

글리치 전력소모감소를 위한 게이트 사이징과 버퍼삽입 혼합기법

(Combination of Gate Sizing and Buffer Insertion Methods to Reduce Glitch Power Dissipation)

김 성 재 * 이 형 우 ** 김 주 호 ***
(Sungjae Kim) (Hyungwoo Lee) (Juho Kim)

요 약 본 논문은 CMOS 디지털 회로에서 글리치(glitch)에 의해 발생하는 전력소모를 줄이기 위한 효율적인 휴리스틱 알고리즘을 제시한다. 제안된 알고리즘은 사이징되는 게이트의 위치와 양에 따라 게이트 사이징을 세 가지 type으로 분류한다. 또한 버퍼삽입은 삽입되는 버퍼의 위치에 따라서 두 가지 type으로 분류한다. 글리치 제거 효과를 극대화하기 위해서 비용과 이득의 상관관계를 고려하여 하나의 최적화 과정 안에서 세 가지 type의 게이트 사이징과 두 가지 type의 버퍼삽입을 혼합한다. 제안된 알고리즘은 0.5 μ m 표준 셀 라이브러리(standard cell library)를 이용한 LGSynth91 벤치마크 회로에 대한 테스트 결과 효율성을 검증하였다. 실험결과는 평균적으로 69.98%의 글리치 감소와 28.69%의 전력감소를 얻을 수 있었으며 이것은 독립적으로 적용된 게이트 사이징과 버퍼 삽입 알고리즘에 의한 것 보다 좋은 결과이다.

Abstract We present an efficient heuristic algorithm to reduce glitch power dissipation in CMOS digital circuits. According to the location and the amount of sizing, gate sizing is classified into three types. The buffer insertion is divided into two types according to the location of the inserted buffer. The proposed algorithm combines three types of gate sizing and two types of buffer insertion into a single optimization process to maximize the glitch reduction effects of separate ones considering cost and gain trade off. The efficiency of our algorithm has been verified on LGSynth91 benchmark circuits with a 0.5 μ m standard cell library. Experimental results show an average of 69.98% glitch reduction and 28.69% power reduction which is much better than those of separate gate sizing or buffer insertion algorithms.

1. 서 론

개인용 계산기기와 무선 통신 시스템의 급격한 성장으로 인해 빠른 속도의 계산능력과 복잡한 기능의 요구가 증가되고 있다. 동시에, 휴대용 기기의 증가에 의해서 저 전력설계에 대한 중요성이 증가되고 있다. CMOS 디지털 회로에서 신호천이에 의해 발생하는 동적 전력 소모(dynamic power dissipation)는 회로전체 전력소모에 있어서 지배적인 양을 차지한다. 신호 천이는 기능성 천이

(functional transition)와 기능에 영향을 미치지 않는 불필요한 천이인 글리치로 나누어진다. 일반적으로 글리치에 의한 전력 소모가 전체 전력 소모의 20% - 70%를 차지하기 때문에 [1], 글리치를 제거함으로써 전체 전력소모 감소의 극대화를 기대할 수 있다.

게이트 사이징을 사용한 효율적인 전력 최적화에 대한 방법들이 [2]-[5]에서 제안되었다. 이 방법들은 주어진 지연시간 제약조건을 고려하면서 트랜지스터나 게이트들의 사이즈를 작게 하여 용량성 부하(load capacitance)의 양을 최적화 한다. 그러나 사이징한 게이트와 트랜지스터는 경로균등화(path balancing)를 깨뜨릴 수 있으며 그로 인해 추가적인 글리치가 생성될 수도 있다. 글리치를 고려한 게이트 사이징 알고리즘은 [5]에서 제안되었다. [5]에서는 글리치 감소에 따른 전력감소의 양에 따라 게이트를 선택하고 이를 사이징한다. 이 방법은 결과가 local

* 비 회 원 : (주)실리콘크래프트 CAD연구실 연구원
sjkim@siliconcraft.com

** 비 회 원 : 서강대학교 컴퓨터학과
feelg@sogang.ac.kr

*** 중신회원 : 서강대학교 컴퓨터학과 교수
jhkim@ccs.sogang.ac.kr

논문접수 : 2000년 9월 28일
심사완료 : 2001년 5월 24일

solution을 벗어나고 global solution에 가까운 결과를 얻기 위해 새로운 방법을 사용한다. 그러나 단지 게이트 사이징만으로 글리치를 감소시키는 것이며 게이트 사이징과 버퍼 삽입을 결합 때 더욱 좋은 결과를 얻을 수 있다. 뿐만 아니라, 게이트 사이징과 버퍼 삽입은 회로의 전력과 지연시간 최적화를 위해 효율적인 방법이 될 수 있다. [6] [7]에서는, 성능을 높이고 큰 용량성 부하를 구동하는 노드의 부하를 줄이기 위해서 버퍼 삽입을 사용하였다.

지금까지 다양한 게이트 사이징과 버퍼 삽입 기법이 제안되었으나, 동시에 단일화된 과정에서 글리치 감소를 위해 두 가지 기법을 적용하는 알고리즘은 없었다. 본 논문은 게이트 사이징과 버퍼 삽입이 혼합된 최적화 알고리즘을 단일화된 과정으로 제안한다. 우선, 게이트 사이징과 버퍼 삽입을 각각 세 가지와 두 가지로 분류한다. 각 게이트 사이징과 버퍼 삽입은 이들의 목적과 효용에 따라서 결합된다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 논문에서 사용되는 게이트 수준의 지연시간, 전력, 글리치 모델에 대해서 논의한다. 제 3장에서는 각각의 게이트 사이징과 버퍼 삽입에 대해서 논의하고 제 4장에서는 글리치 제거를 위한 실질적인 게이트 사이징과 버퍼삽입 혼합기법 알고리즘을 제안한다. 제 5장에서는 구현된 알고리즘에 의한 실험결과를 보여주고 마지막으로 제 6장에서는 휴리스틱 알고리즘으로 인한 한계와 결론으로 끝을 맺는다.

2. 지연시간, 전력, 글리치 모델

2.1 게이트 수준의 지연시간 모델

표준 셀 라이브러리를 사용하여 주어진 네트워크를 기술 매핑(technology mapping) 한 후, 신호지연은 라이브러리 모델 정보를 통해 얻어진다. 기본적인 지연시간 모델은 선형모델(linear model)을 사용한다. SIS[8]에서처럼 게이트 pin i 를 통과하는 지연시간 계산을 다음과 같은 식(1)에 의해 수행한다.

$$Delay(i) = Block(i) + Drive(i) \cdot Load \quad (1)$$

$Load$ 는 선택되어진 게이트에 의해서 구동되어지는 모든 용량성 부하를 말한다. $Block$ 과 $Drive$ 는 상승(rising)과 하강(falling) edge에 따라 바뀔 수 있으며 셀 라이브러리로부터 얻을 수 있다.

2.2 게이트 수준의 전력 모델

CMOS 게이트 i 에 의해서 소모되어지는 동적 전력의 평균값은 식(2)와 같이 나타낼 수 있다.

$$P_i = \frac{1}{2} \cdot C_i \cdot V_{dd}^2 \cdot f \cdot D(i) \quad (2)$$

C_i 는 용량성 부하, V_{dd} 는 공급전압, f 는 clock

frequency 그리고 $D(i)$ 는 게이트 i 천이밀도인데 이것은 매 클럭주기 마다 글리치를 포함하는 신호 천이의 평균적인 횟수를 나타낸다. 천이밀도의 개념은 [9]에서 소개되었다. 그러나 [9]의 모델에서는 신호의 공간(spatial)과 시간(temporal)의 상호관계에 대해서 고려하지 않았다. 천이밀도를 측정하기 위한 확률론적 접근법은 [10] [11]에 제안되었다. 이러한 접근법들은 비록 공간과 시간의 상호관계를 고려대상으로 삼고 있다하더라도, 실행시간(CPU time)이 오래 걸리므로 반복적인 적용에는 적합하지 않다.

전력최적기(power optimizer)는 최적화 과정에서 전력 측정을 반복해서 여러 번 수행한다. 이런 상황에서 가장 중요한 요소는 빠른 실행시간이다. 천이밀도를 계산하기 위한 간단한 방법은 임의로 생성된 입력 벡터(input vector)를 가지고 회로를 시뮬레이션 하는 것이다. 이러한 경우 원하는 정확도를 얻기 위해서 얼마나 많은 입력 벡터가 적용되어야 하는지가 결정되어야한다. 이 입력 벡터의 수를 결정하기 위한 접근법은 [12]에서 제안되었다. 에러 백분율(percentage error) ϵ 과 신뢰수준(confidence level) a 가 주어졌을 때 입력벡터의 최소개수는 식(3)에 의해 구한다.

$$L \geq \left(\frac{\text{erf}^{-1}\left(\frac{\alpha}{2}\right) \times s}{\epsilon \times p} \right)^2 \quad (3)$$

$\bar{\alpha}$ 는 실험에 의한 전력평균, s 는 표준편차를 나타낸다. $\text{erf}^{-1}\left(\frac{\alpha}{2}\right)$ 는 역 에러함수(inverse error function)로 정규분포로부터 얻을 수 있다. 일반적인 경우에 있어서 구해진 벡터의 수는 작아서 이 접근법은 매우 효율적이다. 비록 이 기법은 전체 회로의 평균 천이밀도만을 정확히 측정한다는 한계를 갖고 있다 하더라도, 빠른 실행시간과 정확한 글리치 측정을 위해서 이 기법이 사용되었다.

2.3 글리치 모델

글리치는 회로의 출력에 영향을 주지 않는 천이이나 회로의 출력에 영향을 주는 기능성 천이와 같이 전력을 소모한다. 그림 1 에 글리치의 예를 제시하였다. 글리치 생성전과 생성후의 신호 값의 변화가 없을 때의 글리치를 정적 글리치(static glitch)라 하고 신호 값이 변화했을 때의 글리치를 동적 글리치(dynamic glitch)라 한다. 글리치는 발생원인에 따라 생성 글리치(generated glitch)와 전파 글리치(propagating glitch)로 나누어진다[5].

생성 글리치는 글리치가 아닌 입력 신호에 대하여 도착 시간의 차이가 충분히 크고 입력 신호의 패턴이 글리치를 발생할 조건을 만족할 경우에 발생한다. 전파 글리치는 글리치를 포함한 입력 신호에 의해서 생성된다. 전파 글리치

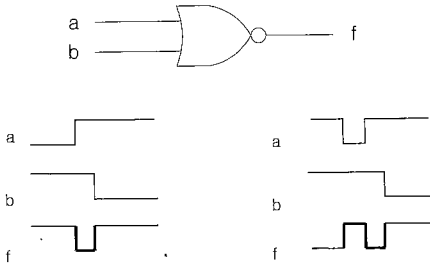


그림 1 Glitches (a) Static glitch, (b) Dynamic glitch

는 부 입력 신호(side-input signal)에 의해서 게이트를 통해 전달되기도 하고 소멸되기도 한다. 글리치의 신호너비(pulse width)또한 글리치가 전달되는지 혹은 소멸되는지를 결정하는 중요한 요소이다. 시뮬레이션을 통한 전력 측정 방법을 채택하였으므로, 신호의 천이 횟수(signal transition number), 도착시간(arrival time), 발생원인 그리고 신호너비를 쉽게 구할 수 있다.

글리치는 한 노드에 대해서 짝수의 신호천이를 만든다. 입력벡터가 한번 변할 때 주어진 하나의 입력 벡터에 대한 시뮬레이션에 의해 한 노드의 신호 천이횟수가 N_{tran} 면, 글리치에 의한 천이횟수는 다음과 같은 식(4)와 같이 나타낼 수 있다.

$$N_{glitch_tran} = \left\lfloor \frac{N_{tran}}{2} \right\rfloor \times 2 \tag{4}$$

Inertial delay $d_i(G)$ 를 갖는 게이트 G가 n 개의 fanin 을 n_1, n_2, \dots, n_n 과 같이 갖고 있고 각 fanin 노드의 도착 시간(arrival time)이 각각 $t_a(n_1), t_a(n_2), \dots, t_a(n_n)$ 이라고 가정하자. 또 각 fanin 노드의 최대 신호 pulse 너비가 $pw(n_1), pw(n_2), \dots, pw(n_n)$ 이라고 각각 가정하자. 이 때 $Max(t_a(n_1), t_a(n_2), \dots, t_a(n_3)) - Min(t_a(n_1), t_a(n_2), \dots, t_a(n_3)) > d_i(G)$ 면 게이트 G의 fanout node 상의 글리치는 입력 신호에 따라서 전파되어 질 수 있다. 비록 새로운 글리치가 생성되지 않았더라도, $Max(pw(n_1), pw(n_2), \dots, pw(n_3)) > d_i(G)$ 면 fanin 노드 상의 글리치는 게이트 G를 통해 전파되어질 수 있다.

3. 게이트 사이징과 버퍼삽입

글리치 감소 알고리즘에 대해 설명하기 전에 게이트 사이징과 버퍼삽입을 두 가지와 세 가지 type으로 분류하고 각각의 활용과 목적을 설명한다.

3.1 게이트 사이징

전력 최적화(power optimization)를 위한 게이트 사이

즈 조정 기법은 용량성 부하를 감소시키기 위해서 게이트의 사이즈를 최소화한다. 그러나 사이징된 게이트는 회로의 지연시간을 변경하므로 추가적인 전력을 소모하는 새로운 글리치를 생성할 수 있다. 따라서 게이트 사이즈 조정에 있어서 글리치의 생성과 소멸을 고려해야 한다. 그림 2는 게이트 사이징의 가능한 세 가지 type을 보인다.

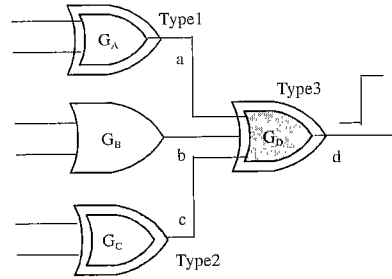


그림 2 Type1, Type2 and Type3 Gate Sizing

3.1.1 Type 1 게이트 사이징

Type1 게이트 사이징은 경로균등화를 위한 게이트 지연사이징이다. 그림 2에서와 같이, $t_a(a) < Max(t_a(b), t_a(c)) - d_i(G_D)$ 일 때 노드 d상에 글리치가 생성될 수 있다. 만약 게이트 G_A 를 $t_a(a) \geq Max(t_a(b), t_a(c)) - d_i(G_D)$ 를 만족하는 작은 게이트로 사이징 한다면 노드 d상의 글리치는 제거된다. 사이징에 의해 게이트 지연시간이 증가하므로 type1 에 의해 사이징되는 게이트는 출력단 노드의 슬랙이 양수이어야 한다. 여기서 슬랙이란 출력단에 안정된 신호의 전파를 위해서 입력단 신호가 도착되어야 하는 시간을 의미하구 요구시간(required time) - 도착시간(arrival time)으로 주어진다.

3.1.2 Type 2 게이트 사이징

Type2 게이트 사이징은 경로균등화를 위한 게이트 업사이징 이다. 그림 2 에서와 같이, $t_a(c) > Min(t_a(a), t_a(b)) + d_i(G_D)$ 일 때 노드 d에 글리치가 생길 가능성이 있다. 만약 게이트 G_C 가 $t_a(c) \leq Min(t_a(a), t_a(b)) + d_i(G_D)$ 을 만족하는 큰 게이트로 사이징 된다면 글리치가 제거될 수 있다. 업사이징으로 인한 용량성 부하의 증가로 회로 전체의 전력소모가 증가하기 때문에 사이징으로 인한 추가적인 전력 소모보다 글리치 제거에 의한 전력감소가 클 때 type 2게이트 사이징을 수행한다.

3.1.3 Type 3 게이트 사이징

작은 게이트로의 사이징은 그 게이트의 inertial delay 를 증가시킨다. 그림 2 에서 노드 d에 글리치가 존재한다

고 가정하자. 이때, 조건 $d_t(G_D) > \text{Max}(t_a(a), t_a(b), t_c(c)) - \text{Min}(t_a(a), t_a(b), t_c(c))$ 와 $d_t(G_D) > \text{Max}(pw(a), pw(b), pw(c))$ 를 만족하도록 작은 게이트로 G_D 를 사이징한다면 노드 d 상에서 글리치는 생성되지 않고 또한 전파 글리치는 inertial delay 증가에 따른 제거효과(filtering effect)에 의해 제거된다.

3.2 버퍼 삽입

3.2.1 Type 1 버퍼 삽입

Type 1 버퍼삽입은 경로균등화를 위해 지연 버퍼(delay buffer)를 삽입하는 것이다. 그림 3 에서, $t_a(b) < \text{Max}(t_a(a), t_a(c)) - d_t(G_D)$ 면 노드 g 상에는 글리치가 생성될 수 있다. Type 1 게이트 사이징은 이와 같은 경우의 글리치를 제거할 수 있다. 그러나 라이브러리에 적당한 게이트가 존재하지 않는 경우에 경로균등화를 위해 버퍼 삽입을 선택한다. 노드 b의 슬랙은 양수이어야 하고 type 1 버퍼 삽입을 선택하기 위해서는 소멸되는 글리치에 의한 전력감소가 삽입된 버퍼에 의한 전력소모 보다 커야 한다.

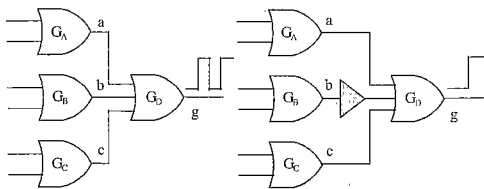


그림 3 Type 1 버퍼 삽입

3.2.2 Type 2 버퍼 삽입

그림 4에 type 2 버퍼삽입을 제시하였다. 노드 g 상에 글리치가 존재하고 $d_t(G) < \text{Max}(pw(a), pw(b))$ 라고 가정하자. 만약 $d_t(G') > \text{Max}(pw(a), pw(b))$ 를 만족하는 게이트 G' 보다 작은 게이트 G' 가 라이브러리에서 존재한다면 노드 g 상의 글리치를 제거하기 위해서 type 3 게이트 사이징이 사용될 수 있을 것이다. 그러나 라이브러리가 이와 같은 게이트를 갖고 있지 않다면 글리치는 제거되지 못하고

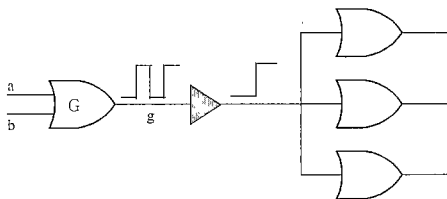


그림 4 Type2 버퍼 삽입

전파되어 동적 전력을 소모하게 된다. 이 경우, $\text{Max}(pw(a), pw(b))$ 보다 큰 전파지연을 갖는 작은 버퍼는 글리치의 전파를 막을 수 있다. 노드 g 상의 글리치는 제거되지 못하나 전파글리치는 제거된다. 버퍼는 그 자체가 전력을 소모하기 때문에 글리치 제거에 의해 감소되는 전력보다 많은 전력을 소모하는 버퍼는 삽입되어질 수 없다.

4. 글리치제거 알고리즘

4.1 전체적인 알고리즘

그림 5에 전체적인 알고리즘을 제시하였다. 먼저 각 노드의 신호 도착시간, 요구시간(required time), 슬랙등의 정보를 얻기 위해서 위상학적 시간 분석(topological timing analysis)이 수행된다. 다음으로 event-driven simulation 에 의해 전력측정이 이루어진다. 입력 벡터의 수는 2.2 절에서 설명한 방법에 의해 결정된다. 전력 측정 과정동안, 각 노드의 기능성천이 횟수가 기록되고 글리치 천이 횟수는 식(4)에 의해 계산된다. 전력측정 후에 글리치를 포함하는 각 게이트에 대해서 전력이득(power gain), 전력비용(power cost), 이득계수(gain factor)를 측정한다. 이러한 요소들은 본 논문의 다음 소절에서 설명한다. 전력소비 측정과정에서, 게이트 사이징과 버퍼삽입은 혼합되고 사이징 후보 게이트와 버퍼 삽입 후보가 채택된다. 이득계수가 양수인 모든 게이트 중 가장 큰 것이 선택되어 지고 게이트 사이징과 버퍼삽입이 포함된 글리치 제거 과정이 적용된다. 이러한 과정은 양수의 이득계수를 가지는 게이트가 없을 때까지 반복된다.

```

1:GlitchReduction()
2:{
3:  repeat {
4:    do Timing Analysis;
5:    do Power estimation;
6:    for each gate G which has  $N_{glitch\_rem}(G) > 0$ 
7:       $\Delta P(G)_{gain} = \text{GainEstimation}(G)$ ;
8:       $\Delta P(G)_{cost} = \text{CostEstimation}(G)$ ;
9:       $GF(G) = \Delta P(G)_{gain} - \Delta P(G)_{cost}$ ;
10:   }
11:    $G' = \text{select the gate which has}$ 
12:     the largest positive  $GF(G)$ ;
13:   if ( $G'$ )
14:     GateSizingBufferInsertion( $G'$ );
15: } until (no gate G has  $GF(G) > 0$ );
16:}
    
```

그림 5 글리치제거 알고리즘

4.2 전력이득측정

게이트 사이징과 비퍼삽입에 의해, 게이트 G의 출력단 글리치가 제거되었을 때, 감소되는 전력의 양이 전력이득 $\Delta P(G)_{gain}$ 이다. 전력 측정이 event-driven simulation에 의해 수행되기 때문에, 전력이득은 시뮬레이션과 정동안 생성된 event들로부터 측정되어진다. 정확한 전력이득측정은 event들 사이의 복잡한 관계 때문에 매우 어렵다. 그림 6에 greedy방식의 전력이득측정 알고리즘을 제시하였다. 게이트 G의 출력단의 event들에 대해서 글리치인지 아닌지를 가리는 검사가 수행된다. 글리치는 노드 상에서 짝수개의 신호천이를 만든다. 따라서, 만약 노드에 짝수개의 event들이 존재하면 모든 event들은 글리치이다. 홀수 개의 event들 중에서는 처음 또는 마지막 event 하나는 글리치가 아닌 천이이다. 게이트 사이징과 비퍼삽입에 의해서 글리치가 제거될 때, type3 게이트 사이징을 제외한 다른 모든 게이트 사이징과 비퍼삽입은 신호지연을 증가시킨다. 따라서 대부분의 경우, 글리치가 아닌 천이는 노드상의 event들 중 가장 늦은 도착신호를 갖는 event가 된다. 한 event가 글리치로 판명되면 이것으로부터 영향받는 모든 event들의 전력소모가 $\Delta P(G)_{cost}$ 에 추가된다.

```

1:GainEstimation(G)
2:{
3:  $\Delta P(G)_{gain} = 0$ ;
4: g = output node of gate G;
5: for each event e on g {
6:   if (e is glitch) {
7:     for each event e' occurred by e {
8:       g' = node to which e' is attached;
9:        $\Delta P(G)_{gain} += \frac{1}{2} \cdot C_{g'} \cdot V_{dd}^2 \cdot f$ ;
10:    }
11:  }
12: }
13: return  $\Delta P(G)_{gain}$ ;
14: }
```

그림 6 전력이득측정 알고리즘

4.3 전력비용측정

전력비용측정 과정에서 게이트 사이징과 비퍼삽입은 하나의 과정 안에서 혼합되어진다. 그림 7에 전력비용측정 알고리즘이 제시되었다. 제안된 알고리즘은 게이트 업사이징과 비퍼삽입을 사용하였기 때문에 글리치를 제거하기 위한 추가적인 전력 소모가 발생한다. 전력비용 $\Delta P(G)_{cost}$

는 게이트 G의 출력단의 글리치를 제거하기 위해 소모되는 전력의 양을 나타낸다. 생성 글리치는 회로를 통해 전파되기 때문에, 전파 글리치 보다 우선적으로 제거하는 것이 더욱 좋은 결과를 산출한다. Type 3 게이트 사이징은 생성 글리치와 전파 글리치를 모두 제거할 수 있다. 그러나 type 3 게이트 사이징이 적용되면, 이 게이트의 출력단에 존재하는 게이트들의 출력단에서 발생할 수 있는 생성 글리치의 제거가 제한될 수 있다. 따라서 type 3 게이

```

1:CostEstimation(G)
2:{
3:  $\Delta P(G)_{cost} = 0$ ;
4: if (generated glitches) {
5:   FindCandidate(type1 gate sizing);
6:   if (candidates found) {
7:      $\Delta P(G)_{cost} += \text{ComputeCost}(G, \text{candidate})$ ;
8:     if (!generate glitch)
9:       goto prop;
10:  }
11: FindCandidate(type2 gate sizing);
12: if (candidates found) {
13:    $\Delta P(G)_{cost} += \text{ComputeCost}(G, \text{candidate})$ ;
14:   if (!generate glitch)
15:     goto prop;
16:  }
17: FindCandidate(type1 buffer insertion);
18: if (candidates found) {
19:    $\Delta P(G)_{cost} += \text{ComputeCost}(G, \text{candidate})$ ;
20:   if (!generate glitch)
21:     goto prop;
22:  }
23: }
24:prop :
25:if (propagating glitches) {
26:  FindCandidate(type3 gate sizing);
27:  if (candidates found){
28:     $\Delta P(G)_{cost} += \text{ComputeCost}(G, \text{candidate})$ ;
29:    if (!propagating glitch)
30:      return  $\Delta P(G)_{cost}$ ;
31:  }
32: FindCandidate(type2 buffer insertion);
33: if (candidates found){
34:    $\Delta P(G)_{cost} += \text{ComputeCost}(G, \text{candidate})$ ;
35:  }
36: }
37: return  $\Delta P(G)_{cost}$ ;
38: }
```

그림 7 전력비용측정 알고리즘

트 사이징은 전파 클리치를 제거하기 위한 용도로만 적용된다. 그림 7에서 8번째 줄과 29번째 줄에서는 게이트 G가 생성 클리치 또는 전파 클리치를 포함하는지를 조사한다. 이것은 event와 이 event를 발생시킨 event들 사이의 상호관계를 분석함으로써 결정된다. 그림 7에서 computecost 함수는 선택되어진 후보게이트가 사이징 또는 버퍼가 삽입되었을 때의 추가적인 전력 소모를 계산한다. 이 또한 event들의 상호 관계를 이용한다. 각각의 event에는 그것을 발생시킨 원인이 되는 event를 저장한다. 게이트가 사이징 되거나 버퍼가 삽입되면 그에 따른 event의 지연 변화를 알 수 있으므로 게이트의 논리특성과 그 게이트의 부 입력 신호 event의 지연 및 논리 형태를 고려하면 그 event의 소멸 혹은 새로운 event의 생성을 예측 할 수 있다. 그 event의 소멸 혹은 다른 event의 생성은 신호 천이의 소멸 혹은 증가를 의미하므로 라이브러리로 부터 얻어진 각각의 게이트들의 용량성 부하를 고려하여 전력의 감소, 증가를 예측한다.

4.4 이득계수

게이트 G의 이득계수 $GF(G)$ 는 전력이득 $P(G)_{gain}$ 과 전력비용 $P(G)_{cost}$ 의 차다. 비교적 큰 양의 이득계수는 적은 전력비용으로 많은 클리치를 제거함을 의미한다. 음의 값의 이득계수는 클리치 제거를 통한 전력 감소보다 클리

치 제거를 위한 전력비용이 더 큰 것을 의미한다. 이 경우, 비록 클리치는 제거되어지나 회로의 전체 전력은 증가된다. 그러므로 게이트 사이징과 버퍼삽입을 포함하는 클리치 감소과정은 제한되어야 한다.

5. 실험 결과

제안된 알고리즘은 Ultra Sparc I workstation상에서 C언어로 구현되었으며 LGSynth91 벤치마크 회로를 사용해 검증되었다. 회로들의 초기 매핑은 SIS[8]를 이용하였고 실제공정에서 사용되는 0.5 μ m 표준 셀 라이브러리를 target으로 하였다. 이 과정에서 사용된 명령어는 'map -n 1.0 AFG'이며 이것은 회로의 크기를 최소화한다. 실험에서 사용된 라이브러리는 buffer, inverter, nand, nor, and, or 게이트로 구성되어있으며 라이브러리 데이터를 SIS 라이브러리 형식으로 변환하여 사용하였다. 전력 측정은 90%의 신뢰수준과 10%의 에러 백분율에 따라 수행되었다. Event-driven simulation을 위한 입력 벡터의 최소 개수는 식(3)에 의해서 계산되었고 모든 회로에 대해서 3이었다. 게이트의 inertial delay는 게이트의 지연을 그대로 사용했다. 표 1에 실험 결과를 제시하였다. 표 1의 'Gate Sizing' 항목은 전력 비용추정을 type1, type2, type3 게이트 사이징만을 이

표 1 실험 결과

Circuit Name	#of gates	Gate Sizing						Buffer Insertion					Gate Sizing + Buffer Insertion							
		T1S	T2S	T3S	Glitch reduction ratio(%)	Power reduction ratio(%)	Run Time (sec)	T1B	T2B	Glitch reduction ratio(%)	Power reduction ratio(%)	Run Time (sec)	T1S	T2S	T3S	T1B	T2B	Glitch reduction ratio(%)	Power reduction ratio(%)	Run Time (sec)
comp	212	3	6	6	69.57	24.40	1	15	5	61.92	15.88	2	2	6	6	8	3	88.23	30.96	1
C432	222	5	4	1	4.73	2.23	1	12	1	63.47	24.37	1	4	4	1	11	1	70.25	26.46	1
C890	378	10	0	4	21.47	9.04	2	19	3	48.53	16.05	3	11	0	4	17	2	57.20	20.38	3
i7	470	82	0	17	58.03	18.31	16	87	0	78.35	11.95	16	82	0	17	26	0	78.35	19.57	16
C499	602	21	17	12	14.03	7.95	15	10	13	5.37	2.70	8	12	16	12	15	8	45.56	22.34	14
C1908	672	13	8	9	41.20	21.06	8	24	12	36.17	15.37	11	12	8	9	17	4	58.64	26.45	10
too_large	764	21	7	18	57.35	18.76	11	53	11	77.69	15.20	16	16	6	17	38	3	84.06	19.91	13
rot	786	28	8	17	56.93	15.28	13	37	10	86.24	18.14	13	26	8	18	31	6	91.18	20.91	15
C1355	978	41	0	10	59.66	32.89	22	44	9	44.48	23.32	25	42	0	10	11	1	61.66	33.40	24
C2607	1024	32	5	21	58.19	23.07	18	44	20	60.51	18.68	25	32	5	19	22	12	74.57	25.04	26
C3540	1321	47	8	29	56.83	31.93	46	95	32	29.31	14.73	87	68	9	34	42	11	71.91	40.17	68
C5315	2123	91	23	80	54.13	27.98	190	167	58	58.94	26.07	241	85	26	73	127	23	70.40	32.19	226
pair	2160	63	5	14	41.18	14.19	71	107	18	58.54	17.51	97	53	4	12	88	4	80.33	25.38	92
k2	2225	64	4	35	79.78	44.46	43	204	18	82.62	37.90	101	60	4	34	59	4	94.77	50.10	58
dalu	2380	61	6	39	71.64	47.73	104	180	96	67.09	37.47	317	60	6	40	92	14	81.06	49.21	159
C7522	3274	215	28	101	24.38	13.47	770	191	64	16.31	6.75	767	204	19	79	158	32	40.65	21.00	819
i10	3728	81	14	24	32.46	17.27	247	149	23	26.33	10.98	410	85	17	27	115	7	51.79	25.24	328
i8	3914	146	11	21	50.83	24.82	204	233	22	40.51	15.93	327	156	11	23	181	3	62.94	27.78	319
des	7061	168	27	89	48.73	21.83	1024	150	38	47.19	20.23	976	124	20	68	91	21	66.02	28.63	1134
Average					47.43	21.93				52.08	18.38							69.98	28.69	

용해서 수행했을 때 얻어진 결과이다. Type1과 type2 버퍼삽입만을 이용한 결과는 'Buffer Insertion' 항목에 수록했다. 'Gate sizing+Buffer Insertion' 항목은 모든 type의 게이트 사이징과 버퍼삽입을 적용해서 글리치를 제거한 결과를 보여준다. 표 1에서 T1S, T2S, T3S, T1B, T2B는 각각 type1, type2, type3 형태로 사이징된 게이트의 수와 type1, type2 형태로 삽입된 버퍼의 수를 의미한다. 글리치감소비율과 전력감소비율은 실행 시간과 함께 각 항목에 제시하였다. 글리치감소는 게이트 사이징만으로 평균적으로 47.43%, 버퍼삽입으로는 52.08%, 게이트 사이징과 버퍼삽입 혼합방법에 의해서는 69.98%의 감소를 얻을 수 있었다. 전력감소에 있어서는 게이트 사이징과 버퍼삽입을 동시에 적용한 경우 평균적으로 28.69%로 가장 좋은 전력감소 결과를 얻었다. 그러나 글리치 감소와 달리 전력감소에 있어서는 버퍼삽입만을 수행한 것보다 게이트 사이징만을 수행한 것이 더욱 좋은 결과를 얻을 수 있었다. Type 2 형태로 삽입된 버퍼 수는 상대적으로 작았다. Type 2 게이트 사이징은 업사이징으로 글리치를 제거하기 위해서 전력을 소모하기 때문에 이러한 형태로 사이징된 게이트 수는 작다. 모든 경우에 있어 실행시간은 매우 빨랐다. 적은 수의 입력벡터를 갖는 event-driven simulation과 빠른 전력이득 측정방법을 사용함으로써 빠른 실행시간을 얻을 수 있었다. 표 2는 초기회로들, 게이트 사이징

후의 회로들, 버퍼삽입후의 회로들, 게이트 사이징과 버퍼삽입의 혼합방법이 적용된 후의 회로들에 대한 각각의 글리치 비율을 보인다. 제안된 알고리즘은 글리치를 줄이기 위한 것이었기 때문에, 적은 글리치를 갖고 있는 회로들은 상대적으로 적은 전력감소 비율을 보였다. 초기 회로에서 *too_large* 회로와 *rot* 회로는 26% 이하의 글리치비율을 갖고 있었다. 이러한 회로들에 대해서 전력감소비율은 글리치감소 비율보다 낮은 결과를 나타내었다.

6. 결론

본 논문에서는 게이트 사이징과 버퍼삽입에 의한 글리치 감소 알고리즘을 제안하였다. 논문의 알고리즘은 하나의 최적화 과정 안에서 게이트 사이징과 버퍼삽입을 혼합한다. 알고리즘에 의해서 69.98%의 글리치 감소와 28.69%의 전력감소를 얻을 수 있다. 게이트 사이징과 버퍼삽입을 각각 적용했을 때는 혼합했을 때 보다 결과가 좋지 않았다. 제안된 알고리즘은 비록 greedy 알고리즘이지만 좋은 결과를 보였다. 만약 전체적으로 최적화된 알고리즘(global optimal algorithm)과 정확한 전력이득 산출방법이 적용된다면 더욱 좋은 결과를 얻을 수 있겠지만 이러한 경우에는 수행시간의 급격한 증가가 문제될 것이다.

참고 문헌

- [1] Amelia Shen, Abhijit Chosh, Srinivas Devadas and Kurt Keutzer, On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Network, in *Proceedings of the 1992 IEEE/ACM International Conference of Computer-Aided Design*, pp.402-407, November 1992.
- [2] Sachin S. Sapatnekar and Weitong Chuang, Power vs Delay in Gate Sizing: Conflicting Objectives?, in *Proceedings of the 1995 IEEE/ACM International Conference of Computer-Aided Design*, pp.463-466, 1995.
- [3] Manjit Borah, Robert Michael Owens and Mary Jane Irwin, Transistor Sizing for Minimizing Power Consumption of CMOS Circuits under Delay Constraint, in *Proceedings of the 1995 International Symposium on Low Power Design*, pp.167-172, 1995.
- [4] D. Sheng Chen, and M. Sarrafzadeh, An Exact Algorithm for Low Power Library-Specific Gate Re-Sizing, in *Proceedings of the 33rd Design Automation Conference*, pp. 783-788, June 1996.
- [5] M. Hashimoto, H. Onodera, and K. Tamaru, A Power Optimization Method Considering Glitch

표 2 글리치 비율

Circuit Name	Initial(%)	Gate Sizing(%)	Buffer Insertion(%)	Gate Sizing + Buffer Insertion(%)
comp	46.89	18.88	21.23	7.99
C432	40.85	39.80	19.73	16.52
C880	40.3	34.79	24.71	21.66
i7	32.63	16.76	8.02	8.01
C499	51.63	48.23	50.22	36.19
C1908	46.89	34.93	35.36	26.37
too_large	24.74	12.99	6.51	4.92
rot	25.96	13.20	4.36	2.89
C1355	48.6	29.21	35.18	27.98
C2607	33.46	18.19	16.25	11.35
C3540	58.93	37.37	48.86	27.67
C5315	50.46	32.14	28.02	22.03
pair	34.58	23.70	17.38	9.12
k2	59.8	21.77	16.73	6.27
dalu	62.79	34.07	33.05	23.42
C7552	52.96	46.28	47.53	39.79
i10	46.75	38.16	38.69	30.15
i8	46.78	30.60	33.10	24.01
des	41.93	27.50	27.76	19.96

Reduction by Gate Sizing, in *Proceedings of the International Symposium on Low Power Design*, pp. 221-226, August 1998.

- [6] N. Hedenstierna and K. Jeppson, CMOS Circuit Speed and Buffer Optimization, *IEEE Transaction on Computer-Aided Design*, Vol. 6, No. 2, pp. 270-281, March 1987.
- [7] J. Kim, C. Bamji, Y. Jiang, and S. Sapatnekar, Concurrent Transistor Sizing and Buffer Insertion by Considering Cost-Delay Tradeoffs, in *Proceedings of the International Symposium on Physical Design*, pp. 130-135, April 1997.
- [8] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R.Murgai, A.Saldanha, H.Savoj, P.R. Stephan, R.K.Brayton, and A.L. Sangiovanni-Vincentelli. SIS: A system for sequential circuit synthesis, Technical Report UCB/ERL M92/41, Electronics Research Lab, University of California at Berkeley, 1992.
- [9] F. Najm, Transition Density, A New Measure of Activity in Digital Circuits, *IEEE Transaction on Computer-Aided Design*, Vol.10, No 4, pp.310-323, February 1993.
- [10] H. Mehta, M. Borah, R. M. Owens and M. J. Irwin, Accurate Estimation of Combinational Circuit Activity, in *Proceedings of the 32nd IEEE/ACM Design Automation Conference*, pp. 618-622, June 1995.
- [11] C-S. Ding, C-Y. Tsui and M. Pedram. Gate-level power estimation using tagged probabilistic simulation, *IEEE Transaction on Computer-Aided Design*, Vol. 17, No 11, pp. 1099-1107, November 1998.
- [12] R. Burch, F. Najm, and T. Trick, McPOWER: A Monte Carlo approach to power estimation, in *Proceedings of the 1992 IEEE/ACM International Conference on Computer-Aided Design*, pp. 90-97, November 1992.



이 형 우

1999년 2월 서강대학교 컴퓨터학과 학사. 2001년 2월 ~ 서강대학교 컴퓨터학과 대학원 석사. 현재 서강대학교 컴퓨터학과 대학원 박사과정. 관심분야는 CAD 및 저전력 VLSI 설계등임



김 주 호

1987년 University of Minnesota at Minneapolis 전산과 학사. 1995년 University of Minnesota at Minneapolis 전산과 박사. 1995 ~ 1996년 Cadence Design Systems Inc. San Jose, California, Senior Member of Technical Staff. 1997년 3월 ~ 현재 서강대학교 컴퓨터학과 교수. 관심 분야는 CAD 및 하드웨어 시스템 설계 등임



김 성 재

1998년 2월 서강대학교 컴퓨터학과 학사. 2000년 2월 ~ 서강대학교 컴퓨터학과 대학원 석사. 현재 (주)실리콘크래프트 CAD 연구실 연구원. 관심분야는 CAD 및 저전력 VLSI 설계등임