

속도 독립 회로를 위한 직접 합성 시스템

(A Direct Synthesis System for Speed-independent Circuits)

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

요약 본 논문에서는 자유 선택 신호 전이 그래프와 비동기 유한 상태기로 기술된 회로 명세로부터 직접 속도 독립 회로를 합성하는 시스템에 대해 기술한다. 기존의 상태 그래프 기반의 합성 시스템은 상태의 수가 지수승으로 증가할 수 있기 때문에 큰 규모의 회로에 대해서는 합성에 실패할 수 있다는 문제점을 가지고 있다. 이를 해결하기 위해 여러 직접 합성 방법들이 제안되었는데, 본 논문의 합성 시스템은 마크드 그래프 분할 방법과 임시 전이의 사용을 허용함으로써 합성할 수 있는 회로의 범위를 넓혔다. 기존의 벤치마크 회로에 대한 실험결과 본 합성 시스템은 기존의 상태 그래프 기반의 합성 시스템에 비하여 현저하게 수행 속도를 단축시킬 수 있었고 기존의 직접 합성 시스템에 비하여 보다 확장된 그리고 보다 실용적인 회로 명세를 처리할 수 있었다.

Abstract This paper presents a new synthesis system which synthesizes speed-independent circuits directly from a free-choice Signal Transition Graph(STG) and an Asynchronous Finite State Machine(AFSM). Synthesis systems based on a State Graph(SG) may fail to synthesize circuits with a large state space because the number of states can be increased exponentially. Some direct synthesis methods have been developed to overcome this problem. Our system extends the class of specifications by using Marked Graph(MG) decomposition method and by allowing dummy transitions. Experimental result shows that our synthesis system can decrease the synthesis time dramatically with respect to SG based synthesis systems. Also, it shows that our synthesis system can synthesize more practical specifications than previous direct synthesis systems.

1. 서론

비동기 회로는 클럭 스쿼나 분배 문제가 없고 평균 속도로 동작할 수 있고 온도나 습도와 같은 처리 환경의 변화에 민감하지 않고 모듈화 설계가 용이하고 전력 소모가 적은 회로 구현에 이용될 수 있는 등 동기 회로에 비하여 여러 가지 장점을 가지고 있기 때문에 비동기 회로에 대한 관심이 증가되고 있다. 비동기 회로는 여러 부류로 분류될 수 있는데, 그 중의 한 부류인 속도 독립 회로(speed-independent circuits)는 논리 게이트의 지연 시간에 관계없이 올바르게 동작하도록 보장되

는 견고성 가지고 있기 때문에 속도 독립 회로의 합성을 위한 많은 방법들이 개발되었다[1,2,3,4,5,6].

기존의 속도 독립 회로 합성 방법들[1,2,3]에서는 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 방법들은 상태 수가 큰 경우에는 합성에 많은 시간이 소요되거나 메모리의 부족으로 인하여 합성에 실패할 수 있는 문제를 가지고 있다. 이러한 문제를 극복하기 위해서 신호 전이 그래프로부터 상태 그래프를 이용하지 않고 직접 신호 전이들의 관계를 이용하여 회로를 합성하는 방법들[4,5,6,7,8]이 제안되었다. 참고 문헌 [4,5]의 방법에서는 유일 상태 코딩 특성을 만족하는 신호 전이 그래프만을 처리할 수 있는 문제점을 가지고 있다. 참고 문헌 [6]의 방법은 처리할 수 있는 신호 전이 그래프가 다른 방법에 비하여 제한적인 문제점을 가지고 있다. 참고 문헌 [7]의 방법은 더 적은 면적의 회로가 존재함에

· 본 연구는 1999년 원광대학교 연구비 지원에 의하여 연구되었음.

[†] 학생회원 : 원광대학교 컴퓨터공학과
hskim@gaebiyok.wonkwang.ac.kr

^{**} 종신회원 : 원광대학교 컴퓨터공학과 교수
stjung@wonkwang.ac.kr
hspark@wonkwang.ac.kr

논문접수 : 2000년 6월 8일
심사완료 : 2000년 12월 25일

도 불구하고 이를 발견하지 못하는 문제점을 가지고 있다. 참고 문헌 [8]의 방법은 복합 게이트로 구성된 회로를 합성한다는 문제점을 가지고 있다.

본 논문의 합성 시스템은 비동기 유한 상태기 및 자유 선택 신호 전이 그래프에 의해서 기술된 회로 명세로부터 단순 게이트 또는 복합 게이트로 구성된 속도 독립 회로를 합성한다. 비동기 유한 상태기는 자유 선택 신호 전이 그래프로 변환한 다음에 합성한다. 자유 선택 신호 전이 그래프의 한 부 집합으로서 선택 행위를 포함하지 않는 신호 전이 그래프를 마크드 그래프라 한다. 본 논문에서는 자유 선택 신호 전이 그래프를 마크드 그래프로 분할한 다음에 각 마크드 그래프 구성 요소에 대하여 회로를 합성한 다음에 그 결과를 합병하는 방법을 사용한다. 마크드 그래프로부터의 합성을 위해서는 먼저 마크드 그래프로부터 직접 신호 전이 사이의 병렬 관계와 활성화 관계를 구한다. 그 다음에 이들 관계에 근거하여 신호 전이 사이의 우선 순위 그래프를 생성한 다음에 그래프 탐색을 통하여 속도 독립 회로를 합성한다.

실험 결과에 의하면 본 논문의 합성 시스템은 기존의 상태 그래프 기반의 합성 방법보다 상태 공간이 큰 신호 전이 그래프에 대하여 수행 시간을 현저히 단축시킬 수 있다. 또한 기존의 직접 합성 방법에 비하여서도 다양한 회로 기술 방법을 제공하고 기존의 방법이 합성할 수 없는 회로도 합성할 수 있었다.

2. 회로 명세

2.1 자유 선택 신호 전이 그래프

그림 1에는 전이가 신호의 전이로 해석된 Petri 넷 [9]인 자유 선택 신호 전이 그래프의 예가 나타나 있다. 그림에서 신호 전이는 $acksend+$, $acksend-$ 등과 같이 이름으로 나타낸다. 한 신호의 전이가 여러 번 발생하면 $sending+ / 1$, $sending+ / 2$ 와 같이 나타낸다. 원은 장소를 나타내고 원안의 검은색 점은 토큰을 나타낸다. 장소와 전이 사이의 아크는 흐름 관계를 나타낸다. 여기에서는 그림을 간단히 하기 위해 입력 전이와 출력 전이가 하나씩인 장소는 생략하였다. 신호 전이 그래프에 의하여 기술되는 시스템의 상태는 표식(marking)에 의해 나타낸다. 한 표식에서 전이 t 의 모든 입력 장소가 하나의 토큰을 가지면 전이 t 가 활성화된다. 활성화된 신호 전이가 일어나면 전이 t 의 각 입력 장소에 있던 토큰들은 제거되고 전이 t 의 각 출력 장소에 토큰이 하나씩 추가된다. 출력 전이가 여러 개인 장소에 토큰이 주어지면 출력 전이 모두가 활성화되지만 그 중의 하나만 일어날 수 있고 어느 전이가 일어날 것인지는 자유롭게

선택될 수 있다. 이러한 이유에서 자유 선택 신호 전이 그래프라 한다.

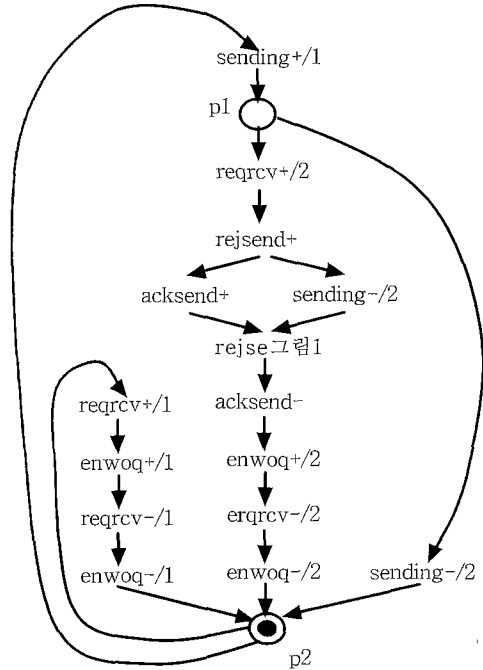


그림 1 자유 선택 신호 전이 그래프의 예

신호 전이 그래프에서 출력 전이가 둘 이상인 장소는 선택 장소라 하고 입력 전이가 둘 이상인 장소를 합병 장소라 하기로 하자. 그림 1의 신호 전이 그래프에서는 장소 $p1$ 은 선택 장소이고 장소 $p2$ 는 선택 장소이면서 합병 장소이다. 신호 전이 그래프에서 전이 $s*$ 의 출력 장소가 전이 $t*$ 의 입력 장소인 경우에 전이 $s*$ 를 전이 $t*$ 의 활성화 전이라 한다. 여기에서 $s*$ 는 $s+$ 또는 $s-$ 를 나타낸다. 그림 1의 신호 전이 그래프에서 $reqrcv+ / 2$ 전이의 활성화 전이는 $sending+ / 1$ 전이이다.

2.2 비동기 유한 상태기

신호 전이 그래프와 함께 비동기 회로의 기술에 많이 사용되는 비동기 유한 상태기는 $\langle S, T, I_0 \rangle$ 의 튜플로서 정의된다. S 는 상태의 집합을 나타낸다. T 는 상태 전이의 집합으로서 $T \subseteq S \times S$ 의 관계를 가지며 각 상태 전이는 입력 신호 전이들의 집합(입력 조건)과 출력 신호 전이들의 집합(출력 조건)을 레이블로 가진다. I_0 는

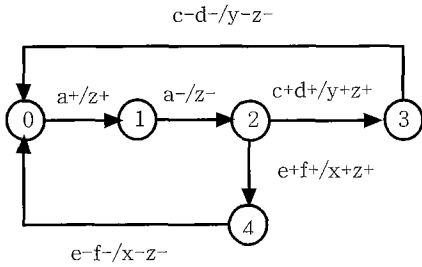


그림 2 비동기 유한 상태기의 예

초기 상태를 나타낸다. 그림 2에는 비동기 유한 상태기의 예가 나타나 있다.

본 논문에서는 비동기 유한 상태기를 신호 전이 그래프로 변환하여 합성한다. 신호 전이 그래프로 변환은 참고 문헌 [10]의 방법을 확장하여 구현하였다. 참고 문헌 [10]에서는 자유 선택 조건과 자유 합병 조건이 만족되는 비동기 유한 상태기만을 신호 전이 그래프로 변환할 수 있다. 자유 선택 조건이란 $(a, b) \subseteq T$ 와 $(a, c) \subseteq T$ 두 조건이 만족될 때와 같이 상태 a 에서 상태 b 또는 상태 c 로 선택적으로 상태 전이가 일어날 경우에 상태 전이의 입력 조건에 단지 하나의 신호 전이만 나타날 수 있다는 것이다. 자유 합병 조건이란 $(a, c) \subseteq T$ 와 $(b, c) \subseteq T$ 두 조건이 만족될 때와 같이 상태 a 와 상태 b 로부터 상태 c 로 상태 전이가 합병될 경우에 이들 상태 전이의 출력 조건에 단지 하나의 신호 전이만 나타날 수 있다는 것이다. 그림 2의 비동기 유한 상태기는 이러한 두 조건을 만족하지 않는다. 여기에서는 상태 전이 (2,3)과 상태 전이 (2,4)의 경우에 입력 조건에 신호 전이의 수가 둘씩이므로 자유 선택 조건이 만족되지 않는다. 그리고 상태 전이 (3,0)과 상태 전이 (4,0)에서는 출력 조건에 전이가 둘씩이므로 자유 합병 조건이 만족되지 않는다. 이와 같이 많은 경우에 비동기 유한 상태기들이 위의 제한 조건을 만족하지 않으므로 본 논문에서는 이 두 조건이 만족되지 않는 비동기 유한 상태기에 대해서도 신호 전이 그래프로 변환이 가능하도록 확장하였다.

이러한 제한 조건을 없애기 위해서 본 논문에서는 신호 전이 그래프에 모의 전이를 사용할 수 있도록 하였고 참고 문헌 [10]의 변환 방법을 확장하여 필요한 경우에 모의 전이를 삽입할 수 있도록 하였다. 실제 신호의 전이만을 허용하는 신호 전이 그래프[1]에서는 자유 선택 조건과 자유 합병 조건이 만족되지 않을 경우에 비동기 유한 상태기에 의해 기술된 회로의 행위를 표현

할 수 없다. 그림 2의 비동기 유한 상태기에 대한 신호 전이 그래프가 그림 3에 나타나 있다. 여기에서는 \$0, \$1, \$2, \$3과 같이 나타낸 것은 실제 신호의 전이가 아니고 모의 전이를 나타낸다. 장소 $p0$ 에 토큰이 주어지면 전이 $e+$ 와 $f+$ 가 병렬로 발생하거나 또는 전이 $c+$ 와 $d+$ 가 병렬로 발생할 수 있다.

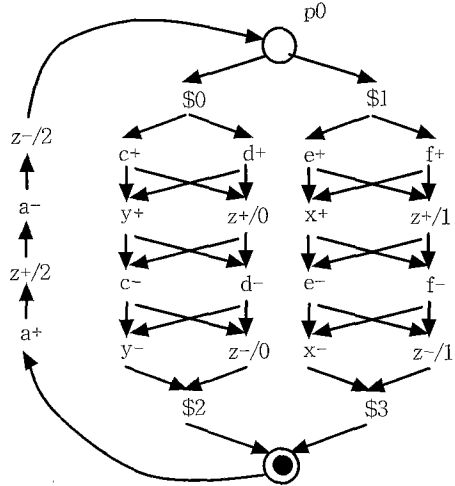


그림 3 그림 2의 비동기 유한 상태기로부터 변환된 신호전이 그래프

2.3 유일 상태 코딩 특성과 완전 상태 코딩 특성

참고 문헌 [4,5]의 합성 시스템은 유일 상태 코딩 특성을 만족하는 신호 전이 그래프로부터 속도 독립 회로를 합성할 수 있다. 자유 선택 신호 전이 그래프가 유일 상태 코딩 특성을 만족하지 않으면 참고 문헌 [11]과 같은 방법을 사용하여 새로운 내부 신호를 삽입함으로써 이 특성이 만족되도록 할 수 있지만, 새로운 신호의 삽입으로 인하여 회로의 면적이 커지게 된다. 또한 신호 전이 그래프에 의해서 기술된 회로가 실제로 구현될 수 있기 위해서 유일 상태 코딩 특성이 반드시 만족되어야 하는 것은 아니다. 참고 문헌 [12]에서는 신호 전이 그래프에 의해서 기술된 회로가 실제로 구현될 수 있기 위해서는 완전 상태 코딩 특성이 필요 조건임을 증명하였다. 완전 상태 코딩 특성이 유일 상태 코딩 특성과 다른 점은 두 상태의 이진 코드가 동일한 것을 허용한다는 것이다. 단, 두 이진 코드가 동일할 때에는 두 상태에서 활성화되는 신호 전이가 동일해야 한다. 그렇지 않으면 동일한 이진 코드를 갖는 두 상태에서 서로 다른 신호 전이가 활성화되도록 해야 하는데, 이런 회로를 구현하는 것은 불가능하다. 그림 1의 신호 전이 그래프에

서는 $enwoq+1$ 전이와 $enwoq+2$ 전이의 입력 장소에 각각 토큰이 있을 때에 두 상태가 동일한 상태 값을 가지므로 유일 상태 코딩 특성을 만족하지 않지만 두 상태에서 활성화되는 전이가 $enwoq$ 신호의 상향 전이로서 동일하므로 완전 상태 코딩 특성을 만족한다. 참고 문헌 [5]의 합성 시스템은 이러한 신호 전이 그래프를 합성하기 위해서 새로운 내부 신호를 추가하여 유일 상태 코딩 특성이 만족하도록 하는데, 이로 인하여 회로의 면적이 커지게 된다.

3. 합성 절차

3.1 회로 모델

본 논문의 합성 시스템은 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. 이 회로 모델에서 설정 네트워크와 해제 네트워크는 두 가지 방법에 의해 구현될 수 있다. 한 가지 방법은 AND, OR, NOT 게이트와 같은 단순 게이트를 사용하여 구현하는 것이다. 일반적으로 단순 게이트를 사용한 구현에서는 회로의 면적이 커지지만 단순한 게이트 라이브러리를 사용할 수 있으므로 실제로 구현하기에 용이하다. 다른 방법은 설정 네트워크와 해제 네트워크 각각을 하나의 복합 게이트로 구현하는 것이다. 복합 게이트를 사용하면 회로의 면적은 작아질 수 있지만 게이트 라이브러리에 여러 가지의 복합 게이트들을 가지고 있어야 하므로 라이브러리 구현이 어렵다는 문제점을 가지고 있다. 본 논문의 합성 시스템은 복합 게이트뿐만 아니라 단순 게이트를 사용한 구현도 지원한다.

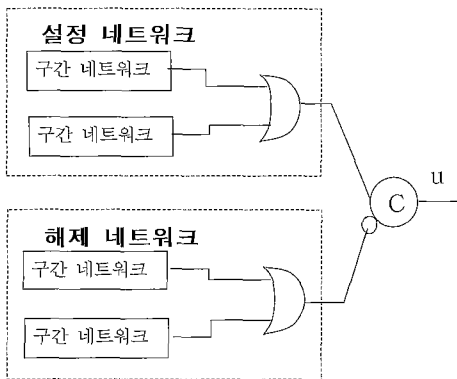


그림 4 목표 회로 모델

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. 참고 문헌 [4,5]에서는 유일 상태 코딩 특성을 만족하는 신호 전이 그래프만 처리하였기 때문에 각 구간 네트워크가 각각 하나의 구간을 구현하였다. 하지만 신호 전이 그래프가 완전 상태 코딩 특성을 만족하는 경우에는 하나의 구간 네트워크가 여러 개의 구간을 구현할 수 있다. 이러한 회로 모델에 의하여 구현된 회로가 해저드를 갖지 않기 위해서는 다음과 같은 조건이 만족되어야 한다.

정의 1 회로 모델의 무해저드(hazard-free) 조건: 신호 u 에 대하여 합성된 회로는 각 구간 네트워크가 다음의 조건을 만족하면 해저드를 갖지 않는다. 여기에서 구간 네트워크가 구간들의 집합 $U = \{u*/1 \mapsto \overline{u*/1}, u*/2 \mapsto \overline{u*/2}, \dots, u*/n \mapsto \overline{u*/n}\}$ 을 구현한다고 하자.

- i) 구간 네트워크의 출력이 각 전이 $u*/i$ ($1 \leq i \leq n$)가 활성화 될 때에 1이 된다.
- ii) 구간 네트워크의 출력이 $u*/i$ 가 일어난 다음에 그리고 $\overline{u*/i}$ 가 활성화될 때까지는 0이 된다.
- iii) 구간 네트워크의 출력이 0이 된 다음에는 임의의 다른 전이 $u*/j$ 가 활성화 될 때까지 구간 네트워크의 출력이 0으로 유지된다.

본 논문의 합성 시스템은 다음의 네 단계로 속도 독립 회로를 합성한다. 첫 번째 단계에서는 자유 선택 신호 전이 그래프를 마크드 그래프로 분할한다. 두 번째 단계에서는 그래프 탐색을 통하여 임의의 두 신호 전이 사이의 병렬 관계 및 인과 관계를 구한다. 세 번째 단계에서는 각 구간 네트워크를 단일 큐브 형태로 구현한다. 네 번째 단계에서는 각 구간 네트워크에 대하여 다중 큐브 형태의 조합형 네트워크가 존재하면 이를 구한다.

3.2 마크드 그래프로의 분할

본 논문에서는 자유 선택 신호 전이 그래프를 마크드 그래프로 분할한 다음에 마크드 그래프에 대하여 합성을 수행하고 그 결과를 합병하는 방법을 사용한다. 합성된 회로가 해저드를 갖지 않기 위해서는 다음과 같은 조건을 만족해야 한다.

정의 2 마크드 분할에 대한 무해저드 조건 : 신호 u 를 신호 전이 그래프 G 의 출력 신호라 하고 U 를 구간 네트워크 N 에 의해서 구현되어야 할 구간들의 집합이라 하자. 그리고 M 을 G 의 마크드 그래프 구성요소

들의 집합이라 하자. 구간 네트워크 N 이 헤저드를 갖지 않기 위해서는 다음의 조건이 만족되어야 한다.

1) 집합 U 에 속한 구간들의 일부 또는 전체를 포함하는 마크드 그래프 구성요소에 대하여 구간 네트워크 N 은 정의 1의 무헤저드 조건을 모두 만족한다.

2) 신호 u 의 구간 중에서 집합 U 의 어떠한 구간도 포함하지 않지만 다른 구간을 포함하는 마크드 그래프 구성요소에 대해서 구간 네트워크 N 은 항상 0의 값을 가진다.

3) 신호 u 의 어떠한 구간도 포함하지 않는 마크드 그래프 구성요소에 대하여 구간 네트워크 N 은 항상 0의 값을 가지거나 항상 1의 값을 가진다.

자유 선택 신호 전이 그래프를 마크드 그래프로 분할하는 단순한 방법으로는 참고 문헌 [1]에서 제안된 분할 방법이 있다. 이 방법에서는 각 선택 장소에 대하여 한 개의 출력 전이를 선택한 다음에, 선택된 출력 전이로부터 도달 가능한 전이들만 남기고 나머지 전이들을 모두 제거함으로써 마크드 그래프를 구한다. 이 방법은 각 선택 장소에 대하여 한 개의 출력 전이를 선택할 수 있는 경우의 수가 지수승으로 증가할 수 있는 반면에 실제로 존재하는 마크드 그래프 구성 요소 수는 상대적으로 작으므로 큰 신호 전이 그래프에 대해서는 많은 시간이 소요된다.

본 논문에서는 연결 그래프를 이용하여 효과적으로 마크드 그래프 구성 요소를 구하는 방법을 사용한다. 이 연결 그래프의 노드는 신호 전이 그래프의 선택 장소와 합병 장소를 나타낸다. 선택 장소와 합병 장소 사이에 존재하는 각각의 경로마다 연결 그래프에서 해당 노드 사이에 하나의 아크가 삽입된다. 이렇게 연결 그래프를 생성한 다음에는 연결 그래프에서 모든 단순 사이클을 구한다. 이렇게 구한 단순 사이클은 하나의 마크드 그래프 구성 요소를 나타낸다. 회로의 합성을 위해서는 신호 전이 그래프에 존재하는 모든 마크드 그래프 구성 요소를 구할 필요 없이 신호 전이 그래프의 모든 신호 전이가 적어도 한 번은 포함되도록 하는 마크드 그래프 구성 요소 집합을 구하면 된다. 따라서, 연결 그래프의 아크가 적어도 한번은 포함되도록 하는 최소의 단순 사이클의 집합을 구함으로써 마크드 그래프 구성 요소의 수를 최소화한다. 그러나, 본 논문에서 제안한 방법은 모든 신호 전이 그래프에 대하여 적용될 수 있는 것은 아니고 선택 장소의 출력 전이가 병렬 관계에 있는 전이를 갖지 않는 경우에만 적용될 수 있다. 따라서 이러한 조건이 만족되지 않는 경우에는 참고 문헌 [1]의 방법을 사용하여 마크드 그래프 구성 요소를 구한다. 그런

데, 기존의 대부분의 벤치마크 회로 명세에 대하여 본 논문에서 제안한 방법을 사용하여 아주 빠른 시간 내에 마크드 그래프 구성 요소를 구할 수 있었다.

그림 5에는 그림 1의 신호 전이 그래프에 대한 마크드 그래프 분할 과정이 나타나 있다. 그림 5의 (a)에는 연결 그래프가 나타나 있고 (b)에는 연결 그래프의 단순 사이

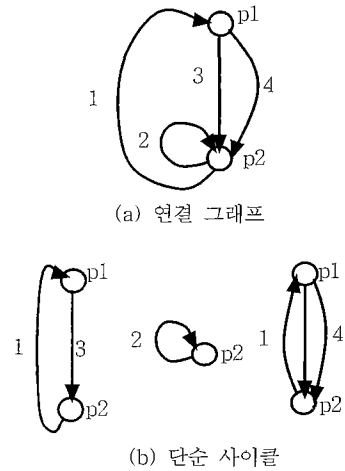


그림 5 그림 1의 신호 전이 그래프의 마크드 그래프 분할 과정

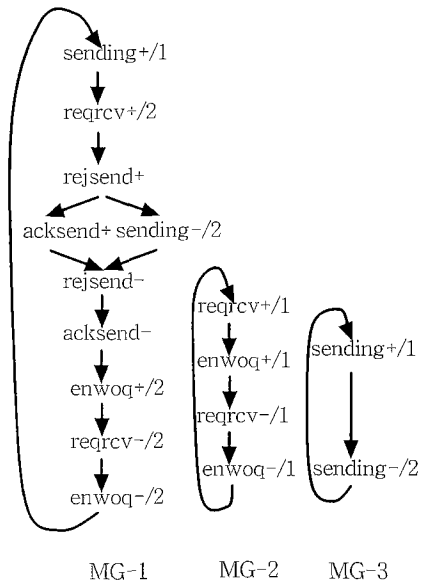


그림 6 그림 1의 신호 전이 그래프에 대한 마크드 그래프 분할

클들이 나타나 있다. 신호 전이 그래프의 모든 신호 전이가 적어도 한 번은 포함되도록 하는 마크드 그래프 구성 요소 집합을 구하기 위해서는 연결 그래프의 아크들이 적어도 한번씩 포함되도록 하는 단순 사이클들의 집합을 구하면 되는데, 그림 5 (a)의 연결 그래프에서는 전체 아크가 포함되도록 하기 위해 모든 단순 사이클이 필요하다. 따라서 그림 1의 신호 전이 그래프에 대해 필요한 마크드 그래프 구성 요소는 그림 6과 같다.

3.3 신호 전이 사이의 관계

3.3.1 병렬 관계

본 논문의 합성 시스템은 병렬 관계와 인과 관계에 기초하여 회로를 합성한다. 병렬 관계는 각 마크드 그래프 구성 요소에 대하여 신호 전이 사이의 도달 가능성을 분석함으로써 구한다. 본 논문에서는 병렬 관계를 구하기 위하여 마크드 그래프 구성 요소를 두 사이클의 비순환 그래프로 펼친다. 원래의 마크드 그래프 구성 요소에 전이 s^* 가 있을 때에, 펼쳐진 그래프의 첫 번째 사이클에서의 s^* 를 $\langle s^*, 0 \rangle$ 로 나타내고 두 번째 사이클에서의 s^* 를 $\langle s^*, 1 \rangle$ 로 나타내기로 하자. 그러면 병렬 관계는 다음과 같이 정의될 수 있다.

정의 3 전이 s^* 와 전이 t^* 가 다음의 관계를 만족하면 두 전이는 병렬관계를 갖는다. 이를 $s^* \parallel t^*$ 와 같이 나타내기로 하자.

- 1) $NR(\langle s^*, 0 \rangle, \langle t^*, 0 \rangle)$
- 2) $NR(\langle t^*, 0 \rangle, \langle s^*, 0 \rangle)$ 또는 $NR(\langle s^*, 0 \rangle, \langle t^*, 1 \rangle)$ 또는 $NR(\langle t^*, 0 \rangle, \langle s^*, 1 \rangle)$

여기서 $NR(x, y)$ 는 x 로부터 출발하여 y 에 도달 가능한 경로가 존재하지 않는다는 것을 나타낸다.

3.3.2 인과 관계

인과 관계는 한 신호 전이가 다른 신호 전이의 발생을 직접적으로 또는 간접적으로 일으킬 수 있는가 그렇지 않은가를 나타내기 위해서 사용된다. 전이 s^* 가 전이 t^* 의 활성화 전이이면 전이 s^* 는 전이 t^* 의 발생을 직접적으로 일으키게 된다. 전이 s^* 가 전이 t^* 를 간접적으로 일으키기 위해서는 전이 s^* 로부터 전이 t^* 로의 경로가 존재하고 그 경로에 전이 s^* 의 역방향 전이 $\overline{s^*}$ 가 있어서는 안 된다는 것이다. 이 조건은 전이 s^* 가 일어난 다음에 전이 t^* 가 일어날 수 있다는 것을 나타내고 전이 $\overline{s^*}$ 는 전이 t^* 가 일어난 다음에 또는 전이 t^* 와 병렬로 일어날 수 있다는 것을 나타낸다. 따라서 전이 t^* 가 일어나기 위해서는 전이 s^* 가 먼저 일어나야 한다. 인과 관계는 자유 선택 신호 전이 그래프에서 깊이 우선 경로 탐색을 수행함으로써 구한다. 이러한

인과 관계를 $s^* \Rightarrow t^*$ 와 같이 나타내기로 한다.

3.4 단일 큐브 구간 네트워크 합성

3.4.1 구간 네트워크의 공유

구간 네트워크를 합성하기 위해 먼저 해야 할 일은 한 구간 네트워크가 구현해야할 구간들을 구하는 것이다. 만약에 두 구간의 시작 전이가 같은 이진 코드를 갖는 두 상태에서 활성화되면 그들은 하나의 구간 네트워크에 의해 구현되어야 한다. 본 논문에서는 전체 상태 공간을 모두 탐색하지 않고서 큐브 정제화 방법을 사용하여 이러한 구간들을 발견한다. 먼저, 신호 전이 그래프의 출력 신호의 각 전이에 대하여 다음과 같이 정의되는 최소 활성화 큐브를 구한다.

정의 4 신호 전이 그래프에서 모든 신호들의 집합을 $S = \{s_1, s_2, \dots, s_n\}$ 이라 하자. 출력 신호의 전이 u^* 에 대한 최소 활성화 큐브, $C_{u^*} = c_1 c_2 \dots c_n$ 은 다음과 같이 정의된다.

- 1) 신호 s_i 의 한 전이가 u^* 와 병렬 관계를 가지면 $c_i = X$ 이다.
- 2) 신호 s_i 의 어떤 전이도 u^* 와 병렬 관계를 가지지 않고 $s_i^* \Rightarrow u^*$ 의 관계가 만족될 때에, s_i^* 가 상향 전이이면 $c_i = 1$ 이다. s_i^* 가 하향 전이이면 $c_i = 0$ 이다.

최소 활성화 큐브를 구한 다음에는 이 큐브가 도달 가능한 상태를 포함하지 않을 때까지 큐브를 정제해 나간다. 큐브의 정제는 그 큐브에 해당하는 전이와 병렬 관계에 있는 장소들의 최소 큐브를 이용하여 구한다. 장소에 대한 최소 큐브는 그 장소에 토큰이 있을 때의 상태들을 나타내는 것으로서 다음과 같이 정의된다.

정의 5 신호 전이 그래프에서 모든 신호들의 집합을 $S = \{s_1, s_2, \dots, s_n\}$ 이라 하자. 장소 p 에 대한 최소 큐브, $C_p = c_1 c_2 \dots c_n$ 은 다음과 같이 정의된다.

- 1) 신호 s_i 의 한 전이가 p 와 병렬 관계를 가지면 $c_i = X$ 이다.
- 2) 신호 s_i 의 어떤 전이도 p 와 병렬 관계를 가지지 않고 전이 s_i^* 로부터 장소 p 로의 경로가 존재하고 그 경로에 s_i^* 의 역방향 전이가 포함되지 않을 때에, s_i^* 가 상향 전이이면 $c_i = 1$ 이다. s_i^* 가 하향 전이이면 $c_i = 0$ 이다.

전이 u^* 가 장소 p_1, p_2, \dots, p_n 과 병렬 관계에 있을 때에 u^* 에 대한 큐브는 다음과 같이 정제한다.

$$C_{u^*}' = C_{u^*} \cap (C_{p_1} \cup C_{p_2} \cup \dots \cup C_{p_n})$$

위와 같이 출력 신호의 각 전이에 대하여 도달 가능한 상태를 나타내는 큐브를 구한 다음에는 다음과 같이 공유되어야 하는 구간들을 구한다. 전이 u^*/i 와 u^*/j 를 신호 u 의 전이라 하고 두 전이의 방향이 동일하다

고 하자. 만약에 두 전이에 대한 큐브가 도달 가능한 상태에서 서로 교차하면 두 전이는 하나의 구간 네트워크에 의해서 구현되어야 한다. 두 큐브 $C_{u^*/i} = c_1c_2...c_n$ 과 $C_{u^*/j} = d_1d_2...d_n$ 이 서로 교차하기 위해서는 모든 $k (1 \leq k \leq n)$ 에 대하여 c_k 와 d_k 가 X 가 아닐 때 $c_k = d_k$ 이어야 한다. 그림 1의 신호 전이 그래프에서 신호의 집합을 $S = \{reqrcv, sending, acksend, enwoq, rejsend\}$ 이라 할 때에 $C_{enwoq+1} = 10000$ 이고 $C_{enwoq+2} = 10000$ 이므로 두 전이는 한 구간 네트워크에 의해 구현되어야 한다. 또한 $C_{enwoq-1} = 00000$ 이고 $C_{enwoq-2} = 00000$ 이므로 두 전이도 한 구간 네트워크에 의해 구현되어야 한다.

3.4.2 초기 큐브

합성하고자 하는 구간 네트워크에 의해 구현되어야 할 구간들의 집합을 $U = \{u^*/1 \mapsto \overline{u^*/1}, u^*/2 \mapsto \overline{u^*/2}, \dots, u^*/n \mapsto \overline{u^*/n}\}$ 라 하자. 무해저드 조건을 만족하는 구간 네트워크를 구하기 위한 다음 단계는 초기 큐브를 구하는 것이다. 초기 큐브는 $u^*/1, u^*/2, \dots, u^*/n$ 의 모든 활성화 신호들의 곱으로 구성된다. 신호 s 의 상향 전이가 u^*/i 의 활성화 전이이면 s 가 활성화 신호가 되고 하향 전이가 활성화 전이이면 \overline{s} 가 활성화 신호가 된다. 그림 1의 신호 전이 그래프의 각 구간 네트워크에 대한 초기 큐브가 그림 7에 나타나 있다. 그러나 초기 큐브들은 무해저드 조건을 만족하지 않을 수 있다. 모든 활성화 신호들이 1이 되면 u^*/i 전이가 활성화되므로 조건 i) 이 만족된다. 그러나 이 활성화 신호들은 $\overline{u^*/i}$ 전이가 활성화되기 전에 0이 되지 않을 수 있다. 또한 0으로 되었더라도 다른 구간의 시작 전이 u^*/j 가 활성화 될 때까지 0으로 유지되지 않을 수 있으므로 무해저드 조건 ii)와 iii)이 만족되지 않을 수

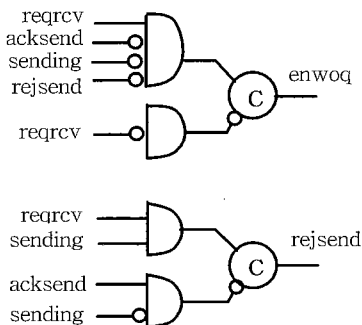


그림 7 그림 1의 신호 전이 그래프에 대한 초기 큐브 네트워크

있다. 이 조건이 만족되지 않을 경우에는 추가의 신호를 초기 큐브에 추가함으로써 조건이 만족되도록 만든다.

3.4.3 추가 신호 발견

추가 신호들을 구하기 위해서는 먼저 각 마크드 구성 요소마다 필요한 추가 신호들을 구한 다음에 이들을 합병하는 방법을 사용한다. 최종 추가 신호의 수를 최소화하기 위해서는 각 마크드 그래프마다 모든 경우의 추가 신호들을 구해야 한다. 정의 2에 나타나 있는 바와 같이 마크드 그래프마다 무해저드 조건을 만족하기 위한 요건이 다르다.

1) U에 속한 구간들을 포함하는 마크드 그래프 구성 요소

U에 속한 구간들을 포함하는 마크드 그래프 구성 요소에 대해서는 구간 네트워크는 정의 1의 무해저드 조건을 모두 만족해야 한다. 이 조건을 만족하도록 하기 위해 추가해야 할 신호를 찾는 것은 참고 문헌 [4]의 방법을 토대로 하였다. 참고 문헌 [4]의 합성 방법에서는 우선 순위 그래프를 생성한 다음에 경로를 탐색함으로써 추가 신호를 발견한다. 우선 순위 그래프는 노드와 아크로 구성된 방향성 그래프이다. 노드는 신호 전이로서 출발 노드, 행선 노드 그리고 중간 노드로 분류된다. 한 구간에 대하여 활성화 신호로 구성된 초기 큐브가 무해저드 조건을 만족하지 않을 때에, 그 이유는 활성화 신호의 값이 1이 되는 시간이 너무 길기 때문이다. 따라서 무해저드 조건을 위반하는 시간 동안에 0의 값을 가지는 신호를 추가함으로써 무해저드 조건이 만족될 수 있도록 하는 것이다. 추가되는 신호가 0의 값을 가져야 하는 시간은 우선 순위 그래프에서 출발 노드의 전이가 발생하는 시간부터 행선 노드의 전이가 발생하는 시간까지로 나타내진다. 그리고 출발 노드로부터 행선 노드의 경로에 있는 전이들로부터 추가 신호를 구한다. 참고 문헌 [4]의 방법에서는 유일 상태 코딩 특성을 만족하는 신호 전이 그래프를 처리할 수 있고 복합 게이트를 사용하므로 본 논문에서는 완전 상태 코딩 특성을 만족하는 마크드 그래프를 처리할 수 있도록 확장하고 또한 단순 게이트를 사용한 회로를 합성하도록 확장하였다. 이와 같이 확장된 합성 절차가 그림 8에 나타나 있다.

그림 8의 합성 절차에서는 먼저 U의 구간들 중에서 G에 포함되어 있는 구간들의 집합 $U_G = \{u^*/1 \mapsto \overline{u^*/1}, u^*/2 \mapsto \overline{u^*/2}, \dots, u^*/m \mapsto \overline{u^*/m}\}$ 을 구한다. 마크드 그래프 구성 요소 G에서 구간들이 $u^*/1 \mapsto \overline{u^*/1}, u^*/2 \mapsto \overline{u^*/2}, \dots, u^*/m \mapsto \overline{u^*/m}$

순서로 나타난다고 하자. 그 다음에는 U_C 에 속한 각 구간 $u^*/i \mapsto \overline{u^*/i}$ 에 대하여 우선 순위 그래프를 이용하여 추가 신호들의 집합을 구한다. 우선 순위 그래프를 생성하기 위해서는 먼저 출발 노드와 행선 노드를 발견한 다음에 중간 노드와 아크들을 추가해 나간다. 전이가 행선 노드 집합에 포함되기 위해서는 다음 구간 시작 전이의 활성화 전이들의 선행 역방향 전이어야 한다. 현재 구간이 i 번째 구간일 때에 다음 구간을 $i+1$ 번째 구간으로 나타내기로 하자. 그러면 다음 구간의 시작 전이는 $u^*/i+1$ 전이가 된다. 전이 e^*/j 가 $u^*/i+1$ 전이의 활성화 전이라면 e^*/j 의 선행 역방향 전이인 $\overline{e^*/j-1}$ 전이가 우선 순위 그래프에서 행선 노드가 된다.

출발 노드는 단순 게이트를 사용하는 경우와 복합 게이트를 사용하는 경우에 서로 다르게 결정된다. 복합 게이트 회로를 합성하는 경우에는 전이 s^*/j 가 출발 노드 집합에 포함되기 위해서는 s^*/j 가 u^*/i 와 $\overline{u^*/i}$ 사이에서 일어나거나 또는 u^*/i 자신이어야 한다. 또한 s^*/j 의 한 역방향 전이 $\overline{s^*/k}$ 에 대하여 $\overline{s^*/k} \Rightarrow u^*/i$ 조건이 만족되어야 한다.

단순 게이트 회로를 합성하는 경우에는 출발 노드가 각 행선 노드별로 독립적으로 결정된다. 전이 $\overline{e^*/j-1}$ 가 행선 노드라 할 때에 구간 네트워크에는 이 행선 노드에 해당하는 신호가 포함되게 된다. 이 신호를 s 라 하자. 신호 e 의 전이중에서 전이 e^*/k 가 $e^*/k \Rightarrow u^*/i$ 의 관계를 만족한다고 하자. 구간 네트워크의 신호 s 는 전이 e^*/k 에 의해 1이 되는데, 이때가 전이 u^*/i 가 활성화되는 때이다. 그런데, 이 활성화 신호는 전이 $\overline{e^*/k}$ 에 의해서 0이 되었다가 전이 $e^*/k+1$ 에 의해서 다시 1이 된다. 전이 $\overline{e^*/k}$ 와 $e^*/k+1$ 가 발생할 동안에 구간 네트워크의 다른 신호들이 1의 값을 가지고 있으면 구간 네트워크의 출력은 0으로 되었다가 다시 1이 되는데 이로 인하여 해저드가 발생하게 된다. 복합 게이트로 구성된 회로를 합성할 때에는 구간 네트워크의 출력은 u^*/i 가 활성화 될 때에 1이 되었다가 $\overline{u^*/i}$ 전이가 활성화 될 때까지 0이 되면 되고 그 사이에는 얼마든지 값이 변해도 해저드가 발생하지 않는다. 따라서 전이 u^*/i 와 전이 $\overline{u^*/i}$ 사이에 있는 전이들이 출발 노드가 되었다. 그러나 단순 게이트로 구성된 회로에서는 구간 네트워크의 출력은 u^*/i 가 활성화 될 때에 1이 되었다가 그 다음에 한번 0이 되면 계속해서 0의 값을 유지해야만 하고 다시 1의 값을 가지게 될 때는 반드시 다음 구간의 시작 전이가 활성화되는 때이어야만 해저드가 발생하지 않는다. 따라서 출발 노드의 전이는

u^*/i 와 $\overline{u^*/i}$ 사이에 있어야 할 뿐만 아니라 전이 e^*/k 와 $e^*/k+1$ 사이에 있어야 한다.

출발 노드들이 세 그룹 SA, SB, SC 로 분류되어 구해진다. 전이 u^*/i 와 전이 $\overline{u^*/i}$ 사이에 있는 출발 노드는 SA 그룹에 포함되고 전이 e^*/k 와 전이 $e^*/k+1$ 사이에 있는 출발 노드는 SB 그룹에 포함된다. 두 조건을 모두 만족하는 출발 노드는 SC 그룹에 포함된다. 그런데, 전이 $e^*/k+1$ 과 전이 $\overline{u^*/i}$ 가 병렬 관계를 갖지 않으면 그룹 SB 에 속하는 노드들은 SA 그룹에도 속하게 되므로 SC 그룹만 구하면 된다.

출발 노드와 행선 노드를 구한 다음에는 합성 절차에 나타나 있는 바와 같이 우선 순위 그래프를 확장한다. 그 다음에는 각 행선 노드에 대하여 해당 출발 노드로부터의 경로를 찾고 경로의 전이로부터 추가 신호들을 구한다. 출발 노드가 SC 그룹에 속할 경우에는 한 출발 노드로부터 행선 노드로의 경로에 있는 전이들만 고려하면 되는 반면에, SA 와 SB 에 속하는 노드들에 대해서는 SA 에 속하는 한 출발 노드로부터의 경로와 SB 에 속하는 한 출발 노드로부터의 경로 둘 다를 모두 고려해야 한다.

find_all_possible_extra_signal_sets(G, U)

```
{
   $U_C = \{ u^*_i \mapsto \overline{u^*_i} \mid u^*_i \mapsto \overline{u^*_i} \in U \text{ and } u^*_i \in G$ 
    and  $\overline{u^*_i} \in G, 1 \leq i \leq m \}$ 
  Foreach  $u^*_i \mapsto \overline{u^*_i}$  in  $U_C$ 
    /* 우선 순위 그래프 생성 */
     $\langle V, E \rangle = \langle \emptyset, \emptyset \rangle$ 
    /* 출발 노드와 행선 노드 결정 */
     $S_N = \{ u^*_i \}$ 
    If (complex_gate_mode)
      Foreach  $s^*_j$  in  $G$ 
        If (  $u^*_i \Rightarrow s^*_j$  and  $s^*_j \Rightarrow \overline{u^*_i}$  and
           $\overline{s^*_k} \Rightarrow u^*_i$  (for some  $k$ ) )
           $S_N = S_N \cup \{ s^*_j \}$ 
        If (Is_an_enabling_transition(  $s^*_j, u^*_{i+1}$  ))
           $D_N = D_N \cup \{ \overline{s^*_{j-1}} \}$ 
           $V = S_N \cup D_N$ 
        else if (simple_gate_mode)
          Foreach  $e^*_h$  (an enabling transition of  $u^*_{i+1}$ )
             $D_N = D_N \cup \{ \overline{e^*_{h-1}} \}$ 
             $SC \overline{e^*_{h-1}} = \{ u^*_j \}$ 
            Find  $e^*_k$  which satisfies  $e^*_k \Rightarrow u^*_i$ 
            If (  $e^*_{k+1} \parallel \overline{u^*_i}$  )
              Foreach  $s^*/j$  in  $G$ 
```



```

If (  $u^*_i \Rightarrow s^*_j$  and not (  $s^*_j \parallel \overline{u^*_i}$  ) and
 $\overline{s^*_l} \Rightarrow u^*_i$  (for some  $l$  ) and
( (  $e^*_k \Rightarrow s^*_j$  or  $\overline{e^*_k} \Rightarrow s^*_j$  )
and not (  $e^*_{k+1} \parallel s^*_j$  )))
 $SC_{\overline{e^*_{k-1}}} = SC_{\overline{e^*_{k-1}}} \cup \{s^*_j\}$ 
Else if (  $u^*_i \Rightarrow s^*_j$  and not(  $s^*_j \parallel \overline{u^*_i}$  )
and  $\overline{s^*_l} \Rightarrow u^*_i$  (for some  $l$  ) )
 $SA_{\overline{e^*_{k-1}}} = SA_{\overline{e^*_{k-1}}} \cup \{s^*_j\}$ 
Else if (  $u^*_i \Rightarrow s^*_j$  and  $\overline{s^*_l} \Rightarrow u^*_i$ 
(for some  $l$  ) and ( (  $e^*_k \Rightarrow s^*_j$  or
 $\overline{e^*_k} \Rightarrow s^*_j$  ) and not (  $e^*_{k+1} \parallel s^*_j$  )))
 $SB_{\overline{e^*_{k-1}}} = SB_{\overline{e^*_{k-1}}} \cup \{s^*_j\}$ 
Else
Foreach  $s^*_j$  in  $G$ 
If (  $u^*_i \Rightarrow s^*_j$  and not (  $s^*_j \parallel \overline{u^*_i}$  ) and
 $\overline{s^*_l} \Rightarrow u^*_i$  (for some  $l$  ) and ( (  $e^*_k \Rightarrow$ 
 $s^*_j$  or  $\overline{e^*_k} \Rightarrow s^*_j$  ) and not (  $e^*_{k+1} \parallel$ 
 $s^*_j$  )))
 $SC_{\overline{e^*_{k-1}}} = SC_{\overline{e^*_{k-1}}} \cup \{s^*_j\}$ 
 $S_{N_{\overline{e^*_{k-1}}}} = SA_{\overline{e^*_{k-1}}} \cup SB_{\overline{e^*_{k-1}}} \cup SC_{\overline{e^*_{k-1}}}$ 
 $V = V \cup S_{N_{\overline{e^*_{k-1}}}}$ 
 $V = V \cup D_N$ 
/* 우선 순위 그래프 확장 */
Foreach unprocessed node  $s^*_j$  in  $V$ 
Foreach  $t^*_j$  in  $G$ 
If ( (  $s^*_j \parallel t^*_k$  or  $\overline{s^*_j} \Rightarrow t^*_k$  ) and
 $t^*_k \Rightarrow \overline{s^*_j}$  and  $\overline{t^*_k} \Rightarrow u^*_i$  (for some  $l$  ) )
 $V = V \cup \{t^*_k\}$ 
 $E = E \cup \{(s^*_j, t^*_k)\}$ 
Foreach  $d_j$  in  $D_N$ 
Foreach  $s_k$  in  $S_{N_{d_j}}$ 
 $EX_{i,j,k} = \text{Find\_all\_possible\_extra\_signals}( u^*_i, d_j, s_k )$ 
 $\text{Find\_all\_possible\_extra\_signal\_sets}(EX)$ 
)

```

그림 8 U에 속한 구간들을 포함하는 마크드 그래프 구성 요소에 대한 추가 신호 발견 절차

3.4.1 절에서 설명한 바와 같이 신호 $enwoq$ 에 대한 설정 네트워크에서 $U = \{enwoq+1, enwoq+2\}$ 이었다. 그림 6에 나타나 있는 마크드 그래프 구성요소에 대하여 $U_{MG-1} = \{enwoq+2\}$ 이고 $U_{MG-2} = \{enwoq+1\}$ 임을 알 수 있다. 따라서 $MG-1$ 과 $MG-2$ 는 집합 U 에 속한 구간을 포함하는 마크드 그래프이므로 위에서

설명한 바와 같이 우선 순위 그래프를 생성하고 경로를 탐색함으로써 추가 신호를 구해야 한다. 그림 9의 (a)에는 $MG-1$ 에서 구간 $enwoq+2 \mapsto enwoq-2$ 에 대하여 생성된 우선 순위 그래프가 나타나 있고 (b)에는 $MG-2$ 에서 구간 $enwoq+1 \mapsto enwoq-1$ 에 대하여 생성된 우선 순위 그래프가 나타나 있다. 여기에서 원은 출발 노드를 사각형은 행선 노드를 나타낸다. 이 구간 네트워크에 대해서는 단순 게이트를 이용한 경우나 복합 게이트를 이용한 경우나 우선 순위 그래프가 동일하다.

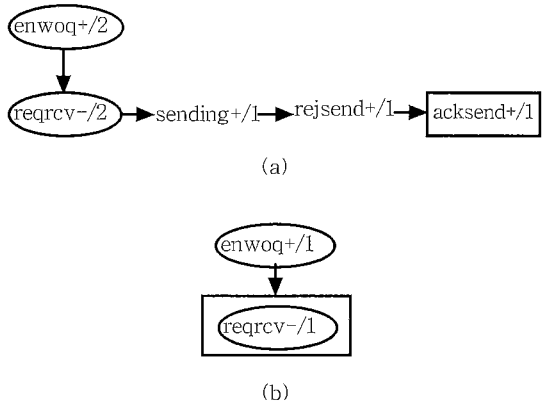


그림 9 우선 순위 그래프의 예 (a) $MG-1$ 에서 구간 $enwoq+2 \mapsto enwoq-2$ 에 대한 우선 순위 그래프 (b) $MG-2$ 에서 구간 $enwoq+1 \mapsto enwoq-1$ 에 대한 우선 순위 그래프

우선 순위 그래프를 생성한 다음에는 각 출발 전이로부터 각 행선 전이로의 경로를 찾음으로써 추가 신호를 구한다. 모든 행선 노드에 대하여 적어도 하나의 출발 노드로부터 그 행선 노드로의 경로가 선택되어야 하고 선택된 경로에 $s+(s-)$ 전이가 존재하면 $\overline{s}(s)$ 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 단계

의 조합형 네트워크 합성 알고리즘에서 다중 큐브 조합형 네트워크를 발견할 가능성을 높인다.

그림 9 (a)에 나타나 있는 구간 $enwoq+1/2 \mapsto enwoq-1/2$ 에 대한 우선 순위 그래프에서 출발 노드로부터 행선 노드로의 경로에 대한 추가 신호는 $E_{MG-1} = \{ reqrcv \overline{sending} \overline{rejsend} \overline{acksend} \}$ 이다. 그리고 그림 9(b)에 나타나 있는 구간 $enwoq+1 \mapsto enwoq-1$ 에 대한 우선 순위 그래프에 대한 추가 신호는 $E_{MG-2} = \{ reqrcv \}$ 이다.

2) U 에 속한 구간들을 포함하지 않지만 신호 u 의 다른 구간을 포함하는 마크드 그래프 구성 요소

U 에 속한 구간들을 포함하지 않지만 신호 u 의 다른 구간을 포함하는 마크드 그래프 구성 요소 G 에 대해서는 구간 네트워크는 항상 0의 값을 가져야 한다. 마크드 그래프 구성 요소 G 에 대하여 구간 네트워크가 항상 0의 값을 갖도록 하는 추가 신호는 두 가지 방법에 의하여 구한다. 첫 번째는 G 가 수행되는 동안에 계속해서 0의 값을 가지는 신호를 찾아서 구간 네트워크에 추가하는 것이다. 이러한 신호를 0-안정 신호라 할 때 이는 다음과 같이 정의된다.

정의 6 마크드 그래프 구성 요소 G 에 대하여 다음의 두 조건을 만족하는 신호 s 를 0-안정 신호라 한다.

- i) G 의 각 전이 t^* 는 신호 s 의 하향 전이들 중의 한 전이 $s-/i$ 와 $s-/i \Rightarrow t^*$ 관계를 갖는다.
- ii) U 에 속하는 각 구간의 시작 전이 u^*/i 는 신호 s 의 상향 전이들 중의 한 전이 $s+/j$ 와 $s+/j \Rightarrow u^*/i$ 의 관계를 갖고 신호 s 의 어떠한 전이와도 병렬 관계를 갖지 않는다.

정의 6과 유사하게 신호 \overline{s} 도 0-안정 신호가 될 수 있다. 만약에 신호 s 또는 \overline{s} 가 G 에 대하여 0-안정 신호일 때에는 이 신호의 어떠한 전이도 G 에 나타나지 않는다. 마크드 그래프 구성 요소 G 에 대하여 여러 개의 0-안정 신호들이 존재하면 이 신호들 중의 한 신호만 구간 네트워크에 추가해도 G 에 대해서는 무해저드 조건이 만족된다. 구간 네트워크가 0의 값을 가지도록 추가해야 할 신호를 발견하는 두 번째 방법은 둘 또는 그 이상의 신호들로 구성되면서 G 에 대하여 항상 0의 값을 가지는 큐브들을 구하여 구간 네트워크에 첨가하는 것이다. G 만을 고려할 때에는 0-안정 신호들만 구하면 추가 신호의 수가 최소화할 수 있지만 여러 마크드 그래프 구성 요소에 대하여 전체적으로 추가되는 신호의 수를 최소화하기 위해서는 0-안정 신호뿐만 아니라 둘 이상의 신호들로 구성된 큐브들도 모두 구해야

한다.

마크드 그래프 구성 요소에 대하여 항상 0의 값을 가지는 둘 이상의 신호로 구성된 큐브를 구하는 문제는 앞 절에서 추가 신호를 발견하기 위한 방법과 유사하게 해결할 수 있다. 앞 절에서 출발 노드로부터 행선 노드까지 한 경로에 대한 추가 신호들로 구성된 큐브는 출발 노드로부터 행선 노드까지의 기간 동안에 항상 0의 값을 가진다. 따라서, 출발 노드로부터 행선 노드까지의 기간이 마크드 그래프 구성 요소의 전체 사이클을 커버할 수 있도록 만들면 마크드 그래프 구성 요소에 대하여 항상 0의 값을 가지는 큐브들을 발견할 수 있는 것이다. 본 논문에서는 출발 노드와 행선 노드를 이와 같이 만들기 위하여 마크드 그래프 구성 요소를 두 사이클의 비순환 그래프로 펼친다. 그 다음에는 펼쳐진 그래프에 대하여 우선 순위 그래프를 생성하고 탐색함으로써 추가 신호들을 발견한다. 원래의 마크드 그래프 구성 요소에 전이 s^* 가 있을 때, 펼쳐진 그래프의 첫 번째 사이클에서의 s^* 를 $\langle s^*, 0 \rangle$ 로 나타내고 두 번째 사이클에서의 s^* 를 $\langle s^*, 1 \rangle$ 로 나타내기로 하자. 펼쳐진 그래프에 대한 우선 순위 그래프에서 $\langle s^*, 0 \rangle$ 와 $\langle s^*, 1 \rangle$ 을 각각 출발 노드와 행선 노드로 선택하면 그 사이의 경로는 마크드 그래프 구성 요소의 전체 사이클을 커버하게 된다. 따라서, 첫 번째 사이클의 전이들 중에서 해당 신호가 추가 신호로 사용될 수 있는 전이를 출발 노드로 선택하고 그 전이에 대한 두 번째 사이클의 전이를 행선 노드로 선택한다. 전이 $\langle s^*, 0 \rangle$ 가 출발 노드로 선택되고 전이 $\langle s^*, 1 \rangle$ 가 행선 노드로 선택되기 위해서는 해당 신호(s^* 가 상향 전이이면 \overline{s} 이고 s^* 가 하향 전이이면 s)가 구간 네트워크의 추가 신호로 사용될 수 있어야 한다. 신호 s 또는 \overline{s} 가 추가 신호로 사용될 수 있기 위해서는 U 에 속한 각 구간의 시작 전이가 활성화 될 때에 그 신호가 1의 값을 가져야 하고 그 구간의 시작 전이가 발생될 때까지는 그 값을 유지해야 한다. 이를 위해서는 s^* 의 한 역방향 전이가 구간의 시작 전이와 인과 관계를 가져야 하고 신호 s 의 어떠한 전이도 구간의 시작 전이와 병렬 관계를 갖지 않아야 한다. 출발 노드와 행선 노드를 생성한 다음에는 노드들과 아크들을 삽입함으로써 우선 순위 그래프를 확장해 나간다. 우선 순위 그래프를 확장해 나가는 기본 원리는 앞 절과 같다. 여기에서는 펼쳐진 그래프에 대해서 우선 순위 그래프를 확장하는 것이 다른 점이다.

3) U 에 속한 구간들을 포함하지 않는 마크드 그래프 구성 요소

이러한 마크드 그래프 구성 요소에 대해서는 구간 네

트위크는 항상 0의 값을 가지거나 항상 1의 값을 가져야 한다. 항상 0의 값을 가지도록 하는 추가 신호는 앞 절과 동일한 방법에 의해 구한다. 마크드 그래프 구성 요소에 대하여 구간 네트워크가 항상 1의 값을 가지기 위해서는 구간 네트워크의 모든 신호들이 마크드 그래프가 수행되는 동안에 항상 1의 값을 가져야 한다. 마크드 그래프 구성 요소 G에 대하여 항상 1의 값을 가지는 신호인 1-안정 신호는 다음과 같이 정의된다.

정의 7 마크드 그래프 구성 요소 G에 대하여 다음의 두 조건을 만족하는 신호 s 를 1-안정 신호라 한다.

- i) G의 각 전이 t^* 는 신호 s 의 상향 전이들 중의 한 전이와 $s+ /i \Rightarrow t^*$ 관계를 갖는다.
- ii) U에 속하는 각 구간의 시작 전이 u^*/i 는 신호 s 의 상향 전이들 중의 한 전이와 $s+ /k \Rightarrow u^*/i$ 의 관계를 갖고 신호 s 의 어떠한 전이와도 병렬 관계를 갖지 않는다.

정의 7과 유사하게 신호 \bar{s} 도 1-안정 신호가 될 수 있다. 구간 네트워크가 1의 값을 갖도록 하기 위해서는 구간 네트워크의 활성화 신호들과 추가 신호들이 1-안정 신호이어야 한다. 그런데, 구간 네트워크에 추가되는 신호는 여러 마크드 그래프 구성 요소에 의하여 결정되므로 여기에서는 일단 1-안정 신호를 모두 구한 다음에 나중에 각 마크드 그래프 구성 요소에 대한 추가 신호들의 집합으로부터 전체적인 추가 신호를 구할 때에 모든 추가 신호들이 1-안정 신호인 경우가 있는지를 검사한다. 위에서 설명한 바와 같이 각 마크드 그래프 구성 요소에 대하여 필요한 추가 신호들의 집합을 구한 다음에 이들로부터 전체적으로 필요한 최소한의 추가 신호를 구한다.

구간의 집합 $U = \{enwoq+1, enwoq+2\}$ 에 대한 구간 네트워크 합성에 있어서 마크드 그래프 $MG-3$ 는 U 의 구간을 포함하지 않는다. 따라서 구간 네트워크는 $MG-3$ 에 대하여 항상 0의 값을 가지거나 항상 1의 값을 가져야 한다. 구간 네트워크의 출력을 항상 0으로 만들기 위한 신호들의 집합은 $E_{OFF_{MG-3}} = \{reqrcv\}$ 이고 1-안정 신호들의 집합은 $E_{ON_{MG-3}} = \{\overline{acksend}, \overline{enwoq}, \overline{rejsend}\}$ 이다. 이들과 앞 절에서 구한 $E_{MG-1} = \{\overline{reqrcv}, \overline{sending}, \overline{rejsend}, \overline{acksend}\}$ 과 $E_{MG-2} = \{reqrcv\}$ 로부터 구간 네트워크가 무해저드 조건을 만족하도록 하기 위한 필요한 추가 신호는 $E = \overline{reqrcv}, \overline{sending}, \overline{rejsend}, \overline{acksend}$ 가 된다. 이와 같은 방법으로 그림 1의 신호 전이 그래프에 대해 합성한 단일 큐브 구간 네트워크가

그림 10에 나타나 있다. 그림 1의 신호 전이 그래프에 대해서는 단순 게이트를 사용한 경우나 복합 게이트를 사용한 경우의 회로가 동일하다.

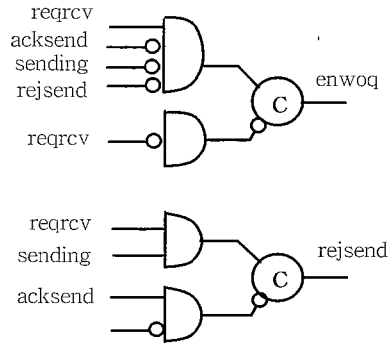


그림 10 그림 1의 신호 전이 그래프에 대한 단일 큐브 구간 네트워크

3.5 다중 큐브 구간 네트워크 합성

본 논문의 합성 시스템은 조합형 네트워크를 발견함으로써 회로의 성능을 향상시킨다. 설정 네트워크가 조합형이면 C-원소와 해제 네트워크를 제거하고 설정 네트워크만으로 출력 신호를 구현함으로써 회로의 면적도 줄이고 속도도 빠르게 할 수 있다. 또한 해제 네트워크가 조합형이면 해제 네트워크만으로 출력 신호를 구현할 수 있다. 앞 단계에서 합성한 단일 큐브 네트워크가 조합형이 아닌 경우에 그 구간 네트워크와 추가의 신호들을 OR 게이트로 연결하여 조합형 네트워크를 합성한다. 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.

조합형 구간 네트워크를 구하는데 있어서도 앞 단계에서와 유사하게 우선 순위 그래프를 이용하여 추가의 신호들을 구한다. 여기에서는 앞 단계의 방법과 출발 노드와 행선 노드를 구하는 조건이 다르고 그래프에서 경로에 $s+$ 전이가 있으면 추가되는 신호가 s_g 되는 것이 다르다. 추가되는 신호의 하나는 앞 단계에서 구한 구간 네트워크의 출력이 0이 되기 전에 1이 되어야 한다. 그리고 구간의 마지막 전이가 활성화 될 때까지 이들 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 신호의 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 $u*/i+1$ 전이로부터 구했지만 여기에서는 $\overline{u*/i}$ 전이가 활성화 될 때까지 1로 유지되는 신호를 찾기 때문에 $\overline{u*/i}$ 전이로부터 행선 노드를 구한다. 그림 1의 신호 전이 그래프에 대해서는 다중 큐브 조합형 구간 네트워크는 존재하지 않는다. 그런데 앞에서 구한 *enwoq* 신호의 설정 네트워크가 이미 조합형이므로 최종적인 합성 결과는 그림 11과 같다.

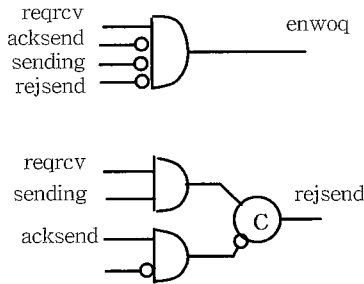


그림 11 그림 1의 신호 전이 그래프에 대한 합성 결과

4. 실험 결과

표 1에는 벤치마크 회로들에 대하여 기존의 합성 시스템과 비교한 실험 결과가 나타나 있다. 이 실험은 주 메모리가 128MB인 동일한 Sun Sparc 워크스테이션에서 수행되었다. 여기에서는 본 논문의 합성 결과와 기존의 합성 시스템 중에서 수행 파일이 공개되어 있는 시스템인 참고 문헌 [3, 8]의 합성 결과를 리터럴의 수와 CPU 시간을 기준으로 비교하였다. 참고 문헌 [3]의 방법은 상태 그래프를 이용하여 단순 게이트와 C-원소로 구성된 회로를 합성한다. 참고 문헌 [8]의 방법은 상태 그래프를 이용하지 않고 신호 전이 그래프로부터 직접 회로를 합성한다. 이 방법에서는 각 출력 신호가 하나의 복합 게이트로 합성된다. 즉, 논리 게이트들과 C-원소가 분리되지 않고 이들이 합쳐져서 전체가 하나의 게이트로 다루어진다. 본 논문의 합성 시스템은 논리 게이트들과 C-원소를 독립된 회로 요소로 다루며 설정 및 해제 네트워크를 단순 게이트 또는 복합 게이트로 합성

한다.

실험 결과에 의하면 작은 상태 공간을 가지는 회로에 대해서는 세 합성 시스템들이 모두 비슷한 합성 시간을 소요하지만 큰 상태 공간을 가지는 회로에 대해서는 직접 합성 방법이 합성 시간을 현저히 단축시킬 수 있다는 것을 알 수 있다. 상태수가 수 만개가 넘어가면 상태 그래프를 이용한 방법은 메모리 부족으로 인하여 합성에 실패한 반면에 본 논문의 방법은 수 십 만개의 상태

표 1 실험 결과

회로 이름	상태 수	참고문헌 [3]		참고문헌 [8]		본 논문		
		리터럴	CPU	리터럴	CPU	복합	단순	CPU
chul33	24	12	0.13	12	0.16	12	12	0.13
chul50	26	11	0.12	11	0.14	11	11	0.11
chul72	13	8	0.04	6	0.12	7	8	0.05
converta	18	20	0.12	16	0.16	20	20	0.14
ebergen	18	14	0.09	8	0.16	12	14	0.11
elatchX	255762	N/A	N/A	52	6.42	52	52	2.65
fail	16	8	0.05	4	0.08	8	8	0.03
half	14	7	0.04	7	0.08	7	7	0.03
hazard	12	10	0.02	4	0.08	8	10	0.05
hybridf	80	14	0.43	8	0.3	14	14	0.17
master-read	2108	33	207.22	28	1.26	33	33	0.84
mp-fwd-pkt	22	19	0.14	14	0.38	14	14	0.17
pipeline15	6006	30	3209.63	30	2.78	30	30	1.83
pipeline17	12376	34	19996.00	34	4.12	34	34	2.45
pipeline18	37128	N/A	N/A	36	4.74	36	36	3.01
pipeline30	N/A	N/A	N/A	60	19.78	60	60	7.13
pipeline60	N/A	N/A	N/A	120	155.90	120	120	47.25
pipeline90	N/A	N/A	N/A	180	520.28	180	180	155.38
nak-pa	58	20	0.44	20	0.32	17	17	0.34
nowick	20	16	0.17	16	0.26	12	12	0.18
qr42	18	14	0.10	8	0.12	12	14	0.12
ram-read-stbuf	39	20	0.32	20	0.50	20	20	0.40
rcv-setup	16	8	0.02	7	0.10	8	8	0.05
rpdiff	22	12	0.07	12	0.24	12	12	0.41
scsi	20	12	0.07	11	0.2	12	12	0.1
scsi4	1126	48	108.46	44	2.18	48	48	1.78
scsi5	4880	60	2580.96	55	4.02	60	60	2.39
scsi6	21910	N/A	N/A	66	5.92	72	72	3.52
scsi10	101072	N/A	N/A	110	23.36	120	120	12.97
scsi20	N/A	N/A	N/A	220	174.84	240	240	175.91
vbe5b	24	10	0.08	10	0.16	10	10	0.08
vbe5c	24	8	0.09	8	0.12	8	8	0.08
vbe6a	192	28	1.85	28	0.46	28	28	0.31
barcode	232	183	44.82	N/A	N/A	183	183	12.68
cd-plaver	824	354	1058	N/A	N/A	354	354	167.96
cache-ctrl	1902	N/A	N/A	N/A	N/A	811	811	615.80
dram-ctrl	109	45	4.70	N/A	N/A	45	45	0.58
hp-ir-it-ctrl	73	53	3.35	N/A	N/A	53	53	0.7
scsi-isend-bm	78	63	1.58	N/A	N/A	63	63	0.71
scsi-trev-bm	77	63	1.65	N/A	N/A	63	63	0.74
scsi-tsend-bm	67	60	1.03	N/A	N/A	60	60	0.57
stetson-p3	27	8	0.08	N/A	N/A	7	8	0.06

를 가지는 회로도 수 분내에 합성할 수 있다는 것이 주

목적 만한 일이다. 그리고 속도가 빠르면서도 합성된 회로의 면적이 대부분 비슷하다는 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. 참고 문헌 [8]의 합성 시스템에 의해 합성된 회로의 면적이 대부분의 경우에 가장 작운데, 그 이유는 한 신호에 대한 전체 회로를 하나의 복합 계이트로 합성하였기 때문이다.

표 1에서 barcode부터 그 아래의 회로들은 비동기 유한 상태기로 기술된 것을 신호 전이 그래프로 변환하여 합성하였다. 그 중에서 barcode[13], cd-player[14], cache-ctrl[15], dram-ctrl[16], scsi[16] 등은 실제로 구현된 회로에 포함되어 있는 비동기 회로들이다. 이들 회로의 기술에 임시 전이가 사용되어야 하는데, 이로 인하여 참고 문헌 [8]의 시스템은 합성에 실패하였다. 참고 문헌 [3]의 합성 시스템은 메모리 부족으로 인하여 cache-ctrl 회로를 합성하는데 실패하였는데, 이는 이 회로가 상태 수는 작지만 전체 회로 크기가 크기 때문인 것으로 분석된다.

이러한 실험 결과로부터 신호 전이 그래프의 크기에 비하여 상태 그래프의 크기는 지수승으로 증가할 수 있음을 알 수 있고 상태 그래프 기반의 합성 시스템은 대규모 회로의 합성에 문제를 가지는 것을 알 수 있다. 그리고 본 논문의 합성 시스템은 합성 시간이 상태의 수에 의존하지 않으므로 이러한 문제를 해결할 수 있음을 보였다. 또한 합성 시간이 빠르면서도 회로의 면적이 더 커지지 않는다는 것이 주목할 만한 일이다.

5. 결 론

본 논문에서는 자유 선택 신호 전이 그래프로와 비동기 유한 상태기로부터 속도 독립 회로를 직접 합성하는 시스템을 개발하였다. 자유 선택 신호 전이 그래프를 마크드 그래프로 분할하여 합성하는 방법을 사용함으로써 완전 상태 코딩 특성을 만족하는 자유 선택 신호 전이 그래프를 합성할 수 있도록 하였고 임시 전이를 사용함으로써 합성할 수 있는 회로 기술의 범위를 확장하였다. 실험 결과에 의하면 본 논문의 합성 시스템이 기존의 상태 그래프 기반의 합성 시스템에 비하여 합성 시간은 현저히 단축시키면서 회로의 면적은 더 커지지 않는 회로를 합성할 수 있었다. 또한 기존의 상태 그래프 기반의 합성 시스템이 합성에 실패한 회로도 합성할 수 있었다. 기존의 직접 합성 시스템에 비해서도 합성 시간을 단축시켰으며 합성할 수 있는 회로의 범위도 확장하였

다. 향후에는 본 논문의 합성 시스템을 보다 상위 단계로 확장하는 연구가 수행되어야 할 것이다. 시스템 단계의 큰 회로를 다루는 상위 단계의 합성에서는 대규모의 신호 전이 그래프를 합성하는 것이 요구되기 때문에 본 논문의 합성 시스템은 상위 단계 합성 시스템 개발에 효과적으로 활용될 수 있을 것이다.

참 고 문 헌

- [1] T.A. Chu, "Synthesis of Self-timed VLSI Circuits from Graph Theoretic Specifications," *Ph.D. Thesis*, Massachusetts Institute of Technology, 1987.
- [2] L. Lavagno, K. Keutzer, and A. Sangiovanni-Vincentelli, "Algorithms for Synthesis of Hazard-Free Asynchronous Circuits," *Proceedings of the 28th Design Automation Conference*, 1991.
- [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] Sung Tae Jung and Chu Shik Jhon, "Direct Synthesis of Efficient Speed-independent Circuits from Deterministic Signal Transition Graphs," *Proceedings of International Symposium on Circuits and Systems*, vol. 1 of 6, pp. 307-310, London, June, 1994.
- [5] Sung Tae Jung, Eun Sei Park, Jung Sik Kim, and Chu Shik Jhon, "Automatic Synthesis of Gate-level Speed-independent Control Circuits from Signal Transition Graphs," *Proceedings of International Symposium on Circuits and Systems*, pp. 1211-1214, May, 1995.
- [6] C. Ykman-Couvreur, B. Lin, G. Goossen, and H. De Man, "Synthesis and Optimization of Asynchronous Controllers based on Extended Lock Graph Theory," *Proceedings of European Design Automation Conference(EDAC)*, pp. 512-517, Feb. 1993.
- [7] K.-J. Kin, C.-W. Kuo, and C.-S. Lin, "Synthesis of Hazard-Free Asynchronous Circuits Based on Characteristic Graph," *IEEE Transactions on Computers*, Vol. 46, No 11, pp. 1246-1263, Nov. 1997.
- [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] T. Murata, "Petri nets: Properties, Analysis and Applications," *Proceedings of IEEE*, Vol. 77, No. 4,

pp. 541-580, Apr. 1989.

[10] Sungtae Jung, "Synthesis of Asynchronous Control Circuits from Asynchronous Finite State Machines," Proceedings of JTC-CSCC'95, pp. 535-538, July 1995

[11] E. Pastor and J. Cortadella, "An Efficient Unique State Coding Algorithm for Signal Transition Graph," in Proceedings of International Conference on Computer Design(ICCD), pp. 174-177, Oct. 1993.

[12] 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.

[13] P.R. Panda and N. Dutt, "1995 High Level Synthesis Repository," Tech. Rep. 95-04, University of California, Irvine, U.S.A, 1995

[14] J. Kessels, K. van Berkel, R. Burgess, M. Roncken, and F. Schalij, "An Error Decoder for the Compact Disc Player as an Example of VLSI Programming," Tech. Rep., Philips Research Laboratories, Eindhoven, The Netherlands, 1992

[15] S.M. Nowick, M.E. Dean, D.L. Dill, and M. Horowitz, "The Design of a High-Performance Cache Controller: A Case Study in Asynchronous Synthesis," Integration, the VLSI Journal, Vol. 15, No. 3, pp 241-262, Oct. 1993.

[16] S.M. Nowick, K.Y. Yun, D.L. Dill, "Practical Asynchronous Controller Design," in Proc. International Conference on Computer Design(ICCD), pp. 341-345, Oct. 1992



박희순

1972년 연세대학교 전기공학과 졸업. 1983년 전남대학교 전기공학과 석사학위 취득. 1986년 전북대학교 전기공학과 박사학위 취득. 1983년 ~ 현재 원광대학교 컴퓨터 및 정보통신공학부 교수. 관심분야는 컴퓨터구조, 병렬처리, 컴퓨터제어

어



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, 멀티미디어

디어



정성태

1987년 2월 서울대학교 컴퓨터공학과 졸업. 1989년 2월 서울대학교 컴퓨터공학과 석사학위 취득. 1994년 8월 서울대학교 컴퓨터공학과 박사학위 취득. 1994년 9월 ~ 1995년 2월 한국전자통신 연구소 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