

Generation of Minimal Leakage Input Vectors with Constrained NBTI Degradation

Pramod Subramanyan¹ Ram Rakesh Jangir²
Jaynarayan Tudu¹ Erik Larsson³ Virendra Singh¹

¹Indian Institute of Science, Bangalore, India

²Government Polytechnic, Hisar, India

³Linköping University, Sweden

7th East-West Design and Test Conference
20 September, 2009

Outline

1 Introduction

- Related Work

2 Problem Formulation

- Overview
- Formulation of Objective
- Formulation of Gate I/O Constraints
- Formulation of Path Delay Constraints
- Identifying Potentially Critical Paths
- Linearization
- Summary

3 Results

- Comparison of PCP Selection Algorithms
- NBTI and Leakage Results

4 Conclusion

Outline

1 Introduction

- Related Work

2 Problem Formulation

- Overview
- Formulation of Objective
- Formulation of Gate I/O Constraints
- Formulation of Path Delay Constraints
- Identifying Potentially Critical Paths
- Linearization
- Summary

3 Results

- Comparison of PCP Selection Algorithms
- NBTI and Leakage Results

4 Conclusion

Challenges in Deep Submicron Technologies

Moore's law is expected to apply for the next 10 years giving us smaller and faster devices with reduced power. But, there is a downside:

- Smaller devices make ICs more susceptible to transient faults
- Wearout and drift effects more prominent
 - e.g., **negative bias temperature instability (NBTI)**, electromigration (EM), gate oxide integrity (GOI)
- Increased **leakage power**
- Increased process variations

NBTI

Negative Bias Temperature Instability (NBTI)

NBTI is a degradation phenomenon in digital circuits, caused by the dissociation of $Si - H$ bonds along the silicon-oxide interface when a PMOS transistor is reverse-biased in inversion. This dissociation generates interface traps which lead to an increase in the threshold voltage of PMOS transistors.

Characteristics of NBTI:

- affects only **PMOS** transistors
- leads to **lower f_{max}** or **higher V_{DD}**
- caused by the application of **reverse bias** (i.e. input 0)
- **recovery** occurs when reverse bias is removed

Input Vector Control (IVC)

Idea

When the circuit is in **inactive/sleep mode**, apply **carefully chosen input vectors** that minimize NBTI degradation. Input vector should be such that most of the **transistors on the critical paths are in recovery mode**.

- IVC can also be used to minimize leakage power
- IV's that minimize leakage power **do not** minimize degradation

This paper presents a **systematic methodology to co-optimize leakage and degradation** minimization.

Related Work

- Gao and Hayes [ICCAD 2004] proposed an ILP formulation for selecting input vectors that minimize leakage power only. We propose a similar formulation that also takes into account NBTI.
- Wang et. al [DATE 2007, ISQED 2009] proposed a search algorithms loosely based on GAs that generates input vectors that minimize leakage power and NBTI. Their algorithms cannot guarantee optimality, unlike our proposal.

Outline

- 1 Introduction
 - Related Work
- 2 Problem Formulation
 - Overview
 - Formulation of Objective
 - Formulation of Gate I/O Constraints
 - Formulation of Path Delay Constraints
 - Identifying Potentially Critical Paths
 - Linearization
 - Summary
- 3 Results
 - Comparison of PCP Selection Algorithms
 - NBTI and Leakage Results
- 4 Conclusion

Overview of Formulation

First we obtain a **nonlinear integer program** as shown below. Then we show how this can be **linearized** to obtain an ILP.

High Level Formulation of IP

variables: inputs of each gate in the circuit

objective: minimize leakage power

constraints:

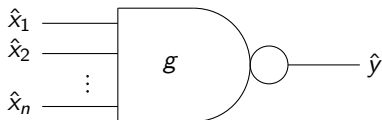
- 1 for each gate g :
 - gate output $\hat{y} = f_g(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$
- 2 for each *potentially critical path* P :
 - degraded path length of $P \leq L$

f_g determines the output of gate g given input values \hat{x}_i .

L is **degradation limit**, the maximum delay after degradation.

L is set by the designer based on frequency guardband.

Formulation of Objective Function: 1



Let $P_g(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$ be the leakage power dissipated by gate g on inputs $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$. In general, P_g can have 2^n different values. Given a gate g and the corresponding values of P_g , we want to express P_g as a **closed-form function** of $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$.

Formulation of Objective Function: 2

Definitions

- 1 Let $I = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$.
- 2 Let $\rho_n = \{S_0, S_1, \dots, S_{k-1}\}$ be the set of all subsets of I .
 - note that $|\rho_n| = k = 2^n$
 - e.g. $\rho_2 = \{\Phi, \{\hat{x}_1\}, \{\hat{x}_2\}, \{\hat{x}_1, \hat{x}_2\}\}$
- 3 Let $\pi(S_i)$ as the product of the elements of S_i .
 - we set $\pi(\Phi) = 1$.
 - $\pi(\{\hat{x}_1, \hat{x}_2\}) = \hat{x}_1 \hat{x}_2$

We can now write:

$$P_g(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) = \sum_{i=0}^{k-1} c_i \pi(S_i) \quad (1)$$

Formulation of Objective Function: 3

$$P_g(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) = \sum_{i=0}^{k-1} c_i \pi(S_i) \quad (2)$$

Equation (2) has 2^n coefficients c_i . There are 2^n values of P_g corresponding to each input of the gate. Substituting each of the values of $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$ and P_g we obtain 2^n equations which can be solved to obtain the values of c_i .

Example: 3-input NAND gate

Input	Leakage (pW)	Input	Leakage (pW)
000	30.1	100	55.1
001	54.9	101	259.2
010	54.7	110	309.8
011	249.1	111	703.3

$$\begin{aligned}
 P(\hat{x}_1, \hat{x}_2, \hat{x}_3) = & 30.1 + 24.8 \cdot \hat{x}_1 + 24.6 \cdot \hat{x}_2 + 25 \cdot \hat{x}_3 \\
 & + 169.6 \cdot \hat{x}_1 \hat{x}_2 + 179.3 \cdot \hat{x}_1 \hat{x}_3 + 230.1 \cdot \hat{x}_2 \hat{x}_3 \\
 & + 19.8 \cdot \hat{x}_1 \hat{x}_2 \hat{x}_3
 \end{aligned}$$

Formulation of Gate I/O Constraints

Need for gate I/O constraints:

- Inputs to all gates are variables of the IP
- But not all inputs can take all values
- Intermediate gate inputs depend on the inputs of the gate from which they are sourced

I/O Constraints relate the gate inputs to the outputs so that variable values assigned by the solver are consistent with logic functions of gates

Examples:

- 1 2-input AND: $y = x_1 x_2$
- 2 3-input NAND: $y = 1 - x_1 x_2 x_3$
- 3 2-input NOR: $y = 1 - x_1 - x_2 + x_1 x_2$

Determining Degraded Gate Delay: 1

To calculate maximum degraded delay of critical paths, we need the maximum degraded delay of each gate. Degraded gate delay depends on:

- 1 Undegraded (nominal) delay of the gate
- 2 Probability that gate inputs are stressed in *normal* mode
- 3 Value of gate inputs during *sleep* mode
- 4 Probability that circuit is in *sleep/inactive* mode

In this paper we tackle the problem of selecting input vectors for use during sleep/inactive mode so that degradation is reduced. (*i.e.*, we try to control 3)

Determining Degraded Gate Delay: 2

Degraded gate delay \mathcal{D}_g for gate g with inputs $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$ is:

$$\mathcal{D}_g(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) = \max_i \left\{ \Delta_d(p_s(x_i), t) \right\} \cdot d_g \quad (4)$$

where:

- $\Delta_d(p_s, t)$ is the factor by which gate delay is increased for stress probability p_s over time t
- $p_s(x_i)$ is the effective probability of stress for input x_i
- d_g is the undegraded delay of gate g

Given the values of \mathcal{D}_g is for each input value, we use the same method as for leakage power to derive a closed-form expression for \mathcal{D}_g in terms of \hat{x}_j .

Path Delay Constraints

We want to ensure that no path in the circuit degrades so that its delay is greater than L , the **degradation limit**. *i.e.*, for each **potentially critical path** $P = (g_1, g_2, \dots, g_k)$, $\sum_{i=1}^k \mathcal{D}_{g_i} \leq L$.

Potentially Critical Path

A potentially critical path (PCP) is any circuit path which could, depending on the input vector selected for use during sleep/standby mode, degrade such that its delay is greater than the degradation limit.

[Wang et al., ICCAD 2007]

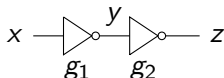
Identifying Potentially Critical Paths: Algorithm 1

Algorithm 1

- Assume all inputs are stressed in sleep mode
- Now stress probability depends only on stress in normal mode
- Worst case degraded delay determined by most-stressed input
- Degraded path delay is sum of worst case degraded gate delays
- PCPs are all paths which have degraded path delay $\geq L$

Identifying Potentially Critical Paths: Algorithm 2

Algorithm 1 overestimates worst-case degraded delays. Consider the following simple circuit.



Algorithm 1 will calculate worst-case degraded delay assuming both inputs are stressed, *i.e.*, $x = y = 0$ during sleep mode. This can never occur.

Algorithm 2 improves over algorithm 1 by tracking the worst case degraded delay separately for each output value (0 and 1). Thus, it reduces the number of PCPs that have to be considered.

Linearization Constraints

Constraints derived so far are non-linear. We convert these to linear form by introducing additional variables and constraints to make them linear. For example consider:

$$z = 1 - xy \quad (5)$$

To linearize (5) we introduce a new variable p representing the product of x and y and replace the constraint (5) with the constraints in (6).

$$\begin{aligned} z &= 1 - p \\ p &\leq x \\ p &\leq y \\ x + y - p &\leq 1 \end{aligned} \quad (6)$$

Formulation of ILP Summary

variables: inputs of all gates + linearization variables

objective: minimize leakage power

constraints:

- 1 gate I/O constraint:
 - for each gate g gate output $\hat{y} = f_g(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$
- 2 path length length constraint:
 - for each PCP P : degraded path length of $P \leq L$
- 3 linearization constraints:
 - introduced to represent product variables

Outline

- 1 Introduction
 - Related Work
- 2 Problem Formulation
 - Overview
 - Formulation of Objective
 - Formulation of Gate I/O Constraints
 - Formulation of Path Delay Constraints
 - Identifying Potentially Critical Paths
 - Linearization
 - Summary
- 3 Results
 - Comparison of PCP Selection Algorithms
 - NBTI and Leakage Results
- 4 Conclusion

Number of PCPs

Circuit	Algorithm 2	Algorithm 1	Wang et al. $p = 12\%$	Wang et al. $p = 15\%$
c1355	196608	786432	2523136	2779136
c1908	128	1077	58979	113563
c2670	168	1864	43832	69292
c3540	414	3468	368292	913898
c432	29430	74844	404946	432702
c499	196608	557056	2523136	2686976
c5315	164	1384	74616	131878
c7552	50	964	41214	63174
c880	12	24	196	329

Number of PCPs selected by our algorithms as compared to the work of Wang et al., DAC 2007.

NBTI and Leakage Power Results

Circuit	RND leakage max (μW)	RND leakage min (μW)	ILP leakage (μW)	ILP degradation	Worst case
c880	0.32	0.29	0.29	10%	16%
c1908	0.71	0.68	0.68	10%	16%
c3540	1.55	1.50	1.51	11%	16%
c432	0.14	0.12	0.12	11%	17%

Results of experiments on ISCAS '85 benchmarks. Our algorithm's results are shown under the "ILP" column. For comparison we show results from a simple random search algorithm, labelled RND.

Outline

- 1 Introduction
 - Related Work
- 2 Problem Formulation
 - Overview
 - Formulation of Objective
 - Formulation of Gate I/O Constraints
 - Formulation of Path Delay Constraints
 - Identifying Potentially Critical Paths
 - Linearization
 - Summary
- 3 Results
 - Comparison of PCP Selection Algorithms
 - NBTI and Leakage Results
- 4 Conclusion

Conclusion and Future Work

Conclusions:

- Proposed an optimal ILP formulation for generating input vectors that minimize NBTI degradation and leakage power
- Proposed two heuristic algorithms to select potentially critical paths that performs better than previous work.
- Experiments show that NBTI degradation can be reduced by 5.75%, while keeping leakage power close to minimum.

Future Work:

- Approximate methods for ILP to reduce solution time
- Take into account stacking effect

Thank You!