

논문 2006-43SD-8-6

통계적 최적화를 위한 확률적 글리치 예측 및 경로 균등화 방법

(Stochastic Glitch Estimation and Path Balancing for Statistical Optimization)

신 호 순*, 김 주 호**, 이 형 우***

(Hosoon Shin, Juho Kim, and HyungWoo Lee)

요 약

이 논문에서는 공정 변이의 고려를 위한 통계적 시간 분석(statistical timing analysis)에서 전력감소를 고려한 회로의 최적화를 위해 글리치 및 지연시간의 확률적 모델 및 연산을 이용하여 각 경로 및 경로상의 게이트의 민감도(sensitivity)를 계산하고 이를 이용한 사이징(sizing)을 통해 회로의 지연시간의 증가 없이 글리치를 감소하는 방법을 제시한다. 제안된 알고리즘은 통계적 시간 분석에 근거한 회로의 전후방 탐색을 이용하여 공정 변수를 고려한 확률적 글리치 발생률을 예측한다. 또한 글리치 발생률을 고려한 게이트의 선택 및 사이징 가능한 지연시간의 최적화된 계산을 통해 효율적인 게이트 사이징 기법과 글리치 감소를 위한 경로균등화 방법을 제시한다. 제안된 알고리즘의 효율성은 0.16 μ m 모델 파라미터를 이용하여 ISCAS85 벤치마크 회로에 대한 실험을 통해 검증되었다. 실험 결과를 통해 제안된 알고리즘은 글리치 예측에 있어 8.6%의 정확도의 개선을 보였고, 경로균등화에 의한 최적화에 있어 9.5%의 개선을 보였다.

Abstract

In the paper, we propose a new method for power optimization that uses path balancing based on stochastic estimation of glitch in Statistical Static Timing Analysis (SSTA). The proposed method estimates the probability of glitch occurrence using tightness probability of each node in timing graph. In addition, we propose efficient gate sizing technique for glitch reduction using accurate calculation of sizing effect in delay considering probability of glitch occurrence. The efficiency of proposed method has been verified on ISCAS85 benchmark circuits with 0.16 μ m model parameters. Experimental results show up to 8.6% of accuracy improvement in glitch estimation and 9.5% of optimization improvement.

Keywords : 통계적 시간 분석(SSTA), 공정변이(Process variation), 글리치(Glitch), 게이트 사이징(Gate sizing) 경로 균등화(path balancing)

I. 서 론

최근 CMOS 공정 기술이 90나노 이하 시대로 접어들면서 반도체 제조 과정에서 발생하는 공정 변이(process variations)의 영향은 더욱 중요시되고 있다. 회로의 타이밍 검증에 있어서 기존에 사용되어 온 코너 기반 또는 사례 기반의 정적 시간 분석(static timing

analysis)은 90나노 이하의 디자인에선 매우 부정확하고 불확실한 결과를 나타내게 되었다. 이에 따라 지연 시간 및 공정 변이의 확률적 모델링과 연산이 가능한 통계적 시간 분석(statistical timing Analysis)이 제시되었다. 통계적 시간 분석 방법에선 공정 변수에 의한 지연시간의 불확실성은 확률 분포를 이용하여 체계적으로 모델링된다.

회로 설계에서의 또 하나의 커다란 과제는 전력 소모이다. Deep sub-micron(DSM)시대에 접어들면서 트랜지스터의 고 집적화는 회로의 크기를 매우 감소시켰고, 이에 따른 전력 소모량의 분석은 더욱 복잡하게 되었다. 신호천이에 의한 동적 전력소모는 전력 소모의 가장 큰 비율을 차지한다. 신호의 천이는 노드의 논리값

* 학생회원, *** 정회원, (주) 삼성전자 반도체총괄 (Semiconductor Business SAMSUNG ELECTRONICS CO., LTD)

** 정회원, 서강대학교 컴퓨터학과 (Department of Computer Science and Eng., SogangUniversity)

접수일자: 2006년2월3일, 수정완료일: 2006년8월18일

의 변화를 의미하는데 이는 회로의 옳은 동작을 위한 기능성 천이(functional transition)와 불필요한 천이인 글리치(glitch)로 구분된다. 이 중 글리치는 신호 천이의 상당한 부분을 차지하여 추가적인 전력 소모 및 논리값의 오류(functional fault)문제를 일으키는데 [1]에 의하면 회로의 전체 전력소모 중 7~43%를 차지한다. 그러므로 이러한 글리치의 제거는 회로의 전력소모를 크게 줄임과 동시에 논리적 안정성을 유지할 수 있는 효과적인 방법이다.

최근에 통계적 시간 분석에 기반 하여 게이트 사이징을 이용한 여러 가지 회로 최적화 방법들이 소개되었다^{[2][3][4][5][6]}. 이들 방법은 통계적인 게이트 사이징을 통해 지연시간 분포의 평균과 분산을 줄임으로써 회로의 지연시간 최적화를 목적으로 하였다. 그러나 [2-4]에서는 게이트 사이징을 통한 최적화의 단계에서 회로의 전력소모량은 전혀 고려되지 않았다. 확률적 글리치를 고려한 게이트 사이징 방법이 [1]에서 제안되었으나 시간 분석의 방법이 정적이며 확률적 모델링이 아닌 계단 함수를 사용했기 때문에 글리치의 예측이 매우 부정확하다.

본 논문에서는 글리치 및 지연시간의 확률적 모델을 이용하여 각 노드의 글리치 발생률을 예측하고 이에 기반한 게이트 사이징을 통하여 글리치 감소를 위한 경로균등화 기법을 제시한다. 제안된 알고리즘은 공정 변수 및 상관관계를 고려한 확률적 글리치 예측 기법과 통계적 민감도 계산을 이용한 게이트 사이징 기법으로 구성된다. 먼저 통계적 시간 분석에 근거한 전후방 탐색을 이용하여 각 노드에 대한 입력신호의 도착 시간(arrival time), 요구시간(required time), 슬랙(slack)을 [1]에 제안된 방법을 응용하여 확률 랜덤 변수로 계산하면서 각 노드에 대한 글리치의 확률적 예측을 병행한다. 다음으로 각 노드에 글리치 발생 확률을 고려하여 게이트를 선택하고 용량성 부하와 글리치를 동시에 줄이면서 제한조건을 만족하는 최적화된 게이트의 사이즈를 결정하고 사이징 하여 경로균등화를 통해 글리치를 감소시킨다.

본 논문은 다음과 같이 구성된다. 제 II장에서는 이론적 배경이 되는 확률적 지연시간 모델 및 통계적 시간 분석의 기본연산, 정적 글리치 모델과 최적화 방법에 대해 설명한다. 제 III장에서는 본 논문에서 제안하는 확률적 글리치 모델링 및 이를 고려한 전후방 탐색 기법과 민감도 계산을 이용한 게이트 사이징 방법을 제안한다. 제 IV장에서 구현된 알고리즘에 의한 실험

결과를 보여준다. 마지막으로 제 V장에서는 추후 과제 및 결론으로 끝을 맺는다.

II. 통계적 시간 분석과 글리치 예측

1. 통계적 시간 분석

반도체의 제조 공정상에서 설계 디자인에 영향을 주는 공정 변수들의 대표적인 예로는 유효 게이트 길이, 도핑 집중, 산화물 두께, 문턱 전압과 같은 변인들이 있으며 서로 다른 칩 상에서 각각의 변화량을 갖는다. 공정 변수는 크게 Inter-die 변수와 Intra-die 변수로 구분된다^[1]. 통계적 시간 분석에서는 일반적으로 공정 변수를 확률 분포 형태로 모델링하며 평균과 분산을 이용하여 표현 한다^{[2][6-10]}. 그림 1은 공정 변수의 확률 밀도 함수(probability density function)와 누적 분포 함수(cumulative density function)를 보여준다. 본 논문에서는 도착시간, 요구시간 및 슬랙과 신호 슬루율을 모두 식 (1)과 같은 정규 형태로 모델링 한다.

$$a_0 + \sum_{i=1}^n a_i \Delta X_i + a_{n+1} \Delta R_a \quad (1)$$

변수 a_0 는 게이트나 와이어의 지연시간의 평균값 혹은 중간값을 뜻한다. $\Delta X_i, i=1,2,\dots,n$ 는 지연시간에 영향을 주는 n 개의 전역 공정 변수 $X_i, i=1,2,\dots,n$ 의 평균이나 중간 값에 대한 편차를 나타내며, $a_i, i=1,2,\dots,n$ 는 각각의 전역 공정 변수에 대한 민감도를 나타낸다. ΔR_a 는 독립 랜덤 변수 R_a 의 평균에 대한 편차를 나타내며, 마지막으로 a_{n+1} 는 R_a 에 대한 민감도를 나타낸다. 독립 랜덤 변수는 전역 공정 변수로 나타낼 수 없는 지연시간의 임의 변동량을 나타낸다. 위와 같이 표현된 지연시간 변수를 전달하기 위해 통계적 시간 분석에서는 두 가지 기본적인 연산을 사용한다. 먼저 ADD연산은 와이어의 지연시간과 게이트의 지연시간을 더함으로써 신호를 전달하기 위해 사용된다. 지연시간 변수 A와 B가 모두 정규형이라 가정할 때 ADD(A,B) 결과 값 C는 다음과 같이 계산된다.

$$\begin{aligned} c_0 &= a_0 + b_0 \\ c_i &= a_i + b_i \quad \text{for } 1 \leq i \leq n \\ c_{n+1} &= \sqrt{a_{n+1}^2 + b_{n+1}^2} \\ C &= c_0 + \sum_{i=1}^n c_i \Delta X_i + c_{n+1} \Delta R_c \end{aligned} \quad (2)$$

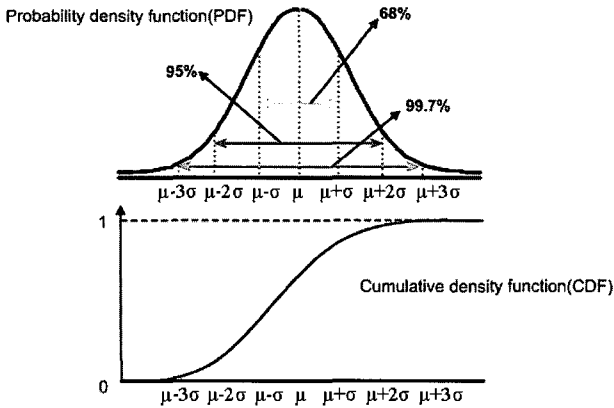


그림 1. 공정 변수의 확률 밀도 함수와 축적 분포 함수
Fig. 1. The probability distributions of process variation.

다음으로, MAX연산은 각 노드에서 최종 도착 시간을 계산하는 연산이다. MAX연산은 통계적 컨볼루션 (statistical convolution)을 통해 결과 값을 계산하기 때문에 함수 형태가 유지되지 않는다^[2]. 따라서 MAX연산 결과의 선형 근사 화를 위해 tightness probability를 사용하여 계산한다^{[7][10]}. 두 입의 변수 A와 B가 주어질 때 A의 tightness probability는 T_A 로 표시되며 A가 B보다 클 확률 혹은 지배적일 확률을 의미한다. A의 tightness probability를 구하는 식은 다음과 같다.

$$T_A = \Phi\left(\frac{a_0 - b_0}{\theta}\right), \Phi(y) = \int_{-\infty}^y \phi(x) dx \quad (3)$$

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), \theta = \sqrt{\sigma_A^2 + \sigma_B^2 - 2r}$$

Clark의 계산법 [11]과 [7]에서의 방법을 이용하여, MAX(A,B)의 평균과 분산은 다음과 같이 계산된다.

$$c_0 = E[MAX(A,B)]$$

$$= a_0 T_A + b_0 (1 - T_A) + \theta \phi\left(\frac{a_0 - b_0}{\theta}\right)$$

$$\sigma_C^2 = var[MAX(A,B)] \quad (4)$$

$$= (\sigma_A^2 + a_0^2) T_A + (\sigma_B^2 + b_0^2) (1 - T_A) + (a_0 + b_0) \theta \phi\left(\frac{a_0 - b_0}{\theta}\right) - c_0^2$$

$$c_i = T_A a_i + (1 - T_A) b_i, \quad i = 1, 2, \dots, n$$

신호의 도착시간 전달을 위해 회로의 전방 탐색에서 ADD, MAX 연산을 사용한 것과 같은 맥락으로 각 노드의 요구시간 계산을 위한 후방 탐색에선 MINUS, MIN 연산을 사용하며 각각의 연산의 계산방법은 ADD, MAX 연산과 거의 동일하다.

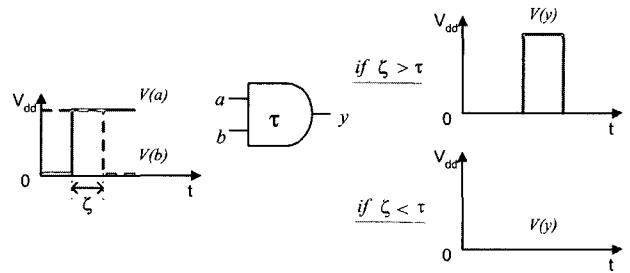


그림 2. 시간적 원인에 의한 글리치 발생
Fig. 2. The glitch generation due to timing cause.

2. 기존의 통계적 글리치 예측 방법

글리치는 두 개의 조건이 동시에 만족될 때 발생한다. 첫번째는 신호의 입력 패턴에 의한 기능적 (functional) 원인이며 두 번째는 도착 지연시간의 관계에 의한 시간적 (timing) 원인이다. 시간적 원인과 기능적 원인이 동시에 만족될 때 발생하는 글리치를 그림 2.9에서 보여주고 있다. 따라서 한 게이트에 대한 글리치의 발생 확률은 다음과 같다^[1].

$$R_{glitch}(i) = P_{pattern}(w_k) \cdot P_{delay}(w_k)$$

$R_{glitch}(i)$: 게이트 i 에서 글리치가 발생할 확률
 w_k : 글리치를 발생시키는 입력 패턴
 $P_{pattern}(w_k)$: w_k 가 발생할 확률
 $P_{delay}(w_k)$: w_k 가 시간적 원인을 만족할 확률

$$(5)$$

글리치의 제거는 동적 전력 소모량을 감소시킬 수 있는 효과적인 방법 중 하나이다. 각 노드의 동적 전력 소모량은 신호 천이의 횟수에 비례하기 때문이다. 따라서 글리치에 의한 동적 전력 소모량은 다음과 같이 계산된다.

$$P_i = \frac{1}{2} \cdot C_i \cdot V_{dd}^2 \cdot N_{glitch_tran}$$

P_i : 게이트 i 에서의 동적 전력 소모
 C_i : 게이트 i 의 용량성 부하
 V_{dd} : 공급전압
 N_{glitch_tran} : 게이트 i 의 글리치 발생 횟수

$$(6)$$

통계적인 글리치 예측 방법들이 [1]과 [13]에서 제안되었다. 그러나 이 방법들은 정적 시간 분석에 기초하였으며 단순 계단 함수를 사용하여 글리치 발생율을 계산하였다. 또한 발생율의 계산에서 공정 변수에 대한 고려는 전혀 이루어지지 않았다.

III. 확률적 글리치 예측

통계적 시간 분석에선 도착시간, 요구시간, 슬랙 및 슬루(slew)는 확률 분포로 표현된다. 따라서 임의의 노드에 대한 입력 신호들의 도착시간의 차이로 인해 발생하는 글리치는 확률적으로 모델링되어야 한다. 그림 3은 도착시간과 게이트 내부 지연시간의 분포를 보여준다. 간격 시간을 두 입력 신호간의 도착시간의 차이로 정의하자 그러면 글리치가 발생할 확률은 간격이 이 게이트 소자의 내부 지연시간보다 클 확률로서 계산된다.

$$\text{Prob}(\text{glitch}) = \begin{cases} \text{Prob}(A-B > G_1) & \text{if } A \text{ is the latest signal} \\ \text{Prob}(B-A > G_1) & \text{if } B \text{ is the latest signal} \end{cases} \quad (7)$$

$\text{Prob}(X)$: probability of X

따라서 한 노드에서의 글리치 발생 확률의 예측은 $\text{Prob}(\text{glitch})$ 의 값을 계산하는 것이다. 글리치 발생의 기준이 되는 게이트 내부 지연시간 G_1 은 정규 분포의 형태이다. 그림 4는 글리치 발생과 게이트 내부 지연시간간의 관계를 보여준다. 따라서 정규 분포를 따르는 내부 지연시간의 크기에 따라 발생하는 글리치에 대한 예측의 정확성은 식 (8)과 같이 나타낼 수 있다.

$$G_1 = \begin{cases} \mu_{G_1} & \text{estimate glitch in 50\% accuracy} \\ \mu_{G_1} + \sigma_{G_1} & \text{estimate glitch in 84\% accuracy} \\ \mu_{G_1} + 2\sigma_{G_1} & \text{estimate glitch in 97.7\% accuracy} \\ \mu_{G_1} + 3\sigma_{G_1} & \text{estimate glitch in 99.9\% accuracy} \end{cases} \quad (8)$$

식 (8)의 관계를 이용하여, 회로 설계자는 글리치 예측의 정확성과 효율성 사이에서 선택이 가능하다. 글리치 발생확률을 구하기 위한 식 $\text{prob}(B-A > G_1)$ 는 $\text{prob}(B > G_1 + A)$ 와 같다. 이 식에서 지연시간 변수 G_1 및 도착시간 변수 A 는 모두 정규형 지연시간 변수

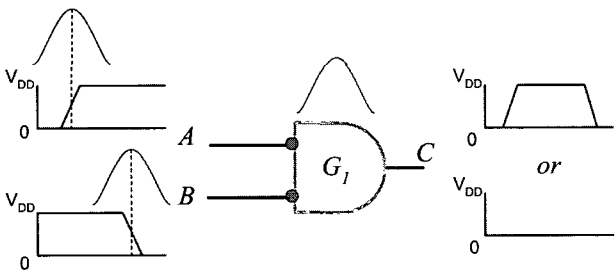


그림 3. 글리치가 발생할 시간적 조건
Fig. 3. The timing condition to generate glitch.

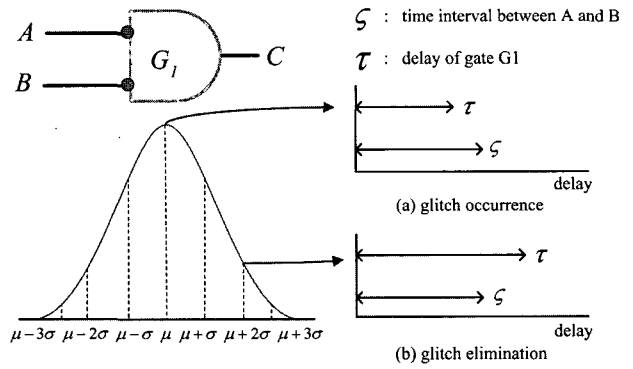


그림 4. 글리치 발생과 게이트 내부 지연시간과의 관계
Fig. 4. The relation between gate delay and glitch occurrence.

이다. 따라서 (G_1+A) 의 결과를 A' 라 할 때 이는 $\text{SUM}(G_1,A)$ 의 결과와 같다.

$$\begin{aligned} A' &= \sum(A, G_1) \\ &= (a_0 + \sum_{i=1}^n a_i \Delta X_i + a_{n+1} \Delta R_a) + \\ &\quad (g_0 + \sum_{i=1}^n g_i \Delta X_i + g_{n+1} \Delta R_g) \\ &= (a_0 + g_0) + \sum_{i=1}^n (a_i + g_i) \Delta X_i + \\ &\quad (\sqrt{a_{n+1}^2 + g_{n+1}^2}) \Delta R_a \\ &= a_0' + \sum_{i=1}^n a_i' \Delta X_i + a_{n+1}' \Delta R_a \end{aligned} \quad (9)$$

따라서 $\text{prob}(B > G_1 + A)$ 는 $\text{prob}(B > A')$ 와 같고 글리치의 발생 확률은 $\text{prob}(B > A')$ 의 값이 된다. 지연시간 변수 A' 와 B 는 모두 정규형 지연시간 변수이므로 $\text{prob}(B > A')$ 의 값은 B 가 A' 보다 클 확률이다. 그러므로 $\text{prob}(B > A')$ 는 변수 B 의 tightness probability인 T_B 와 같다.

$$\begin{aligned} \text{prob}(B > A') &= T_B \\ &= \Phi\left(\frac{b_0 - a_0'}{\theta}\right), \Phi(y) = \int_{-\infty}^y \phi(x) dx \\ \phi(x) &= \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right), \theta = \sqrt{\sigma_B^2 + \sigma_{A'}^2 - 2r} \end{aligned}$$

$$\begin{aligned} B &= b_0 + \sum_{i=1}^n b_i \Delta X_i + b_{n+1} \Delta R_b \\ A' &= a_0' + \sum_{i=1}^n a_i' \Delta X_i + a_{n+1}' \Delta R_a \\ \sigma_B^2 &= \sum_{i=1}^{n+1} b_i^2, \sigma_A^2 = \sum_{i=1}^{n+1} a_i'^2, r = \sum_{i=1}^n a_i' b_i \end{aligned}$$

(9)

```

For each gate  $G_i$ 
glitch_estimation()
{
     $I_i$  : input signals ( $i=1,2,\dots,n$ )
     $D_g$  : gate delay PDF  $\approx N(\mu_G, \sigma_G)$ 
    Sorting  $I_i$ , ( $i=1,2,\dots,n$ ) using  $\mu_{t_i}$  (mean value) of arrival time PDF ;
    For each sorted input signal  $I_j$  ( $j=1,2,\dots,n$ )
    {
         $I'_j = \text{Sum}(I_j, D_g)$ ;
        Calculate probability of ( $I_{j+1} > I'_j$ ) =  $T_{I_{j+1}}$ 
         $T_{I_{j+1}}$  = probability of glitch ( $I_j, I_{j+1}$ ) at  $G_i$ 
    }
}
    
```

그림 5. 확률적 글리치 예측 알고리즘
 Fig. 5. Stochastic glitch estimation algorithm.

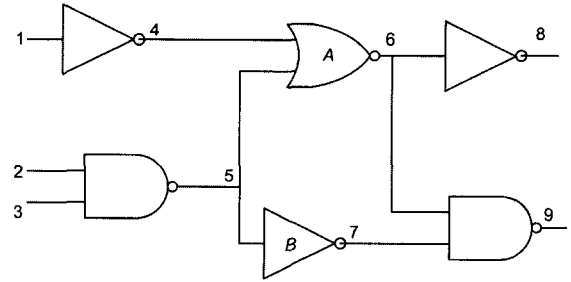
따라서 제안된 확률적 글리치 예측 알고리즘의 요약은 그림 5와 같다.

IV. 통계적 최적화를 위한 경로 균등화

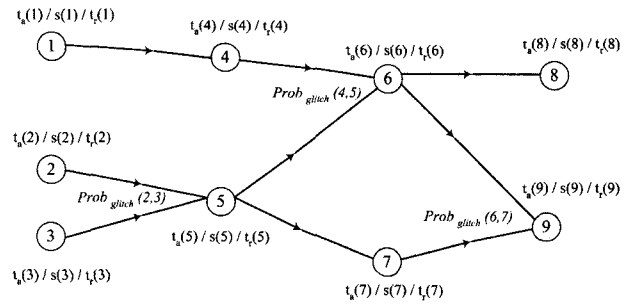
도착시간, 요구시간 및 슬랙 정보는 회로의 타이밍 그래프에 대한 전후방 탐색의 수행에 의해 구해진다. 또한 각 노드에 대한 글리치 발생률도 전방탐색 시 구해진다. 그림 6은 예제 회로의 타이밍 그래프에 대한 전후방 탐색 이후의 각 노드에 대한 정보를 나타낸다. $ta(i)$, $tr(i)$, $s(i)$ 는 노드 i 에 대한 도착시간, 요구시간, 슬랙을 각각 나타낸다. 만약 사이징 할 게이트의 수가 한정적이며 사이징 이후에 면적 등의 제한조건을 만족해야 한다고 할 때 전력 감소량이 큰 게이트를 사이징하는 것이 효율적이며 제한조건에 위배되지 않을 가능성이 높다. 따라서 본 논문에서는 리사이징에 있어 각 게이트의 글리치 발생율을 가장 주요한 요인으로 하여 우선순위를 정한다.

그림 6에서 $prob_{glitch}(2,3) < prob_{glitch}(4,5) < prob_{glitch}(6,7)$ 의 조건을 가정하면, 게이트 A 혹은 B가 가장 먼저 리사이징 된다. 사이징 우선순위가 정해진 이후에는 사이징 할 게이트를 결정해야 한다. 다운사이징 할 게이트의 선택에선 각 노드의 tightness probability를 사용한다. 그림 7에서 6번 노드가 7번 노드보다 tightness probability가 작다. 이는 6번 노드의 도착시간이 7번 노드보다 빠름을 의미한다. 따라서 6번 노드의 도착시간을 지연하기 위해 게이트 A가 다운사이징 할 게이트로 선택된다.

사이징 할 게이트가 선택되면 사이징 할 지연시간의 양을 계산해야 한다. 그림 7에서 게이트 A는 단일 팬아웃 노드를 갖는다. 따라서 경로균등화를 위한 게이트 A



(a) sample circuit



(b) timing graph of sample circuit

그림 6. 통계적 시간 분석 이후 예제 회로의 타이밍 정보

Fig. 6. Timing information after SSTA.

의 지연시간 증가량은 6번과 7번 노드의 도착시간의 차이와 같다. 지연시간의 변화량을 D_{sizing} 이라 하면 다음과 같이 계산된다.

$$\begin{aligned}
 D_{sizing} &= MINUS(t_a(v_j) - t_a(v_i)) \\
 v_i &= fanout(G_i), fanin(G_j) \\
 v_j &= latest arrival fanin(G_j)
 \end{aligned}
 \tag{11}$$

여기서 사이징 할 다양한 크기의 게이트는 셀 라이브러리에서 선택된다고 가정한다. 만약 리사이징된 크기를 가진 게이트가 라이브러리에 없을 경우엔, 수정된 변화량에 가장 근접한 크기를 가진 게이트가 선택된다.

다음으로 사이징 할 게이트가 여러 개의 팬아웃 노드를 갖는 경우를 가정하자. 이 경우는 사이징 할 게이트의 다중 출력 노드가 다음 레벨의 여러 게이트에 입력 노드로 사용되므로 부 입력 노드들과의 경로 상관관계(path correlation)를 고려해야 한다. 이러한 관계를 고려하여 지연시간의 변화량을 계산하기 위해 본 논문에서는 [12]에서 제안된 최소 입력 요구시간(MIN_REQ)의 개념을 사용한다. 임의의 게이트 G에 대한 최소 입력 요구시간은 G에 대한 입력신호들의 요구시간 중 최소의 값으로 정의되며 $t_{min_req}(G)$ 로 표기된다. 그러나 통계적 시간 분석에서는 요구시간이 정규형 지연시간 모델로 표현된다. 따라서 통계적 시간 분석에서의 게이

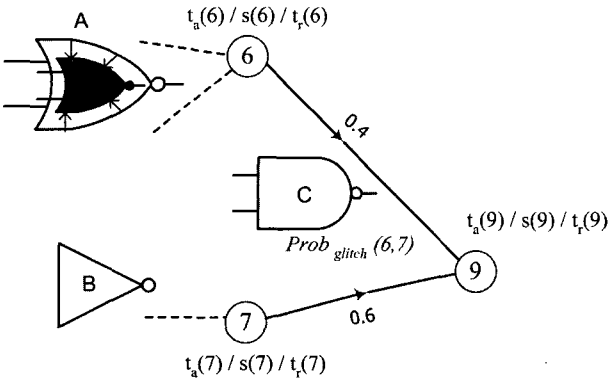


그림 7. 다운사이징 할 게이트의 선택
Fig. 7. Choosing gate for downsizing.

트 G에 대한 최소입력 요구시간은 다음과 같다.

$$t_{min_req}(G) = t_r(v_i) \quad (12)$$

if $\mu_i < \mu_j$ for $j = 1, 2, \dots, n \wedge j \neq i$

$$t_r(v_i) \approx N(\mu_i, \sigma_i)$$

*sizable_delay*는 주어진 제한 조건에서 글리치를 제거하기 위한 게이트의 사이징 가능한 최대 지연시간의 양을 나타낸다^[12].

통계적 시간 분석의 여러 지연 시간 변수들과 모델링 형태의 동일성을 유지하기 위하여, *sizable_delay*를 계산하는 방법도 수정되어야 한다. 따라서 본 논문에서는 게이트 G_i 에 대한 *sizable_delay*를 $D_{sizable_max_G_i}$ 라 할 때 다음과 같은 통계적 방법으로 계산한다.

$$D_{sizable_max_G_i} = MINUS(t_{r_min}, t_{v_i}) \quad (13)$$

$G_{i+1} \in fanout(G_i)$
 $t_{r_min} = \text{minimum value of } t_{min_req}(G_{i+1})$
 $v_i = fanout(G_i), fanin(G_{i+1})$

식 13의 모든 지연시간 변수들은 같은 정규형이다. $D_{sizable_max_G_i}$ 는 지연시간의 제한조건을 위배하지 않기 위해 사용된다. 그러나 계산된 지연시간의 양은 부 입력 노드들의 최소 입력 요구시간은 고려되었지만 도착시간에 대한 고려는 이루어지지 않았다. 그림 8에서 $D_{sizable_max_G_A}$ 는 2이며 사이징 결과 A번 노드의 도착시간은 4가 된다. 그러나 부 입력 노드 B의 도착시간인 3보다 값이 커져 게이트 G_B 에 대한 최대 도착시간이 증가된다. 이러한 도착시간의 변화는 다음 레벨의 게이트들의 도착시간에도 영향을 주어 또 다른 글리치 발생의 원인이 된다. 따라서 부 입력들의 최소 도착 시간들 또한

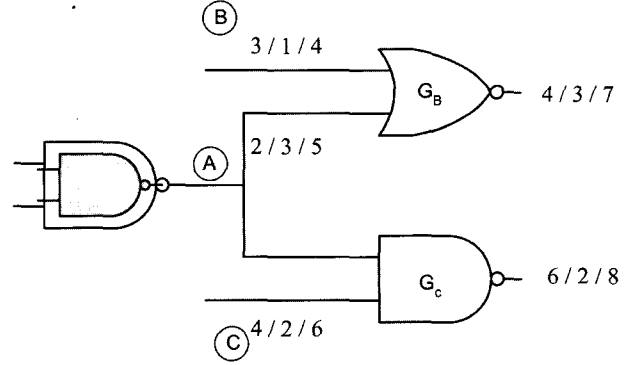


그림 8. 경로 균등화를 위한 게이트의 크기 선택
Fig. 8. Determining the size of a gate for path balancing.

고려되어야 하며 이는 다음과 같이 계산된다.

$$t_{min_arr}(G) = t_a(v_i) \quad (14)$$

if $\mu_i < \mu_j$ for $j = 1, 2, \dots, n \wedge j \neq i$

$$t_a(v_i) \approx N(\mu_i, \sigma_i)$$

결과적으로 사이징 가능한 지연시간의 범위는 최소 입력 요구시간과 도착시간의 확률 분포에 의해 결정된다. 따라서 임의의 게이트 G_i 에 대하여 도착시간과 요구시간의 제한조건을 모두 만족하는 사이징 가능 지연시간 $D_{sizable_max_G_i}$ 는 다음과 같다.

$$D_{sizable_max_G_i} = \begin{cases} MINUS(t_{r_min}, t_a(v_i)) & \text{if } \mu_{t_{r_min}} \leq \mu_{t_{a_min}} \\ MINUS(t_{a_min}, t_a(v_i)) & \text{if } \mu_{t_{r_min}} > \mu_{t_{a_min}} \end{cases} \quad (15)$$

$$G_{i+1} \in fanout(G_i)$$

$t_{r_min} = \text{minimum value of } t_{min_req}(G_{i+1})$
 $t_{a_min} = \text{minimum value of } t_{min_arr}(G_{i+1})$
 $v_i = fanout(G_i), fanin(G_{i+1})$

요약하여 전후방 탐색 및 게이트 사이징을 포함한 전체적인 경로 균등화 알고리즘은 그림 9와 같다. 제안된 알고리즘은 전 후방 탐색이후에 글리치 발생을 이용하여 사이징 할 게이트를 선택한다. 그리고 주어진 제한 조건 하에서 식 (15)를 이용하여 선택된 게이트를 리사이징한다.

V. 실험 결과

제안된 알고리즘은 C와 MATLAB으로 Ultra Sparc workstation에서 구현되었다. 실험 데이터는 0.16 μ m 공

```

levelize the circuit;
/* glitch estimation and calculate arrival time of all node */
forward_propagation();
/* calculate required time and slack of all node */
backward_propagation();

/* statistical gate sizing */
Statistical_path_balancing()
{
    Max_change : the number of sizable gates not to violate constraints
    Sorting all gates  $G_i$  using  $Prob_{glitch}(input\ nodes)$ ;
    For(i=1; i<=Max_change; i++)
        For each  $G_i$ 
        {
            select input node  $v_j$  that has minimum  $T_j$ 
            ( $j=1,2,\dots,n \wedge fanin(G_i)=v_j$ );
            select gate  $G_j$  that  $fanout(G_j)=v_j$ ;
            calculate  $D_{sizing} = D_{stable\_max\_G_j}$ ;
            downsize  $G_j$  amount of  $D_{sizing}$ ;
        }
}
    
```

그림 9. 통계적 경로 균등화 알고리즘
 Fig. 9. Statistical path balancing algorithm.

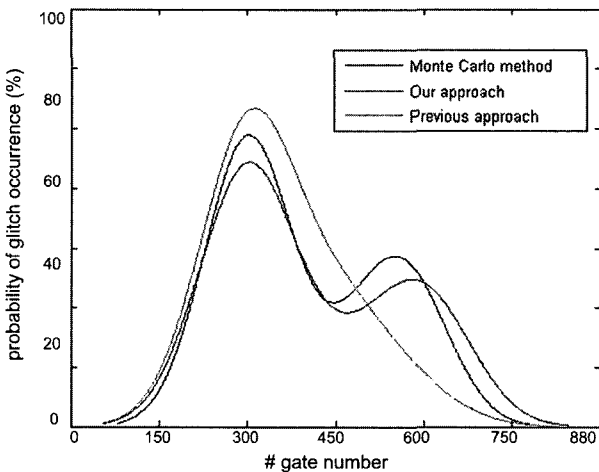


그림 10. C1908 회로에 대한 글리치 발생을 예측 비교
 Fig. 10. The comparison of the distributions of estimated glitches in C1908 circuit.

표 1. 제안된 글리치 예측 기법의 정확성
 Table 1. Results of glitch estimation.

circuit	#gate	SPICE simulation (Monte Carlo method) (%)	Previous approach (uniform distribution) (%)	Our approach (gaussian distribution) (%)	Improvement (%)
C432	160	52.93	30.7	57.20	17.9
C499	202	41.58	21.1	46.42	15.6
C880	383	40.15	22.3	44.88	13.1
C1355	546	31.64	44.0	44.97	-0.9
C1908	880	32.87	19.6	35.84	10.3
C2670	1193	35.04	28.2	37.81	4.1
C3540	1669	35.33	22.8	38.84	9.0
C5315	2307	40.97	33.7	45.16	3.1
C6288	2406	44.26	34.8	52.15	1.5
C7552	3512	40.76	24.6	44.55	12.3
	average error rate (%)	-	13.8	5.2	8.6

표 2. 각 방법의 글리치 제거율 비교
 Table 2. The results of glitch reduction rate.

circuit	#gate	#max_gate (10% of #gate)	Previous algorithm (circuit level order) (%)	Our algorithm ($prob_{glitch}$ order) (%)	Improvement (%)
C432	160	16	7.9	17.4	9.5
C499	202	20	21.3	21.3	0
C880	383	38	6.9	22.1	15.2
C1355	546	54	8.6	21.9	13.3
C1908	880	88	20.1	27.9	7.8
C2670	1193	119	13.3	26.3	13
C3540	1669	166	10.8	25.6	14.8
C5315	2307	230	16.1	22	5.9
C6288	2406	240	10.7	19.1	8.4
C7552	3512	351	15.1	22.4	7.3
	average reduction rate (%)	-	13.1	22.6	9.5

정을 사용한 ISCAS'85 벤치마크 회로를 사용하였으며 HSPICE 시뮬레이션을 통해 지연시간을 특성화 하였다. 실험에 사용된 공정변수는 게이트 유효 길이 $\Delta L_{eff} = \pm 10\%$, 공급 전압 $V_{DD} = \pm 5\%$, 산화물 두께 $\Delta TOX = \pm 5\%$, 온도 $\Delta T = \pm 15\%$, 노드의 임의 랜덤 변수 $\Delta R = \pm 5\%$ 를 고려 하였다.

먼저 확률적 글리치 예측 기법의 정확성을 [1]에서 제안한 기존의 방법과 비교하여 검증하였다. 표 1은 제안된 글리치 예측 알고리즘과 기존의 방법의 비교를 보여준다. 몬테 카를로 방법의 결과를 기준으로 하여 제안된 방법은 최대 17.9%, 평균 8.6% 정확성의 개선을 나타냈다. 그림 10은 C1908의 회로 내부의 모든 게이트에 대한 글리치 발생을 제안된 방법과 기존의 방법, 몬테 카를로 방법으로 예측하여 비교하였다. 예상대로 제안된 방법의 글리치 발생을 분포가 몬테카를로 시뮬레이션의 분포에 매우 근접함을 알 수 있다.

다음으로 제안된 경로균등화 방법의 효율성을 검증 하였다. 기존의 방법을 글리치 발생을 고려하지 않고 단순히 회로의 레벨만을 고려하여 순서대로 게이트를 리사이징 하였다. 또한 각 게이트 사이징 방법의 타당한 성능 측정을 위해, 사이징 가능한 게이트의 수는 전체 게이트 수의 10%로 제한하였다. 표 2는 제안된 방법과 기존 방법의 글리치 제거율을 비교하여 보여준다. 기존 방법과 비교하여 제안된 방법은 최대 15.2%, 평균 9.5%의 글리치 제거율 증가를 기록하였다.

VI. Conclusion

본 논문에서는 글리치의 확률적 예측 기법 및 이를 고려한 통계적 경로 균등화 방법을 제안하였다. 제안된 경로 균등화 알고리즘은 각 노드의 글리치 발생 확률을

고려하고 최적화된 사이징 양을 결정하는 통계적인 게이트 사이징 기법을 이용하여 글리치를 제거한다. 확률적 글리치 예측 알고리즘은 몬테카를로 시뮬레이션의 결과를 기준으로 평균 5.23%의 오차를 보였고, 기존 방법의 결과보다 평균 8.6% 정확하였다. 또한 경로균등화 알고리즘은 글리치 제거율에서 기존 연구 방법과 비교하여 평균적으로 9.5%정도의 효율 향상을 보였다.

제안된 확률적 글리치 예측 기법은 선형적인 정규분포의 형태를 가정하여 실제 글리치 발생 분포와 어느 정도의 오차를 보였다. 계산시간의 고려와 함께 실제 시뮬레이션 결과와의 오차를 줄일 수 있는 체계적이고 비선형적인 확률 분포로의 모델링이 가능하다면 더욱 빠르고 정확한 글리치 예측이 가능할 것이다. 또한 회로의 최적화 알고리즘에선 사이징 할 게이트의 선택에서 확률적 글리치 발생률과 동시에 각 노드의 임계 경로 결정성(criticality)를 고려하여 게이트 사이징을 수행한다면 전력소모량 및 지연시간의 최적화를 동시에 수행하여 회로 수율 상의 큰 개선이 가능하리라 생각된다. 마지막으로 확률적 경로 균등화 방법에선 글리치 발생률과 같이 노드사이에 크로스톡(crosstalk) 발생 문제를 확률적으로 고려한다면 회로의 면적, 지연시간 및 전력소모량을 모두 고려한 최적화 알고리즘이 가능할 것이다.

참 고 문 헌

- [1] M. Hashimoto, H. Onodera, and K. Tamaru, "A practical gate resizing technique considering glitch reduction for low power design," presented at Design Automation Conference, 1999. Proceedings. 36th, 1999.
- [2] A. Agarwal, K. Chopra, and D. Blaauw, "Statistical timing based optimization using gate sizing," presented at Design, Automation and Test in Europe, 2005. Proceedings, 2005.
- [3] M. Hashimoto and H. Onodeva, "Increase in delay uncertainty by performance optimization," presented at Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on, 2001.
- [4] X. Bai, C. Visweswariah, P. N. Strenski, and D. J. Hathaway, "Uncertainty aware circuit optimization," presented at Design Automation Conference, 2002. Proceedings. 39th, 2002.
- [5] E. T. A. F. Jacobs and M. R. C. M. Berkelaar, "Gate sizing using a statistical delay model," presented at Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, 2000.
- [6] S. Raj, S. B. K. Vrudhula, and J. Wang, "A methodology to improve timing yield in the presence of process variations," presented at Design Automation Conference, 2004. Proceedings. 41st, 2004.
- [7] C. Visweswariah, K. Ravindran, and K. Kalafala, "First-Order Parameterized Block-Based Statistical Timing Analysis," presented at TAU'04, 2004.
- [8] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," presented at Computer Aided Design, 2003. ICCAD-2003. International Conference on, 2003.
- [9] F. N. Najm, "Transition density, a stochastic measure of activity in digital circuits," presented at Design Automation Conference, 1991. 28th ACM/IEEE, 1991.
- [10] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah, "Parameterized Block-Based Statistical Timing Analysis with Non-Gaussian Parameters, Nonlinear Delay Functions," presented at DAC 2005, Proceedings, 2005.
- [11] C. E. Clark, "The Greatest Of A Finite Set Of Random Variables," Operations Research, vol. vol.9, pp. 85-91, 1961.
- [12] S. Kim, J. Kim, and S.-Y. Hwang, "Efficient algorithm for glitch power reduction [CMOS logic circuits]," Electronics Letters, vol. 35, pp. 1040-1041, 1999.
- [13] Y. J. Lim and M. Soma, "Statistical estimation of delay-dependent switching activities in embedded CMOS combinational circuits," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 5, pp. 309-319, 1997.

저 자 소 개



신 호 순(학생회원)
 2004년 서강대학교 컴퓨터학과 졸업.
 2006년 서강대학교 컴퓨터학과 대학원 졸업.
 2006년~현재 삼성전자 반도체총괄 재직

<주관심분야 : 회로 시간 분석, 회로 전력소모 및 시간 최적화, 회로 잡음 분석>



김 주 호(정회원)
 1987년 Univ. of Minnesota 학사
 1995년 Univ. of Minnesota 박사
 1996년 미국 Cadence Design System 수석연구원
 1997년~현재 서강대학교 조교수/부교수/정교수

<주관심분야 : CAD, 하드웨어 시스템 설계, 저전력 회로 설계, 통계적 시간 분석>



이 형 우(정회원)
 1999년 서강대학교 컴퓨터학과 학사 졸업.
 2001년 서강대학교 컴퓨터학과 석사 졸업.
 2005년 서강대학교 컴퓨터학과 박사 졸업.

2005년~현재 삼성전자 반도체총괄 책임연구원.
 <주관심분야 : CAD 및 회로 시간분석 및 검증 등>