

시간 제한 조건을 가진 결정성 신호 전이 그래프로부터 비동기 회로의 합성

(Synthesis of Asynchronous Circuits from Deterministic Signal Transition Graph with Timing Constraints)

김희숙[†] 정성태^{**}
(Hee-Sook Kim) (Sung-Tae Jung)

요약 본 논문에서는 시간 제한 조건을 가진 신호 전이 그래프로부터 비동기 회로를 합성하는 방법을 기술한다. 이 방법에서는 기존의 방법과는 달리 상태 그래프를 생성하지 않고 신호 전이 그래프로부터 직접 신호 전이들간의 관계를 구하여 비동기 회로를 합성한다. 본 논문의 합성 과정에서는 먼저 타이밍 분석을 통하여 임의의 두 신호 전이 사이에 시간 제한 조건 내에서 병렬 관계와 인과 관계가 있는지를 구한다. 그 다음에는 이들 관계들로부터 우선 순위 그래프를 생성하고 이 그래프 상에서 경로들을 구함으로써 해저드가 없는 회로를 생성한다. 실험 결과에 의하면 본 논문에서 제안한 합성 방법은 상태 수가 많은 회로에 대해서 현저하게 합성 시간을 단축시킬 수 있을 뿐만 아니라 기존의 합성 방법과 비교하여 거의 같은 면적의 회로를 합성한다.

Abstract This paper presents a new method to synthesize timed asynchronous circuits directly from the specification without generating a state graph. The synthesis procedure begins with a deterministic signal transition graph specification with timing constraints. First, a timing analysis extracts the timed concurrency and timed causality relations between any two signal transitions. Then, a hazard-free implementation under the timing constraints is synthesized by constructing a precedence graph and finding paths in the graph. The major result of this work is that the method does not suffer from the state explosion problem, achieves significant reductions in synthesis time, and generates circuits that have nearly the same area as compared to previous methods.

1. 서론

비동기 회로는 클럭 스쿼나 분배 문제가 없고 평균 속도로 동작할 수 있고 전력 소모가 적은 회로 구현에 이용될 수 있는 등 여러 가지 장점을 가지고 있기 때문에 많은 관심을 끌고 있다. 비동기 회로는 여러 부류로 분류될 수 있는데, 그 중의 한 부류인 속도 독립 회로(speed-independent circuits)는 논리 게이트의 지연 시

간에 관계없이 올바르게 동작하도록 보장되는 견고성을 가지고 있기 때문에 속도 독립 회로의 합성을 위한 많은 방법들이 제안되었다[1,2,3,4,5].

그러나 회로 신호 사이의 타이밍 정보를 알 수 있는 경우에는 주어진 시간 제한 조건 내에서 올바르게 동작하는 회로를 구현하는 것이 보다 효율적일 수 있다. 주어진 시간 제한 조건을 이용하는 합성 방법[6,7]들이 제안되었는데, ATACS[7]는 시간 제한 조건을 이용함으로써 속도 독립 회로보다 면적과 속도 면에서 성능이 우수한 회로를 보다 빠른 시간 내에 합성할 수 있었다.

기존의 시간 제한 조건을 이용하는 합성 방법들에서는 중간 단계 표현으로 상태 그래프를 이용하는데, 상태 그래프의 한 노드는 회로의 전체 신호들의 값을 나타낸다. 따라서, 신호의 수가 N 일 때 상태 그래프의 노드 수는 2^N 개가 될 수 있다. 이러한 이유로 이들 합성 방

· 본 연구는 1998년도 후반기 한국과학재단 해외 Post-doc 지원 사업과 1999년도 원광대학교 연구비 지원에 의하여 연구되었음

† 학생회원 : 원광대학교 컴퓨터공학과

hskim@gaebnyok.wonkwang.ac.kr

** 종신회원 : 원광대학교 컴퓨터및정보통신공학부 교수

stjung@wonms.wonkwang.ac.kr

논문접수 : 1999년 4월 29일

심사완료 : 1999년 12월 15일

법들은 상태 수가 큰 경우에는 합성에 많은 시간이 소요되거나 메모리의 부족으로 인하여 합성에 실패할 수 있는 문제를 가지고 있다. 예를 들어, ATACS의 경우에는 상태수가 100만 개만 넘어가도 합성에 실패하였는데, 신호의 수가 20개만 되어도 상태 수는 100만개 이상이 될 수 있기 때문에 문제점으로 지적되고 있다. 속도 독립 회로의 경우에도 같은 문제가 있기 때문에 이를 해결하기 위하여 상태그래프를 이용하지 않고 신호 전이 그래프의 구조적 분석을 이용하여 속도 독립 회로를 합성하는 알고리즘이 제안되었다[8]. 이 방법은 큐브 근사(cube approximation)에 의거한 방법이라 할 수 있다. 즉, 신호 전이들의 병렬 관계를 이용하여 상태들의 집합을 하나의 큐브로 근사시키고 이 큐브들을 이용하여 회로를 합성한다. 그런데, 해저드를 일으키지 않는 큐브 근사치를 구하기 위해 신호 전이 그래프를 상태기(state machine) 구성 요소로 분할하는 구조적 분석 방법을 사용하고 있다. 이 방법에서는 지수승으로 증가하는 상태 대신에 큐브를 사용함으로써 합성 시간을 현저히 줄일 수 있었다. 그러나 이 알고리즘은 시간 제한 조건을 가진 신호 전이 그래프로부터 회로를 합성할 수 없다.

따라서 본 논문에서는 시간 제한 조건을 갖는 결정성 신호 전이 그래프로부터 상태 그래프를 이용하지 않고 해저드가 없는 비동기 회로를 합성하는 방법을 제안한다. 결정성 신호 전이 그래프란 신호 전이 그래프의 한 부 집합으로서 선택 행위가 없는 그래프를 말한다. 본 논문의 방법에서는 신호 전이들 사이의 관계를 이용하여 회로를 합성한다. 여기에서 이용되는 주요 관계는 주어진 시간 제한 조건 내에서 두 신호 전이가 병렬로 일어날 수 있는가와 한 신호 전이가 다른 신호 전이와 인과 관계에 있는가 이다. 이러한 관계를 구하기 위해서는 타이밍 분석을 통하여 신호 전이들 사이의 타이밍 정보를 추출해야 하는데, 본 논문에서는 참고 문헌 [7]의 타이밍 분석 알고리즘을 사용한다. 타이밍 분석을 이용하여 신호 전이 사이의 관계들을 구한 다음에는 이들 관계를 이용하여 회로 합성에 적합한 정보를 표현할 수 있는 우선 순위 그래프를 생성한다. 그 다음에는 이 그래프에서 경로 탐색을 통하여 회로를 생성한다.

본 논문에서는 SCSI 제어기 회로 명세[7]와 FIFO 회로 명세[9]를 이용한 실험을 통하여 상태 수의 증가와 수행 시간과의 관계를 보인다. 상태 수가 점점 증가되는 회로 명세를 구하기 위해서 SCSI 제어기 회로 명세를 병렬로 연결하여 여러 제어기가 병렬로 동작하는 회로 명세를 만들었고 FIFO 회로 명세를 직렬로 연결

하여 여러 단계의 FIFO 회로 명세를 만들었다. 이러한 회로 명세에 대하여 ATACS와 본 논문의 합성 방법을 적용해본 결과 상태 수가 큰 회로에 대하여 경우에 본 논문에서 제안한 방법이 합성 시간을 현저하게 줄일 수 있었다. 또한 합성된 회로의 면적이 대부분 같았다.

1.1 신호 전이 그래프

그림 1에는 시간 제한 조건을 가진 결정성 신호 전이 그래프의 예가 나타나 있다. 이 신호 전이 그래프는 SCSI 프로토콜 제어기에 대한 명세이다. 이 회로 명세에서 노드는 신호 전이를 나타낸다. 예를 들어 신호 s 의 상향 전이는 $s+$ 로 나타내고 하향 전이는 $s-$ 로 나타낸다. 만약에 신호 s 의 전이가 여러 번 일어날 경우에는 $s+/1, s-/1, s+/2, s-/2, \dots, s+/n, s-/n$ 과 같이 나타낸다. 노드 사이의 아크는 신호 전이 사이의 순서 관계를 나타낸다. 아크에 나타나 있는 검은 색 원은 토큰을 나타내는데, 이는 신호 전이의 활성화 관계를 나타내기 위해 사용된다. 또한 아크에는 시간 제한 조건이 “[하한값, 상한값]”과 같은 형식으로 주어진다. 신호 전이는 모든 입력 아크들이 토큰을 하나씩 가지고 있고 시간이 “하한값”을 지났으면 활성화된다. 활성화된 신호 전이가 일어나게 되면 입력 아크에 있던 토큰들은 제거되고 출력 아크에 토큰이 하나씩 추가된다. 다음에서 s^* 는 전이 $s+$ 또는 $s-$ 를 나타낸다고 하자. 전이 s^* 로부터 전이 t^* 로의 시간 제한 조건을 $TC(s^*, t^*)$ 로 나타내기로 한다. 그리고 $TC_L(s^*, t^*)$ 는 시간 제한 조건의 하한값을, $TC_U(s^*, t^*)$ 는 상한값을 나타내기로 한다.

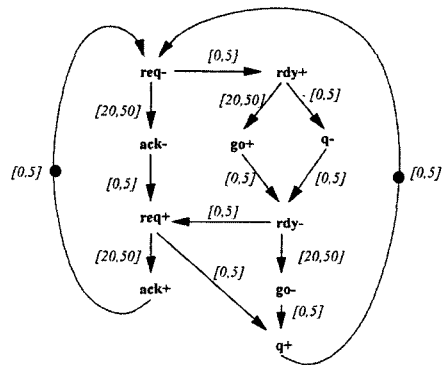


그림 1 SCSI 프로토콜 제어기에 대한 시간 제한 조건이 첨가된 신호 전이 그래프

그림 1에서는 req- 전이가 활성화된 상태를 나타내고 있다. 이 전이가 일어나면 ack- 전이와 rdyl+ 전이의

입력 아크에 토큰이 추가된다. 그런데, $TC_L(req-, rdy+)=0$ 이므로 rdy+ 전이는 바로 활성화되지만 ack- 전이의 경우에는 $TC_L(req-, ack-)=20$ 이므로 시간이 20 단위만큼 지난 다음에 활성화된다.

1.2 타이밍 분석

시간 제한 조건을 이용한 회로 합성을 위해서는 필요한 타이밍 정보를 얻기 위해서 타이밍 분석을 수행해야 한다. 본 논문의 합성 알고리즘에서 필요한 타이밍 정보는 회로 명세에서 임의의 두 신호 전이에 대하여 두 신호 전이가 발생할 수 있는 시간 차이의 최소 값과 최대 값이다. 비순환 그래프로부터 임의의 두 신호 전이에 대하여 두 신호 전이가 발생할 수 있는 시간 차이의 최소 값과 최대 값을 구하는 다항식 복잡도의 알고리즘이 개발되었다[10,11]. 그러나 그림 1에 나타나 있듯이 회로 명세는 순환 그래프의 형태를 갖기 때문에 이 알고리즘들을 회로 합성에 적용하기가 어려웠다. 순환 그래프에서 시간 차이를 구하는 알고리즘들[7, 12]이 제안되었는데, 참고 문헌 [12]에서 제안된 알고리즘은 지수승의 복잡도를 가지는 반면에 참고 문헌 [7]에서 제안된 알고리즘은 다항식의 복잡도를 갖는다.

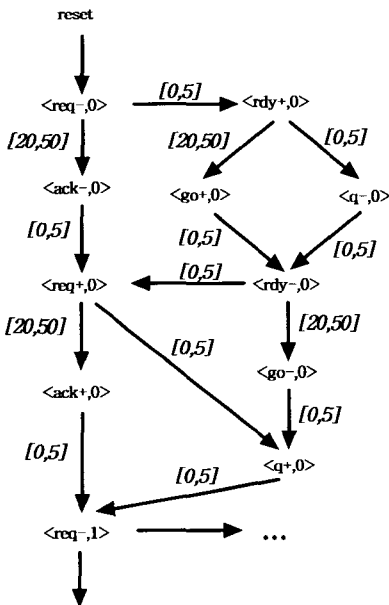


그림 2 SCSI 프로토콜 제어기에 대한 무한의 비순환 그래프의 부그래프

따라서 본 논문에서는 타이밍 정보를 얻기 위해 참고 문헌 [7]의 타이밍 분석 알고리즘을 이용한다. 이 타이

밍 분석 알고리즘은 그림 1과 같은 순환 그래프를 그림 2와 같은 비순환 그래프로 펼친 다음에 비순환 그래프에 대하여 참고 문헌 [10,11]의 알고리즘을 적용하여 시간 차이를 구한다. 순환 그래프는 무한대의 사이클을 갖는 비순환 그래프로 펼쳐질 수 있지만 참고 문헌 [7]의 알고리즘에서는 무한대의 비순환 그래프에서 유한의 부그래프만을 사용하여 타이밍 분석을 수행하는 효과적인 방법을 이용한다.

신호 전이 $s+$ 와 $t+$ 에 대하여 타이밍 분석을 이용하여 구한 시간 차이를 " $TD(s+, t+)=\{최소값, 최대값\}$ "과 같이 나타내기로 한다. 이는 신호 전이 $s+$ 에 대하여 상대적으로 $t+$ 가 발생할 수 있는 시간 범위를 나타낸다. 예를 들어 $TD(s+, t+)=\{10, 20\}$ 이면 $t+$ 전이는 $s+$ 전이가 일어난 다음으로부터 10 단위 시간에서 20 단위 시간 사이에 일어날 수 있는 것을 의미한다. 만약에 $TD(s+, t+)=\{-10, 10\}$ 이면 $t+$ 전이는 $s+$ 전이보다 10 단위 시간 만큼 빨리 일어날 수도 있고 나중에 일어날 수도 있다. 즉, $s+$ 전이와 $t+$ 전이는 병렬로 일어날 수 있다는 것을 의미한다. 시간 범위의 최소값은 " $TD_L(s+, t+)$ "와 같이 나타내기로 하고 최대값은 " $TD_U(s+, t+)$ "와 같이 나타내기로 하자.

시간 차이를 구하는 알고리즘은 참고 문헌 [7]에 자세히 설명되어 있다. 여기에서는 그림 1의 신호 전이 그래프에서 $TD(q-, rdy-)$ 를 구하는 예를 설명하기로 하자. 그림 1의 그래프를 살펴보면 rdy- 전이가 일어나는 시간 범위를 구하기 위해서는 rdy+ 전이를 기준으로 rdy- 전이와 q- 전이가 일어날 수 있는 시간 범위를 고려해야 한다는 것을 알 수 있다. 신호 전이 그래프에서 주어진 타이밍 제한 조건에 의하면 rdy+ 전이가 발생한 다음에 rdy- 전이가 활성화되기까지는 20에서 55 단위 시간이 소요된다. 그리고 rdy+ 전이가 발생한 다음에 q- 전이가 활성화되기까지는 0에서 5 단위 시간이 소요된다. 이에 의하면, 실제로는 q- 전이가 발생한 다음에 rdy- 전이는 최소 15에서 최대 55 단위 시간 사이에 발생할 수 있는데 이는 다음과 같이 구해진다.

$$TD_L(q-, rdy-) = TD_L(rdy+, rdy-) - TD_U(rdy+, q-) = 20 - 5 = 15$$

$$TD_U(q-, rdy-) = TD_U(rdy+, rdy-) - TD_L(rdy+, q-) = 55 - 0 = 55$$

1.3 관련 연구

참고문헌 [8]에서는 상태 대신에 큐브를 이용하여 속도 독립 회로를 합성하는 방법을 제안하였다. 그림 3(a)에는 이 방법에서 사용하는 신호 전이 그래프가 나타

나 있다. 본 논문에서 사용하는 신호 전이 그래프에 비하여 이 신호 전이 그래프에는 시간 제한 조건이 주어지지 않는다. 그림 3 (a)의 신호 전이 그래프에서 p_1 부터 p_8 까지의 명칭이 붙여져 있는 원은 토큰이 위치하는 곳으로서 장소(place)라 불리는데, 그림 1의 신호 전이 그래프에서는 단지 그림을 간단히 하기 위하여 장소를 생략하였다. 참고 문헌 [8]의 합성 방법에서는 먼저 신호 전이 그래프의 각 장소에 대한 큐브를 구한다. 이들 큐브는 장소와 신호 전이들 사이의 병렬 관계와 순서 관계를 이용하여 구한다. 예를 들어, 장소 p_2 는 $y+$ 전이와 병렬 관계에 있기 때문에 신호 y 에 해당하는 값은 '- '가 되고 $x+$ 전이와 $z-$ 전이가 장소 p_2 바로 전에 나타나 있으므로 x 에 해당하는 값은 1이되고 z 에 해당하는 값은 0이 된다. 이와 같이 각 장소에 대해 구해진 큐브가 그림 3 (a)의 장소 명칭 아래에 표시되어 있는데, 이들 큐브가 나타내는 신호의 순서는 x, y, z 이다. 예를 들어, 장소 p_2 에 해당하는 큐브 1-0은 xz' 을 나타낸다.

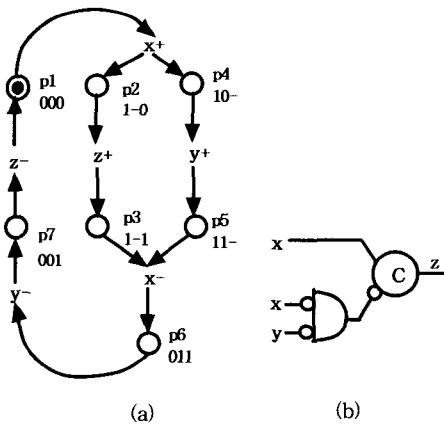


그림 3 (a) 신호 전이 그래프의 예 (b) 참고 문헌 [8]의 합성 방법에 의해 합성된 회로

각 장소에 대한 큐브를 구한 다음에는, 각 출력 신호 전이의 구현에 필요한 장소를 구하고 그들에 대한 큐브의 합을 구함으로써 출력 신호에 대한 회로를 합성한다. 예를 들어, $z+$ 전이를 구현하는 회로는 장소 p_2 와 장소 p_3 에 토큰이 있을 때에 출력 값이 1이 되어야 한다. 따라서 두 장소에 대한 큐브의 합 $S=(1-0)+(1-1)=(1--)$, 즉 $f=x$ 가 전이 $z+$ 를 구현하는 회로가 된다. 그리고 전이 $z-$ 에 대해서는 장소 p_7, p_1 에 해당하는 큐브의 합 $R=001+000=(00-)$, 즉 $f=x'y'$ 이 전이 $z-$ 를 구현하는 회로가 된다. 이렇게 합성된 회로가 그림 3

(b)에 나타나 있다. 여기에서 C-원소는 저장 장치로서 다음에 설명된다.

여기에서는 각 장소에 대한 정확한 큐브를 병렬 관계와 순서 관계에 의해서만 구할 수 있는 단순한 경우에 대하여 합성 방법을 설명하였다. 한 장소에 대한 큐브는 토큰이 그 장소에 있을 때에 전체 신호가 가질 수 있는 상태들의 집합을 나타낸다고 할 수 있다. 그런데, 많은 경우에는 위와 같이 구한 큐브가 이 상태들의 집합과 정확하게 일치하지 않고 포함되어서는 안될 상태를 포함하거나 또는 포함되어야할 상태를 포함하지 않을 수 있다. 이러한 경우에는 큐브가 정확한 상태 집합을 포함하도록 수정해 나가는데, 이때 필요한 추가 정보는 신호 전이 그래프를 상태기 구성 요소로 분할함으로써 구한다.

위에서 설명한 바와 같이 참고 문헌 [8]의 방법은 신호 전이 그래프의 구조적 분석을 통하여 상태의 집합을 큐브로 근사시킴으로써 합성 시간을 단축시키는 방법이라 할 수 있다. 그런데, 타이밍 제한 조건이 추가된 신호 전이 그래프에서는 구조적 분석에 의해서 합성에 필요한 정보를 얻기가 용이하지 않다. 본 논문에서는 타이밍 분석을 통하여 합성에 필요한 정보를 얻은 다음에 이를 효과적으로 이용할 수 있는 방법을 제안한다.

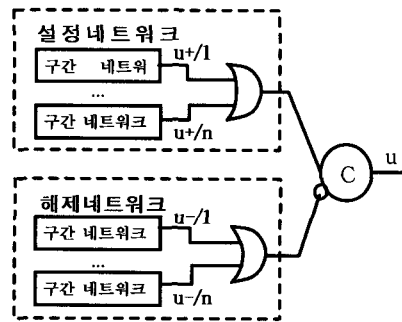


그림 4 목표 회로 모델

2. 합성 절차

2.1 회로 모델

본 논문의 합성 알고리즘에서는 출력 신호 u 에 대하여 그림 4와 같은 모델의 회로를 합성한다. 여기에서 C-원소는 저장 장치로서 두 입력이 모두 1이 되면 출력이 1이 되며, 두 입력이 모두 0이 되면 출력이 0이 되고 그 외의 경우에는 이전의 출력 값을 그대로 유지한다. 따라서 설정 네트워크의 출력이 1이 되고 해제 네트워크의 출력이 0이 되면 C-원소의 출력은 1이 된다.

반대로 해제 네트워크의 출력이 1이 되고 설정 네트워크의 출력이 0이 되면 C-원소의 출력은 0이 된다.

설정 네트워크와 해제 네트워크는 구간 네트워크들을 OR 게이트로 연결하여 구성된다. 여기에서 구간이란 출력 신호의 한 전이, u^*/i 가 활성화되는 시간부터 그 전이의 다음의 역 방향 전이, $\overline{u^*/i}$ 가 활성화되는 시간 사이를 나타낸다고 하자. 그리고 이를 $u^*/i \mapsto \overline{u^*/i}$ 와 같이 나타내기로 하자.

정의 1: 무해저드(hazard-free) 조건 - 신호 u 에 대하여 합성된 회로는 각 구간 네트워크가 다음의 조건을 만족하면 주어진 타이밍 제한 조건 내에서 해저드를 갖지 않는다

- i) 구간 네트워크의 출력이 u^*/i 가 활성화 될 때에 1이 된다.
- ii) 구간 네트워크의 출력이 $\overline{u^*/i}$ 가 활성화될 때까지는 0이 된다.
- iii) 구간 네트워크의 출력이 0이된 다음에는 u^*/i 가 다시 활성화 될 때까지 구간 네트워크의 출력이 0으로 유지된다.

본 논문의 합성 알고리즘은 다음의 세 단계로 구성된다. 첫 번째 단계에서는 타이밍 분석을 통하여 신호 전이 그래프로부터 불필요한 아크를 제거하고 임의의 두 신호 전이 사이의 관계를 구한다. 두 번째 단계에서는 각 구간 네트워크를 단일 큐브 형태로 구현한다. 세 번째 단계에서는 각 구간 네트워크에 대하여 다중 큐브 형태의 조합형 네트워크가 존재하면 이를 구한다.

2.2 불필요한 아크 제거 및 신호 전이들의 관계 계산

본 논문의 합성 알고리즘에서는 타이밍 분석을 이용하여 신호 전이 그래프로부터 불필요한 아크를 제거하고 시간 제한 조건 내에서 신호 전이 사이의 병렬 관계와 인과 관계를 구한다.

정의 2: 전이 s^* 로부터 전이 t^* 로의 아크에 대하여 $TD_L(s^*, t^*) > TC_U(s^*, t^*)$ 이면 이 아크는 불필요한 아크이다.

예를 들어 그림 1의 신호 전이 그래프에서는 $TD_L(q^-, rdy^-) = 15$ 이고 $TC_U(q^-, rdy^-) = 5$ 이므로 q^- 로부터 rdy^- 로의 아크가 불필요한 아크이다. 신호 전이 그래프에서 주어진 타이밍 제한 조건에 의하면 q^- 전이가 일어난 다음으로부터 0에서 5 단위 시간 사이에 rdy^- 전이가 활성화된다고 되어 있다. 그러나

$TD_L(q^-, rdy^-) = 15$ 에 의하여 rdy^- 는 q^- 전이가 일어난 때로부터 적어도 15 단위 시간 후에 일어날 수 있기 때문에 이 아크는 회로의 행위에 아무런 영향을 미치지 못하게 된다. 따라서 이 아크는 불필요한 아크가 되는 것이다. 이를 다른 관점에서 보면 go^+ 전이가 발생했을 때는 타이밍 제한 조건에 의하여 이미 q^- 전이가 발생했다는 것이 보장되므로 rdy^- 전이가 발생하기 위해서는 go^+ 전이가 발생했는가만 인지하면 된다는 것을 의미한다.

정의 3: 두 전이 s^* 와 t^* 가 다음 조건을 만족하면 시간 제한 조건 내에서 병렬 관계를 갖는다. 이를 $s^* \parallel t^*$ 와 같이 나타내기로 한다.

- 1) 두 전이 s^* 와 t^* 가 시간 제한 조건을 고려하지 않을 때에 서로 병렬 관계에 있다.
- 2) $TD_L(s^*, t^*) \leq 0$ 이고 $TD_U(s^*, t^*) \geq 0$ 이다.

정의 3의 조건이 만족되면 전이 t^* 가 전이 s^* 보다 전에 일어날 수도 있고 후에 일어날 수도 있다는 것을 의미하므로 두 전이는 병렬 관계를 가지는 것이다. 예를 들어, 그림 1의 신호 전이 그래프에서 ack^- 전이와 go^+ 전이는 시간 제한 조건을 고려하지 않을 때에 서로 병렬 관계에 있고 $TD(ack^-, go^+) = [-35, 30]$ 이므로 두 전이는 주어진 시간 제한 조건 내에서 병렬 관계를 가진다.

정의 4: 전이 s^* 의 바로 다음의 역방향 전이를 $\overline{s^*}$ 라 하자. 전이 s^* , $\overline{s^*}$, t^* 가 다음의 조건을 만족하면 주어진 시간 제한 조건 내에서 전이 s^* 는 전이 t^* 와 인과 관계를 가진다. 이를 $s^* \Rightarrow t^*$ 와 같이 나타내기로 한다.

- 1) $TD_L(s^*, t^*) > 0$ 이거나 전이 s^* 로부터 t^* 로의 경로가 존재하고 $TD_L(s^*, t^*) = 0$ 이다.
- 2) $TD_U(t^*, \overline{s^*}) \geq 0$.

그림 1의 신호 전이 그래프에서는 시간 제한 조건을 고려하지 않으면 q^+ 전이가 go^+ 전이와 인과 관계를 가진다. 그러나 타이밍 분석에 의하면 $TD(q^+, go^+) = [20, 60]$ 이고 $TD(go^+, q^+) = [-50, -15]$ 이다. 따라서 정의 4의 조건이 만족되지 않으므로 q^+ 전이와 go^+ 전이는 시간 제한 조건 내에서는 인과 관계를 갖지 않는다. 반면에 $TD(q^-, go^+) = [15, 50]$ 이고 $TD(go^+, q^+) = [20, 60]$ 이므로 q^- 전이가 go^+ 전이와 인과 관계를 갖는다.

2.3 단일 큐브 구간 네트워크 합성

2.3.1 초기 큐브

본 논문에서는 각 구간 네트워크를 단일 큐브 형태로 합성한다. 논문 [3]에서는 각 구간 네트워크가 단일 큐브 형태로 합성될 수 있는 조건을 규명하였다. 그리고 논문 [13]에서는 새로운 신호의 삽입에 의해 신호 전이 그래프들이 그러한 조건을 만족할 수 있도록 변환될 수 있다는 것을 증명하였다. 본 논문에서 제안하는 합성 알고리즘은 각 구간 네트워크가 단일 큐브 형태로 구현될 수 있는 신호 전이 그래프만 다룬다.

여기에서는 $u^*/i \mapsto \overline{u^*/i}$ 구간에 대하여 합성 과정을 설명하기로 한다. 무해저드 조건을 만족하는 구간 네트워크를 구하기 위한 첫 단계는 u^*/i 의 여기 신호들로 구성된 초기 큐브를 구하는 것이다.

정의 5: 신호 전이 그래프에서 전이 $s+(s-)$ 로부터 u^*/i 로의 아크가 있고 이 아크가 불필요한 아크가 아니면 $s+(s-)$ 를 u^*/i 의 여기 전이라 하고 $s(s)$ 를 여기 신호라 한다.

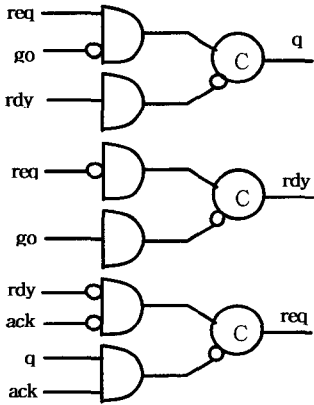


그림 5 그림 1의 신호 전이 그래프에 대한 초기 큐브 네트워크

이렇게 구한 초기 큐브가 그림 5에 나타나 있다. 그러나 초기 큐브들은 무해저드 조건을 만족하지 않을 수 있다. 모든 여기 신호들이 1이 되면 u^*/i 전이가 활성화되므로 조건 i)이 만족된다. 그러나 이 여기 신호들은 $\overline{u^*/i}$ 전이가 활성화되기 전에 0이 되지 않을 수 있다. 또한 0으로 되었더라도 나중에 다시 1이 될 수 있기 때문에 위의 조건 ii)와 iii)이 만족되지 않을 수 있다. 이 조건이 만족되지 않을 경우에는 추가의 신호를 초기 큐브에 추가함으로써 조건 ii)와 iii)이 만족되도록 만든다.

그림 6에는 이러한 추가 신호를 찾는 알고리즘이 나타나 있다. 이 알고리즘에서는 우선 순위 그래프를 생성한 다음에 이 그래프에서 경로를 탐색함으로써 최소한의 추가 신호를 구한다.

2.3.2 우선 순위 그래프 생성

그림 6의 알고리즘에서 첫 부분에는 우선 순위 그래프를 생성하는 과정이 나타나 있다. 여기에서, 한 신호 s 의 전이가 여러 번 일어나는 경우에는, 전이가 일어나는 순서가 $s+1, s-1, s+2, s-2, \dots, s+n, s-n$ 과 같은 순서로 일어난다고 가정한다.

정의 6: 우선 순위 그래프는 노드와 아크로 구성된 방향성 그래프이다. 노드는 신호 전이로서 출발 노드, 행선 노드 그리고 중간 노드로 분류된다. $\overline{u^*/i}$ 활성화되기 전에 구간 네트워크의 출력을 0으로 만들 수 있는 신호의 전이가 출발 노드가 된다. u^*/i 의 여기 전이의 선행 역방향 전이가 행선 노드가 된다. 노드 A로부터 노드 B로의 아크는 A에 해당하는 신호에 이어서 B에 해당하는 신호가 계속해서 구간 네트워크의 출력을 0으로 유지할 수 있다는 것을 의미한다.

전이 s^*/j 가 출발 노드 집합에 포함되기 위해서는 s^*/j 가 u^*/i 와 $\overline{u^*/i}$ 사이에서 일어나거나 또는 u^*/i 자신이거나 한다. 또한 s^*/j 의 한 역방향 전이가 $\overline{s^*/k}$ 에 대하여 $\overline{s^*/k} \Rightarrow u^*/i$ 조건이 만족되어야 한다. 예를 들어, 구간 $rdy+ \mapsto rdy-$ 에 대해서는 $go+, a-, rdy+$ 전이가 이 조건을 만족시킬 수 있다. 그림 7에 나타나 있는 $rdy+ \mapsto rdy-$ 구간에 대한 우선 순위 그래프에서 이 세 출발 노드들이 원으로 표시되어 있다.

전이가 행선 노드 집합에 포함되기 위해서는 u^*/i 전이의 여기 전이들의 선행 역방향 전이어야 한다. 그 이유는 이 전이들에 의하여 여기 신호들의 값이 0이 되고 u^*/i 가 다시 활성화 될 때까지는 그 값이 0으로 유지되기 때문이다. 즉, 이 행선 노드의 전이들이 발생한 다음에는 구간 네트워크의 출력이 0으로 유지되는 것이 보장되는 것이다. 구간 $rdy+ \mapsto rdy-$ 에서는 $req-$ 전이가 $rdy+$ 의 여기 전이이므로 그것의 선행 역방향 전이 $req+$ 가 행선 노드가 된다. 그림 7에 나타나 있는 우선 순위 그래프에서 행선 노드는 사각형으로 표시되어 있다.

출발 노드와 행선 노드 집합을 구한 다음에는 출발 노드에서 시작하여 중간 노드들과 아크들이 더해져 간다. 중간 노드가 삽입되기 위해서는 기존 노드의 신호를 뒤따라서 중간 노드의 신호가 구간 네트워크의 출력

을 0으로 유지할 수 있어야 한다. 또한 기존 노드들 사이에서도 노드 A의 신호를 뒤따라서 노드 B의 신호가 구간 네트워크의 출력을 0으로 유지할 수 있으면 A로부터 B로의 아크가 삽입된다. 구간 $rdy+ \mapsto rdy-$ 에 대하여 이와 같이 확장된 우선 순위 그래프가 그림 7에 나타나 있다.

```

find_a_single_cube_interval_network(STG G, transition u*/i)
{
  /* 우선 순위 그래프 생성 */
  <V,E>=<∅,∅>

  /* 출발 노드와 행선 노드 결정 */
  SN={u*/i}
  Foreach s*/j in G
    If ( u*/i ⇒ s*/j and s*/j ⇒  $\overline{u*/i}$  and
           $\overline{s*/k} \Rightarrow u*/i$  (for some k) )
      SN=SN ∪ {s*/j}
    If (Is_a_non_redundant_trigger_transition(s*/j, u*/i))
      DN=DN ∪ {s*/j-1}

  V=SN ∪ DN

  /* 우선 순위 그래프 확장 */
  Foreach unprocessed node s*/i in V
    Foreach t*/j in G
      If (( s*/i || t*/j or s*/i ⇒ t*/j ) and
            t*/j ⇒ s*/i and  $\overline{t*/k} \Rightarrow u*/i$  (for some k) )
          V=V ∪ {t*/j}
          E=E ∪ {(s*/i, t*/j)}

  Foreach si in SN
    Foreach dj in DN
      Ei,j = Find_all_possible_extra_signals(si, dj )

  Find_a_minimal_extra_signal_set( E )
}

```

그림 6 단일 큐브 구간 네트워크 합성 알고리즘

2.3.3 우선 순위 그래프 탐색

우선 순위 그래프를 생성한 다음에는 각 출발 전이로부터 각 행선 전이로의 경로를 찾음으로써 추가 신호를 구한다. 추가 신호는 다음 정의와 같이 정해진다.

정의 7: 모든 행선 노드에 대하여 적어도 하나의 출발 노드로부터 그 행선 노드로의 경로가 선택되어야 하고 선택된 경로에 $s+(s-)$ 전이가 존재하면 $s'(s)$ 가 추가 신호에 포함된다.

본 논문에서는 추가 신호의 수를 최소화하기 위하여

각 출발 노드로부터 각 행선 노드로의 모든 경로를 구한다. 이렇게 함으로써 각 행선 노드에 대하여 가능한 모든 추가 신호들의 집합을 구할 수 있다. 그 다음에는 집합의 곱 연산을 이용하여 전체 행선 노드에 대하여 최소한의 추가 신호 집합을 구한다. 같은 수의 신호로 구성된 추가 신호 집합이 여러 개인 경우에는 출발 노드의 전이가 구간 내에서 가장 뒤에 위치한 경우를 선택한다. 이렇게 함으로써 구간 네트워크가 조합형 네트워크인 것을 우선적으로 선택할 뿐만 아니라 다음 단계의 조합형 네트워크 합성 알고리즘에서 다중 큐브 조합형 네트워크를 발견할 가능성을 높인다.

그림 7에 나타나 있는 구간 $rdy+ \mapsto rdy-$ 에 대한 우선 순위 그래프에서는 출발 노드로부터 행선 노드로의 최단 경로는 $q- \rightarrow req+$ 이고 행선 노드가 하나이므로 이 구간에 대한 최소의 추가 신호는 q 가 된다. 그림 8에는 각 출력 신호의 구간 네트워크들에 대하여 합성된 구간 네트워크들이 나타나 있다.

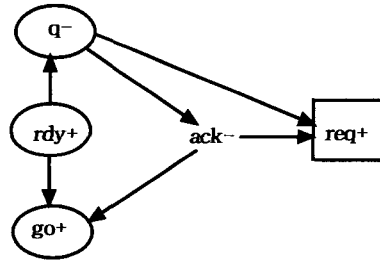


그림 7 구간 $rdy+ \mapsto rdy-$ 에 대한 우선 순위의 그래프

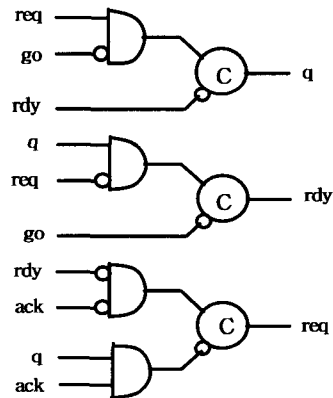


그림 8 그림 1의 신호 전이 그래프에 대한 단일 큐브 구현

2.4 조합형 네트워크 합성

본 논문의 알고리즘은 조합형 네트워크를 발견함으로써 회로의 성능을 향상시킨다. 앞 단계의 단일 큐브 형태의 구간 네트워크 합성 알고리즘에서는 같은 수의 추가 신호들의 경우가 여러 개 있을 때에 조합형으로 만들어 주는 것을 우선적으로 선택하였다. 이 절에서는 앞 단계에서 합성한 단일 큐브 네트워크가 조합형이 아닌 경우에 그 구간 네트워크와 추가의 신호들을 OR 게이트로 연결하여 조합형 네트워크를 합성하는 알고리즘을 설명한다. 구간 네트워크가 조합형이라는 것은 구간네트워크의 출력이 u^*/i 전이가 활성화 될 때에 1이 되고 $\overline{u^*/i}$ 전이가 활성화될 때에 0이 된다는 것을 의미한다. 따라서, 앞 단계에서 구한 구간 네트워크가 조합형이 아니라는 것은 구간 네트워크의 출력이 빨리 0이 된다는 것을 의미한다. 따라서 OR게이트를 추가하여 출력이 1이 되는 기간을 $\overline{u^*/i}$ 전이가 활성화되는 때까지 연장시키는 것이다.

그림 9에는 조합형 네트워크의 추가 신호를 구하는 알고리즘이 나타나있다. 여기에서도 앞 단계에서와 유사하게 우선 순위 그래프를 이용하여 추가의 신호들은 구한다. 이 알고리즘은 앞 단계의 알고리즘과 출발 노드와 행선 노드를 구하는 조건이 다르고 그래프에서 경로에 s^+ 전이가 있으면 추가되는 신호가 s^+ 가 되는 것이다. 추가되는 신호의 하나는 앞 단계에서 구한 구간 네트워크의 출력이 0이 되기 전에 1이 되어야 한다. 그리고 구간의 마지막 전이가 활성화 될 때까지 이 들 중의 적어도 한 신호의 값은 1의 값을 가져야 한다. 또한 추가되는 신호의 값은 이 구간 내에서만 1이 되어야 한다. 따라서 상향 전이와 하향 전이가 한 번만 일어나는 신호 전이만을 고려한다. 출발 노드를 구하는 조건에서는 앞 단계에서 구한 단일 큐브 C 를 0으로 만드는 전이보다 먼저 일어나는 가를 검사하는데, 이는 추가되는 신호가 구간 네트워크의 출력이 0이 되기 전에 1이 되도록 하기 위해서이다. 앞 단계에서는 u^*/i 전이가 활성화 될 때까지 0으로 유지되는 신호를 찾았기 때문에 행선 노드를 u^*/i 전이로부터 구했지만 여기에서는 $\overline{u^*/i}$ 전이가 활성화 될 때까지 1로 유지되는 신호를 찾기 때문에 $\overline{u^*/i}$ 전이로부터 행선 노드를 구한다.

```

find_a_multi_cube_network(STG G, transition  $u^*/i$ , cube C)
{
    /* 우선 순위 그래프 생성 */
    <V, E> = <∅, ∅ >

    /* 출발 노드와 행선 노드 결정 */

```

```

Foreach  $s^*$  in G
    If (occurrence_of_transition(s) = 1 and
         $u^*/i \Rightarrow s^*$  and  $\overline{s^*} \Rightarrow \overline{u^*/i}$  and
        ( $t^*/j \Rightarrow s^*$  and not ( $\overline{t^*/j} \parallel s^*$ ) for all  $t^*/j$ ,
        전이  $t^*/j$ 는 C의 상향전이를 일으키는 전이임))
         $S_N = S_N \cup s^*$ 
Foreach  $s^*/j$  in G
    If (Is_a_non_redundant_trigger_transition( $s^*/j, \overline{u^*/i}$ ))
         $D_N = D_N \cup \{s^*/j-1\}$ 

 $V = S_N \cup D_N$ 

/* 우선 순위 그래프 확장 */
Foreach unprocessed node  $s^*$  in V
    Foreach  $t^*$  in G
        If (occurrence_of_transition(t) = 1 and
            ( $s^* \parallel t^*$  or  $s^* \Rightarrow t^*$ ) and  $t^* \Rightarrow \overline{s^*}$  and
             $\overline{t^*} \Rightarrow \overline{u^*/i}$  and  $u^*/i \Rightarrow t^*$ )
             $V = V \cup \{t^*\}$ 
             $E = E \cup \{(s^*, t^*)\}$ 

Foreach  $s_i$  in  $S_N$ 
    Foreach  $d_j$  in  $D_N$ 
         $E_{i,j} = \text{Find\_all\_possible\_extra\_signals}(s_i, d_j)$ 

```

Find_a_minimal_extra_signal_set(E)

그림 9 다중 큐브 조합형 네트워크 합성 알고리즘

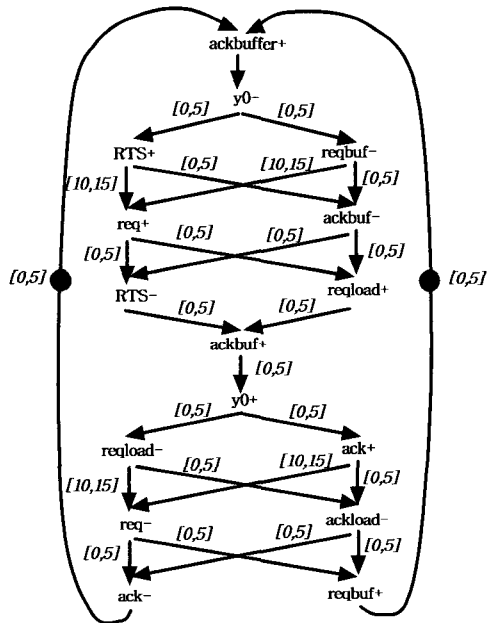


그림 10 패킷 전송 제어기에 대한 신호 전이 그래프

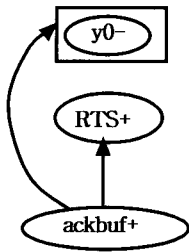


그림 11 구간 $ack- \rightarrow ack+$ 에 대한 우선 순위 그래프

그림 1의 신호 전이 그래프에 대해서는 조합형 구간 네트워크가 존재하지 않으므로 그림 8의 회로가 최종 합성 결과이다. 그림 10에는 HP에 의해서 개발된 통신용 칩[14]에 포함되어 있는 패킷 전송 제어기에 대한 신호 전이 그래프가 나타나 있다. 이 신호 전이 그래프에서는 구간 $ack- \rightarrow ack+$ 과 구간 $RTS- \rightarrow RTS+$ 가 다중 큐브 조합형 네트워크로 구현될 수 있다. 그림 11에는 $ack- \rightarrow ack+$ 구간에 대한 우선 순위 그래프가 나타나 있다. 이 구간에서는 출발 노드가 $y0-, RTS+, ackbuf+$ 이고 행선 노드가 $y0-$ 이다. 이 그래프에서는 $y0-$ 가 출발 노드이면서 행선 노드이므로 최소의 추가 신호는 $y0'$ 이 된다. 그림 12에는 패킷 전송 제어기에 대

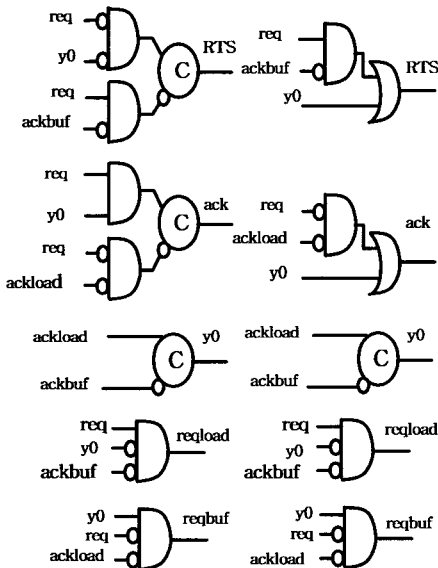


그림 12 패킷 전송 제어기에 대한 구현: (a) 단일 큐브 구간 네트워크 합성 결과 (b) 다중 큐브 조합형 구간 네트워크 합성 결과

하여 단일 큐브 구간네트워크와 최종 합성 결과가 나타나 있다.

표 1 실험 결과 : ATACS와의 비교

회로 이름	상태수	ATACS		본 논문	
		리터럴	CPU	리터럴	CPU
Half Handshake	14	8	0.03	8	0.03
Full Handshake	16	8	0.03	8	0.01
2-4 phase 변환기	19	20	0.04	20	0.04
sender-done	9	5	0.03	5	0.03
mp-forward-pkt	22	16	0.05	14	0.05
Multiple region	14	7	0.02	8	0.02
MMU	92	22	0.06	22	0.05
master-read	2108	34	2.01	34	0.15
AtoD	24	12	0.04	12	0.04
sbuf-ram-write	56	18	0.11	18	0.11
ram-read-sbuf	39	21	0.09	20	0.09
elatch	44	10	0.05	10	0.06
VME	19	20	0.05	20	0.05
1 SCSI 제어기	16	30	0.02	30	0.02
4 SCSI 제어기	806	40	1.17	40	0.22
8 SCSI 제어기	404006	80	4937.36	80	1.29
9 SCSI 제어기	N/A	N/A	N/A	90	1.96
10 SCSI 제어기	N/A	N/A	N/A	100	2.91
60 SCSI 제어기	N/A	N/A	N/A	600	1019.0
120 SCSI 제어기	N/A	N/A	N/A	1200	16395.79
180 SCSI 제어기	N/A	N/A	N/A	1800	82151.49
FIFO 1-단계	29	9	0.06	9	0.02
FIFO 5-단계	67392	45	456.6	45	1.23
FIFO 6-단계	N/A	N/A	N/A	54	2.2
FIFO 10-단계	N/A	N/A	N/A	90	16.98
FIFO 40-단계	N/A	N/A	N/A	360	1240.98
FIFO 100-단계	N/A	N/A	N/A	900	19079.43

3. 실험 결과

표 1에는 벤치마크들에 대한 실험 결과가 나타나 있다. 여기에서는 본 논문의 합성 결과와 참고 문헌 [7]의 합성 결과를 리터럴의 수와 CPU 시간을 기준으로 비교하였다. 큰 상태 공간을 가지는 회로를 만들기 위해 그림 1의 신호 전이 그래프를 병렬로 연결하였고 또한 여러 단계의 FIFO 회로를 합성하였다. 나머지 회로들은 기존의 논문들에서 벤치마크로 사용된 회로들이다. 실험 결과에 의하면 큰 상태 공간을 가지는 회로에 대해서는

본 논문에서 제안하는 알고리즘이 합성 시간을 현저히 단축시킬 수 있다는 것을 알 수 있다.

이 실험 결과는 주메모리가 384MB 이고 스왑 메모리가 700MB인 400MHz 펜티엄 II 컴퓨터를 이용하여 얻어진 것이다. 기존의 방법이 6단계의 FIFO 와 9개의 SCSI 제어기와 같이 그리 크지 않은 회로에 대하여도 메모리 부족으로 인하여 합성에 실패한 반면에 본 논문의 방법은 40 단계의 FIFO와 60 SCSI 제어기를 수십 분 안에 합성할 수 있다는 것이 주목할 만한 일이다. 그리고 속도가 빠르면서도 합성된 회로의 면적이 대부분 같다는 것도 주목할 만한 일이다. 제안된 방법의 회로의 면적이 작은 경우는 본 논문에서는 다중 큐브 형태의 조합형 네트워크를 합성하는 것에 반하여 기존의 방법에서는 이러한 회로를 합성하지 않기 때문이다. 기존의 방법이 보다 작은 회로를 합성하는 경우는 한 신호에 대하여 여러 구간이 존재할 경우에 여러 구간을 동시에 고려하는 것에 반하여 본 논문의 방법에서는 한 구간의 합성 시에 다른 구간을 고려하지 않기 때문이다. 이러한 두 경우는 드물었고 대부분의 경우에는 두 합성 방법이 같은 면적의 회로를 합성하였다. 본 논문에서 제안한 방법에 의하여 합성된 회로가 해저드가 없다는 것을 재확인하기 위하여 기존의 검증기[15]를 사용하였다. 검증기를 사용하여 회로의 동작을 확인한 결과 아무런 문제없이 올바르게 작동하는 것을 알 수 있었다.

표 2 실험 결과 : 참고 문헌 [8]과의 비교

회로 이름	상태수	참고문헌 [8]		본 논문	
		리터럴	CPU	리터럴	CPU
Half Handshake	14	8	0.1	8	0.2
Full Handshake	16	8	0.1	8	0.2
2-4 phase 변환기	19	20	0.1	20	0.3
sender-done	9	5	0.1	5	0.3
mp-forward-pkt	22	18	0.3	14	0.4
MMU	92	22	0.2	22	0.4
master-read	2108	31	0.9	34	0.9
1 SCSI 제어기	16	12	0.1	12	0.3
5 SCSI 제어기	3646	60	3.2	60	1.6
10 SCSI 제어기	N/A	120	20.6	120	13.6
20 SCSI 제어기	N/A	240	150.1	240	150.7
40 SCSI 제어기	N/A	480	1080.5	480	1319.4
60 SCSI 제어기	N/A	720	3868.8	720	6419.8
70 SCSI 제어기	N/A	N/A	N/A	840	11631.2
80 SCSI 제어기	N/A	N/A	N/A	960	22718.1
90 SCSI 제어기	N/A	N/A	N/A	1080	37738.9

만약에 신호 전이 그래프에서 모든 시간 제한 조건이 $[0, \infty]$ 로 주어지면 합성된 회로는 속도 독립 회로가 된다. 다시 말하면 본 논문에서 제안한 합성 방법은 시간 제한 조건을 만족하는 회로를 합성할 수 있을 뿐만 아니라 속도 독립 회로도 합성할 수 있다. 표 2에는 참고 문헌 [8]의 속도 독립 회로 합성 방법과 본 논문의 합성 방법을 비교한 결과가 나타나 있다. 참고 문헌 [8]의 합성 프로그램으로는 Sun 워크스테이션에서 수행되는 프로그램만 공개되어 있기 때문에, 두 프로그램을 주 메모리가 128MB 인 동일한 Sun Sparc 20 워크스테이션에서 수행하였다.

실험 결과에 의하면 두 합성 방법 모두 상태 그래프를 사용하지 않기 때문에 아주 큰 신호 전이 그래프를 합성할 수 있었다. 참고 문헌 [8]의 방법이 본 논문의 방법 보다 더 많은 메모리를 사용하기 때문에 70 SCSI 제어기부터는 메모리 부족으로 인하여 합성에 실패하였다. 수행 속도 면에서는 참고 문헌 [8]의 방법이 본 논문의 방법보다 대체적으로 빠른 것을 알 수 있다. 회로 면적에 있어서는 두 방법이 대부분의 경우 비슷하였다. 회로 *mp-forward-pkt* 같은 경우에 본 논문의 합성 방법이 다중 큐브 형태의 조합형 네트워크를 합성하는 것에 반하여 참고 문헌 [8]의 방법에서는 이를 처리할 수 없기 때문에 본 논문의 합성 방법이 보다 작은 면적의 회로를 생성하였다. 회로 *master-read*의 경우에는 회로 최적화 과정에서 참고 문헌 [8]의 방법이 보다 개선된 방법을 사용하기 때문에 작은 면적의 회로를 생성하였다.

4. 결론

본 논문에서는 시간 제한 조건을 가진 신호 전이 그래프로부터 비동기 회로를 합성하는 방법을 제안하였다. 기존의 방법들이 지수승의 데이터 복잡도를 갖는 상태 그래프를 이용하는 반면에 본 논문에서는 신호 전이들의 관계를 이용하여 회로를 합성할 수 있음을 보였다. 그리고 필요한 신호 전이들의 관계는 타이밍 분석과 그래프 탐색에 의하여 구할 수 있음을 보였다. 실험 결과를 통하여 본 논문에서 제안하는 방법이 기존의 방법들의 문제점중의 하나인 상태 수의 급증 문제를 극복할 수 있음을 보였다. 또한 시스템 단계의 큰 회로를 다루는 상위 단계의 합성에서는 대규모의 신호 전이 그래프를 합성하는 것이 요구되기 때문에 본 논문의 합성 알고리즘은 상위 단계 합성 알고리즘 개발에 효과적으로 활용될 수 있을 것이다. 본 논문에서 제안한 합성 방법은 결정성 신호 전이 그래프만을 다룰 수 있다. 앞으

로 본 논문의 알고리즘을 확장하여 선택 행위가 포함된 신호 전이 그래프도 처리할 수 있도록 해야 할 것이다.

참고 문헌

- [1] T.A. Chu, "Synthesis of Self-timed VLSI Circuits from Graph Theoretic Specifications," *Ph.D. Thesis*, Massachusetts Institute of Technology, 1987.
- [2] A.J. Martin, "Programming in VLSI: From communicating processes to delay-insensitive circuits," In C. A. R. Hoare, editor, *Developments in Concurrency and Communication, UT Year of Programming Series*, pp. 1-64. Addison-Wesley, 1990.
- [3] P.A. Beerel and T. H.-Y. Meng, "Automatic Gate-Level Synthesis of Speed-independent Circuits," *Proceedings of International Conference on Computer Aided Design*, pp. 581-586, Nov. 1992.
- [4] V.I. Varshavsky, V.B. Marakhovshy, and V.V. Smolensky. "Designing self-timed devices using the finite automaton model," *IEEE Design & Test of Computers*, Vol. 12, No. 1, pp. 14-23, Spring 1995.
- [5] K.Y. Yun, "Automatic synthesis of extended burst-mode circuits using generalized C-elements," *In Proc. European Design Automation Conference (EURO-DAC)*, pp. 290-295, Sep. 1996
- [6] L. Lavagno, K. Keutzer, and A. Sangiovanni-Vincentelli, "Algorithms for Synthesis of Hazard-Free Asynchronous Circuits," *Proceedings of the 28th Design Automation Conference*, 1991.
- [7] C.J. Myers, T. H.-Y. Meng, "Synthesis of Timed Asynchronous Circuits," *IEEE Transactions on VLSI Systems*, pp. 106-119 Jun. 1993.
- [8] E. Pastor, J. Cortadella, A. Kondratyev, and O. Roig, "Structural Methods for the Synthesis of Speed-Independent Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 17, No. 11, pp. 1108-1129, Nov. 1998.
- [9] C.E. Molnar, I.W. Jones, B. Coates, and J. Lexau. "A FIFO ring oscillator performance experiment," *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, April 1997.
- [10] K. McMillan, D.L. Dill, "Algorithms for Interface Timing Verification," *In Proceedings of International Conference on Computer Design*, 1992
- [11] P. Vanbekbergen, G. Goossens, D. De Man, "Specification and Analysis of Timing Constraints in Signal Transition Graphs," *In Proceeding of the European Design Automation Conference*, 1992
- [12] T. Amon, H. Hulgaard, G. Borriello, S. Burns, "Timing Analysis of Concurrent Systems," *Technical Reports UW-CS-TR-92-11-01*, University of Washington, 1992
- [13] A. Kondratyev, M. Kishinevsky, B. Lin, P. Vanvekbergen, and Yakovlev, "Basic Gate Implementation of Speed-independent Circuits," *In Proceedings of Design Automation Conference*, pp. 56-62 Jun. 1994.
- [14] K.S. Stevens, S.V. Robinson, and A.L. Davis, "The Post Office - Communication Support for Distributed Ensemble Architectures," *In Proceeding of 6th International Conference on Distributed Computing Systems*, pp. 567-571, 1986
- [15] T.G. Rokicki and C.J. Myers, "Automatic Verification of Timed Circuits," *In Proceedings of the conference on Computer Aided Verification*, June 1994



김희숙

1991년 2월 원광대학교 전자계산공학과 졸업. 1995년 8월 원광대학교 교육대학원 전자계산교육전공 석사학위 취득. 1996년 8월 ~ 현재 원광대학교 컴퓨터공학과 박사과정 수료. 관심분야는 비동기 회로 설계, 컴퓨터 그래픽스, 멀티미

디어



정성태

1987년 2월 서울대학교 컴퓨터공학과 졸업. 1989년 2월 서울대학교 컴퓨터공학과 석사학위 취득. 1994년 8월 서울대학교 컴퓨터공학과 박사학위 취득. 1994년 9월 ~ 1995년 2월 한국전자통신연구소 박사후 연수 연구원. 1995년 3월 ~ 현재 원광대학교 컴퓨터 및 정보통신공학부 교수 1999년 1월 ~ 1999년 12월 미국 Univ. of Utah Post-Doc. 관심분야는 비동기 회로 설계, 컴퓨터 그래픽스