be used while low switching activity, gate sizing-up is a better selection.

To understand if our observation is correct, we perform an experiment on two invertors, A and B. Invertor A has higher Vth and larger size than invertor B. The size and Vth are tuned so that the two invertors have the same

0-7803-9451-8/06/$20.00

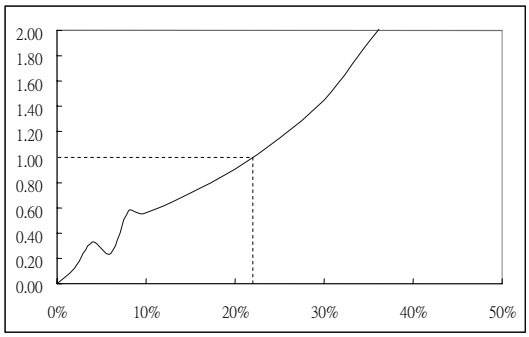

Figure 1: Relation between switching activity and $\frac{dyn(A)-dyn(B)}{lea(B)-lea(A)}$

Table 1: Switching activity(α) distribution of cells

α	Ratio(%)						
	TOP	MAC	AVG	GCC	RSA	AES	Avg.
$\alpha \leq 22\%$	71.0	48.9	70.9	55.3	84.5	60.8	65.3
$\alpha > 22\%$	29.0	51.1	29.1	44.7	15.5	39.2	34.7

delay under the same driving capability and output loading. The experiment is performed using *45nm* model from *PTM*[9] by *Hspice*. The result is shown in Figure 1. In this figure, X axis represents the switching activity, and Y axis is the result calculated by $\frac{dyn(A)-dyn(B)}{lea(B)-lea(A)}$, where $dyn(A)$ and $dyn(B)$ represent the dynamic power of A and B, and $lea(A)$ and $lea(B)$ represent the leakage power of A and B. Notice that invertor B is a better choice when the value of Y is larger than 1.

This figure shows that when the switching activity is lower than 22%, using invertor A (high Vth and larger cell) will produce less amount of total power. However, when the switching activity is more than 22%, invertor B (low Vth and smaller cell) will produce less power consumption. Based on the result in Figure 1, next, we would like to know if the switching activity of gates are indeed distributed in these two groups, with the switching activity less than and more than 22%. Table 1 shows the switching-activity distribution of gates in a set of circuits. In this table, we do see that on the average, 65% of the total gates are with switching activity between 0% to 22% and 34% with switching activity greater than 22%.

Moreover, previous work regarding minimizing power focused only on non-critical path. Here, we will minimize power on both critical and non-critical paths. On critical path, we will assign Vth to high and up-size gates which has small switching activity. On non-critical path, slack will be used to down-size gate or assign Vth to high.

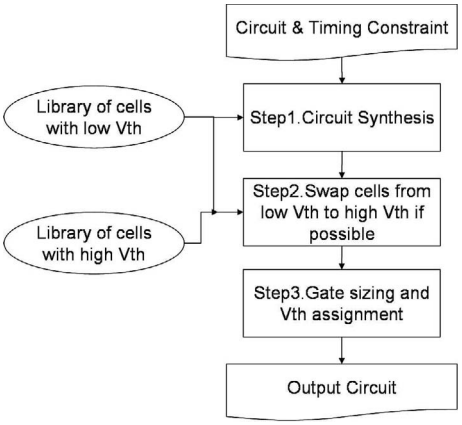

Figure 2: Design flow of the algorithm

3 Design Flow

Based on our motivation in Section 2, we propose a design flow to determine how to perform gate sizing and threshold voltage assignment. In Section 3.1, we first define the problem and show the design flow. In Section 3.2, the detailed algorithm will be presented.

3.1 Problem Definition and Design Flow

The problem can be defined as follows. Given a circuit, timing requirement, time profile of active and idle modes (i.e., the percentage of the total time that the circuit is in active mode and in idle mode) and cell library, to minimize the power consumption (including leakage in idle mode and dynamic and leakage in active mode) by gate sizing and threshold-voltage assignment.

To solve this problem, a design flow shown in Figure 2 is proposed. First, with the timing constraint, the circuit is synthesized using only cells with low Vth by synthesis tools. The reason to use only cell with low Vth is that a circuit synthesized using cells with the best timing (low threshold voltage) but minimal size allows gates to be sized up in the later optimization steps. Then, in the second step, all gates with low Vth are swapped to their corresponding high Vth cells. After this step, the timing constraint is no longer satisfied but the leakage is maximally reduced (the gate with low Vth has larger delay and less leakage). Finally, the last step is to perform gate sizing and threshold voltage re-assignment to restore the original timing performance. For nodes on critical path , our decision to choose gate for up-sizing or low threshold voltage assignment, will take the minimal increase in power consumption and area into consideration. For nodes on non-critical path, the algorithm utilize the slack of nodes to save more power consumption.

The first and the second steps are well understood. The detailed algorithm of the third step, **gate sizing and Vth assignment**, will be explained in the next subsection.

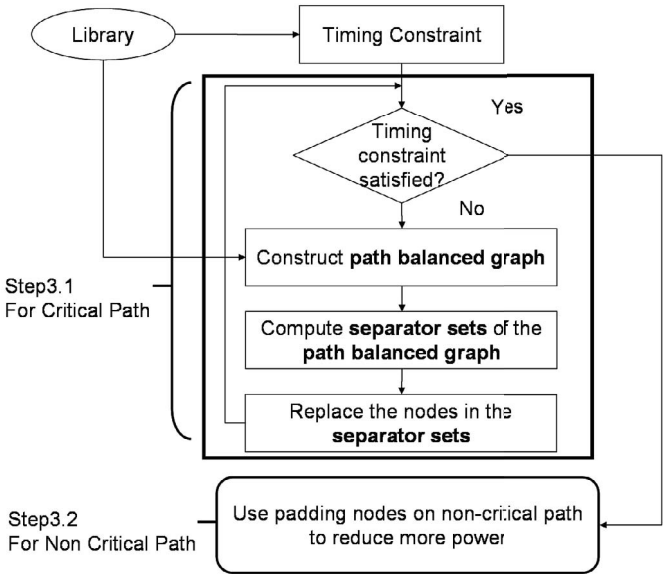

Figure 3: Step of **gate sizing and Vth assignment**

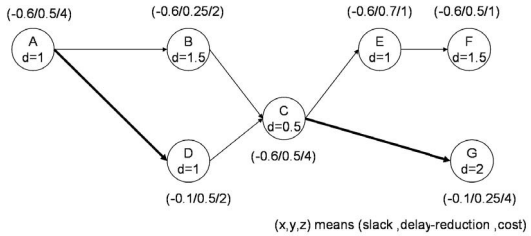

Figure 4: A circuit graph

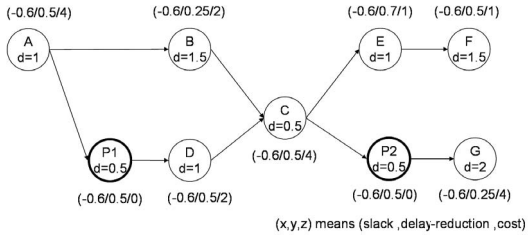

Figure 5: A *path balanced graph*

3.2 Gate Sizing and Threshold Voltage Assignment

Our third step, **gate sizing and Vth assignment**, is conducted in two phases. In the first phase, gate sizing or Vth voltage re-assigning is performed on nodes on critical path and in the second phase, on nodes on non-critical path. First, timing analysis on the circuit is performed. The arrival time, the required time and the slack of each gate are computed. Then, based on this timing information, a path balanced graph *G = (V, E)* for the circuit is constructed. Next, separator sets of the graph are computed. The nodes in the separator sets are candidates for sizing or threshold voltage re-assignment. This step continues until the timing constraint is satisfied. Once the timing constraint is met, we continue to minimize the power consumption of circuit in the second phase by utilizing the remaining timing slack on non-critical paths.

The algorithm is depicted in Figure 3. The details are described in the following.

3.2.1 Optimization on Critical Path

A circuit can be viewed as a directed graph *G = (V, E)*, as shown in Figure 4, where *x, y, z* in *(x, y, z)* denote *slack*, *delay-reduction*, and *cost*, of the nodes, respectively (*delay reduction*, and *cost* will be defined later). After timing analysis, the arrival, required times and slack of nodes are computed. Based on this timing information, to improve the timing performance, a set of nodes can be selected to speed up. Since there is usually more than one critical path, the selection step requires a lot of attention. The objective is usually to select a set of nodes with minimum cost.

One way to select nodes which guarantee the circuit timing improvement is to select a separator set. However, simply selecting a separator set will not produce low cost result because slack on short path may not be fully utilized. Instead, we will select a separator set based on a *path balanced circuit graph* [7] in which slack of short paths can be fully exploited.

A *path balanced graph* is defined as follows [7]. First, for all edges *e*, *ds(e)* is computed. *ds(e)* is defined as the slack difference of nodes at the two ends of an edge *e*. It is computed as

$$ds(e) = slack(tail\_node(e)) - slack(head\_node(e)) \quad (1)$$

If *ds(e) > 0*, it means that input is from a short path. A *padding node* is inserted at *e* whose delay is *ds(e)* and the cost of this node is *0*. By doing so, slack of all nodes become equal. To select a separator set with least cost, padding nodes are more likely to be selected. When the padding nodes are indeed selected in a separator for timing improvement, the cost of replacing *padding nodes* is *0*. Figure 5 shows a *path balanced graph*, $G_{balanced} = (V, E)$ with padding nodes constructed from circuit graph, *G = (V, E)* of Figure 4. In this figure, padding node *P1* is added between node *A* and node *D* because the slack difference of the edge $A \rightarrow D$ is *0.5* (*slack(node D) - slack(node A)*). Similarly padding node *P2* is added because the slack difference of the edge $C \rightarrow G$ is *0.5* (*slack(node G) - slack(node C)*).

Once the *path balanced graph* is constructed, we need to set the cost of a node. The cost is to be defined so that the less the cost of a node, the more likely the node is to be replaced for performance improvement. Before we present how to set the cost of a node, we need first to decide what the next candidate-change is for a node, in order to solve the timing violation problem. The objective of our algorithm is to select nodes with minimal power

and area increase for replacement. There are two ways to solve the timing violation: up-sizing gate or changing the gate with high Vth to its corresponding cell with low Vth. However, either way will increase the total power (dynamic and leakage) in which up-sizing gate increases the load capacitance of fan-ins and hence dynamic power while assignment of low threshold voltage increases leakage power. To make a choice between these two options, based on the observation in Section 2, we should take the switching activity of gates into consideration. For gates with fan-ins of low switching activity, up-sizing should be selected because the increase in dynamic power of fan-ins may be very small. On the other hand, for gates with fan-ins of high switching activity, assignment of low threshold voltage should be considered.

Based on this observation, we define a power penalty function, *penalty(g)* which is the penalty for the gate g when up-sizing or low Vth assignment is selected to replace the current gate g. It is calculated as

$$penalty(g) = \alpha \cdot p\_penalty(g) + \beta \cdot a\_penalty(g) \quad (2)$$

In this equation, $p\_penalty$ and $a\_penalty$ are the power and area increase overhead, respectively, and α and β are parameters to control the weights of power and area penalty. The $p\_penalty(g)$ is further defined as

$$p\_penalty(g) = per \cdot (\sum_{j \in fanin(g)} E(j) \cdot C_{inc}(g) V^2 + leak_{inc}(g)) + (1 - per) \cdot leak_{inc}(g) \quad (3)$$

where *per* is the percentage of the total time that the circuit is in the active mode, *E (j)* is the switching activity of signal *j*, $C_{inc}(g)$ is the increased capacitance, V is the supply voltage, $leak_{inc}(g)$ is the increased leakage. The first term represents the power increase when the circuit runs in active mode and the second term the power increase when the circuit is in idle.

We compute the *penalty(g)* for the gate g for both up-sizing and low Vth assignment options. The option that has less penalty is selected as a candidate for replacement of the current gate g. Then, it is used to compute the *cost* of nodes in the *path-balanced graph*, $G_{balanced} = (V, E)$. Moreover, the delay reduction of the selected replacement is modelled as *delay-reduction* in the *path-balanced graph* .

Now, we show how to compute the *cost* of the *path-balanced graph* . The cost of a *padding node* is set to *0* and all other nodes g are computed as,

$$cost(g) = \gamma \cdot penalty(g) + \delta \cdot delay\_reduction(g) \quad (4)$$

where γ and δ are control parameters. Once the cost of the *path balanced graph* is computed, we will find a separator set of the graph. The nodes in the separator set are selected for replacement. Note that the *delay improvement* of this separator set, which is defined as the delay improvement of the circuit after the nodes of the separator set are replaced, is the minimum *delay-reduction* among the nodes in the separator set.

The next iteration will start with the timing analysis. If the timing constraint is not satisfied, the procedure continues.

Take the example shown in Figure 6 to demonstrate our selection algorithm. Figure 6(a) shows a separator

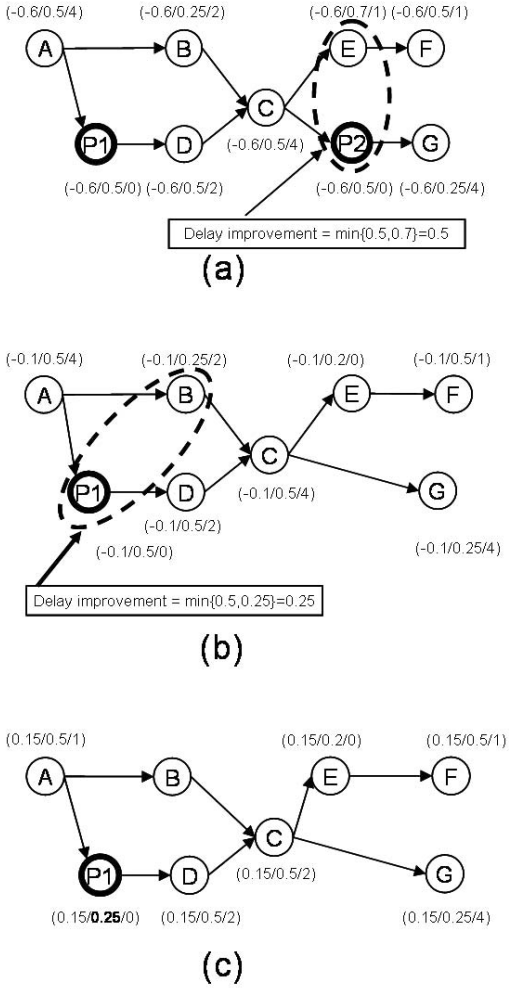

Figure 6: An example of running step 3.1
(a) iteration one (b) iteration two (c)a final *path balanced graph* after replacement

set of *path balanced graph* of the original graph shown in Figure 5. {*E,P2*} is selected to be the separator set with the minimal cost. The *delay improvement* of this set is *0.5*. The *delay-reduction* of node *E* and *P2* are decreased by *0.5* as shown in Figure 6(b). Since the *delay-reduction* of node *P2* equals to *0*, it is removed from the graph in Figure 6(b). In the next iteration, the separator {*B,P1*} is selected as shown in Figure 6(b) and it results in *0.25* delay reduction. We continue finding separator sets and updating *path balanced graph* until timing constraint is met. The final *path balanced graph* is shown in Figure 6(c).

3.2.2 Optimization on Non-Critical Path

After sizing and Vth re-assignment are performed on critical path, the timing constraint of circuit is met now. The objective of the next step is to utilize the remaining timing slack on *padding nodes* to reduce more power consumption. There are two ways to save the power consumption of a gate: down-sizing gate or changing the gate

to high threshold voltage.

Node A can be delayed ϵ time if every path going out from A has ϵ slack. Recall that adding a *padding node* between node A and node B means that there is slack on the edge $A \rightarrow B$. In other words, if and only if all the *fan-out nodes* of node A are *padding nodes* with slack at least ϵ, node A can be delayed ϵ without affecting the timing of circuit. The maximum ϵ time of node A is computed as the minimum *delay reduction* of its fan-out *padding nodes*.

Therefore, for a node *Ni* in the graph, its available slack is computed by checking if the fan-outs of the output are all padding nodes. If they are, the available slack of *Ni* is computed as the minimum *delay-reduction* of *Nj*, where *Nj* is the fan-out node of *Ni*. Otherwise, the available slack of *Ni* is *0*.

After we compute the available slack of each node, the next step is to utilize the available slack to reduce power consumption. First, we compute the delay penalty which will be caused by down-sizing nodes or re-assigning Vth to high. The *delay-penalty* of *Ni* is computed as $Delay(new\_Ni) - Delay(Ni)$. *Delay(Ni)* is the delay of *Ni* with the current size and Vth, and *Delay(new_Ni)* is the delay of *Ni* after *Ni* is down-sized or re-assigned Vth to high.

If the delay penalty of both options are less than the available slack, we have a choice to select either down-sizing or re-assigning Vth to high. The choice will be based on the power saving of these two options. From our observation presented in Section 2, we know that if the switching activity is high, smaller gate with lower Vth should be selected while if the switching activity is low, high Vth with larger size gate is more power efficient. Therefore, we define a cost function, *p_saving(g)* for a gate g, to determine which option is better. For both down-sizing and re-assigning Vth, we compute

$$p\_saving(g) = p\_penalty(g) - p\_penalty(new\_g)$$

where *new_g* is a new implementation of gate g by either down sizing or high Vth voltage assignment and *p_penalty* is the cost function defined in equation (3).

The implementation which has more *p_saving* is selected for replacement. The procedure repeats until no more slack can be used or no more cell can be replaced.

Take the circuit in Figure 7 as an example. Let us consider nodes A and E. For node A, its fan-outs, *P1* and *P2*, are both *padding nodes*. Since the minimum *delay reduction* of *P1* and *P2* is *0.4*, the available slack of node A is *0.4*. Suppose the delay penalty of node A be *0.3*, which is less than the available slack *0.4*. The size of node A can be reduced. On the contrary, for node E, it has two fan-outs but only one of them is *padding node*. Hence, the available slack of node E is *0*, hence node E cannot be delayed without affecting the timing of circuit.

4 Experimental Results

Our power reduction algorithm is implemented in C language. The experimental process proceeds as follows. First, circuits are synthesized with low Vth library by *DesignCompiler*. Next, we obtain the critical paths of each synthesized circuit using *PrimeTime*. This critical timing plus ε will be used as timing constraint, where ε is set to be the timing-variance tolerance of the optimization process. In this experimental, ε is set to 2% of the critical timing

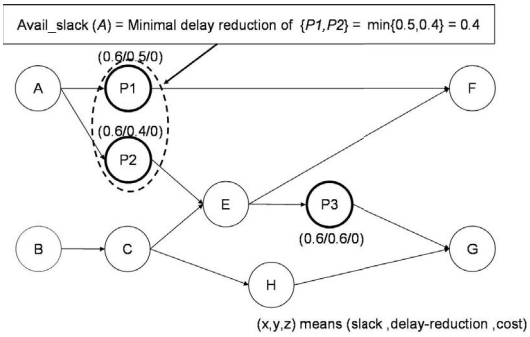

Figure 7: A updated *path balanced graph*

Table 2: Circuit descriptions

Cir.	CN	Characteristics
TOP	463	An Alarm Clock
MAC	2425	Multiplier and Accumulator
AVG	6361	Average Number Calculator
GCC	8204	Gravity Center Calculator
RSA	14815	Asymmetric Crypto-Processor
AES	16824	Advanced Encryption Core

of the synthesized circuit. Then, we change Vth from low to high and form a new circuit. Next, we use *PrimeTime* again to get the output capacitance of cells. Besides, we randomly generate some input patterns which are stored in *VCD* file. By reading this *VCD* file, *PrimePower* is used to report the toggle rate of every cell.

With the information mentioned above, we execute our program and output a *Verilog* file using cells of high Vth and low Vth cells from *TSMC 0.13*μm library. Finally, *PrimeTime* and *PrimePower* are used to report timing and total power consumption of the circuit, respectively.

Six circuits are used to examine the effectiveness of our algorithm. Table 2 shows the characteristic of our benchmark set. The columns labelled **CN**, and **Characteristics** are the cell number of the design, and the characteristics of the design, respectively.

The following tables show the power consumption and reduction after performing our algorithm for different cases where the percentages of active time and idle time are different. Let the percentage of active time and idle time be α. α is computed by $\alpha = \frac{active\_time}{active\_time+idle\_time} * 100\%$

Table 3, and Table 4 are the results for the cases where α is 100%, and 50%, respectively. The columns labelled P_O, P_A and P_B are the original power consumption, power consumption after executing **Steps 2 + 3.1** and power consumption after executing **Step 3.2**, respectively.

The original power is the result of the circuit with low Vth cells mapped by **DesignCompiler**. The columns labelled **A (Steps 2 + 3.1)** are the results that we change the Vth of cell from low to high and repair the timing violation problem on critical path. The columns labelled **B**

Table 3: The percentage of active time (α) is 100%

Cir.	P_O (mW)	A (Steps 2+3.1) P_A(mW)	Red_A	B (Step3.2) P_B(mW)	Red_B	Red_{total}
TOP	0.413	0.383	7.10%	0.363	4.85%	11.95%
MAC	0.970	0.843	13.06%	0.790	5.50%	18.56%
AVG	1.750	1.70	2.86%	1.65	3.07%	5.75%
GCC	0.806	0.764	5.19%	0.753	1.29%	6.48%
RSA	3.490	2.97	14.82%	2.12	24.38%	39.20%
AES	15.90	14.3	10.18%	13.4	5.42%	15.60%
Avg.	-	-	8.84%	-	7.42 %	16.26%

Table 4: The percentage of active time (α) is 50%

Cir.	P_O (mW)	A (Steps 2+3.1) P_A(mW)	Red_A	B (Step3.2) P_B(mW)	Red_B	Red_{total}
TOP	0.209	0.191	8.39%	0.179	5.85%	14.24%
MAC	0.504	0.424	15.89%	0.397	5.19%	21.09%
AVG	0.916	0.862	5.87%	0.835	2.92%	8.79%
GCC	0.451	0.418	7.39%	0.412	1.30%	8.69%
RSA	1.84	1.50	18.16%	1.08	23.24%	41.40%
AES	8.07	7.16	11.19%	6.70	5.79%	16.99%
Avg.	-	-	11.15%	-	7.38%	18.53%

(Step 3.2) are the results that we utilize the remaining slack on non-critical path to minimize the power consumption of the circuit.

The columns labelled Red_A are the power reduction rate computed as $Red_A = \frac{(P_O - P_A)}{P_O} * 100\%$ and the columns labelled Red_B and Red_{total} are similarly defined.

On the average, the power reduction rates of **A (Steps 2 + 3.1)**, **B (Step 3.2)**, and **total** of Table 3 are 8.84%, 7.42%, and 16.26%, respectively. Similar results are also produced in Table 4 where reduction rates are 11.15%, 7.38%, and 18.53%.

Table 5 shows the circuit timing information. The columns labelled **N_T** represent the timing of the new circuit. The columns labelled **TP** represent the timing penalty after performing our algorithm, which is calculated as $TP = 1 - \frac{(N\_T)}{Original\_T}$.

Table 5: Timing comparisons

Cir.	Original T	α is 100% N_T	TP	α is 50% N_T	TP	α is 10% N_T	TP
TOP	1.43	1.37	-4.2%	1.39	-2.8%	1.40	-2.1%
MAC	3.30	3.33	0.8%	3.33	0.8%	3.33	0.8%
AVG	23.78	23.13	-2.7%	23.46	-1.3%	23.54	-1.0%
GCC	26.30	26.65	1.3%	26.73	1.6%	26.34	0.2%
RSA	10.00	10.08	0.8%	10.03	0.3%	10.10	1.0%
AES	2.29	2.21	-3.5%	2.27	-0.9%	2.27	-0.9%

If **TP** is less than *0*, it means that timing becomes better than the original circuit. From the table, we can see that our algorithm do not cause serious timing violation problem. The largest timing penalty is 1.6% (circuit GCC when $\alpha = 50\%$), which is within our timing-variance tolerance, 2%.

From the table above, we found that our algorithm can effectively decrease the power consumption with very small timing penalty.

5 Conclusions

We have studied the problem that under a timing constraint, to minimize total power consumption. We have found that switching activity of a gate plays an important role in making decision as to choosing gate sizing or threshold assignment to improve timing performance. For high switching activity gates, Vth assignment should be used while for low switching activity gates, gate sizing should be utilized. The results showed that under the timing constraint, our circuits have 16.26%, and 18.53% improvement as compared the original circuits for cases where the percentage of active time are 100%, and 50%, respectively.

References

[1] Shekhar Borkar, "Low Power Challenges for the Decade", *Proceedings of ASP-DAC*, pp. 293-296, 2001.

[2] A. P. Chandrakasan and R. W. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits", *Proceddings of the IEEE*,pp. 498-523, 1995.

[3] L. Wei, Z. Chen, M. Johnson and K. Roy, "Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits", pp. 489-494, *Proceedings of the 35th DAC*, 1998.

[4] Vijay Sundararajan and Keshab K. Parhi, "Low Power Synthesis of Dual Threshold Voltage CMOS VLSI Circuits", *Proceedings of 1999 ISLPED*, pp. 139- 144, 1999.

[5] N. Tripathi, A. Bhosle, D. Samanta and A. Pal, "Optimal Assignment of High Threshold Voltage for Synthesizing Dual Threshold CMOS Circuits", *The 14th International Conference on VLSI Design*, pp. 227-232, 2001.

[6] Yen-Te, and TingTing Hwang, "Low Power Design Using Dual Threshold Voltages", *Proceedings of ASP-DAC 2004*, pp. 205-208, Japan, Jan. 2004.

[7] Yutaka Tamiya, "Performance Optimization Using Separator Sets", *Proceedings of ICCAD 1999*, pp. 191-194, 1999.

[8] D. Nguyen, A. Davare, M. Orshansky, D. Chinnery, B. Thompson, and K. Keutzer, "Minimization of Dynamic and Static Power Through Joint Assignment of Threshold Voltages and Sizing Optimization", *Proceedings of ISLPED* , pp. 156-163, 2003.

[9] Predictive Technology Model "http://www-device.eecs.berkeley.edu/p̃tm/".