Available online at www.sciencedirect.com
SCIENCE DIRECT®
Current Applied Physics 5 (2005) 373–380
Current Applied Physics
Physics, Chemistry and Materials Science
www.elsevier.com/locate/cap

Unified low power optimization algorithm by gate freezing, gate sizing and buffer insertion ☆

Hyungwoo Lee *, Hakgun Shin, Juho Kim

Department of Computer Science, Sogang University, C.P.O. Box 1142, Seoul, South Korea

Received 9 August 2003; accepted 12 March 2004
Available online 2 July 2004

Abstract

One of the major factors contributing to the power dissipation in CMOS digital circuits is the switching activity. Many of such switching activities include spurious pulses, called glitches. In this paper, we propose a new method of glitch reduction by gate freezing, gate sizing, and buffer insertion. The proposed method unifies gate freezing, gate sizing, and buffer insertion into a single optimization process to maximize the glitch reduction. The effectiveness of our method is verified experimentally using LGSynth91 benchmark circuits with a 0.5 μm standard cell library. Our optimization method reduces glitches by 65.64% and the power by 31.03% on average.

PACS: 84.30.Bv
Keywords: Power optimization; Glitch; Gate freezing; Gate sizing; Buffer insertion

1. Introduction

Low power circuit design is one of the major design constraints due to the increased demand in the portable consumer electronics. For CMOS digital circuits, the dynamic power dissipation is the dominant source of the power dissipation. A signal transition can be classified into two categories; a functional transition and a glitch. A signal might go through several state changes before it reaches its steady state within a clock cycle. These spurious transitions are called glitches. Since the glitch power can occupy 20–70% [1] of total power dissipation, glitch should be eliminated for low power design.

In [2,3], gate freezing is introduced for glitch power minimization in circuits. The technique relies on the availability of modified library cells, so called F-gates. None of the previous approaches explicitly reduce the power dissipation by applying gate freezing, gate sizing and buffer insertion techniques in a single process concurrently to reduce glitches. Our optimization method unifies gate sizing, buffer insertion, and gate freezing into a single optimization process.

Gate sizing is an effective method for power optimization [4–7]. These techniques optimize the amount of capacitive load by down sizing transistors or gates considering the given delay constraints [4–6]. However, the sized gates or transistors might break path balancing and cause additional glitches. The power optimization method considering glitch by gate sizing is proposed in [7]. This approach selects gates according to the amount of power reduction achieved by glitch reduction. It is basically a gate-sizing algorithm that utilizes perturbations for escaping a bad local solution.

In addition to gate freezing, buffer insertion is a well-known method to optimize power and delay of circuits. In [8,9], buffer insertion is employed to increase the driving capability of a node that drives a large capacitive load. However, these approaches did not consider the glitches caused by signals having different arrival times. Gate sizing method is combined with buffer insertion using linear programming in [10]. But it may suffer from too much computation time.

Gate sizing algorithm considering glitch has been presented in [7]. This approach selects gates according to the amount of power reduction achieved by glitch

☆ Original version was presented at the 10th Korean Conference on Semiconductors, Seoul, Korea, 27–28 February 2003.

* Corresponding author. Fax: +82-2-706-3997.
E-mail address: hwlee@sogang.ac.kr (H. Lee).

1567-1739/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.cap.2004.03.003

reduction and resizes them. It is novel method for glitch reduction, which utilizes perturbations for escaping a bad local solution. But it is only a gate sizing algorithm and better result can be achieved when gate sizing and buffer insertion are combined. Besides gate sizing, buffer insertion can be a good choice to optimize power and delay of circuits. In [8,9], buffer insertion is employed to enhance performance and reduce load capacitance of nodes, which drive large capacitance. Although many gate sizing or buffer insertion techniques have been proposed, there is no algorithm to apply both techniques in a single process concurrently to reduce glitches. Our optimization algorithm interleaves both gate sizing, gate freezing and buffer insertion into a single optimization process. First we classify gate sizing and buffer insertion into three and two types respectively. Each type of gate sizing and buffer insertion is combined according to its purpose and utility. We also defined gain factor in Section 4. A negative gain factor means that power cost to remove glitches is greater than the power reduction through glitch reduction. In this case, although glitches are removed the total power of a circuit is increased, so the glitch reduction process including gate freezing, gate sizing and buffer insertion should be restricted.

The circuit under the optimization is a CMOS combinational circuit designed in a synchronous design style. The rest of the paper is organized as follows: Section 2 gives basic terminologies, notations, and formulation. Section 3 discusses gate freezing, the different types of gate sizing, and buffer insertion methods. Section 4 explains the unified optimization algorithm of gate freezing, gate sizing, and buffer insertion. Section 5 shows experimental results followed by the conclusion in Section 6.

2. Preliminaries

2.1. Gate-level power model

For CMOS circuits, dynamic power is the dominant source of power dissipation. The average dynamic power consumed by a CMOS gate is given by

$$P_{\text{average}} = 0.5 \cdot \frac{V_{\text{dd}}^2}{T_{\text{cycle}}} \cdot C_{\text{load}} \cdot N. \tag{1}$$

The notations C_{load}, V_{dd}, and T_{cycle} are the load capacitance, the supply voltage, and the global clock period respectively. N is the switching activity, i.e., the number of gate output transitions per clock cycle (also known as transition density). A straightforward method eliminating transition density is simulating a circuit with arbitrary generated input vectors.

Input vectors are generated randomly. We generated input vectors randomly and carefully monitored the average power to find out input vectors used. So, we think Eq. (2) is suitable for our experiments. The input vectors applied final power evaluation were newly random generated using Eq. (2). The accuracy and confidence level used were 10% and 90% respectively. The minimum number of input vectors found was 3 for all circuits. Although 3 seems very small, if it is achieved properly, it can characterize the power dissipation of the circuitry. The probabilistic approaches to estimate transition density with glitches have been proposed in [11,12]. Although theses approaches take the spatial and temporal correlation into consideration, the CPU times are significant and iterative applying is infeasible. For an optimizer performing estimation at each step of iteration, the most important factor is the run time. A straightforward method to estimate transition density is to simulate a circuit with arbitrary generated input vectors. In this case, the problem is how may input vectors have to be applied to achieve given accuracy level. For a typical logic circuit and reasonable error and confidence level, the numbers of vectors is small, making this approach very efficient. Although this technique is some limitation that it only guarantees accuracy for the average switching activity over all gates, we employed it because of its fast run time and accurate glitch estimation.

The most important aspect of simulation-based switching activity estimation is deciding how many input vectors for simulate in order to achieve a given accuracy level. Given a user-specified allowable percentage error ε and confidence level α, the approach computes the number of input vectors to be used to simulate the circuit. With $\alpha \times 100\%$ confidence, $|\overline{p} - P| < \text{erf}^{-1}(\alpha/2) \times s/\sqrt{L}$, where $\overline{p}$ and s are the measured average and standard deviation of the power, P is the average power dissipation, L is the number of input vectors and $\text{erf}^{-1}(\alpha/2)$ is the inverse error function obtained from the normal distribution. Since we require $|\overline{p} - P|/\overline{p} < \varepsilon$, it follows [14]:

$$L \geqslant \left(\frac{\text{erf}^{-1}(\alpha/2) \times s}{\varepsilon \times \overline{p}} \right)^2. \tag{2}$$

In a typical logic circuit with reasonable error and confidence level, the number of vectors needed is usually small, making this approach very efficient. We have employed it because of its fast run time and accurate glitch estimation.

2.2. Glitch model

Glitches can be separated into a generated glitch and a propagating glitch depending on its precedence [7]. Generated glitch is generated by functional transitions and a propagating glitch is caused by an input signal, which contains glitches. Fig. 1 shows an example.