Static Leakage Reduction through Simultaneous Threshold Voltage and State Assignment

Dongwoo Lee, David Blaauw
University of Michigan, Ann Arbor, MI

Abstract

We propose a new method that uses a combined approach of sleep-state assignment and threshold voltage (Vt) assignment in a dual-Vt process. While each of these methods has previously been used individually, their combined effect has not been leveraged to date. By combining Vt and sleep-state assignment, leakage current can be dramatically reduced since the circuit is in a known state in standby-mode and only transistors that are off need to be considered for high-Vt assignment. A significant improvement in the leakage/performance trade-off is therefore achievable using such a combined method. We formulate the optimization problem for simultaneous state and Vt assignment under delay constraints and propose both an exact method for its optimal solution as well as a number of practical heuristics with reasonable run time. We compare our results with Vt and sleep state assignment only and demonstrate an average decrease in leakage current of 3.5X compared to previous approaches.

Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance analysis

General Terms

Algorithms, performance, design, reliability

1 Introduction

There is a growing need for low-voltage, high-performance and low-power system, especially for portable and battery-powered applications. Since these applications often remain in stand-by mode significantly longer than in active mode, their stand-by (or leakage) current has a dominant impact on battery life. Subthreshold leakage current reduction therefore has been a concern for some time and a number of such methods have been proposed [1]-[4]. With continued process scaling, lower supply voltages necessitate reduction of threshold voltages to meet performance goals and result in a dramatic increase in subthreshold leakage current. New methods for reducing the leakage current in standby mode are therefore critically needed. A number of such methods have been proposed.

In [1], the circuit input state that minimizes total leakage current is determined and special flip-flops are inserted in the design to produce this state in standby mode. The required modification to a flip-flop is minor and can be incorporated in the feedback path of the slave latch with minimal impact on performance [5]. In general, determining the minimum sleep state is a difficult problem due to the inherent logic correlations in the circuit. However, a number of efficient heuristics for this problem have been proposed. The limitation of this approach is that for larger circuits, the reduction in leakage current is typically in the range of only 10 to 30% [2]. In recent years, a number of methods for leakage reduction using individual assignment of transistor threshold voltages in a dual-Vt process have been proposed [3][4]. In these approaches, circuit paths that are non-critical are assigned high-Vt while critical circuit portions are given low-Vt assignments. It was demonstrated that with a modest performance reduction of 5–10%, significant reduction of 3-4x or more in leakage could be obtained [3].

In this paper, we proposed a new approach that combines the concepts of Vt-assignment and sleep state assignment. Our approach is based on the key observation that, given a known input state for a gate, the leakage of that gate can be dramatically reduced by setting only a single OFF-transistor on each path from Vdd to Gnd to high-Vt. Since all other transistors in the gate are kept at low-Vt and continue to have high drive current, the performance degradation is limited while significantly gains in leakage current is obtained. This approach therefore provides a much better trade-off between leakage and performance compared to Vt-assignment with unknown input state where most or all of the transistors must be set to high-Vt before a significant improvement in the leakage current is observed. The link between the effectiveness of Vt-assignment and state assignment was previously observed for Domino logic [5], since these circuits are by their own nature in a known state in standby mode. However, we extend this concept to general CMOS circuits by actively controlling the circuit state in standby mode, thereby dramatically increasing the effectiveness of leakage reduction.

Due to the interaction between circuit state and threshold voltages, it is necessary to consider their assignment simultaneously. For instance, the optimal Vt assignment depends on the leakage/performance trade-off of transistors in the circuit that are turned OFF. On the other hand, the state assignment controls this set of OFF-transistors. The objective of the state assignment is therefore not to simply maximize the number of OFF-transistors but to turn OFF specific transistors with favorable leakage/performance trade-offs in close interaction with the Vt-assignment algorithm. We therefore formulate the problem as an integer optimization problem under delay constraints. The search space consists of all input state/Vt assignments and hence is very large. We develop an exact solution to the problem and also propose a number of heuristics. The proposed methods were implemented on benchmark circuits synthesized using an industrial cell library in 0.18um technology. On average, the proposed method improved leakage current by a factor of 3.5X over the traditional approach using Vt-assignment only.

2 Simultaneous Vt and State Assignment

Consider the leakage and performance of the simple NAND2 circuit shown in Figure 1 under different input states and Vt-assignments. It is clear that given a particular input state, only those transistors that are OFF need to be considered for high-Vt assignment as the ON-transistors are not leaking. For instance, in state AB = 01, only transistor *tn*1 needs to be considered for high-Vt assignment. Assigning other transistors to high-Vt will only decrease the performance of the gate with no reduction in leakage current. On the other hand, in state 11 *both tp*1 and *tp*2 must be assigned high-Vt in order to reduce leakage, since they are parallel devices.

We can partition the transistors into so-called vt-groups, corresponding to the minimum sets of transistors that need to be set to high-Vt to reduce leakage in a particular state assignment. For the 2-input NAND gate in Figure 1, three vt-groups exist as shown. The concept of vt-groups can be easily applied to more complex structures in which case it may be possible that a transistor belongs more than one vt-group. It is clear that we can restrict ourselves to setting only entire vt-groups to either high or low-vt. By considering only

DAC 2003, June 2-6, 2003, Anaheim, California, USA.
Copyright 2003 ACM 1-58113-688-9/03/0006...$5.00.

Table 1. Leakage current of NAND2 gate

Input State	Assigned Group	Leakage current[pA]		
		w/ Group Assign.	w/ All High Vt	w/All Low Vt
00	2	24.9	7.2	286.7
	3	9.8		
	2 and 3	7.2		
01	2	26.6	26.6	1054.0
10	3	25.7	24.4	922.6
11	1	14.2	14.2	357.2

vt-groups, instead of individual transistors, we therefore significantly reduce the number of possible vt-assignment and the optimization complexity. In Table 1, we show the leakage current for the NAND2 in Figure 1 for different input states and vt-group assignments. Column 3 shows the leakage current when we use high-Vt for one or more Vt-groups that are OFF in a particular input state. In column 4 and 5, the leakage current with all transistors assigned to, respectively, high-Vt and low-Vt is shown. We can see that in states 01, 10, and 11 only a single Vt-group is a candidate for high-Vt assignment. Also, setting only this one vt-group to high-Vt results in equal or nearly equal leakage compared with the leakage when all transistors are assigned high-Vt demonstrating the effectiveness of the approach. In state 00, three high-Vt assignments are possible: group 2, group 3, and both group 2 and 3. However, the leakage current with both groups assigned to high-Vt is only slightly better than that with only one group set to high-Vt, and assigning group 3 to high-Vt reduces leakage somewhat more than assigning group 2 to high-Vt. Hence, it is clear that we need to only consider assignment of group 3 to high-Vt without significant loss in optimality.

Table 1 shows that the leakage current varies considerably as different groups associated with different input states are set to high-Vt. At the same time, the impact of different high-Vt group assignments on the performance of the circuit must be considered. By setting only a single group to high-Vt, the performance degradation is restricted to only a single signal transition direction and is also reduced compared to high-Vt assignments where most or all transistors are set to high-Vt. Therefore, the performance/power trade-off of Vt assignment with known input state is much improved compared with that with unknown input state.

The input state of a gate effects which transition direction is degraded by a high-Vt group assignment to a gate. Also, the position of the high-Vt group in a stack of transistors changes the impact of a high-Vt group assignment on the different input to output gate delays. Therefore, the input state of a gate must be chosen such that its associated high-Vt group results in the least degradation of the critical paths in the circuit. However, only the input state of the circuit as a whole can be controlled and the logic correlations of the circuit restrict the possible assignments of gate input states. Therefore, selection of the circuit input state and of which gate is assigned a high-Vt group must be made simultaneously to obtain the maximum improvement in leakage current with minimum loss in performance.

3 Exact Solution to Vt and State Assignment

The size of the input state space is 2^n, where n is the number of circuit inputs. For each input state assignment, there are two possible Vt assignments for each gate (one high-Vt group which is pre-determined by its input state, and all low-Vt). The total number of possible Vt-assignment is therefore 2^m, where m is the number of gates in the circuit and the total size of the search space is 2^{n+m}.

In order to find an exact solution to the problem, we developed an efficient branch-and-bound method that simultaneous explores the state and Vt assignments and that exploits the characteristics of the problem to obtain efficient pruning of the search space to improve the run time. Due to the exponential nature of the problem, an exact solution is only possible for very small circuits. However, the exact approach is still useful as the proposed heuristics are based on it.

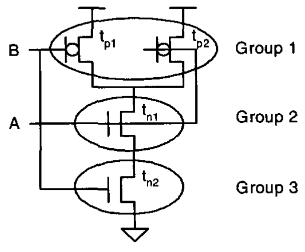

Figure 1. The concept of group at NAND2 gate

We use two types of branch and bound trees. The first branch-and-bound tree determines the input state of the circuit and is referred to as the *state tree*. The nodes of the state tree correspond to the input variables of the circuit inputs. Each node of the state tree is associated with a so-called *gate tree* which is searched to determine the group Vt-assignment. In other words, for a state tree with *k* nodes, there exist *k* copies of the gate tree. Each node in a particular gate tree corresponds to a gate in the circuit, as shown in Figure 2. Each node has two fanout edges, representing the assignment of that gate with all low-Vt groups (left branch) or with one high-Vt group, as determined by the input state of the gate (right branch).

At the root of the state tree, the state of all input variables is unknown. As the algorithm proceeds down the tree, the state of one input variable becomes defined with each level that is traversed. At each node in the state tree, a solution of leakage current can be obtained by traversing the gate tree. Note that the gate tree may be traversed both with a completely known input state at the bottom of the state tree as well as with a partially or completely unknown input state, at higher levels of the state tree.

For each node in the state and gate tree, an upper and low bound on the leakage current is computed incrementally as explained in Section 3.1. Note that early in the state tree the bounds on leakage will be very loose since the state of the circuit is only partly defined. As the algorithm traverses down the state tree, the input state becomes more defined and the leakage bounds become closer. Similarly, the leakage bounds are very wide at the top of each gate tree, as the Vt-assignment of all gates are unknown, and becomes progressively tighter as the algorithm traverses down the tree. Only at the bottom of *both* the state tree and its associated gate tree do the upper and low bounds on leakage coincide. The algorithm first traverses down to the bottom of the tree and then returns back up, to traverse down unvisited branches in DFS manner. During the search, a tree branch is pruned when if it has a lower bound on leakage that is worse than the best upper bound on leakage that has been observed so-far. In addition to pruning based on leakage bounds, we also compute a lower bound on the circuit delay at each node in the gate tree

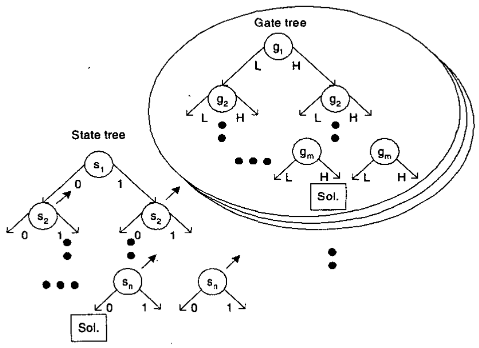

Figure 2. State tree with gate tree at each node

traversal and prune all branches whose lower bound exceeds the specified delay constraint. Computation of the delay bounds is also performed incrementally and is discussed in Section 3.2.

Also, early in the state tree, computation of the exact minimum Vt-assignment by traversing the gate tree is not meaningful since even at that bottom of the gate tree there is considerable uncertainty in the leakage current due to the unknown input state. Therefore, the gate tree is searched only partially at the higher levels of the state-tree which results in slightly more conservative bounds, but an overall improvement in the run time of the algorithm.

The gate tree is also searched in DFS manner and edges are pruned based on the computed leakage bounds. During the downward traversal of the gate tree, the high Vt branch is always selected, provide it meets the delay constraint. This is due to the fact that the high-Vt branch always has less leakage current than the low Vt branch. Only if the lower bound on the delay of the high-Vt branch exceeds the delay constraint, is the low-Vt branch selected and is the high-Vt branch pruned.

Finally, the gates in the circuit are assigned to nodes in the gate tree in topological order to enable incremental delay computation. Gates of equal topological level are further sorted by decreasing leakage to improve the pruning of the search space. The input signals of the circuit are also assigned to nodes in the state tree in specific order. We want to place inputs whose state assignment strongly influences the total leakage of the circuit near the top of the state tree. We estimate the influence of each input signal on the circuit leakage by taking the sum of the leakage current of all gates connected to the input signal. This input variable ordering is similar to that used in [6].

3.1 Incremental leakage bound computation

During the traversal of the gate tree, some of the gates will have a known Vt assignment and others, which have not been visited, will have an unknown Vt-assignment. As shown in Figure 3, a lower bound on the leakage is computed by assuming all unknown gates have a high-Vt group assignment and an upper bound is computed by assuming all unknown gate have a low-Vt group assignment. As the high branch is taken in the downward traversal, only the upper bound is update (decreased) while when a low branch is taken, only the lower bound must be updated and is increased.

3.2 Incremental delay bound computation

Similar to the leakage current bounds, a lower bound on the delay is computed assuming all unknown gates have low-Vt group assignments. Delay is changed only when a high branch is taken in the traversal and is computed incrementally. We first compute the slack of the circuit for all circuit nodes at the start of the tree traversal with all Vt-assignments assumed to be low-Vt. When a group changes from a low to a high-Vt group assignment during the traversal, the slack of that gate will be updated. However, the Vt change of the gate will affect not only the gate itself but also the delays of fanout gates due to the slope change at the output of the changed gate. Since the slope at the output of the changed gate will become slower due to its high-Vt assignment, the delay of all fanout gates will increase, resulting an overall increased circuit delay. Ignoring the effect of slope change will therefore result in the computation of an *optimistic* lower bound which ensures that the optimal solution is not accidentally pruned. It also enables incremental delay computation, given that the gates are visited in topological ordering. As gates are visited, the changed input slope, due to high-Vt assignments of a fanin gate, is processed to ensure that an exact delay bound is computed at the bottom of the gate tree.

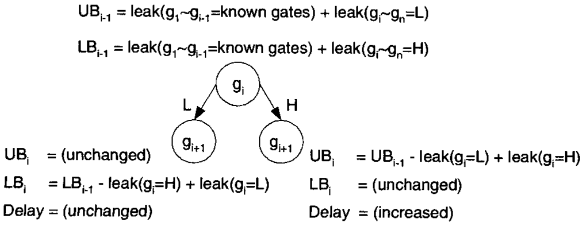
UB_{i-1} = leak(g_1~g_{i-1}=known gates) + leak(g_i~g_n=L)
LB_{i-1} = leak(g_1~g_{i-1}=known gates) + leak(g_i~g_n=H)
g_i
L
H
g_{i+1}
g_{i+1}
UB_i = (unchanged)
LB_i = LB_{i-1} - leak(g_i=H) + leak(g_i=L)
Delay = (unchanged)
UB_i = UB_{i-1} - leak(g_i=L) + leak(g_i=H)
LB_i = (unchanged)
Delay = (increased)

Figure 3. Incremental leakage bound computation

4 Heuristic solution to Vt and State Assignment

We propose three fast heuristics that can be applied to large circuits and that produce high quality solutions. The proposed heuristic are based on the exact method described in Section 3, and are discussed below.

Heuristic 1

In this heuristic, the state and gate tree search is limited to only one downward traversal. Note that while only a single traversal of the state tree is performed, at each node of the state tree the decision to follow the left or right child node is based on the computed bounds of the leakage using the gate tree. Each downward traversal of the gate tree visits *m* nodes, where *m* is the number of gates in the circuit. We perform exactly two such traversals at each state tree node, leading to a total run time complexity that is *O(nm)*, where *n* is the number of circuit inputs. Since the number of inputs is generally thought to grow approximately as the sqrt(m), the total complexity of this heuristic is *O(msqrt(m))*.

Heuristic 2

In the second heuristic, the state tree is searched more extensively, subject to a fixed run time constraint, while the gate tree search is kept to a single downward traversal for each state tree node. Experimentally, it was found that the quality of the first bottom node reached in the gate tree search is near the optimal Vt-assignment. This is due to the fact that the gate tree always chooses the high-Vt child in its downward traversal which tends to produce a high quality result. This is in contrast to the state tree, where choosing the correct child during the downward traversal was found to be much more difficult. Therefore, the solution quality was found to improve most by searching the state tree more extensively, subject to a run time constraint, while limiting the gate tree search to a single downward traversal.

Heuristic 3

In the third heuristic, input state assignments are generated randomly and for each input state, and the gate tree is searched using a single downward traversal.

5 Results

The proposed methods for simultaneous state and Vt-assignment were tested on the ISCAS benchmark circuits [7] and MCNC benchmark circuits [8], synthesized using a 0.18 industrial library using Synopsys. The leakage current for each Vt version of a cell was computed using SPICE simulation and stored in precharacterized tables. Delay computation was performed based on the Synopsys table delay model and was verified to match with Synopsys timing analysis delay reports. In addition to the proposed methods, the traditional methods using only state assignment or Vt-assignment were also implemented for comparison. The state-only assignment was implemented using the approach discussed in [6] while for Vt-only assignment a method similar to the sensitivity based approach discussed in [3] was used.

Table 2 compares the leakage results obtained with the three proposed heuristics for three delay constraints. The columns marked 10%, 25% and 50% refer to leakage minimization where the delay constraints were set at, respectively, 10, 25 and 50% from the all low-Vt circuit delay. The 10% column is therefore the most strin-

Table 2. Leakage current comparison between heuristics

	Minimized leakage current [nA]								
	10%			25%			50%		
	Heu1	Heu2	Heu3	Heu1	Heu2	Heu3	Heu1	Heu2	Heu3
C880	6.02	5.33	5.50	5.99	4.89	5.17	4.97	4.80	4.91
C1355	13.88	11.53	11.97	13.77	9.20	9.00	7.25	7.25	6.72
C1908	14.48	11.42	11.22	12.09	8.73	9.03	8.67	6.55	5.95
C2670	12.00	11.85	11.59	11.85	11.71	11.59	11.39	10.97	11.45
C3540	19.09	17.11	16.70	16.74	14.01	14.55	12.12	11.88	12.25
C5315	25.84	25.63	25.91	24.79	23.42	24.08	21.93	21.68	22.16
C6288	107.54	95.24	81.10	61.17	51.01	49.61	43.16	35.72	33.59
C7552	31.89	31.06	30.66	31.18	27.91	28.20	27.10	26.59	27.49
alu64	43.30	41.82	46.82	31.91	31.37	35.99	24.02	23.77	23.70
i2	1.69	1.69	2.03	1.31	1.31	1.76	1.25	1.25	1.70
i3	2.10	2.10	2.35	1.87	1.87	1.99	1.78	1.75	1.89
i4	2.44	2.44	2.97	1.89	1.89	2.16	1.74	1.70	1.87
i5	1.79	1.79	2.60	1.60	1.60	2.44	1.57	1.57	2.41
i6	5.26	5.26	5.41	5.13	5.13	5.34	5.13	5.13	5.34
i7	6.61	6.61	13.44	5.67	5.67	6.82	5.61	5.61	6.54
i8	11.81	10.29	8.20	10.94	8.57	8.16	10.31	8.53	8.16
i9	4.96	4.83	4.93	4.96	4.84	4.51	4.95	4.81	4.51
i10	26.98	26.98	27.30	24.85	24.85	25.80	24.43	24.42	25.46

Table 3. Leakage current comparison

	Circuits		Minimized leakage current [nA]						
	Number of		State only	Vt only & proposed heuristic (reduction factor: vs. Vt					
				10%		25%		50%	
	Input	Gate		Vt	Heu2	Vt	Heu2	Vt	Heu2
C880	60	364	63.7	19.2	5.33(3.6x)	13.9	4.89(2.8x)	7.57	4.80(1.6x)
C1355	41	528	81.4	13.2	11.5(1.2x)	12.7	9.20(1.4x)	8.63	7.25(1.2x)
C1908	33	432	74.6	21.0	11.4(1.8x)	13.8	8.73(1.6x)	10.4	6.55(1.6x)
C2670	233	825	146	31.2	11.9(2.6x)	28.1	11.7(2.4x)	11.7	11.0(1.1x)
C3540	50	940	156	59.9	17.1(3.5x)	18.1	14.0(1.3x)	15.9	11.9(1.3x)
C5315	178	1627	283	36.5	25.6(1.4x)	32.3	23.4(1.4x)	23.1	21.7(1.1x)
C6288	32	2470	412	184	95.2(1.9x)	51.4	51.0(1.0x)	46.0	35.7(1.3x)
C7552	207	1994	352	52.5	31.1(1.7x)	33.8	27.9(1.2x)	27.3	26.6(1.0x)
alu64	131	1803	294	86.9	41.8(2.1x)	45.1	31.4(1.4x)	29.7	23.8(1.3x)
i2	201	189	16.5	16.3	1.69(9.6x)	15.6	1.31(12x)	11.5	1.25(9.2x)
i3	132	120	22.8	7.32	2.10(3.5x)	4.77	1.87(2.6x)	2.87	1.75(1.6x)
i4	192	160	17.1	13.3	2.44(5.4x)	11.2	1.89(5.9x)	6.84	1.70(4.0x)
i5	133	198	15.7	26.1	1.79(14x)	5.23	1.60(3.3x)	2.93	1.57(1.9x)
i6	138	390	66.3	11.1	5.26(2.1x)	6.43	5.13(1.3x)	6.43	5.13(1.3x)
i7	199	510	71.2	16.1	6.61(2.4x)	16.1	5.67(2.8x)	8.01	5.61(1.4x)
i8	133	749	135	17.5	10.3(1.7x)	12.2	8.57(1.4x)	10.3	8.53(1.2x)
i9	88	473	71.3	8.5	4.83(1.8x)	7.54	4.84(1.6x)	6.80	4.81(1.4x)
i10	257	1912	324	59.2	27.0(2.2x)	37.9	24.9(1.5x)	26.3	24.4(1.1x)
Avg.					(3.5x)		(2.6x)		(1.9x)

gently constrained optimization while the 50% column is constraint such that the delay falls exactly in the middle between the all low-Vt and all high-Vt circuit delay.

Heuristic 1 is the fastest heuristic but also produces an inferior optimization results. Comparing heuristics 2 and 3, heuristic 2 produces better results for most benchmark circuits.

In Table 3, we show the comparison between the proposed approach and the Vt-assignment only method. The results demonstrate that dramatic improvement in the leakage current can be obtained using the proposed methods, with an average improvement of 3.5X to for the 10% delay constraints. The relative reduction of the proposed method reduces to 1.9X for the 50% delay constraint. This is due to the fact that, as the delay constraint becomes looser, more transistors can be set to high-Vt in both approaches, and the relative advantage of the proposed approach reduces. However, the critical concern for leakage reduction methods is to do so at points with minimal performance loss. Table 3 also shows the results using state assignment only yields higher leakage current than both the proposed approach and the Vt-only approach.

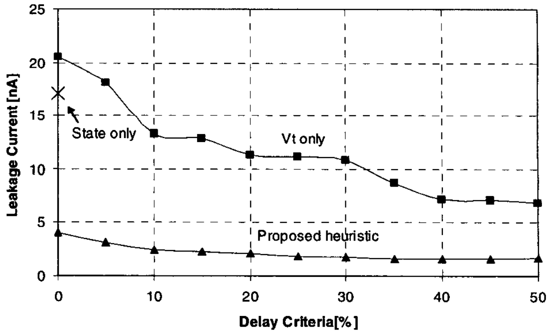

Figure 4. Leakage current comparison for i4

Finally, Figure 4 plots the leakage results for the proposed method and the two traditional methods as a function of the delay constraint for circuit i4. The optimization was performed for a number of delay constraints.

6 Conclusions

In this paper, we have proposed a new approach for standby leakage current minimization under delay constraints. Our approach uses simultaneous state assignment and Vt-assignment. An efficient method for computing the state and Vt assignment leading to the minimum circuit leakage was presented. The proposed approach was shown to reduce the leakage current by a factor of 3.5X compared to previous approaches over different benchmark circuits where the highest gains were obtained for more stringent delay constraints.

Acknowledgements

This research was performed with the support of MARCO research grant 98-DF-660.

References

[1] J.Halter and F.Najm, "A gate-level leakage power reduction method for ultra-low-power CMOS circuits," Proc. CICC, pp. 475-478, 1997.

[2] M.C. Johnson, *et al.*, "Models and algorithms for bounds on leakage in CMOS circuits," IEEE Trans. CAD, pp. 714-725, June 1999.

[3] S.Sirichotiyakul, *et al.*, "Duet: an accurate leakage estimation and optimization tool for dual Vt circuits," IEEE Trans. VLSI, pp. 79-90, April 2002.

[4] M.Ketkar and S.Sapatnekar, "Standby power optimization via transistor sizing and dual threshold voltage assignment," Proc. ICCAD, 2002, pp. 375-378.

[5] V.De, *et al.*, "Techniques for leakage power reduction," in *Design of high-performance microprocessor circuits*, New York: IEEE press, 2001

[6] H.Kriplani, *et al.*, "Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: algorithms, signal correlations, and their resolution," IEEE Trans. CAD, pp998-1012, Aug. 1995.

[7] F. Brglez and H.Fujiwara, "A Neutral Netlist of 10 Combinatorial Benchmark Circuits", Proc. ISCAS, 1985, pp.695-698.

[8] http://www.cbl.ncsu.edu