IR-drop Reduction Through Combinational Circuit Partitioning

Hai Lin, Yu Wang, Rong Luo, Huazhong Yang, and Hui Wang

EE Department, Tsinghua University, Haidian District
Beijing, 100084, P.R. China*
{linhai99, wangyuu99}@mails.tsinghua.edu.cn,
{luorong, yanghz, wangh}@tsinghua.edu.cn

Abstract. IR-drop problem is becoming more and more important. Previous works dealing with power/ground (P/G) network peak current reduction to reduce the IR-drop problem only focus on synchronous sequential logic circuits which consider the combinational parts as unchangeable [4],[5]. However, some large combinational circuits which work alone in one clock cycle can create large current peaks and induce considerable IR-drops in the P/G network. In this paper, we propose a novel combinational circuit IR-drop reduction methodology using Switching Current Redistribution (SCR) method. A novel combinational circuit partitioning method is proposed to rearrange the switching current in different sub-blocks in order to reduce the current peak in the P/G network, while circuit function and performance are maintained. Experimental results show that, our method can achieve about 20% average reduction to the peak currents of the ISCAS85 benchmark circuits.

Keywords: IR-drop, circuit partitioning, Static Timing Analysis.

1 Introduction

With technology stepping into submicron region, circuit design for single-chip integration of more complex, higher speed, and lower supply voltage systems has made the on-chip signal-integrity (SI) problem to be a tough task. Among all the sources of SI problem, the dynamic voltage drop caused mainly by *Ldi/dt* and IR-drop draws much attention in recent years.

As the supply voltage goes down continuously, ignoring the dynamic voltage drop through supply networks will cause run-time errors on real chips. These errors may include that transistors may not turn on with an unexpected voltage drop, and a timing constraint violation because of a delay increase of the standard gates with lower supply voltage. Some publications have already paid attention to reduce the voltage variation on P/G network for all kinds of purposes. Early publications focus directly on the optimization of the P/G network of the circuit, such as supply wire sizing [1] and P/G network decoupling capacitance (DC) insertion [2], [3] strategies. However as the technology feature

* This work was sponsored in party by NSFC under grants #90207001 and #90307016.

J. Vounckx, N. Azemard, and P. Maurine (Eds.): PATMOS 2006, LNCS 4148, pp. 370–381, 2006.

scales down, such efforts become insufficient and suffer from the drawback of large on-chip resource occupation.

In recent years, a few researchers have focused on the optimization of the logic blocks of the circuit[4],[5]. In publication [5], a synchronous digital circuit is first divided into "clock regions" and then these regions are assigned with different-phase clocks, in this case the author tried to spread the original simultaneous switching activities on the time axis to reshape the switching current waveform and reduce the current peak.

However, those algorithms using clock as the controlling signal to distribute the switching activity have an essential defect. As mentioned in [4], these algorithms lack the ability to control combinational circuit. Even in sequential circuits, the combinational part which triggered by flip-flops works alone in one clock cycle and draw corresponding currents from power network. When these combinational parts are large enough, the current peak created by one single combinational part is quite considerable. This problem cannot be settled by algorithms dealing with clock skew assignment.

In this paper, we present our IR-drop reduction method in combinational circuits. And the paper mainly has three contributions:

1. We derive a formal problem definition of IR-drop reduction in the combinational circuits and propose a novel combinational circuit IR-drop reduction methodology using *Switching Current Redistribution* (SCR) method based on circuit partitioning.

2. We give out a combinational circuit decomposition algorithm with better circuit slack utility to support our SCR method. Combinational block is partitioned into sub-graphs based on a new partitioning criterion called *slack sub-graph partitioning* to rearrange the switching time of different parts. STA tool is used to insure the original timing constraints and critical paths, in this way the exact logic function and the highest working frequency are both preserved.

3. A simple and proper additional delay assignment strategy is proposed. Then we compare some methods which modify the decomposed circuits to redistribute the switching current while the logical function and the performance constraints of the circuit are maintained.

The paper is organized as follows. The definition of combinational circuit IR-drop reduction problem is proposed in Section 2. Our novel circuit decomposition method is presented in Section 3. In Section 4 we present the additional delay assignment and the exact circuit modification strategy to achieve the additional delay. The implementation and experimental results are shown and analyzed in Section 5. In Section 6, we give the conclusion.

2 Problem Definition of Combinational Circuit IR-drop Reduction

2.1 Preliminary

Our research focuses on gate level combinational circuits. At the gate level, a combinational circuit can be represented by a directed acyclic graph (DAG), *G=(V, E)*. A vertex $v \in V$ represents a CMOS transistor network which realizes a single output logic function (a logic gate), while an edge*(i; j)*$\in$*E*, *i*, *j*$\in V$ represents a connection from vertex i to vertex j.

We define three attributes for every vertex $v \in V$, they are , the arrival time $t_a(v)$, the required time $t_{req}(v)$, and the slack time $t_{slk}(v)$. The arrival time $t_a(v)$ is the worst case signal transfer time from the primary inputs to vertex v. $t_{req}(v)$ is the latest time the signal needs to arrive at vertex v. We define them as:

$$t_a(v) = \begin{cases} t_0 \text{ given time of arrival if } v \text{ is the primary input} \\ \max_{i \in fanin(v)} \{t_a(i) + d(i)\} \text{ otherwise} \end{cases} \tag{1}$$

$$t_{req}(v) = \begin{cases} t_a(v) \text{ if } v \text{ is the virtual output} \\ \min_{i \in fanin(v)} \{t_{req}(i) - d(v)\} \text{ otherwise} \end{cases} \tag{2}$$

The signal propagation delay of a vertex *d(v)* can be respectively represented as:

$$d(v) = \frac{K C_L V_{DD}}{(V_{DD} - V_{TH})^{\alpha}} \tag{3}$$

Where C_L and V_{TH} are the output load capacitance and the transistor threshold voltage of the gate, respectively; K and α are technology dependent constants. The slack time of a gate v is defined as the difference of its arrival time and required time.

$$t_{slk}(v) = t_{req}(v) - t_a(v) \tag{4}$$

The slack time of a gate v represents the timing laxity of the graph at this point. The performance will not be harmed if a circuit modification still maintains the $t_{slk}(v) \leq 0$. We can call it a *slack time limitation.*

If we define a working frequency, the *critical path* of the circuits is constituted by the set of gates that has the minimum slack time value. And with the highest working frequency, this minimum slack value is zero. Our analysis focuses on the highest working frequency situation to ensure the original best performance of the circuit.

2.2 Problem Definition

The IR-drop $\Delta V(t)$ under a certain input can be represented as:

$$\Delta V(t) = I(V,t) \times R_{P/G} = (\sum_{v \in V} I_v(t, input_v, t_a(v), d(v))) \times R_{P/G} \tag{5}$$

Where $R_{P/G}$ is the P/G network resistance; *I(V,t)* is the current of the combinational circuits; I_v is the switching current of the individual gate $v \in V$, which is determined by its input state $input_v$, input signal arrival time $t_a(v)$ and propagation delay *d(v)*. From the equation (5), we can modify I_v through $t_a(v)$ and $d_(v)$ in order to minimize the current peak of the combinational circuit. However if we adjust every gate to get the optimal result, the IR-drop reduction problem will be unacceptably difficult.

As a result, in our method the combinational circuit *G=(V, E)* is partitioned into independent blocks $G_{sub} = G_1, G_2, , G_n$in order to simplify the IR-drop problem. Thus the IR-drop can have an alternative definition as below:

$$\Delta V(t, G_k, D_k) = \left\{ \sum_{1 \le k \le n} (t, input_k, T_{a,k}, D_k) \right\} \times R_{P/G} \tag{6}$$

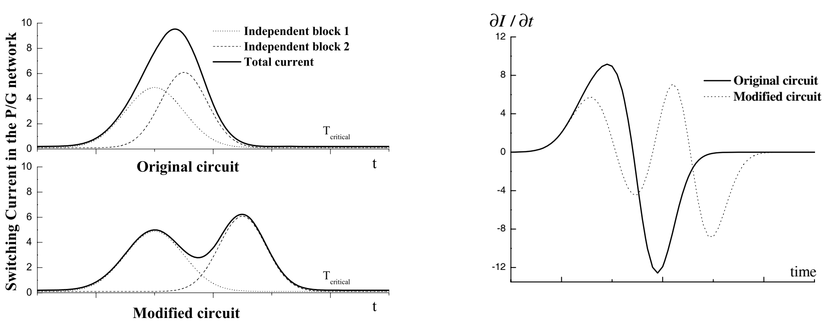
Where I_k is the switching current of block $G_k = (V_k, E_k)$,$G_k \subset G$, $1 \le k \le n$; $input_k$ is the input state of block G_k; $T_{a,k} = \{t_a(v_in), v_{in} \in V_k\}$ and $D_k = \{d(v_{in}), v_{in} \in V_k\}$ are the arrival time set and the propagation delay set for all the input vertexes of block G_k respectively. Therefore we only need to modify the delay value of all the input vertexes of the independent blocks to redistribute the switching current. Thus the IR-drop reduction problem of a combinational circuit can be defined as:

$$\min_{G_k, D_k} \left\{ \max_t \{\Delta V(t, G_k, D_k)\} \right\} \tag{7}$$

while satisfies the circuit performance constraints:
$t_a(m) = 0$, $\forall m \in$ Primary input, $m \in V$
$t_a(u) + d(u) \le T_{critical}$, $\forall u \in$ Primary input, $u \in V$
$t_a(i) + d(i) \le t_a(j)$, $\forall (i, j) \in E$, $i, j \in V$
where $T_{critical}$ is the delay of the circuit critical path.

2.3 Switching Current Distribution Methodology

As in the problem definition, the IR-drop reduction problem can not be easily solved. Based on circuit partition we presented our own method to solve the problem in a smart way of combinational blocks' switching current distribution.



(a) Current amplitude comparison (b) *di/dt* comparison

Fig. 1. Switching current redistribution

Shown by Fig. 1, if the combinational circuit are partitioned into two independent blocks without signal dependence, their switching current can be adjusted independently, by separate the switching time of the two blocks the current peak can be considerably reduced. Moreover, as mentioned above, the *Ldi/dt* noise is becoming significant in the P/G network. To smooth the currents waveforms in this way may also help reduce such noise when inductance of the P/G network is considered (see Fig. 1). We call this *Switching Current Redistribution.* To achieve this specific partitioning goal, we present a new algorithm combining static timing analysis (STA) information into the partitioning algorithm and make sure to maintain the critical paths after partitioning to ensure the circuit performance. And a simple and proper additional delay time assignment method is proposed to realize the redistribution of the switching current of different blocks.

3 Combinational Circuit Partitioning Method

The combinational circuit should be partitioned into independent blocks. These blocks should have no signal dependence between each other and their switching current can be modified independently to reduce the total switching current peak of the circuit. However, traditional partitioning algorithms [8], [9] are not capable for this specific partitioning requirement. First, traditional partitioning algorithms focus mainly on mini-cut or weighted mini-cut, while our partitioning requires awareness of signal independent characteristics of each block. Second, random assignment and element exchanging strategy in traditional partitioning algorithms can easily break critical paths of a combinational circuit. We develop our partitioning algorithm through which the critical paths are not cut off or modified in order to preserve original performance.

We first propose a concept of *slack sub-graph.* A sub-graph is called *slack sub-graph* if and only if all of its vertexes (gates) are of non-zero slack time at highest working frequency situation. And on the contrary, sub-graphs that consist of all zero slack time vertexes are defined as *critical sub-graphs.* According to this definition, *critical sub-graphs* consist of all the critical paths. If we only modify the *slack sub-graphs* under the timing constraints-all vertexes obey the *slack time limitation* discussed in the preliminary part, then the original critical paths in the circuit will not be affected, which conditionally satisfy our requirement of the "independence" characteristics between sub-graphs. Therefore our algorithm proposed a way to divide the combinational circuit into *slack sub-graphs* (G_{SLK}) and *critical sub-graphs* (G_{CRI}) which are independent under the timing constraints obtained by STA.

With the definition of *slack sub-graph*, our specific partitioning process is expressed as below:

Combinational-circuit-partitioning (G)

1 Perform STA to G and get the slack time of all the vertexes;
2 $V_{CRI} = \{v \mid t_{slk}(v) = 0\}$;
3 Get all the critical edges E_{CRI} ;
4 $G_{CRI} = (V_C RI, E_{CRI})$; // construct the critical block

5 $V_{SLK} = V - V_{CRI}$;
6 While (V_{SLK} not empty)
Begin while:
$\forall v_i \in V_{SLK}$; //randomly choose a vertex v_i
Get all the vertexes connected with v_i in V_{SLK}, and put them in set $V_{SLK}(i)$;
Get all the edges generated by vertexes in $V_{SLK}(i)$, and put them in set $E_{SLK}(i)$;
$G_{SLK}(i) = (\ V_{SLK}(i), E_{SLK}(i))$; // construct a slack sub graph
$V_{SLK} = V_{SLK} - V_{SLK}(i)$;
End while;
7 Return G_{sub}= $\{G_{CRI}, \{G_{SLK}(i)\}\}$; // return the independent blocks

Therefore, we obtain two kinds of blocks. One independent block consists of $G'_{CRI}s$, which should not be modified in order to maintain the circuit performance. We consider this block as *critical block.* And we obtain the other kind of independent blocks- slack blocks, consists of $G_{SLK}(i)'s$. The Switching Current Redistribution can be implemented through modifications of $G_{SLK}(i)'s$. From our partitioning algorithm, at least one slack block can be obtained. And always, the slack blocks whose size is comparable to the critical block are targeted to be modified, in this way we can achieve a high efficiency of current redistribution.

4 Strategy of Switching Current Redistribution

After circuit decomposition, it is important to modify the targeted *slack blocks* so as to redistribute their currents. Our attempt can be illustrated in Fig. 2.

For combinational circuits, there is no controlling signal like clock, so referring to equation (6), the slack block current $I(V_{SLK}, t)$ is determined by $T_{a,SLK}$, D_{SLK}. Thus in our method, we modify the input vertexes' delay in the *slack block* to modify the switching time of this block. We artificially delay the input signal transferring from the input to the next stage , in this way the switching time of this block is controlled. A simple and effective delay time assignment method is proposed to determine the amount of additional time that the input signal should be delayed.

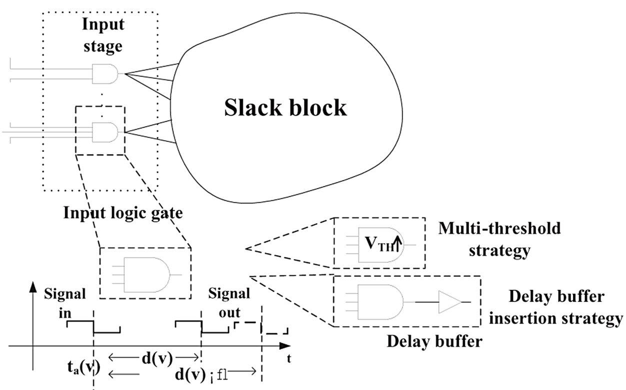


Fig. 2. Modify the input gates through two strategies

4.1 Additional Delay Assignment

Currents of different blocks are estimated using a simplified switching current estimation model similar with the one used in [10]. The switching current model of every logic gate is represented as a production of the switching activity α multiplied by the current waveform which is modeled as a trapezoid starting from the earliest possible switch time of the gate and ends at the latest. The trapezoid wave model is derived from the gate's original switching current waveform -a triangle representation. (see Fig. 3). The current model $I^i_{model}(t)$ for gate i is presented as follow, α is the switching activity of gate i.

$$I^i_{model}(t) = \alpha I'_{gi}(t) \tag{8}$$

The total current from one slack block is the sum of all the gates' current within it. And we can easily calculate the peak when we actually store the current waveform by discrete value at each time interval.

We perform a simple but practical additional delay time assignment strategy to achieve a considerable large reduction in the switching current peak of the combinational circuit.

Here, we propose the experimental based assignment of the artificial additional delay value of input gate in every targeted sub-graph. We assign the additional delay of input vertex to the amount of its slack time to form the initial solution SOL_{DLY}, so that we may spread more switching current to the entire circuit switching period and reduce the overlapping of switching currents of critical block and the targeted slack blocks. Then a small nearby region search for better solution of this assignment is made based on the evaluation of I_{peak}. Experimental results show that little change of the initial solution is needed and this simple and practical additional delay assignment strategy can appropriately redistribute the switching current of the blocks, utilize the total circuit switching period more equably and reduce the peak current of the whole circuit to a considerably lower value.

The circuit performance is maintained since the *critical block* is not changed and the *slack blocks* are adjusted following the timing constraints. In the slack time assignment, slack information is extracted by our STA tool [7]. The final delay time information of each input gate is saved in specific data file for circuit modification procedure.

4.2 Additional Delay Achievement

One practical strategy to realize in circuit the additional delay insertion is to change the transistor threshold voltage V_{TH} of the input gates so as to change $d(v)$ of them. Referring to equation (3), signal transfer delay of a logic gate is related to the threshold voltage of its transistors and the adequate threshold voltage can be calculated according to the required delay. However, as the threshold values of transistors can not be continuously changed in reality and are often fixed to several threshold levels in multi-threshold design due to process

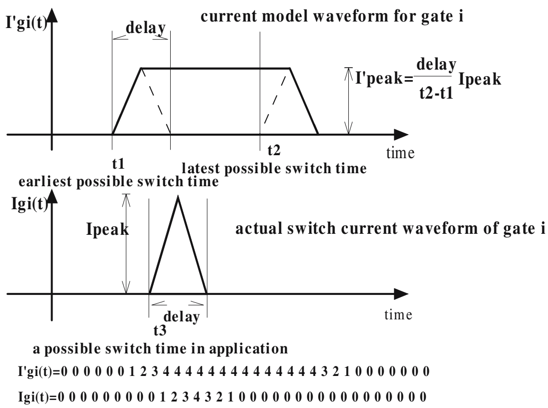


Fig. 3. Current model for logic gate

limitation, it is necessary to adjust V_{TH} of input gates to acceptable discrete values.

Therefore, here we propose a three-level discrete V_{TH} assignment to achieve the additional delay.

Discrete V_{TH} assignment
1 Set discrete V_{TH} value: V_{THO}, $V_{THO} + \Delta V_{THL}$, $V_{THO} + \Delta V_{THH}$;
2 Read the required additional delay Δd_i for v_i, $v_i \in$ input vertexes of slack blocks;
3 V_{input}=$\{v_i \mid v_i \in input\_vertexes\_of\_slack\_blocks\}$
4 While ($V_{input}! = \phi$)
Begin while:
Random select $v_i \in V_{input}$,
$\Delta V'_{TH}$ =f(Δd_i) //calculate actual ΔV_{TH} required for gate i according to equation(3)
Set ΔV_{TH}: if 0$\leq$ $\Delta V'_{TH} < \Delta V_{THL}$, ΔV_{TH}= 0
if $\Delta V_T HL \leq \Delta V'_{TH} < \Delta V_{THH}$, $\Delta V_{TH} = \Delta V_{THL}$
Else if $\Delta V_{THH} \leq \Delta V'_{TH}$
(1) Set $\Delta V_{TH} = \Delta V_{THH}$
(2) $\Delta d = d(\Delta V'_{TH}) - d(\Delta V_{THH})$ Calculate the delay difference caused by $\Delta V'_{TH}$ and ΔV_{THH};
(3) Get all the output vertexes of v_i, assign Δd information to all them and put them in set V_out; //propagate the overflow delay to the next stage.
End while;

As in our experiment we are using TSMC 0.18m standard cell library for simulation, the three discrete V_{TH} values are set to V_{THO}, $V_{THO} + \Delta V_{THL}$ and $V_{THO} + \Delta V_{THH}$. V_{THO} is the library determining original transistor threshold voltage. And in our experiment, we set $\Delta V_{THL} = 0.2V$, $\Delta V_{THH} = 0.4V$. In reality, the actual discrete values can be determined by process limitation. First of all, the input gate is assigned one of the discrete V_{TH} values that is just below the calculated value. Then, if the required additional delay exceed the maximum value that can be achieved by a single input gate, the overflow delay would be assigned to gates in the following stage of the slack block. In our experiment, we only allow two stages of the slack block to be modified (input stage and the stage following that) to reduce the modification complexity. The simulation result of this circuit modification strategy is presented in Table.1 and compared to buffer insertion strategy.

Buffer insertion strategy is a backup strategy for multi-threshold strategy, and reduces the fabrication process cost. Instead of change *d(v)* of the input gates, specialized buffers are inserted right after original input gates, thus change the arrival time of the other gates in the slack block. However, this strategy has two major drawbacks: additional area occupation and more power dissipation. Thus we consider using it only if we can not use the multi-threshold strategy.

5 Implementation and Experimental Results

The implementation of our algorithm can be illustrated in Fig. 4. Our gate level netlists are synthesized using Synopsys Design Compiler and a TSMC 0.18μm standard cell library. The DAG extraction and customized circuit partitioning procedure have been implemented in C++ under a customized STA environment according to the TSMC standard cell delay library. We implemented a small tool to automatically generate the modified gate list including the delay time assignment and the two circuit modification strategies. Both the original and modified circuits are simulated using HSPICE with TSMC 0.18μm CMOS process and a 1.67V supply condition. The P/G network is modeled as RC network.

As our algorithm focuses on the redistribution of switching current from logic blocks, the architecture of P/G network model does not have much influence in

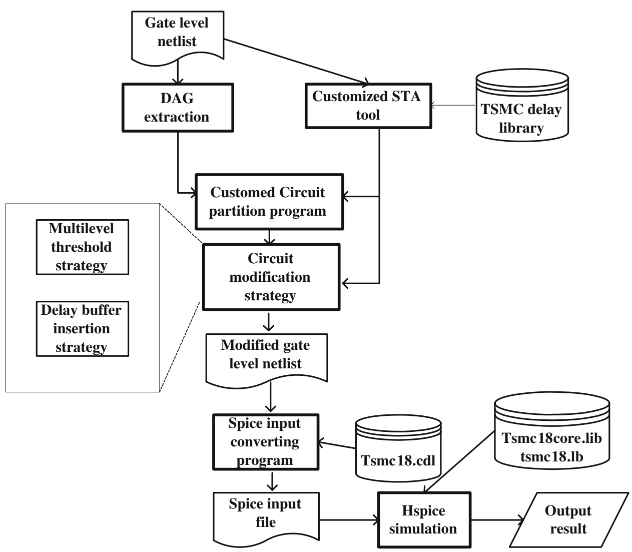


Fig. 4. Implementation procedure

the peak current reduction ability. We actually compared the simulation results from the circuit with simple model(single R and C) and complex model(multiple R and C connected as a mesh) of P/G network in several circuits. It shows that the detailed waveform of the current is changing slightly with the P/G network variation but the reduction rate remains approximately the same (see Fig. 5). As a result, in our simulation we simply model the P/G network as a 100 Ω resistance connected between V_{DD} and the logic block, and a capacitance of 0.3pf parallel connected to the logic block to reduce the simulation complexity. We apply the proposed method to ISCAS85 benchmark circuits and all the circuits are simulated with large number of random input vectors. And we are running the program on a PC with P4 2.6GHz and 512M memory.

We show in Fig. 5 the transient on-chip current waveform in one processing cycle of the modified circuit compared with the original circuit of C1355 simulated with both simple P/G network and complex P/G network. As we expected, the current waveform of both unmodified (figures above) and modified (figures below) circuit with complex P/G network (dotted line) are different from the ones with simple P/G network (real line). However the peak current reduction rate remains approximately the same. And comparing the waveforms with the same P/G model, we can find that the current curves with single peak in one processing cycle change into curves with two or more lower swing peaks after circuit modification. Thus the switching current of the two major kinds of blocks (the *slack blocks* and the *critical block*) is actually separated and the peak current of the circuit is significantly reduced.

Table.1 shows the current peak reduction results of both multi-threshold and buffer insertion strategies. We can see that the reduction of current peak varies with the circuit structures, from 15% up to 33% by multi-threshold and from 12% up to 32% by buffer insertion, which are very impressive. The circuits with more slack to be utilized get a better optimization result through our algorithm.

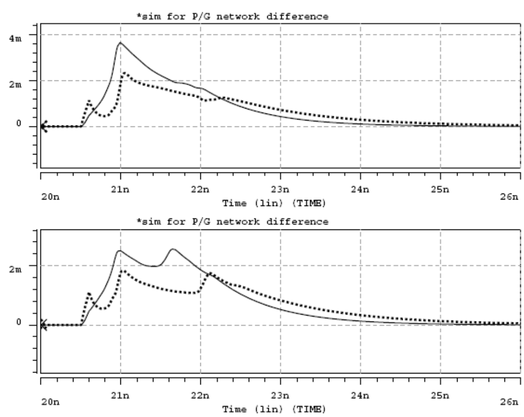


Fig. 5. Simulation current waveforms of C1355

Table 1. Comparison of the multi-threshold and buffer insertion strategies

ISCAS85 Circuit	Original	Multi-V_{TH}		Buffer insertion		
	Average I_{peak}(mA)	Average I_{peak}(mA)	I_{peak} Reduction	Average I_{peak}(mA)	I_{peak} Reduction	Area Overload ($N_{buf}/N_{totalgates}$)
C432	2.45	2.08	15%	2.15	12%	6/160
C499	4.27	2.98	31%	3.01	29%	14/202
C880	3.05	2.28	25%	2.31	24%	38/383
C1355	4.21	3.01	29%	2.85	32%	22/545
C1908	3.96	3.07	22%	3.04	23%	31/880
C2670	3.68	2.95	20%	2.98	19%	49/1269
C3540	2.78	2.27	18%	2.33	16%	84/1669
C5315	4.94	3.29	33%	3.75	24%	170/2307
C6288	4.78	3.87	19%	3.98	17%	119/2416
C7552	5.26	4.43	16%	4.46	15%	201/3513
average			23%		21%	

Here, we comment that the algorithm would have a limit of applicability if the slack blocks have too little slack amount to be utilize, which would be very rare for functional combinational circuits. Even in that case, we suggest a circuit slow down be induced to achieve more slack utility if reducing switching current peak is the most urgent problem for a application.

Although the peak current reduction is nearly the same, the average current of the circuit shows that buffer insertion strategy induces more on-chip current besides the draw back of on chip area overload due to the insertion of buffers. Some other strategies, such as gate sizing or transistor stacking, can also be considered in order to avoid large addition current meanwhile achieve the equivalent required delay.

6 Conclusions

IR-drop reduction is becoming essential in deep submicron circuit design today. The efficient reduction methodology needs to be improved imperatively. In this paper, we have presented a novel methodology for IR-drop reduction in combinational circuits through circuit partitioning and switching current redistribution. The original circuit is partitioned into independent blocks and the switching time of the blocks is carefully arranged to ensure that switching current redistribution is achieved for IR-drop reduction. The additional delay assignment and insertion is achieved without affecting the circuit performance under timing constraints. The experimental results for ISCAS85 benchmark circuits show an average current peak reduction on P/G network around 20%. The only drawback of this switching current redistribution method is that as the slack in the circuit is used for current redistribution, the circuit is going to lose some tolerance ability to process variations which affect the path delay. As the statistic effect on physical design is becoming not neglectableto induce a slight circuit slow down or to maintain a certain amount of the original slack according to the specific manufacturing technique would be both applicable in order to insure a process variation tolerance ability. Since our method does not have any performance loss and do not require modifications on P/G network or circuit clock trees, it can be used with other commonly used methods such as P/G network DC insertion and clock skew assignment in synchronous circuits to achieve further reduction ability of on-chip IR-drop.

References

1. Sheldon X.-D, Tan, C.-J, Richard Shi, Jyh-Chwen Lee: Reliability-Constrained Area Optimization of VLSI Power/Ground Networks Via Sequence of Linear Programmings. IEEE Trans. on CAD, Vol. 22, No. 12, pp. 1678-1684, 2003.
2. Mondira Deb Pant, Pankaj Pant, Donald Scott Wills: On-Chip Decoupling Capacitor Optimization Using Architectural Level Prediction. IEEE Trans. on VLSI, Vol. 10, No. 3, pp 772-775, 2002.
3. Howard H. Chen, J. Scott Neely, Michael F. Wang, Grieel Co: On-Chip Decoupling Capacitor Optimization for Noise and Leakage Reduction. Procs of IEEE ISCAS, 2003.
4. P. Vuillod, L. Benini, A. Bogliolo, G. De Micheli: Clock-skew optimization for peak current reduction. Procs of ISLPED.1996 Monterey CA USA, 1996.
5. Mustafa Badaroglu, Kris Tiri, Stphane Donnay, Piet Wambacq, Ingrid Verbauwhede, Georges Gielen, Hugo De Man: Clock Tree Optimization in Synchronous CMOS Digital Circuits for Substrate Noise Reduction Using Folding of Supply Current Transients. Procs of DAC 2002, June 10-14, New Orleans, Louisiana, 2002.
6. Amir H. Ajami, Kaustav Banerjee, Amit Mehrotra, Massoud Pedram: Analysis of IR-Drop Scaling with Implications for Deep Submicron P/G Network Designs. Procs of ISQED 2003.
7. Yu Wang, Huazhong Yang, Hui Wang: Signal-path level Assignment for Dual-Vt Technique. Procs of IEEE PRIME 2005, pp 52-55.
8. George Karypis, Rajat Aggarwal, Vipin Kumar, Shashi Shekhar: Multilevel Hypergraph Partitioning: Applications in VLSI Domain. IEEE Trans. on VLSI, Vol. 7, No. 1, pp 69-79, 1999.
9. Navaratnasothie Selvakkumaran, George Karypis: Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization. Procs of ICCAD'03, November 11-13, San Jose, California, USA, 2003.
10. Mohab Anis, Shawki Areibi, and Mohamed Elmasry: Design and Optimization of Multithreshold CMOS (MTCMOS) Circuits. IEEE Trans on CAD, Vol. 22, No. 10, pp 1324-1342, Oct 2003.