Leakage power reduction through dual V_{th} assignment considering threshold voltage variation

Xukai Shen, Yu Wang*, Rong Luo, Huazhong Yang
NICS, EE Dept., Tsinghua University, Beijing 100084, P. R. China
*Email: yu-wang@mail.tsinghua.edu.cn

Abstract[†]

In today's sub-100nm CMOS technologies, leakage current has become an important part of the total power consumption, affecting both yields and lifetime of digital circuits. Dual V_{th} assignment, which is proven to be an effective method of reducing leakage power in the past, is also effective in today's technologies with certain modifications. In the paper, based on a statistical timing analysis (SSTA) framework we presented a dual V_{th} assignment method which can effectively reduce the leakage power even in the presence of large V_{th} variation. Besides, we use a statistical DAG pruning method which takes correlation between gates into account to speed up the dual V_{th} assignment algorithm. Experimental results show that statistical dual V_{th} assignment can reduce on average 40% more leakage current compared with conventional static method without affecting the performance constraints. Our DAG pruning method can reduce on average 30% gates in the circuit and save up to 50% of the total run time.

1. Introduction

As IC technologies scales down, leakage power has become a dominant contributor to the total power consumption [1]. With the fast development of dual V_{th} process, dual or multiple threshold voltage assignment techniques are becoming more popular due to their ability of reducing leakage power while maintaining the overall high performance, which can be achieved by setting different transistors in the circuit with different threshold voltages. The delay between high and low V_{th} transistors is roughly twice, and the leakage current differs by almost thirty times [2]; while dual V_{dd} may only lead to 2 or 3 times leakage difference with many additional level converters. Therefore, dual V_{th} assignment is proven to be a potent way of leakage power reduction.

Deterministic dual V_{th} assignment approaches were first proposed: some heuristic algorithms for dual V_{th} assignment [3]–[5] provided local optimal solutions; while in [6], [7], global optimizations were derived from linear programming formulation. As technology scaling, process variation can severely affect both power and timing yield. Hence, some researchers devoted themselves to probabilistic dual V_{th} assignment approaches [8]–[10] which statistically optimized the leakage power and circuit performance. However, little work has been done to improve the speed of circuit optimization based on statistical timing analysis.

This paper presents a novel SSTA-based dual V_{th} assignment considering V_{th} variation and distinguishes itself in the following aspects:

(1) V_{th} correlation is taken into account in the delay model of the statistical timing analysis framework. An SSTA-based dual V_{th} assignment technique which can effectively reduce the leakage power in the presence of large V_{th} variation is proposed.

(2) A statistical DAG pruning method which takes correlation between gates into account is used to reduce the gate number in the circuits. Therefore the problem size is reduced and our dual V_{th} assignment algorithm is sped up greatly.

Experimental results show that statistical dual V_{th} assignment can reduce on average 40% more leakage current compared with conventional static method. Our DAG pruning method can reduce 30% gates in the circuit on average. The improvements suggest that statistical optimization is vital for high performance designs.

This paper is organized as follows. Section 2 presents the V_{th} variation model considering correlation, and the gate delay/leakage models as the preliminaries for our SSTA based dual V_{th} assignment. Section 3 first compares the problem definitions of dual V_{th} assignment under different timing constraints; and then proposes our SSTA-based DAG pruning method. The implementation and experimental results are given and discussed in Section 4. Section 5 concludes this paper.

2. Preliminaries

Given a digital circuit, timing analysis tools translate it into a directed acyclic graph (DAG) $G=(V,E)$, where each node $v_i \in V$ denotes a primary input, primary output or internal gate, each edge $e_i=(v_m, v_n) \in E$ denotes the interconnect between gate v_m and v_n. In addition, a source/sink node is conceptually added before/after the primary inputs/outputs so that the graph can be analyzed as a single-input single-output network.

2.1 V_{th} variation model considering correlation

In gate level circuit optimization, for a path from source to sink with n nodes v_1 , v_2 ,…, v_n, we assume that the logically close nodes are also spatially close. We may further assume that the threshold voltage of node v_i is correlated with node v_{i+2}, v_{i+1}, v_{i-1}, v_{i-2}, (which can be generalized to more nodes), and each correlation coefficient is:

$$\begin{aligned}&\rho(v_i, v_{i\pm1}) = \rho_1\\ &\rho(v_i, v_{i\pm2}) = \rho_2\\ &(\rho_1 > \rho_2)\end{aligned} \tag{1}$$

The threshold voltage of each node v_i is expressed as linear combinations of the mean value and random variables:

$$V_{i,th} = V_i + \alpha\Delta V_{i-1} + \beta\Delta V_i + \alpha\Delta V_{i+1} \tag{2}$$

where ΔV_0, ΔV_1,…, ΔV_{n+1} are random variables (*i.i.d* ~$N(0,1)$); V_i is the mean value of $V_{i,th}$; α, β are constants. If there are n nodes in a path from source to sink, there will be n+2 random variables: ΔV_0, ΔV_1,…, ΔV_{n+1} to illustrate the correlation between the nodes in the path. Once the mean and variance of the threshold voltage are determined, α, β can be

[†] This work was supported by grant from National Natural Science Foundation of China (No.60506010) and Basic Research Foundation of School of Information Science and Technology of Tsinghua (No SIST2017).

calculated using Equation (1) and (2):

$$\rho_1 = \frac{2\alpha\beta}{2\alpha^2+\beta^2} \quad \rho_2 = \frac{\alpha^2}{2\alpha^2+\beta^2} \tag{3}$$

Notice that the physical level information can also be used to get the real spatial and correlation information for each gate. Therefore, the method proposed in this paper can be also applied in physical level design.

2.2 Delay model

The load dependent delay of gate v is given by [11]:

$$D(v) = \frac{KC_L V_{dd}}{(V_{dd}-V_{th})^{\alpha}} \tag{4}$$

where C_L, V_{th}, α, K are the load capacitance at the gate output, the threshold voltage, the velocity saturation index and the proportionality constant respectively. Using first order approximation, the delay can be expressed as

$$D(v) = KC_L V_{dd}{}^{1-\alpha}(1+\alpha V_{th}/V_{dd}) \tag{5}$$

Since delay can be expressed as linear function of the threshold voltage as shown in equation (5), the correlation between gate delay and the threshold voltage variation can also be easily expressed.

2.3 Leakage model

For the gates with low/high V_{th}, a leakage lookup table is created by simulating all the gates in the standard cell library under all possible input patterns. Thus the leakage current $I_{leak}(v)$ can be expressed as:

$$I(v)= \Sigma\, I_l(v,IN) \times PB(v,IN) \tag{6}$$

where $I_l(v, IN)$ and $PB(v, IN)$ are the leakage current and the probability of gate v under input pattern IN.

The sub-threshold leakage current of each gate depends exponentially on V_{th}, which is:

$$I_{leak}(v) = \mu C(W/L)V_T{}^2 \exp(\frac{V_{th}}{V_T}) \tag{7}$$

where μ V_T are constants, and C is the capacitance of the depletion region under the gate area, if we assume ΔV_{th} is Gaussian random variable, Equation (7) will become：

$$I_{leak}(v) = \mu C(W/L)V_T{}^2 \exp(\frac{V_{th,mean}}{V_T})(1+\frac{\Delta V_{th}}{V_T}) \tag{8}$$

where $V_{th,mean}$ is the mean value of the threshold voltage and ΔV_{th} ~N(0,σ^2).

3． SSTA based Dual Vth Assignment

3.1 Problem definition

In STA, optimization object function is:

$$I_{leak}(G) = \sum_{v\in V}\left(I_{leak,l}(v)\times(1-H(v)) + I_{leak,h}(v)\times H(v)\right) \tag{10}$$

where $I_{leak,l}$ and $I_{leak,h}$ are the leakage current of gate with high and low threshold voltage, respectively; H(v) equals to one if V_{th} of node v is high, otherwise it equals to zero. The timing constraints for the circuits are:

$$\begin{array}{ll} t_a(m)=0 & \forall m \in PI \\ t_a(n) < D_{req} & \forall n \in PO \\ t_a(i)+D_{wc}(i) < t_a(j) & \forall i,j \in V \\ & (i,j)\in E \end{array}$$

where t_a is the arrival time, $D_{wc}(i)$ is the worst case delay of gate i.

In our SSTA framework, the objective function is almost the same as the STA-based object function:

$$I_{leak}(G) = \sum_{v\in V}\left(I_{leak,l}(v)\times(1-H(v)) + I_{leak,h}(v)\times H(v)\right) \tag{11}$$

where H(v) shares the same meaning as the one in Equation (10), $I_{leak,l}$ and $I_{leak,h}$ are random variables whose mean and standard deviation are known. The minimization of the above equation may be expressed as minimization of linear combination of mean value and standard deviation:

$$\min(I_{leak,mean} + k\sigma_{I_{leak}}) \tag{12}$$

where k is a constant. The timing constraints also change as follows:

$$\begin{array}{ll} t_a(m)=0 & \forall m \in PI \\ P(t_a(n) < D_{req}) > p & \forall n \in PO \\ P(t_a(i)+D_{wc}(i) < t_a(j)) > p & \forall i,j \in V \\ & (i,j)\in E \end{array}$$

where t_a is the arrival time (random variable); $D(i)$ is the delay (random variable) of gate i; $P(y)$ is the probability that y is true; p is a given percentage between 0 and 1 representing the timing yield requirement of the circuit. If $p \approx 1$, the above condition represents the worse case scenario for SSTA.

3.2 DAG pruning method

For path-based SSTA, the main problem is that there are too many paths in the graph, while actually only part of these paths should be inspected when timing requirement is concerned. So if the number of paths which needs to be considered can be reduced, the SSTA-based optimization will be more efficient. Therefore a DAG pruning method is used here to reduce the problem size and to speed up the algorithm.

The DAG pruning method is based on the following assumptions: **if the max delay (no matter static or statistical) of all paths (with all the gates in high V_{th} condition) associated with one specific node v_i satisfy timing requirement, then this node v_i and edges connected to it can be removed from the graph without affecting the timing analysis.**

Denote source and sink as v_{source}, v_{sink} respectively. For node v_i, delay of the paths passing through v_i can be expressed as:

$$\begin{aligned} &D(v_{source}, v_i, v_{sink}) \\ &= D(v_{source}, v_{i-1}) + D(v_i) + D(v_{i+1}, v_{sink}) \end{aligned} \tag{13}$$

If delays are not random variables, there is

$$\begin{aligned} &\max D(v_{source}, v_i, v_{sink}) \\ &= \max D(v_{source}, v_{i-1}) + D(v_i) + \max D(v_{i+1}, v_{sink}) \end{aligned} \tag{14}$$

And $maxD(v_x, v_y)$ can be represented using the arrival time from v_x to v_y which can be easily calculated in STA tool with linear scale time. However, in SSTA-based framework, the above equation holds only as long as the mean value is concerned. Yet with our V_{th} variation assumption, we can derive an upper bound for the variance of $maxD(v_{source}, v_i, v_{sink})$, which will help us to prune the DAG in a statistical manner.

First, consider a path with m (m>2) node $v_1, v_2, \ldots, v_m$, according to Section 2, the path delay can be expressed as

$$\begin{aligned} &D(v_1,\ldots,v_m) \\ &= D_{nom} + B\left(\alpha(\Delta V_0 + \Delta V_{m+1}) + (\alpha+\beta)(\Delta V_1 + \Delta V_m) + \sum_{i=2}^{m-1}(2\alpha+\beta)\Delta V_i\right) \end{aligned} \tag{15}$$

where D_{nom} is the mean value of the path delay; and ΔV_0, $\Delta V_1, \ldots, \Delta V_{n+1}$ are random variables with $i.i.d$ ~$N(0,1)$. Add another node v_{m+1} to this path and the new delay has the form:

$$D(v_1,\ldots,v_{m+1}) \tag{16}$$
$$= D_{nom}' + B\left(\alpha(\Delta V_0+\Delta V_{m+2})+(\alpha+\beta)(\Delta V_1+\Delta V_{m+1})+\sum_{i=2}^{m}(2\alpha+\beta)\Delta V_i\right)$$

which can also be expressed as

$$D_{mean}(v_1,\ldots,v_{m+1}) = D_{mean}(v_1,\ldots,v_m)+D_{mean}(v_{m+1}) \tag{17}$$
$$Var(D(v_1,\ldots,v_{m+1})) = Var(D(v_1,\ldots,v_m))+Var(D(v_{m+1}))$$
$$+2\rho\sqrt{Var(D(v_1,\ldots,v_m))\times Var(D(v_{m+1}))}$$

where ρ is the correlation coefficient of $D(v_1,\ldots,v_m)$ and $D(v_{m+1})$.

Next, we derive an upper bound of ρ from (14) and (15):

$$\rho = \frac{E\left(\left(D(v_1,\ldots,v_m)-D_{mean}(v_1,\ldots,v_m)\right)\left(D(v_{m+1})-D_{mean}(v_{m+1})\right)\right)}{\sqrt{Var(D(v_1,\ldots,v_m))\times Var(D(v_{m+1}))}} \tag{18}$$
$$= \frac{2\alpha\beta+\alpha^2}{\sqrt{\left((m-2)(2\alpha+\beta)^2+2\alpha^2+2(\alpha+\beta)^2\right)\times\left(2\alpha^2+\beta^2\right)}}$$
$$\le \frac{2\alpha\beta+\alpha^2}{\sqrt{(m-1)(2\alpha+\beta)^2\times\left(2\alpha^2+\beta^2\right)}} \le \frac{1}{\sqrt{m-1}}$$

From equation (18), it can be seen that the correlation coefficient is smaller than the inverse square of the length of old path minus 1, similar method can be applied to equation (15) to derive an upper bound for the variance of $maxD(v_{source},v_i,v_{sink})$

Assume the path of source to *i* has *n* (*n*>2) node, *i* to sink has *m* (*m*>2) node, there are:

$$D_{mean}(v_{source},v_i) = D_{mean}(v_{source},v_{i-1})+D_{mean}(v_i)$$
$$Var(D(v_{source},v_i)) = Var(D(v_{source},v_{i-1}))+Var(D(v_i)) \tag{19}$$
$$+2\rho_1\sqrt{Var(D(v_{source},v_{i-1}))\times Var(D(v_i))}$$
$$D_{mean}(v_{source},v_i,v_{\sin k}) = D_{mean}(v_{source},v_i)+D_{mean}(v_{i+1},v_{\sin k})$$
$$Var(D(v_{source},v_i,v_{\sin k})) = Var(D(v_{source},v_i))+Var(D(v_{i+1},v_{\sin k}))$$
$$+2\rho_2\sqrt{Var(D(v_{source},v_i))\times Var(D(v_{i+1},v_{sink}))}$$

Using procedures similar to the derivation of (17) the following holds for all paths from source passing through node *i* to sink:

$$\rho_1 \le \frac{1}{\sqrt{n-1}} \qquad \rho_2 \le \frac{1}{\sqrt{(n-1)(m-1)}} \tag{20}$$

Therefore we can derive an upper bound for correlation coefficients and use it as an upper bound for the variance of $maxD(v_{source},v_i,v_{sink})$.

From the above, assuming that circuit delays are all Gaussian random variables if gate delays are Gaussian random variables, we can use the mean value plus *kσ* as the nominal delay for the random variables.

So the pseudo code for graph simplification is as follows:

```
For (each gate $v_i$)
  Calculate $maxD(v_{source},v_i)$ and $maxD(v_{i+1},v_{sink})$
  Using the upper bound of correlation coefficient to calculate the delay of $maxD(v_{source},v_i,v_{sink})$
If (meet timing requirement)
    Delete node $v_i$ and edges associated to $v_i$
```

3.2 Dual V_{th} assignment

There are many existing literatures[2]-[10][12] on this topic, the basic idea is to find out gates which should be assigned high V_{th} by queuing all the gates in certain order, and then to assign these gates high V_{th} until the queue is empty. In this paper, a dual V_{th} assignment scheme modified on our previous work [12] is used in the reduced DAG after our DAG pruning phase.

4. Experimental results and discussions

This section describes the implementation and shows the experimental results for ISCAS85 benchmark circuits. The simulation results are based on a standard cell library constructed using the PTM 90nm bulk CMOS model [20]. V_{dd} =1.0V, $|V_{thl,mean}|$=300mV $|V_{thh.mean}|$=500mV, σ_{vth}=10%$V_{th,mean}$ are set for all the transistors in the circuits. The leakage current lookup tables are created using HSPICE and PTM 90nm bulk CMOS model. All ISCAS85 benchmark circuit netlists are synthesized using a commercial synthesis tool and mapped to the 90nm standard cell library. A static timing analysis (STA) tool [12] is modified in a statistical way mentioned above to analyze the timing constraints of each circuit. The experiments were performed on a computer with Intel 2GHz CPU and 2GB RAM.

We compared three different conditions: corner based STA, worst case SSTA (p≈1), moderate constraint (p=95%) SSTA under 5% and 0% timing performance constraint (D_{req}) relaxation, respectively.

In table 1, it is shown that both our worst case SSTA and moderate constraint SSTA can achieve very impressive leakage reductions and graph reduction potentials when the performance constraint relaxation is 5%. N_r represent gates remained after DAG pruning procedure. N_H represents gates that are assigned high V_{th} after the optimization. Normalized I_{leak} is the leakage current normalized to maximum leakage current, i.e. the leakage current where all gates are with low V_{th}. After DAG pruning, only about 60% gates remain using worst case SSTA and moderate constraint SSTA, whereas more than 70% gates remain using STA. Secondly, about 90% gates can be assigned high V_{th} in SSTA, while only about 80% gates can be assigned high V_{th} in STA. Finally, 58.0% and 61.8% more leakage saving can be achieved by worst case SSTA and moderate constrain SSTA compared with that by STA.

The SSTA-based optimization is better than STA-based optimization for the following two reasons: (1) STA does not consider correlation of delay between gates, i.e. it treats delay as independent of each other, while in actual circuits, spatially close gates tend to affect the delay of each other; (2) STA specifies that delay MUST satisfy the timing constraints under the worst case delay, thus leads to more strict timing requirement at the expense of less leakage reduction. The same reasons also explain why the DAG pruning procedure is more effective in the two SSTA conditions than in the STA condition.

Table 2 shows the condition that the performance constraint relaxation is 0%. The worst case SSTA- and moderate constraint SSTA-based optimization can still achieve on average 38.9% and 42.1% more leakage saving than the STA -based optimization. And after DAG pruning procedure, only less than 70% gates remains in worst case SSTA and moderate constraint SSTA conditions, whereas more than 75% gates remains in STA condition.

We further compare the runtime of optimization w and w/o DAG pruning in Table 3. Step 1 is the DAG pruning phase, Step 2 is the dual V_{th} assignment in the remaining DAG. If the DAG pruning procedure is not used, the run time can be up to 2X of that with DAG pruning for larger circuits in ISCAS85 benchmark circuits. Furthermore the runtime shown in table 3 consists with the remaining gates number in Table 1 and 3: that is: the more gates remain after graph reduction, the longer the total runtime of the algorithm.

5. Summary

We present an approach to statistically assign dual threshold voltages while performing low-power optimization. The DAG pruning procedure of our algorithm makes it especially efficient in SSTA without losing its accuracy. The effectiveness and efficiency of our approach have been demonstrated by the experimental results, which show that at least 40% leakage power can be reduced compared with STA-based method. In addition, by using the statistical DAG pruning procedure, approximately 30~40% gates can be removed and up to 50% runtime can be saved without affecting the timing yields of the circuit. Furthermore, the correlation between gates can be easily adapted to more complex situations according to the actual circuit layout information.

References

[1] K. Roy, S. Mukhopadhay, and H. Mahmoodi -Meimand, “Leakage current mechanisms and leakage reduction techniques in DSM CMOS circuits,” Proceedings of the IEEE, vol. 91, no.2, pp. 305–327, Feb. 2003.

[2] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. V. Panda, and D. Blaauw, Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing, DAC (1999), pp. 436-441.

[3] L. Wei, Z. Chen, M. Johnson, and K. Roy, “Design and optimization of low voltage high performance dual threshold cmos circuits,” in Proc. DAC, 1998, pp. 489–494.

[4] Q. Wang and S. B. K. Vrudhula, “Static power optimization of deep submicron cmos circuits for dual vt technology,” in Proc. ICCAD, 1998, pp. 490–496.

[5] V. Sundararajan and K. K. Parhi, “Low power synthesis of dual threshold voltage cmos vlsi circuits,” in Proc. ISLPED, 1999, pp. 363–368.

[6] D. Nguyen, A. Davare, M. Orshansky, D. Chinney, B. Thompson, and K. Keutzer, “Minimization of dynamic and static power through joint assignment of threshold voltages and sizing optimization,” in Proc. ISLPED, 2003, pp. 158–163.

[7] F. Gao and J. P. Hayes, “Gate sizing and vt assignment for activemode leakage power reduction,” in Proc. ICCD, 2004, pp. 258–264.

[8] A. Srivastava, D. Sylvester, and D. Blaauw, “Statistical optimization of leakage power considering process variations using dual-vth and sizing,” in Proc. DAC, 2004, pp. 773–778.

[9] M. Liu, W. Wang, and M. Orshansky, “Leakage power reduction by dual-vth designs under probabilistic analysis of vth variation,” in Proc. ISLPED, 2004, pp. 2–7.

[10] A. Davoodi and A. Srivastava, “Probabilistic dual-vth optimization under variability,” in Proc. ISLPED, 2005, pp. 143–147.

[11] S. Mutoh et al. “1-V Power Supply High Speed Digital Circuit Technology with Multithreshold Voltage CMOS,” in IEEE JSSC, Vol. 30, No. 8 August 1995

[12] Y. Wang, H.Z. Yang, H. Wang, ”Signal-path Level Dual-Vt Assignment for Leakage Power Reduction,” Journal of Circuits, Systems and Computers, Vol. 15, No. 2 (2006), pp 197 -216.

Table 1 Comparison between STA and SSTA based dual V_{th} assignment under 5% performance constraint relaxation

ISCAS Bench Circuit	Gate num	Nr 95% (SSTA)	Nr 99.7% (SSTA)	Nr (STA)	N_H 95% (SSTA)	N_H 99.7% (SSTA)	N_H (STA)	Normalized I_{leak} 95% (SSTA)	Normalized I_{leak} 99.7% (SSTA)	Normalized I_{leak} (STA)
C432	169	128	128	131	145	139	119	24.4%	26.3%	47.0%
C499	204	196	196	196	184	183	158	9.9%	10.3%	21.0%
C880	383	208	208	234	374	371	358	2.8%	3.3%	5.9%
C1355	548	492	492	540	423	413	345	15.7%	16.8%	25.2%
C1908	911	493	498	631	857	853	753	5.3%	5.7%	21.7%
C2670	1279	690	710	801	1264	1264	1180	1.9%	2.0%	9.0%
C3540	1699	756	779	1026	1663	1656	1542	3.6%	4.1%	11.5%
C5315	2329	1290	1303	1377	2277	2275	2090	3.2%	3.3%	9.2%
C6288	2447	2011	2042	2301	1977	1937	1129	18.0%	19.4%	48.0%
C7552	3566	1188	1233	1707	3491	3483	3199	3.3%	3.5%	11.2%
average	NA	**59.2%**	**64.6%**	**72.7%**	91.8%	90.7%	79.6%	**61.3%**	**58.1%**	NA

Table 2 Comparison between STA and SSTA based dual V_{th} assignment under 0% performance constraint relaxation

ISCAS Bench Circuit	Gate num	Nr 95% (SSTA)	Nr 99.7% (SSTA)	Nr (STA)	N_H 95% (SSTA)	N_H 99.7% (SSTA)	N_H (STA)	Normalized I_{leak} 95% (SSTA)	Normalized I_{leak} 99.7% (SSTA)	Normalized I_{leak} (STA)
C432	169	128	128	131	130	126	116	42.1%	43.4%	49.4%
C499	204	196	196	196	167	162	151	17.4%	19.6%	22.4%
C880	383	218	218	272	364	362	351	5.0%	5.3%	7.4%
C1355	548	492	492	540	374	364	317	21.4%	22.6%	36.3%
C1908	911	528	531	696	821	819	726	8.0%	8.1%	26.9%
C2670	1279	770	780	806	1241	1238	1153	3.6%	3.8%	11.1%
C3540	1699	947	962	1044	1601	1596	1476	7.8%	8.2%	15.0%
C5315	2329	1359	1365	1447	2200	2190	2070	6.0%	6.2%	9.7%
C6288	2447	2142	2172	2374	1454	1425	1106	36.5%	37.5%	48.8%
C7552	3566	1538	1567	1847	3407	3388	3054	5.5%	6.0%	14.2%
average	NA	**69.0%**	**69.5%**	**76.4%**	85.3%	84.2%	76.0%	**42.1%**	**38.9%**	NA

Table 3 Runtime comparison under 5% performance constraint relaxation (time: s)

ISCAS Bench Circuit	With DAG pruning phase									W/O DAG pruning phase		
	Step1 95% (SSTA)	Step1 99.7% (SSTA)	Step1 (STA)	Step2 95% (SSTA)	Step2 99.7% (SSTA)	Step2 (STA)	Total 95% (SSTA)	Total 99.7% (SSTA)	Total (STA)	Total 95% (SSTA)	Total 99.7% (SSTA)	Total (STA)
C432	0.016	0.031	0.016	0.23	0.32	0.23	0.3	0.3	0.3	0.30	0.30	0.31
C499	0.016	0.031	0.016	0.40	0.42	0.45	0.4	0.5	0.5	0.52	0.53	0.53
C880	0.031	0.031	0.031	0.67	0.67	0.80	0.7	0.7	0.8	1.1	1.1	1.1
C1355	0.094	0.094	0.094	3.38	3.42	3.52	3.5	3.5	3.6	4.5	4.6	4.6
C1908	0.19	0.20	0.19	3.94	3.92	6.17	4.1	4.1	6.4	6.6	6.6	8.8
C2670	0.20	0.20	0.19	7.64	7.75	8.86	7.8	8.0	9.0	11.7	11.8	13.7
C3540	0.39	0.38	0.39	13.78	13.73	15.05	14.2	14.1	15.4	23.3	23.3	25.6
C5315	0.59	0.61	0.61	26.33	26.55	27.03	26.9	27.2	27.7	43.5	44.1	43.6
C6288	1.97	1.95	1.89	135.52	140.73	186.23	137.5	142.7	188.1	174.9	182.4	255.1
C7552	1.86	1.89	1.86	60.23	62.31	82.83	**62.1**	**64.2**	**84.7**	**120.3**	**121.9**	**188.4**