IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 3, Issue 1 (Sep. – Oct. 2013), PP 48-55
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Leakage Reduction of Scaled CMOS Circuits Using Efficient Control Point Insertion Technique

D. Vijayalakshmi, Dr. P. C. Kishore Raja,
BIT Bangalore ECE, Saveetha University,Chennai,India

Abstract: *Leakage power reduction is extremely important in the design of scaled CMOS logic circuits. The dominant leakage components of such circuits are the sub threshold leakage and the thin-oxide gate leakage. This paper describes an efficient leakage reduction method that considers both these components, and is based on the selective insertion of control points. The selection is based on the leakage reduction potential and the delay insensitivity of the candidate gates. Simulations on the ISCAS85 benchmark circuits show that this method results in 67% leakage reduction with no speed degradation when control points are added to 93% of the gates compared to the leakage of the baseline circuit whose inputs have been subjected to the minimum leakage vector.*
Index Terms: *Control point insertion, leakage current reduction, leakage sensitivity (LS), low power design, minimum leakage vector (MLV).*

I. Introduction

TECHNOLOGY scaling enables us to integrate huge number of transistors on chip for higher performance. This comes at the price of increase in both static and Dynamic power consumption. It is predicted that the leakage will increase 7.5 fold and the total power consumption will Increase five fold for every new microprocessor-chip generation [1].Thus, for scaled technologies, leakage power reduction is an essential design component.

There are different mechanisms that contribute to leakage power. These include subthreshold leakage, thin-oxide gate leakage, band-to-band-tunneling (BTBT) leakage, etc. [2]. The existing work on leakage power reduction is aimed at reducing the subthreshold component of leakage. These include techniques based on: 1) identifying and employing the minimum leakage vector (MLV) [3], [4], and control point insertion on the MLV circuit to reduce leakage further [4]; 2) introducing multiple threshold voltage CMOS (MTCMOS) for a cluster of transistors to gate the power supply OFF [5]; 3) using dual V_{TH} to balance performance and leakage reduction [6]; 4) inserting transistors in a stack to reduce leakage [7]. However, with technology scaling especially when gate oxide thickness is less than 20 A , thin-oxide gate leakage is an important component of the total leakage power. A transistor-level technique to reduce leakage, including thin-oxide gate leakage, was proposed in [8]. This technique uses pin re-ordering to reduce the gate

TABLE I
LEAKAGE OF 3-INPUT NAND GATE.

Input State	Subthreshold leakage (nA)	Gate leakage (nA)	Total Leakage (nA)
000	0.49	6.58	7.07
001	0.81	19.68	21.49
010	0.81	6.79	7.60
011	2.68	34.78	37.46
100	0.81	3.15	3.96
101	2.68	16.8	19.48
110	2.68	1.84	4.52
111	16.85	45.3	62.15

oxide leakage. Another technique for gate leakage reduction that was recently proposed uses efficient transistor stacking [9]. This paper describes an efficient technique to reduce the leakage power (subthreshold and thin-oxide gate leakage) in CMOS circuits. The method is based on gate level restructuring and selective insertion of control points. The control points are selected based on their leakage reduction potential. The main contributions are as follows.

1) Development of a heuristic algorithm for control point insertion that maximizes leakage reduction when *delay is not a constraint*. Simulation results on ISCAS85 benchmarks show that the average leakage reduction is25 % when control points are added to 20% of the gates, compared to the baseline circuit whose inputs are the MLV.

2) Development of a heuristic algorithm for control point insertion that maximizes leakage reduction *given a delay constraint*. For ISCAS85 benchmark circuits, control points can be added to about 93% of gates without any speed degradation resulting in an average leakage reduction of67 %.
3) Development of an efficient algorithm that select the gates from a pre-select group of gates for control point insertion. This algorithm has comparable performance, considers only a subset of candidate gates and is considerably faster.
The rest of the paper is organized as follows. Section II introduces the control point insertion based technique for leakage power reduction. Section III shows the simulation results for the leakage reduction methods. Section IV concludes the paper.

II. Leakage Control Using Control Point Insertion

The leakage power consumption during the standby mode can be significantly reduced if the inputs to the circuit are chosen carefully. Table I describes the leakage of a three input NAND gate for different input combinations for a technology with feature size 65 nm and 17 A gate-oxide thickness. The leakage corresponding to input vector "100" is 16 times smaller than that corresponding to vector "111."Thus, choosing the input corresponding to minimum leakage (referred to in the literature as the MLV) is the first step toward reducing standby power. For large circuits, the variance in the leakage energy for different input combinations is not very large. Application of the MLV thus may not result in significant reduction in leakage energy. Greater reduction can be achieved if the state of the gates deep in the circuit can be manipulated. One way of achieving it is by control point insertion [4]. The method in [4] uses a SAT solver to find the MLV. For a specified delay constraint, it adds maximum number of control points to a randomly selected set of gates.

Our approach, based on a single pass algorithm, is significantly simple compared to the SAT-based method. It adds control points to a selective number of gates; the gate selection is based on leakage reduction potential and delay insensitivity. Furthermore, our approach is scalable. The input to the control point insertion method is a circuit with MLV applied to its input, referred to the *baseline circuit*. MLV is computed using the method in [10]. Since we consider both subthreshold leakage which depends on the number of OFF transistors in the stack and thin-oxide gate leakage which depends on the position of the OFF transistor in the stack [8], MLV is likely to be unique.

In the rest of the section, we discuss the proposed method that employs a gate level strategy and a circuit level strategy to reduce the leakage efficiently. At the gate level, the structure of the gate is modified (when a control point is added) in a way that maximizes leakage reduction with minimum delay penalty. At the circuit level, gates are selected based on their leakage and delay sensitivity.

A. Gate-Level Strategy

Since the subthreshold current can be reduced if the drain to source and gate to source bias is reduced, adding more "OFF" transistors in a transistor stack can significantly reduce the subthreshold current. The gate tunneling current, on the other hand, depends on the gate to source/drain bias, and this bias can be reduced if the transistor near to the supply rail is made OFF. So stacking helps to reduce both the subthreshold leakage and the thin-oxide gate leakage [8], [9]. However, stacking more number of transistors results in additional delay. So, stacking is a good solution for leakage reduction at the expense of speed degradation.

To reduce both the subthreshold leakage and gate-tunneling leakage current, we need to add the transistor (called "controltransistor") in a stack near the supply rail. Fig. 1 shows three options of adding control transistors in a gate. Methods (b) and (c) in Fig. 1 were first proposed for leakage control in [11] and have been used for control point insertion in [4] based on whether the gate structure is a NAND or a NOR. Since in CMOS logic circuits, the leakage current for nMOS devices is much greater than those of pMOS devices, an effective solution for leakage reduction is to add an nMOS transistor as a control transistor at the bottom of the nMOS transistor stack (near the ground) as
shown in Fig. 1(a) and (b)

Table II shows the average leakage and delay of a 2-input NAND gate and a 2-input NOR gate before and after control point insertion using the methods (a), (b), and (c) in Fig. 1. Method (a)

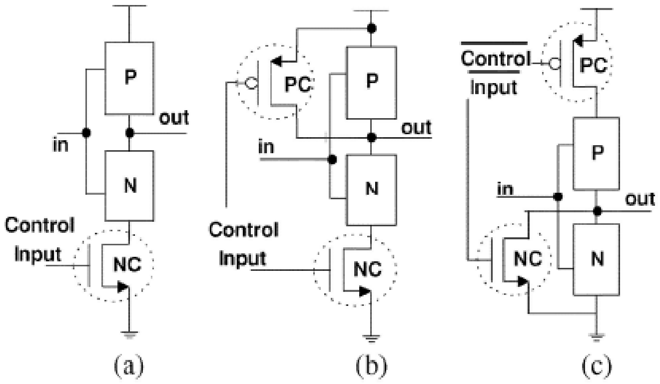

Fig. 1.Illustration of control point insertion in a gate

TABLE II
Average Leakage And Delay Of 2-Input Nand And Nor Gate Without And With Control Point Insertion

Example circuit	Leakage (nA)				Delay (ps)			
	w/o CP	ckt (a)	ckt (b)	ckt (c)	w/o CP	ckt (a)	ckt (b)	ckt (c)
NAND2	18.94	5.14	5.79	31.11	36.51	41.01	46.6	42.29
NOR2	22.07	5.10	10.18	28.74	38.1	40.87	45.49	36.95

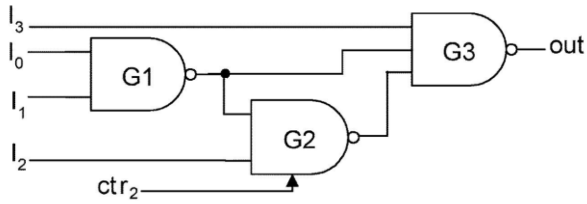

Fig. 2.Example circuit to demonstrate the calculation of LS.

achieves highest leakage reduction with minimum speed degradation. However this method is not used since during standby, if all inputs of the gate are high, the output of the gate could cause the pMOS and nMOS transistor of the next stage to be simultaneously ON, resulting in large short circuit current. Method (c) has significant leakage due to the higher gate leakage of the nMOS and is not used either. The insertion method in Fig. 1(b) offers a good compromise between leakage reduction and delay increase, and is used throughout this work.

B. Circuit Level Strategy

Control points cannot be added to all the gates in the circuit
because of the increased area and delay penalty. Next we describe a method of choosing gates that would maximize leakage reduction.
Let "leakage sensitivity (LS)" of a gate be defined as
the amount of leakage reduction due to the insertion of a control point to gate i $LS_i = \sum$. leakage after control point insertion in gate i -∑leakage before control point insertion in gate i
Consider the example logic circuit shown in Fig. 2. Suppose, MLV for this circuit is $\{I_0I_1I_2I_3\} = \{0110\}$ The leakage currents of three gates are— L_{gate1} =4.77nA, L_{gate2} =41.45 nA, L_{gate3}=7.6nA If we add a control input to gate2, the leakage currents are L_{gate1} =4.77nA, L_{gate2} =4.52 nA, L_{gate3}=4.52nA The LS of gate2= {4.77+41.45+7.6 }-{4.77+4.52+4.52}=40.01nA

C. Leakage Without Delay Constraint

This is the baseline algorithm that chooses the candidate gates in a greedy manner. The algorithm is iterative: in each iteration, the gate with highest LS is chosen. The number of iterations depends on the desired leakage energy reduction. *ALG basic:*
1) Compute LS and make a priority queue based on LS
2) Until the required leakage performance is satisfied, do
a) Add a control point to the gate at the head of the queue

b) Update the priority queue.

This algorithm has been applied to the example circuit with devices having 17 A gate oxide (shown in Fig. 2). In the first iteration,LS1=8 nA,LS2=40.014 nA, and LS3=0.08 nA. Gate 2 is chosen since it has the highest LS. In the second iteration, LS1=40.82nA, and LS3=40.73 nA and gate 1 is chosen and so on. If the number of the gates in a circuit is N, the desired gate is chosen from N gates and the complexity of the algorithm is$\Theta(N^2)$

D. Leakage Under Delay Constraint

In this section, we describe an algorithm for leakage reduction given a delay bound. The idea is not to add a control point to the gate at the head of the priority queue, but to add it to the first one in the priority queue that satisfies a specific "incremental delay bound."The incremental delay bound is a function of the overall delay bound and the number of gates in the critical path. After each assignment both LS and delay (due to addition of control point) has to be calculated. The complexity of this algorithm is almost twice as that of ALG basic. The modified algorithm is given below.

ALG delay:

1) Compute LS and make a priority queue based on LS
2) Set an incremental delay bound
3) Until the required leakage performance is satisfied, do
a) Until the incremental delay bound is satisfied, do
i) Determine delay if a control point is added to gate on the priority queue
b) Add a control point to the gate
c) Increment the delay bound if the delay is increased
d) Update the priority queue.

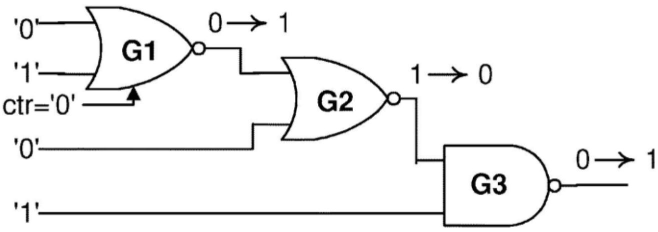

Fig. 3. Example circuit to demonstrate the effect on next level of gates for the addition of control point to a gate f111g.

E. Efficient Selection of Candidate Gates

In order to reduce the time required for the selection of candidate gates, LS is calculated for a *select* group of gates. These include:

i) gates that have a high leakage due to the input settings;
ii) gates at a lower level (that are closer to the primary input) whose output states change after inserting control points;
iii) larger gates that have more number of transistors; iv) gates that have large fan-out.

Case i) Consider a NAND3 gate inside a large circuit whose inputs have been forced {011}to{111} or after application of MLV. This NAND3 gate is very leaky and is a candidate for control point insertion. In general, gates that have all inputs in logic high state, and gates that have an OFF transistor in the stack near to the output contribute greater leakage.

Case ii) The gate whose output will change after insertion of the control point is likely to have a larger effect on the leakage of the next level of gates. Consider the example logic circuit shown in Fig. 3 which is a part of large circuit whose MLV sets the inputs $\{I_0I_1I_2I_3\}$ to {0101}. Inserting the control input to gate G1 changes its output from logic "0" to logic "1."The change in the output of G1 affects gates G2 ("1" to "0") and G3 ("0" to "1") and causes their leakage to decrease by(3.71+36.68) nA. Thus, the LS1=42.6nA of G1 is highLS2=3.83 nA, compared to LS3=36.93nA .

Case iii) A large gate with multiple transistors is also a prime candidate since inserting an OFF control transistor to a large stack results in higher reduction in leakage. For instance, adding a control transistor to a 3-input NAND gate whose inputs are high results in leakage reduction 57.76 nA, compared to leakage reduction of 36.93 nA for a 2-input NAND gate.

Case iv) Adding a control point to a gate with large fan out affects more number of gates in the next level, and also causes an increase in the LS.

III. Results

The proposed leakage reduction methods were implemented and tested for ISCAS85 benchmark circuits. Nine benchmark circuits were used with the largest having 3837 gates. Each of the circuits was optimized and synthesized using Berkeley SIS tools and mapped into technology library for feature size 65 nm and 17 A gate oxide thickness. Inverter, NAND and NOR gates were used from the cell library; the fan-in was at most three and the fan-out was at most five. For leakage and delay calculation, each type of gate was pre-characterized for all possible input combinations using the Berkeley Predictive SPICE model (BPSIM4) in HSPICE simulation. The leakage of all gates were summed to calculate the overall leakage. The delays were calculated using the "transition mode delay" model [12]. This is distinctly different from the delay calculation method used in [4], which measures topological longest path delay. The delays of all possible paths from input to output node are determined and the maximum delay is taken as the delay of the circuit.

A. Leakage Without Delay Constraint

Fig. 4 shows the average leakage reduction for all ISCAS85benchmark circuits after control point addition using two methods: one in which the gates are randomly selected (using MATLAB random number generator) and one in which gates are selected based on the LS (Section II-C). Note that with random selection of gates, leakage reduction is significantly lower when control points are added to 0–60% of the gates.
In fact, for the first 0%–20%, leakage is barely reduced with random selection and reduced by ~25% with LS–based selection. If control points are added to all gates of the circuit, the average leakage reduction is ~74%, which is the same for both methods. Fig. 5 shows the average delay penalties for all benchmark circuits due to the addition of control points. Note that the average delay overhead when control points are inserted in 100% of the gates is % if delay constraint is not considered.

B. Leakage Under Delay Constraint—Complete Set

Next, we describe the leakage behavior of the benchmark circuits *under delay constraint* (Section II-D). Fig. 6 shows the maximum percentage of gates in which the control points can be added without any speed degradation. We obtain this by simulating each benchmark a number of times using different input combinations to activate different critical paths and identifying in each case the number of gates that are not on the critical path. Adding control points to the gates that are not on the critical path results in significant leakage reduction without causing any speed degradation. Fig. 7 shows the average leakage reduction for various benchmarks for different delay tolerances. Note that for control point insertion in 0%–25% gates, there is little difference in leakage reduction for the three delay tolerances. The difference increases as more control points are added to the gates. The largest leakage reduction is found if the delay is not considered at all. If the delay tolerance is higher, then in each iteration during gate selection, there is more flexibility to select the gate that has higher LS.

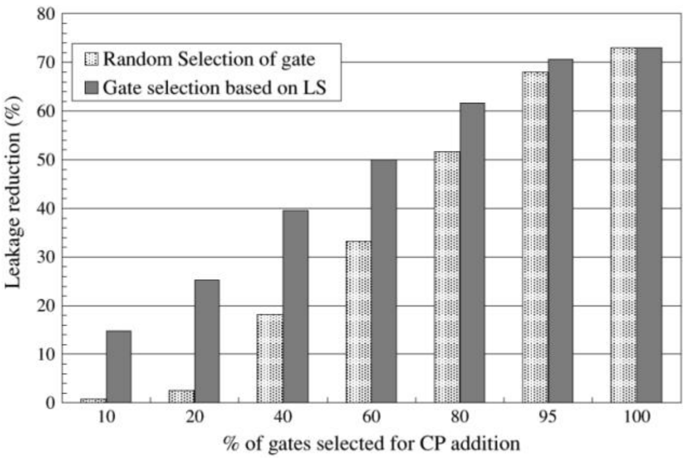

Fig. 4. Leakage reduction after adding control points to gates selected randomly and gates selected based on LS using ALG basic.

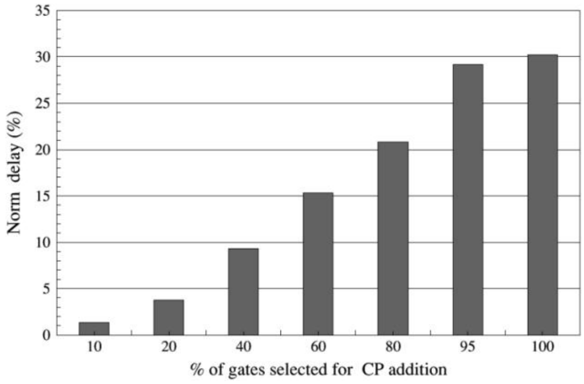

Fig. 5. Delay penalties after adding control points to the gates selected using ALG basic.

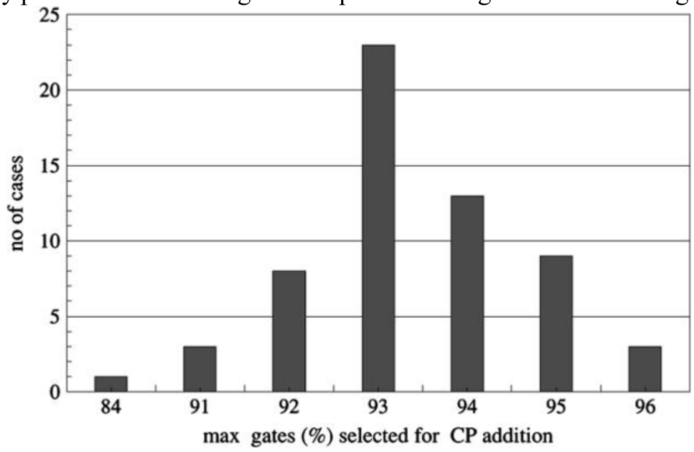

Fig. 6. Maximum number of gates for control point addition with 0% speed degradation.

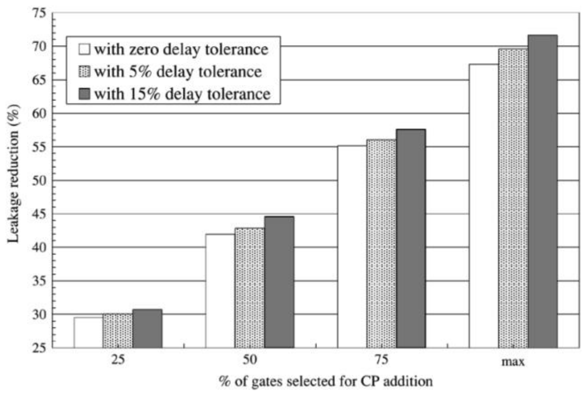

Fig. 7. Average leakage reduction for all benchmarks for different delay tolerances

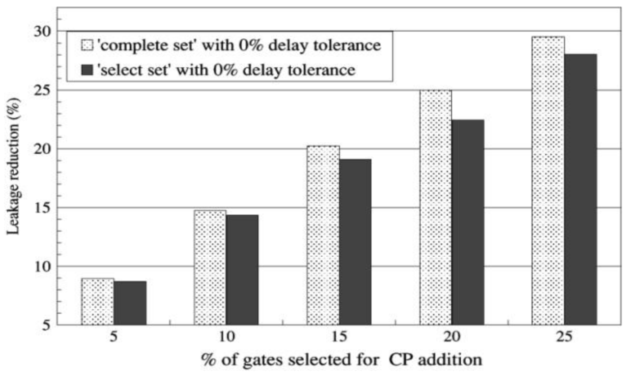

Fig. 8. Leakage reduction for control point insertion using “complete set” and “select set.”

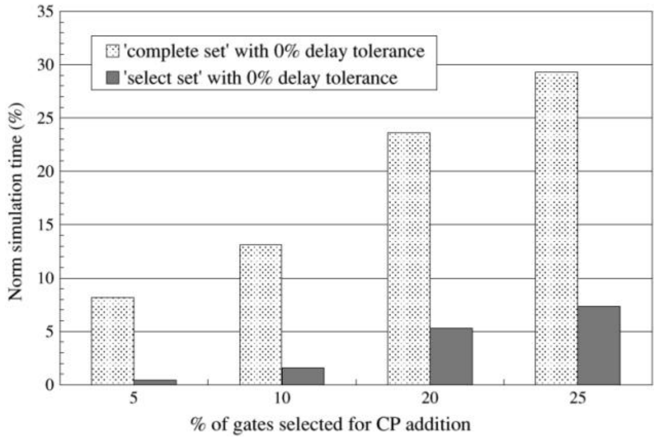

Fig. 9. Comparison of simulation time between "complete set" and "select set."

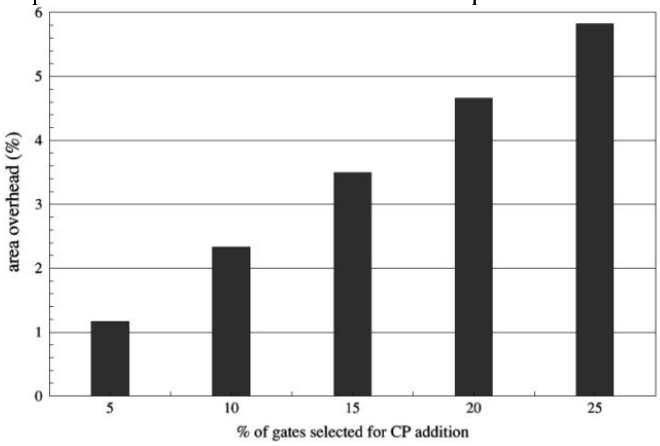

Fig. 10. Average area overhead for control point insertion.

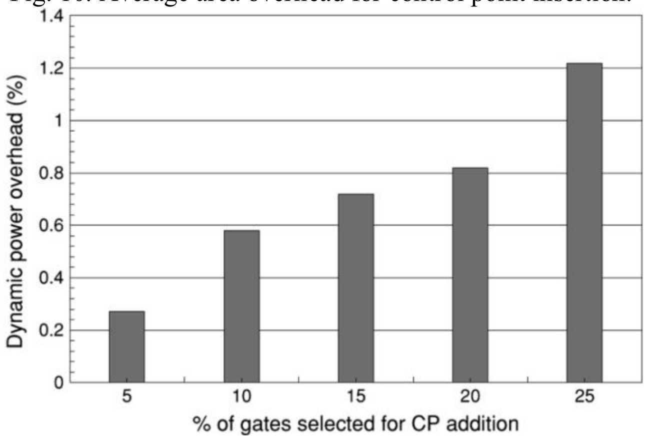

Fig. 11. Average active power overhead for control point insertion.

C. Leakage With Delay Constraint—Select Set

Next, we present the results based on efficient selection of candidate gates described in Section II-E. Fig. 8 shows the average leakage reduction of all benchmark circuits obtained using ALG delay (referred to as "complete set" in Section II-D) and

the efficient selection (referred to as "select set" in Section II-E) with zero delay tolerance. For 0%–10% of the gates, the two methods give comparable results. If control points are added to 25% gates, "complete set" and "select set" result in average leakage reduction of 29.5% and 28%, respectively, with 0% delay tolerance. In comparison, the SAT-solver-based technique in [4] results in an average leakage reduction of less than 20% for 0% delay tolerance on MCNC91 benchmark suite. Efficient selection method results in LS being calculated for only 23%–38% of the gates in the benchmark circuits. Thus, using the "select set" method results in significant time being saved during simulation. For instance, for control point addition to 25% gates, almost 75% time is saved if "select set" is used compared to if "complete set" is used. Fig. 9 compares the average simulation times between "complete set" and "select set."All values are normalized with respect to the simulation time needed for selection of entire set of gates.

D. Overheads

1) Area: Since the standard cell for feature size of 65-nmgate length is not available yet, area is calculated using the dimensions of pMOS and nMOS devices that are used for the design and analysis. The area overhead was obtained by summing up transistor widths and the area due to routing has not been considered.

Fig. 10 shows the average area overhead for all bench-marks after control point insertion. The average area overhead when control points are inserted in 25% of the gates is %.

2) Dynamic Power: The dynamic power overhead is calculated using HSPICE by applying a large set of randomly generated inputs to the benchmark circuits before and after the control point insertion. Fig. 11 shows the average dynamic power overhead for all benchmarks. The average dynamic power overhead when control points are inserted in 25% of the gates is %.

3) Control: The control overhead of the proposed method is minimal. The standby signal that is generated by the processor is routed to all the control inputs. Thus, no additional signals are required, and there is no power and area overhead except that of routing the control signal.

IV. Conclusion

In this paper, we proposed an efficient leakage control method based on selective insertion of control points to a circuit whose inputs have been subjected to MLV. Simulation results on ISCAS85 benchmarks show that for 0% delay tolerance, this method results in an average leakage reduction of 29.5 % when control points are added to 25% of the gates and 67 % when control points are added to 93% of the gates.The penalty, in each case, is the area due to additional control transistors and routing, and dynamic power.

Acknowledgment

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Borker, "Design challenges of technology scaling," *IEEE Micro*, vol. 9, no. 4, pp. 23–29, Jul.–Aug. 1999.

[2] S.23 Mukhopadhyay, A. Raychowdhury, and K. Roy, "Accurate estimation of total leakage current in scaled CMOS logic circuits based on compact current modeling," in *Proc. Design Automation Conf.*, Jun. 2003, pp. 169–174.

[3] M. Johnson, D. Somasekhar, and K. Roy, "Models and algorithms for bounds on leakage in CMOS circuits," *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.*, vol. 18, no. 6, pp. 714–725, Jun. 1999.

[4] A. Abdollahi, F. Fallah, and M. Pedram, "Runtime mechanisms for leakage current reduction in CMOS VLSI circuits," in *Proc. ISLPED*, Aug. 2002, pp. 213–218.

[5] J. Kao, A. Chandrakasan, and D. Antioniadis, "Transistor sizing issues and tool for multi-threshold CMOS technology," in *Proc. Design Automation Conf.*, 1997, pp. 409–414.

[6] L. Wei, Z. Chen, M. Johnson, K. Roy, and V. De, "Design and optimization of low voltage high performance dual threshold CMOS circuits," in *Proc. 35th Design Automation Conf.*, Jun. 1998, pp. 489–494.

[7] S. Narendra, S. Borker, V. De, D. Antoniadis, and A. P. Chandrakasan, "Scaling of stacked effect and its application for leakage reductions," in *Proc. ISLPED*, 2001, pp. 195–200.

[8] D. Lee, W. Kwong, D. Blaauw, and D. Sylvester, "Analysis and minimization techniques for total leakage considering gate oxide leakage," in *Proc. Design Automation Conf.*, Jun. 2003, pp. 175–180.

[9] S. Mukhopadhyay, C. Neau, R. T. Cakici, A. Agarwal, C. H. Kim, and K. Roy, "Gate leakage reduction for scaled devices using transistor stacking," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 4, pp. 716–730, Aug. 2003.

[10] K. Chopra, S. B. K. Vrudhula, and S. Bhardwaj, "Efficient algorithms for identifying the minimum leakage states in CMOS combinational logic," in *Proc. 17th Int. Conf. VLSI Design*, Jan. 2004, pp. 240–245.

[11] M. Johnson, D. Somasekhar, and K. Roy, "Leakage control with efficient use of transistor stacks in single threshold CMOS," in *Proc. 36th Design Automation Conf.*, 1999, pp. 442–445.

[12] S. Devadas, K. Keutzer, and S. Malik and A. Wang, "Certified timing verification and the transition delay of a logic circuit," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 2, no. 3, pp. 333–342, Sep. 1994.