기능적 오류방지를 위한 크로스톡 글리치 제거 알고리즘

이 형 우, 김 유 빈, 김 주 호
서강대학교 컴퓨터학과
전화 : 02-706-3997 / 핸드폰 : 011-1761-1267

Crosstalk Glitch Elimination Algorithm for Functional Fault Avoidance

Hyungwoo Lee, Youbean Kim, Juho Kim
Dept. of Computer Science, Sogang University
E-mail : hwlee@sogang.ac.kr

Abstract

Our paper focus on crosstalk noise problem, especially crosstalk glitch that occurs when victim is stable state and aggressor is transitive state. This generated glitch weigh with the functional reliability if the glitch is considerable . In this paper, we use buffer insertion, down sizing, buffer insertion with up-sizing methods concurrently. These methodologies use filtering effects which gates that have bigger noise margin than glitch width eliminates glitch. In addition, we do limited optimization in boundary of node's slack. Therefore, the operated node's changes are for nothing in other node's slack.

I. 서론

반도체 기술의 고집적화(deep-submicron)로 과거에 비해 같은 면적에 집적되는 트랜지스터의 수가 수천만 개 이상으로 월등히 높아졌다. 그에 따라 기존에는 크게 심각하게 고려하지 않았던 요소들이 회로의 기능이나 성능에 중대한 영향을 미치는 새로운 문제로 떠오르게 되었는데, 그 중 대표적인 것이 신호 무결성(SI : Signal Integrity) 문제이다. 신호 무결성 문제에는 IR-drop, electomigration, 크로스톡(crosstalk) 등이 있으며, 본 논문에서는 이 중 회로에 가장 큰 영향을 미칠 수 있는 크로스톡 노이즈 문제에 주목한다.

크로스톡은 인접한 연결선들의 신호 흐름이 상호 영향을 주고받아서 생기는 현상으로 지연 시간 변화 (delay degradation), 글리치, overshooting, undershooting 등의 문제를 일으킨다. 이 중 글리치는 불필요한 천이가 발생 하는 것으로, 추가적인 전력 소모 및 논리 값의 오류(functional fault)문제를 발생 시키게 되는데, 본 논문은 버퍼 삽입[1]과 게이트 사이징[2]을 통해 이러한 글리치를 제거하여 논리 값의 오류 확률을 낮추는 것이 목적이다.

본 논문은 수학적 글리치 모델링[3]을 통해 빠르고 정확하게 글리치의 최대 전압(voltage peak)과 폭(width)을 측정하여 일정 기준 이상의 고려 대상 네트, 즉 크로스톡 노이즈에 의해 글리치가 발생 가능한 네트들을 추려내도록 하였으며, 후배선(post-route) 과정에서의 버퍼 삽입과 게이트 사이징을 동시에 고려하여 노이즈 마진(noise margin)을 이용한 효율적인 글리치 제거가 가능하도록 하였다. 또한 각 노드의 슬랙(slack)한도 내에서 최적화를 수행 하도록 하여, 한 노드의 변화가 인접한 노드에 미치는 영향을 최소화 시켰다.

II. 이론적 배경

크로스톡 노이즈는 인접한 연결선들 간의 상호 전기용량(coupling capacitance)로 인해 서로 영향을 주고받아 생기는 전기적 현상으로 신호 천이 상태에 따라 지연 시간 또는 논리 값에 영향을 미치게 된다. 과거에 비

해 면적당 집적되는 트랜지스터의 수가 월등히 늘어남에 따라 크로스톡에 의한 영향은 상대적으로 그 비중이 점차 높아져 가고 있다. 크로스톡 현상을 분석함에 있어 영향을 주는 연결선을 어그레서라 부르고, 영향을 받는 연결선을 빅팀이라 부른다[4]. 본 논문의 알고리즘은 단일 어그레서와 단일 빅팀을 전제로 하였다.

글리치의 크기는 높이와 폭으로 결정되며, 본 논문은 V와 A의 전압 변화량을 미분 방정식으로 수식화하여 V와 A의 시간에 따른 전압 파형 함수를 구하는 방법으로 글리치를 모델링한다.

V, A의 최종적인 함수:

$$v(t) = Vdd\frac{\alpha_{va}}{\tau_1 - \tau_2}(e^{-t/\tau_1} - e^{-t/\tau_2})$$

$$a(t) = Vdd(\frac{\tau_v - \tau_1}{\tau_1 - \tau_2}e^{-t/\tau_1} + \frac{\tau_v - \tau_2}{\tau_2 - \tau_1}e^{-t/\tau_2} + 1)$$

최종 유도된 식은 빅팀이 0로 안정화 되어 있고 어그레서가 상승 할 때의 각각 V와 A의 파형을 그려주는 함수식이다. 이 경우 $v(t)$함수에서 그려 주는 파형이 결국 크로스톡에 의한 글리치의 파형이 된다.

글리치 모델링의 목적은 일정 범위 이상의 고려 대상이 되는 글리치 탐색에 있으며, 그 범위는 앞서 말한 바와 같이 글리치의 높이와 폭으로 결정한다.

III. 글리치 제거를 위한 게이트 사이징 및 버퍼 삽입

3.1 버퍼 삽입

그림 3-1에서는 보는 경우와 같이 어그레서의 신호 도착 시간이 빅팀의 신호 도착 시간보다 빠르고 빅팀의 슬랙에 여유가 있으며, 빅팀에서 파생되는 reconvergent fan-in이 없을 때, 버퍼 삽입을 한다. 이 때 삽입 되는 버퍼는 기존 게이트의 노이즈 마진을 합하였을 때, 발생된 글리치의 모델링에서 얻어낸 글리치의 폭보다 큰 것을 선택해야 한다. 물론 빅팀 슬랙의 범위 내에서 해당 버퍼가 선택되어져야 한다.

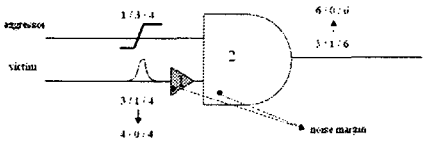

그림 3-1. 버퍼 삽입의 조건과 실제 버퍼 삽입의 예

위의 그림에서 보듯, 어그레서의 신호 도착 시간이 1ns였고, 빅팀의 신호 도착 시간이 3ns였으로 이 때 크로스톡 글리치가 발생한 것으로 본다. 또한 빅팀의 슬랙에 1만큼의 여유가 있고 지연 시간 1ns의 특성을 가진 버퍼를 삽입하여 노이즈 마진을 증가시켜 글리치가 전파 되면서 제거된다. 버퍼 삽입의 경우 삽입으로 인한 이후 노드의 신호 도착 시간에 영향을 미쳐 슬랙의 변화가 올 수 있기 때문에 이후 노드들에 대한 슬랙 업데이트 과정이 필요하다. 그러나 빅팀의 슬랙 한도 내에서 버퍼를 삽입하였고 이후 노드에 미치는 영향도 결국은 빅팀의 요구 시간(requirement time)을 넘어 설 수 없으므로 회로 전체에 큰 혼란을 주는 것은 아니다. 만약 슬랙의 여유 내에서 버퍼 삽입만으로 글리치 폭보다 큰 노이즈 마진을 얻을 수 없다면, 뒤에서 설명할 버퍼 삽입과 업 사이징을 동시에 사용하는 알고리즘을 적용하여 충분한 노이즈 마진을 확보하는 방법을 취한다. 이 때 버퍼 삽입과 업 사이징을 동시에 사용하는 이유는 버퍼 삽입으로 가능한 빅팀의 슬랙을 다 사용하고, 부족한 노이즈 마진에 대해서 업 사이징 하도록 하여, 해당 네트의 슬랙 변화가 이후 네트에 미치는 영향을 최소화하기 위해서이다.

3.2 다운사이징

버퍼 삽입의 조건과 같이 어그레서의 신호 도착 시간이 빅팀의 신호 도착 시간보다 빨라 크로스톡 글리치가 발생되는 조건이기는 하나, 빅팀의 슬랙에 여유가 없어 버퍼를 삽입할 수 없는 경우에는 그림 3-2에서 보는 바와 같이 어그레서 네트의 다운사이징을 통해 어그레서와 빅팀의 신호 도착 시간을 같게 조정하는 방법을 취한다. 이 때 라이브러리(library)에 다운사이징을 통해 빅팀의 신호 도착 시간과 같게 만들 수 있을 만큼의 사이즈를 가진 게이트가 존재해야 한다. 이 경우도 글리치가 통과하는 게이트를 업 사이징해서 노이즈 마진을 확보하면 되겠지만, 앞서 말했듯이 본 논문은 해당 네트의 슬랙의 변화가 이후 네트의 슬랙 변화를 최소화하는 방향으로 알고리즘을 적용한다.

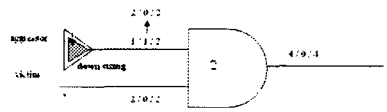

그림 3-2. 다운사이징의 조건과 실제 다운 사이징의 예

위의 그림에서 보듯이 어그레서의 신호 도착 시간이 1ns로 빅팀의 그것보다 빠르지만 빅팀에 버퍼를 삽입할만한 슬랙의 여유가 없다. 이 경우 어그레서이전 노드를 다운사이징 하여 빅팀의 신호 도착 시간인 2ns와 같게 조정해 주고 있다. 이 경우 주의해야 하는 것은 그림 3-3에서 보는 바와 같이 만약 다운사이징을 하는 노드 이전의 네트에 글리치 제거를 위한 버퍼 삽입이 이루어

져 있다면, 이 네트에 다운사이징을 하게 되면 이전 네트의 노이즈 마진 균형을 깨어 버리게 되어 글리치 제거 효과가 무산 되는 수가 있다. 이 때는 다음에서 설명할 업 사이징과 게이트 사이징의 혼합 방법을 통한 노이즈 마진 확보와 같은 방법을 취하게 되는데, 이 경우 앞서 계속 언급 한 바와 같이 해당 네트 이후의 네트에 슬랙의 변화가 생기게 되므로 전체적인 슬랙 정보의 업데이트(update) 과정이 필요하다.

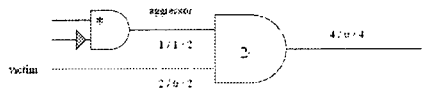

그림 3-3. 이전 노드의 버퍼 삽입으로 인해 다운사이징을 하지 못하는 예

그림 3-3에서는 어그레서의 신호 도착 시간이 빅팀보다 빠르지만 빅팀의 슬랙에 여유가 없어 어그레서의 *gate에 다운사이징을 하려고 했으나, 그림에서 보는 바와 같이 *gate 이전 네트에 글리치 제거를 위한 버퍼가 삽입 되어 있으므로 이 때 *gate를 다운사이징을 하게 되면 노이즈 마진 값도 틀려서 글리치 제거 효과가 무산 되어 버리게 되는 경우를 보여 주고 있다.

3.3 버퍼 삽입과 업 사이징의 동시 수행

업 사이징과 버퍼 삽입의 동시 수행은 그 방법에 따라 크게 두 가지로 구분 할 수가 있다. 3.1의 버퍼 삽입에서 언급한 바와 같이 삽입된 버퍼의 노이즈 마진과 게이트의 노이즈 마진을 합쳐도 글리치의 폭보다 크지 않을 때에 이 방법을 사용하게 된다. 그림 3-4는 버퍼 삽입 후에도 글리치 폭보다 큰 노이즈 마진을 얻을 수 없어 게이트의 업 사이징을 통해 노이즈 마진을 확보하는 예를 보여주고 있다. 이 경우 게이트 업 사이징으로 인해 이후의 슬랙이 변하게 되므로 업데이트를 통해 최신의 슬랙 정보를 유지해야 한다. 그렇지 않을 경우 빅팀을 탐색하면서 잘못된 슬랙 정보로 인해 글리치 발생의 예측에 있어 오류가 생길 수 있다.

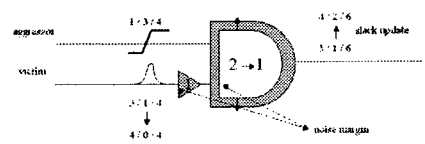

그림 3-4. 버퍼 삽입과 업 사이징의 동시 수행으로 노이즈 마진 확보 (버퍼 삽입법의 보완책)

위의 그림은 삽입된 버퍼의 노이즈 마진과 사이징 전의 게이트의 노이즈 마진의 합이 글리치의 폭보다 작아 글리치를 제거 할 수 없어서, 게이트의 업 사이징으로 좀 더 큰 노이즈 마진을 확보하려 하는 예를 보이고 있다. 이 때 업 사이징으로 인해 게이트의 지연 시간이 2ns에서 1ns로 빨라져 이후의 네트의 슬랙이 5/1/6에서 4/2/6으로 변화 되었다. 변화된 슬랙 정보는 후단의 모든 네트에 전파 되어 새로운 슬랙 정보를 유지토록 하는 업데이트 과정이 필요하다. 즉, 5/1/6으로 계산되어진 슬랙정보를 4/2/6으로 다시 재 계산하여 최신 정보로 업데이트 하여야만 정확한 글리치 발생 예측이 가능한 것이다. 여기까지의 과정은 빅팀의 슬랙이 여유가 있을 때의 조건이 전제되는 것이며 3.1에서 언급한 버퍼 삽입의 보완책이라고 보면 된다.

또 다른 버퍼 삽입과 업 사이징의 동시 수행의 경우는 빅팀 네트의 슬랙에 여유가 없는 경우이다. 즉, 다운사이징 방법의 보완책이다. 이 경우에는 앞의 방법대로 버퍼를 바로 삽입할 수가 없고 글리치 폭과 현재 게이트의 노이즈 마진의 차를 구하여 그 만큼의 노이즈 마진에 해당하는 버퍼를 라이브러리에서 찾아, 해당 버퍼의 지연 시간 만큼의 슬랙을 이전 노드의 업 사이징으로 통해서 확보해 주는 방법을 취한다.

그림 3-5에서 보는 것과 같이 어그레서의 신호 도착 시간이 빅팀의 그것보다 빨라 크로스톡 글리치가 발생될 수 있어 버퍼를 삽입하려 했으나 빅팀의 처음 슬랙이 4/0/4에서 보듯 여유가 전혀 없는 상태라 버퍼를 삽입 할 수가 없었다. 이 경우 글리치 폭과 게이트가 가진 노이즈 마진의 차를 구하여 그림과 같이 지연 시간 1ns의 버퍼 삽입으로 노이즈 마진을 확보 할 수 있다고 판단되었을 때, 이전 노드의 업 사이징으로 통해 지연 시간 1ns 만큼의 슬랙을 만들어 줘야 한다. 만약 이전 노드의 사이징 과정에서 라이브러리에 버퍼 삽입으로 인한 지연 시간을 보상해 줄 만한 사이즈를 가진 게이트가 없다면 버퍼의 업 사이징을 통해 노이즈 마진은 더 커지면서 지연 시간을 줄어드는 버퍼를 선택하도록 하여 게이트 사이즈의 라이브러리 부재를 보완토록 하였다.

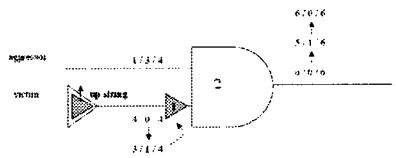

그림 . 빅팀 네트의 슬랙 여유 없을 경우의 버퍼 삽입과 업 사이징 동시 수행의 예 (다운 사이징법의 보완책)

VI. 실험 결과

본 논문의 실험을 위해 상호 전기 용량 값과 주위 연결선 및 각 게이트들의 저항, ground capacitance 값을 가진 네트리스트(netlist)를 입력으로 받았으며 게이트나 버퍼는 각 사이즈별로 노이즈마진과 지연 시간 등의 정보를 담은 라이브러리와 연동하여 테이블 참조 방식(table-look-up)으로 참조하도록 하였다.

표1에서는 실험에 쓰인 벤치마크 회로에 대한 기본 정보와 각 회로에서 발생하는 크로스톡 글리치의 개수 및 본 논문에서 제시한 알고리즘을 적용한 후 얼마만큼의 글리치 감소가 일어났는지 또 그 비율은 얼마만큼 되는지를 보여 주고 있다. 감소된 글리치의 수가 많을수록 크로스톡 글리치에 의한 논리 값의 오류가 일어날 확률이 감소한다. 표1에서와 같이 평균 5.3%의 크로스톡 글리치 감소 효과를 나타내었다.

표1 . 실험에 사용된 벤치마크회로의 특성 및 크로스톡 글리치 감소 결과

circuit name	Max level	#gates	#nodes	#crosstalk glitch	#considerable glitch	#glitch reduction	ratio(%)
c432	19	222	258	240	216	12	5.4
c880	19	378	438	320	288	16	5.5
c499	25	602	643	514	463	24	5.2
c1908	32	672	705	564	508	25	4.9
c1355	36	978	1.019	815	733	39	5.3

표2에서는 본 논문에서 글리치 제거 알고리즘으로 사용 된 버퍼 삽입, 다운 사이징, 버퍼 삽입과 업 사이징의 동시 수행 적용 횟수를 표시하였다. total은 표1에서 나온 제거된 글리치의 개수이다. 버퍼 삽입의 경우는 삽입 할 빅팀 네트에 슬랙의 여유가 있어야 하는데 실제 회로에서는 업 사이징이나 다운사이징의 혼용 없이 초기 단계에서 슬랙의 확보가 쉽지 않아서, 실험 결과 버퍼 삽입과 업 사이징의 동시 수행의 빈도가 가장 높은 것으로 나타났다. 이 방법의 경우 이후 노드에 대한 슬랙 변화를 일으키지 않아 최적화 측면에서도 바람직한 결과를 나타내었다.

V. 결론 및 향후 과제

어그레서가 천이 중이고 빅팀이 안정화 되어 있을 때

표2 . 글리치 제거에 사용된 각 알고리즘 별 개수

circuit name	total	buffer	down sizing	buffer + up sizing
c432	12	3	2	7
c880	16	4	3	9
c499	24	6	5	13
c1908	25	7	4	14
c1355	39	9	10	20

크로스톡에 의한 글리치가 발생하는 것을 이용하여, 어그레서와 빅팀의 신호 도착 시간을 비교한 후 수학적으로 모델링 된 글리치 파형을 참조하여 고려 대상 글리치 인지 아닌지를 판단한다. 이렇게 선별된 네트는 노이즈 마진의 증가를 통한 글리치 제거를 위해 버퍼 삽입, 다운사이징을 우선적으로 고려하였고, 각 알고리즘의 조건을 만족 하지 못할 경우에도 버퍼 삽입과 업 사이징 동시 수행 알고리즘을 적용하여 해당 네트의 변화가 이후 네트에 가장 최소한으로 영향을 미치도록 하였다.

본 논문에서는 기능(functional) 값을 고려하지 않고, 신호 도착 시간이라는 시간적 정보만을 고려하여 글리치를 제거 하도록 하여, 비록 어그레서가 빨리 도착했지만 실제로는 고려하지 않아도 되는 신호들에 대한 제거(pruning)를 할 수가 없었다. 또한 실제 회로에서는 단일 빅팀, 다중 어그레서 환경 내지는 다중 빅팀, 다중 어그레서 환경이 의미를 가지겠으나 본 논문은 단일 빅팀, 단일 어그레서로 한정을 두었다. 이는 다중 개념의 경우 경우의 수가 너무나 많음으로 인해 제거의 개념이 함께 고려되어야 하고, 이를 위해 기능 값 및 다중 어그레서에 대한 타이밍 윈도우(timing window)와 같은 새로운 개념이 되어야 한다. 향후 알고리즘에 이를 반영한다면 실제 회로에도 적용 가능하면서도 효율적인 알고리즘이 되리라 생각한다.

참 고 문 헌

[1] Sanjay Dubey, Joel Jorgenson, "crosstalk Reduction Using Buffer Insertion", IEEE, 2002.

[2] Murar R. Becer, David Blaauw, Ilan Algor, "Post-Route gate Sizing for crosstalk Noise Reduction", DACm 2003.

[3] Pirouz Bazargan Sabet, "Modeling crosstalk Noise for Deep Submicron Verification Tools", University of Paris 6, 2002.

[4] P.Saxena and C. L. Liu., "crosstalk minimization using wire perturbation", In Proceedings of Design Automation Conference DAC, pages 100 - 103, 1999.