

게이트 레벨 동기 회로의 자동 합성에 관한 연구

°김현기* · 신원철* · 안종복** · 이천희***
극동전문대* · 춘천기능대** · 청주대학교***

Automatic synthesis of gate-level timed circuits

°Kim, Hyun-gi* · Shin, Won-cheol* · Ahn, Jong-bok** · Yi, Cheon-hee***
Keugdong College* · Chunchon Polytechnic College** · Chongju University***

요약

본 논문은 gate-level timed circuits의 자동 합성과 검증에 대한 것으로, 동기 회로는 디자인을 최적화하기 위해 합성 절차가 사용된 동안 설계서에 명시된 시간 정보에 속한 비동기 회로의 일부로서 이 시스템은 열거된 일반적인 회로 작용과 시간의 요구 조건에 대해 설계를 해석한다. 이 설계는 영향을 미치는 상태 공간을 구하기 위해 정확하고 효과적인 시간 해석 알고리즘을 사용해 해석할 수 있는 그래픽 표현으로 자동적으로 변환된다.

이 상태공간으로부터 합성 절차는 standard-cells과 gate-arrays와 같은 반 주문형 반도체로 매핑을 용이하게 하기 위해 기본 게이트만을 사용해 어려움을 해결하는 시간에 대한 회로 유도된다.

1. 서론

최근 몇 년간 클럭의 왜곡 문제를 제거, 보통의 경우 성능을 이루고 처리와 환경변수를 적용하고 성분의 모듈화가 준비할 수 있는 능력 때문에 비동기 회로의 디자인이 관심을 불러일으켰다. 비동기 회로는 동시 전력을 줄이기 위해 요구되는 시스템 전력을 줄이고 사용되지 않는 성분은 자동적으로 전력을 절약시키고 비논리적인 변환을 제거하고 동적인 전력을 공급하기 위해 쉽게 조절할 수 있다. 비동기 설계는 인터페이스 회로에 너무 오래 사용되는 동안에 저전력 잇점에 기인한 컨트롤러와 휴대용 장치가 가지는 저전력으로 디자인하기 위해 사용되었다.

전통적인 비동기 디자인 방법론은 해방된 지연 가정 즉 정확하게 검증할 수 있는 회로를 사용한다. 그러나 간단하게 하기 위해 희생된 시간 즉 불필요하게 보수적인 디자인을 유도한다. 하지만 제조업에서 타이밍은 칩의 면적과 회로 지연을 낮추기 위해 매우 중요하다.

정확히 타이밍 정보를 다루기 위해 형식적인 방법의 부족에 기인하여 타이밍이 제한된 회로는 디자인에서 이득을 확신하기 위해 대규모의 시뮬레이션이 항상 요구된다.

기존 연구에서 효과적인 타이밍 해석 알고리즘은 동기 회로의 합성에 타이밍의 고찰을 위해 개발되었으며 본 연구의 동기 회로 완성에는 Burch 동기 회로 검증기를 사용해 입증하였다.

먼저 타이밍 해석은 이 연구에서만 제한되므로 본 합성 절차는 회로에 제한적으로 적용된다. 두 번째 본 연구의 동기 회로 완성에는 standard-cells과 gate-arrays와 같은 반주문형 성분을 사용하기에 어렵게 만드는 복잡한 게이트를 사용하게 된다. 즉 이것은 time-to-market을 향상하기 위해 점점 중요하게 된다. 세 번째는 Burch 검증기에 의해 사용된 이산 시간 검증 접근의 사용은 그것의 응용에 제한을 받는다. 이산 시간 상태의 수가 동시에 발생되는 사건의 수에 대해서 지수적으로 증가한다.

본 논문에서 동기 회로의 합성을 위해 일반적인

적용 절차를 나타내었다. 먼저 정확하고 효과적인 동기 해석 알고리즘으로 해석할 수 있도록 하기 위해 디자인에 자동 변환을 적용하여 조건적인 동작 또는 선택을 위해 확장하였다. 둘째로 우리는 우리의 합성 절차를 부가함으로 반주문형 성분에 완성을 용이하게 한다. 그러므로 생성된 hazard-free timed 회는 AND gate, OR gate와 C-elements와 같은 기본적인 게이트만을 사용하였다.

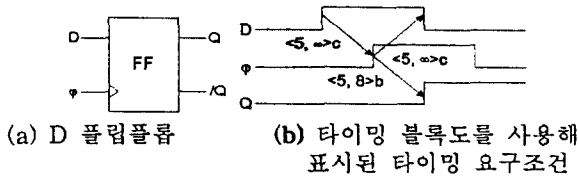
2. 동기가 적용된 디자인

2.1 Orbital nets

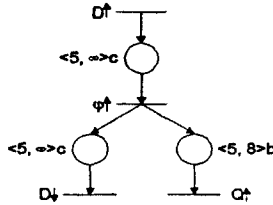
Orbital net에서 타이밍은 lower bound, upper bound와 어떤 형태로 이루어진 타이밍 요구조건처럼 place와 관련된다. Lower bound는 음수가 아닌 정수이고 upper bound는 lower bound보다 크거나 또는 ∞ 와 같은 정수이다. 실수값을 어떤 요구되는 정밀도에서 유리수와 같이 표현할 수 있으므로 정수가 되기 위한 타이밍 요구조건의 제한된 bound는 orbital net의 표현을 줄인다. 이것들은 타이밍 파라미터의 유한수이므로 어떤수가 유리수라면 최소 공통 분모에 의해 그것들의 모두를 곱할 수 있다.

타이밍 요구조건의 두가지 형태 즉 이동(b)과 속박(c)를 갖는다. 이동 타이밍 요구조건은 검증된 타이밍 이동을 설명하는 데 사용된다. 그 반면에 속박 타이밍 요구조건은 설명된 타이밍의 이동을 설명하는 데 사용된다. 그리고 그것들은 실제 타이밍 이동에 영향이 없다. 그림 1(a)에 그려진 D flip-flop을 고찰하면 FF의 타이밍 요구조건은 그림 1(b)에서 타이밍 다이어그램을 사용하여 그림 1(c)에서 orbital net을 사용해 그렸다. 이 FF는 입력 D에서 상승 변이로부터 클럭 ϕ 에서 상승 변이까지 속박된 타이밍 요구조건을 가지고 재표현된 5 time units의 setup time을 가진다. 유사하게 5 time units의 hold time은 클럭 ϕ 에서 상승 변이의 입력 D에서 상승 변이까지 제한된 타이밍 요구조건을 가지고 재표현된다. 이것들은 환경을 만족해야만 하는 요구조건들임에 주의하고 FF는 이들 작용을 검증할 수 없다. FF의 지연은 클럭

ϕ 에서 상승 변이의 출력 Q에서 상승 변이까지 작용 time 요구조건과 같이 표현된다. 이들 요구조건은 FF 회로가 $\phi \uparrow$ 후에 5와 8 time units간에 Q가 검 증된다.



(a) D 플립플롭 (b) 타이밍 블록도를 사용해 표시된 타이밍 요구조건



(c) orbital net를 사용한 타이밍 요구조건
그림 1 타이밍 블록도와 orbital net의 비교

2.2 Orbital net에서 THSE로 변환

각 수반되는 orbital net의 첫 번째 변환에서 THSE로 변환하기 위한 절차는 독립적으로 처리된다. 완전한 orbital net의 재표현을 얻기 위해 orbital net을 구성한다.

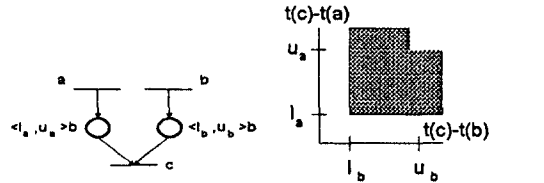
Orbital net을 위해 변환 과정의 절차는 초기에 설명되었다. 먼저 신호에서 상승과 하강 변화의 각 신호는 자동적으로 추가된다. 다음에 변화는 event이든 wait이든 작용의 발생을 위해 net에 더해진다. 그리고 변화는 상용한 작용으로 나타낸다. 각 event을 위해 절차는 작용 place에 더해지고 event에서 각 작용으로부터 상용한 변화를 위해 상용한 예지는 타이밍 요구조건 $\langle 0, \infty \rangle c$ 와 wait에 의해 분리된 어떤 다른 event을 위해 preset을 위한 상용한 예지를 가지고 속박 place에 더해진다. 외각선을 다루기 위해 절차를 초기 표시된 것을 더한 다른 각 place를 가진 첫 번째 event가 선행하는 것과 같이 마지막 event들을 관찰한다. 어드레스를 선택하기 위해 care가 새로운 place가 필요한지 이미 추가된 place가 분할될 것인지 결정해야만 한다. 예를 들면 SEL로부터 방법을 선택하기 위해 event sel \downarrow 와 sel \uparrow 는 event sel \uparrow 의 postset에서 place를 분할해야만 한다.

이 net가 SEL을 위해 orbital net로 구성될 때 한 방법에서 작용의 다른 발생은 그들 차수의 유지되는 트랙에 의해 다른 방법에 상용한 발생과 결합된다.

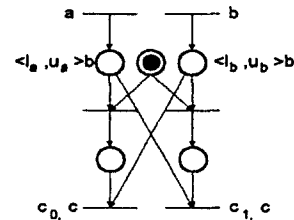
2.3 단일 작용 위치 요구조건을 만족하기 위한 변환

Orbital net가 단일 작용 place 요구조건을 만족하기 위한 것으로 변환된다. 이것을 성취하기 위해 그림 2(a)에 나타낸 변화의 preset에서 두 작용 place를 가진 orbital net를 나누어 고찰한다. 원하는 타이밍 작용은 그림 2(b)에 나타낸 것과 같이 그래프적으로 그려질 수 있다. 이 net는 단일 작용 place 요구조건을 만족하는 그림 3(a)에 나타낸 것과 같이 변환될 수 있다. 기본적으로 이 net 변환 뒤의 개념은 net를 통한 패스가 preset에서 변이의 각 가능한 순서를 위해 만들어진다. 이것은 영향을 preset에서 각 변이는 발생하는 postset에서 변이를 막는 기회가 주어진다. 목적을 설명하기 위해서 추가되는 event c_0 과 c_1 은 c에 연합된 두 변이를 가지고 동시에 발생하

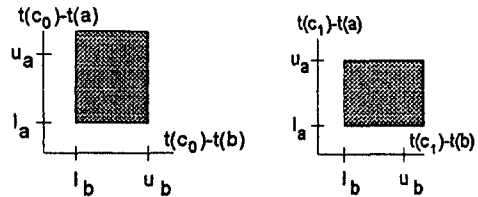
기 위한 net에 추가된다. c_0 과 c_1 의 타이밍 작용은 그림 3(b)과 (c)에서 그래프로 나타내었다. 이들 두 개의 작용은 c의 요구하는 타이밍 작용에 알맞다. n 작용 place을 위해서 net는 n 가능한 event의 n! 가능한 차수 모델을 위해 변환된다. 이 변환이 net 크기에 중요하게 파생되는 동안 n의 값이 실제의 예에서 아주 작은 영향을 미친다. 이 변환은 preset에서 작용 place의 하나는 그것의 postset에서 배수가 되는 변이를 가지는 경우에 더 복잡하다.



(a) 단일 작용 위치 요구조건의 분해
그림 2 단일 작용 위치 요구조건의 변환



(a) 단일 작용 위치 요구조건



(b) C_0 의 타이밍 작용의 (c) C_1 의 타이밍 작용의 그래프 표현
그림 3 단일 작용 위치 요구조건의 표현

3. 동기 상태 공간의 탐구

3.1 Unit-cubes

Alur의 unit-cubes 기술은 일반적인 동기 시스템^[4]의 동기 상태 공간 연구를 위해 최악의 경우를 고려한 복잡성으로 잘 알려져 있다. 이 기술은 클럭의 분수의 규칙적인 선형 차수와 같은 정수 클럭 값을 가진 동기 state의 등가 클래스를 고찰한다. 두 표시된 place와 두 클럭 clk_1 와 clk_2 가 있는 경우 등가 클래스는 그림 4(a)에 나타내었다. 즉 모든 점, line segment과 안의 삼각형은 등가 클래스이다.

3.2 Discrete-time

일반적으로 unit-cube 기술은 정수 event times만이 continuous-time 작용^[3]에 전체 특성이 주어지므로 orbital net를 위해 필요성이 증명된다. 이 증명은 동기 transition systems를 위한 Henzinger^[5]에 의해 주어진 것과 유사하다. 다른 말로 그림 4(b)에서 두 요소의 경우를 위한 것과 같이 표현된 각 discrete-time instance와 결합된 모든 동기 상태는 고찰하는데 필요하다. 이것은 증명된 동기 회로에 대한 Burch에 의해 사용된 기술이다. 이 기술은 n!보다

더 큰 unit-cube 방법보다 더한 $|S|(k+1)^n$ 의 최악의 경우도 고려한 상태 공간 크기를 가진다.

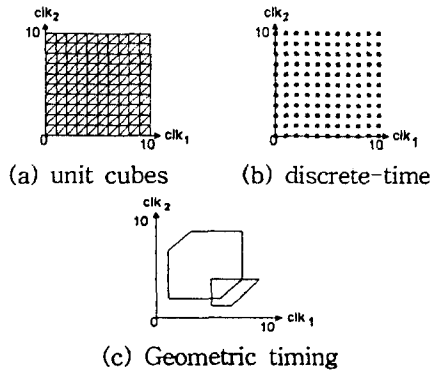


그림 4 동기 상태 공간의 표현

3.3 Geometric 타이밍

Unit-cubes와 discrete-time 둘다 상태 공간의 크기가 net에서 일시에 지수 함수적으로 증가되기 때문에 이론적인 흥미는 거의 없다. 일반적으로 해석을 하는 동안 모든 가능한 정수는 각 상태에서 모든 변화를 위해 고찰되어야만 한다. 두 표시된 디지털을 위해 6개의 독립되어 동시에 일어나는 사건으로 인한 타이밍의 정확한 값을 가진 회로에서 state space는 가장 안정된 합성 기술의 특성을 가진 10^{12} 상태를 초과하기가 쉬웠다.

3.4 Geometric regions

각 단계에서 단일 이산 시간 상태나 또는 시상태의 최소 일치 등급을 고려하는 것보다는 차라리 geometric 방법은 병렬에서 시상태의 무한집합을 고려한다. 특히 특정 클럭값과 특정클럭값의 쌍사이에서 상한과 하한경계로 표시된 시상태의 블록 기하영역은 시상태 공간의 표현으로 사용되었다. 그러한 조건들의 set는 매트릭스 A로 표현되며 클럭 $\{clk_1, \dots, clk_n\}$ 상에 제약조건들은 $clk_i - clk_j \leq a_{ij}$ 형태이다. 항상 영의 값을 갖는 허위의 클럭 clk_0 이 채택되어서 특정 클럭상의 상하한 한계가 같은 형태로 표현될 수 있다. 어느 블록 영역이라도 그러한 매트릭스로 표현될 수 있으며 같은 블록 영역을 표현하는 많은 매트릭스들이 있다. Floyd 알고리즘을 사용한 정규의 진행이 독특한 제약조건 매트릭스들을 만들기 위하여 수행되었다. 일반적으로 Floyd 알고리즘은 $O(n^3)$ 시간에 진행되며 단지 증가의 변화만이 분석중에 매트릭스에 일어나서 Floyd 알고리즘의 특수한 경우는 $O(n^2)$ 시간에 수행된다. 두 예의 영역이 그림 4의 c)에 나타나 있다.

4. 결론

본 논문은 gate-level 동기 circuits의 자동 합성을 위한 방법들을 설명한다. 합성 절차는 표시된 업무의 중단, 동시 발생과 선택권이 가능한 일반적인 THSE 설계로서 시작된다. 설계는 모델링된 동기 circuit 작용을 위해 게도 통신망 설명으로 자동적으로 변환된다. 따라서 합성을 위해 이루어질 수 있는 상태 공간을 결정하기 위해 기하학적인 영역을 사용해 타이밍 해석 절차를 개발했다. 상태 공간 영역에서 분석된 방법을 사용하여 합성 절차를 기본 게이트만을 사용하여 hazard-free 동기 회로로 완성이 된다. 또한 용이하게 하기 위해 반주문형 성분을 사용하였다. 부분

차수 타이밍에서 이 접근은 동기 상태 공간의 거의 최적 재표현으로 성취되는 것을 의미하는 모든 비동기 상태를 위해 두 기하학적 영역에 매우 밀접하게 관계됨을 알 수 있었다. 비동기 회로 설계에 시간을 부가한 규칙적인 방법을 적용함으로써 본 연구의 절차는 동기 회로를 사용하여 효과적이며 안정적임을 알 수 있었다.

참고문헌

1. A. J. Martin. Programming in VLSI : from communicating processes to delay-insensitive circuits. In C.A.R. Hoare, editor, UT Year of Programming Institute on Concurrent Programming. Addison-Wesley, 1990.
2. T.-A. Chu. Synthesis of Self-d 동기 VLSI Circuits from Graph-theoretic Specifications. PhD thesis, Massachusetts Institute of Technology, 1987.
3. T. G. Rokicki. Representing and Modeling Circuits. PhD thesis, Stanford University, 1993.
4. R. Alur. Techniques for Automatic Verification of Real-Time Systems. PhD thesis, Stanford University, August 1991.
5. D. L. Dill. Timing assumptions and verification of finite-state concurrent systems. In Proceedings of the Workshop on Automatic Verification Methods for Finite-State Systems, June 1989.
6. H. R. Lewis. "Finite-State Analysis of Asynchronous Circuits with Bounded Temporal Uncertainty". Technical report, Harvard University, July 1989.
7. B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time petri nets. IEEE Transactions on Software Engineering, 17(3), March 1991.