

# 이산 시스템의 설계와 해석을 위한 확장된 마크흐름선도의 재정의와 회로변환

여 정 모<sup>†</sup>

## 요 약

확장된 마크흐름선도(EMFG: Extended Mark Flow Graph)는 이산 시스템을 개념적으로 또는 상세하게 설계할 수 있는 강력한 도구일 뿐만 아니라 시스템을 직접적으로 구현하기 위한 적당한 표현이다. 그러나 EMFG의 다소 복잡한 표현으로 인해 수학적 해석이 불가능하여 시스템의 설계나 해석에 어려움이 있다. 따라서 본 연구에서는 수학적 해석이 가능하도록 EMFG의 표현을 재정의하였고, 표현 불가능한 부분이 가능하도록 트랜지션에 출력 역 아크를 도입하였다. 또한 EMFG의 회로변환에서 잘못 변환되는 부분들을 해결하였으며, 수식을 사용한 회로변환 방법을 제안하였다. 마지막으로 기존의 EMFG와 재정의된 EMFG를 비교하였고, 재정의된 EMFG로써 0101 감지기 회로를 설계하고 대응하는 회로로 변환하였다.

## A Redefinition and Circuit Translation of the Extended Mark Flow Graph for the Design and Analysis of the Discrete-event System

Jeong Mo Yeo<sup>†</sup>

### ABSTRACT

The EMFG (Extended Mark Flow Graph) is not only a powerful tool for designing the discrete-event system conceptually or specifically but also a good representation tool for implementing the system directly. But it is difficult to design or analyze the system because the impossibility of mathematical analysis results from some complicated representation of the EMFG. In this paper, therefore we redefined EMFG representations to provide the possibility of mathematical analysis, and employed an output inhibit arc of a transition, which makes possible the adequate representations for the indescribable parts. We also solved trouble parts in the circuit translations of the EMFG and proposed the circuit translation method by using the mathematical formula. Finally, we compared the conventional EMFG with the redefined EMFG, and we designed a 0101 detector with the redefined EMFG and translated it to the corresponding circuit.

### 1. 서 론

시스템의 병렬성과 동시성을 표현할 수 있는 Petri Net[1]가 J. L. Petri에 의해 제안되었으며, 여러 선도(graph)들이 Petri Net에서 파생되었다. 마크선도(Mark Graph)[2,4-7]는 Petri Net에서 제한을 가한 선도이며, 방향성 마크선도(Marked Directed Graph)[3]는 마크선도에서 충돌(conflict)하지 않고 동시성을 표

현할 수 있을 뿐만 아니라 그래프의 안전성(safeness), 활성(liveness) 및 도달가능성(reachability)을 해결하는 알고리즘 등을 제시하였다. 그리고 마크흐름선도(MFG: Mark Flow Graph)[8,11,12,16]는 Petri Net에서 모든 place의 마크가 안전(safe)하도록 제한을 가한 선도이다. Petri Net 및 각종 선도들은 회로의 분석 및 해석[4-7]에 이용되었으며, 이산제어 시스템의 표현 및 구현[12,16-18]에도 이용되었고, 논리 제어기(logic controller)[19]나 라다(ladder)[22]를

<sup>†</sup> 부경대학교 전자계산학과

구성할 때 사용되기도 하였으며, 공장 자동화[11,13, 14,20,21,23] 등에도 응용되었다.

그리고 동기나 비동기의 이산제어 시스템을 설계하여 구현하거나 분석하는데 적합하도록 확장된 마크흐름선도(EMFG: Extended Mark Flow Graph)가 [9]에서 제안되었다. EMFG는 Petri Net나 각종 선도로서 시스템을 설계하는 것과 마찬가지로 시스템의 동시성이나 병렬성을 잘 표현할 수 있을 뿐만 아니라 시스템을 개념적으로 설계할 수 있게 하고, 시스템의 동작을 상세하게 설계할 수 있게 한다. 즉 EMFG는 설계자의 생각을 그대로 표현하기에 적합한 강력한 설계 도구이다. 또한 EMFG를 구성하고 있는 요소를 각각 일대일로 변환하면 직접적인 회로를 얻을 수 있어 시스템의 구현이 아주 쉽다. 응용 예로 [9]에서는 4층 리프트 제어회로가 EMFG로 설계되어 구현되었다. 그리고 EMFG는 순서 제어회로에도 그 적용[10]이 쉬울 뿐만 아니라 설계된 EMFG 회로도 간략화[15]될 수 있다.

그러나 기존의 EMFG[9,10,15]에서는 박스가 게이트 박스와 레벨 박스 및 펄스 박스로 나누어져 있고, 아크도 일반 아크와 보존 아크 및 역 아크로 나누어져 있어 그 표현이 복잡하다. 이로 인해 수학적 표현이 불가능하여 시스템의 해석이나 설계에 어려움이 있었고, 아크들간의 역할이 명백하지 않아 표현이 불가능한 부분도 존재하였다. 또한 기존의 EMFG가 회로로 변환될 때에도 잘못 변환되는 경우도 있었고, 기존의 EMFG를 구성하는 원소들이 일대일로 변환되어 회로가 구성되므로 복잡한 시스템에서는 그 구현 과정이 복잡하였다.

따라서 본 연구에서는 표현이 다소 복잡한 기존의 EMFG를 재정의하여 그 표현을 단순화함으로써 시스템의 설계가 용이하도록 하고, 또한 수학적으로 표현할 수 있게 함으로써 시스템의 분석이나 해석이 쉽게 이루어지고 컴퓨터 프로그램에 의해서도 그 분석 및 해석이 가능하도록 하였다. 그리고 트랜지션의 출력 아크로 역 아크를 도입하여 아크들간의 역할을 명백히 하고, 표현 불가능하던 부분들도 표현 가능하게 하였다. 또한 회로변환도 GD 플립플롭을 도입하여 잘못 변환되는 경우가 올바르게 변화되도록 하였으며, 더욱이 수식으로 회로로 변환하게 함으로써 설계된 EMFG가 쉽고 간편하게 회로로 변환되도록 하였다. 마지막으로 기존의 EMFG와 재정의된 EMFG

를 비교하고, 응용 예로 0101 감지회로[25]를 재정의된 EMFG로 설계하고 그에 대응되는 회로로 변환하였다.

## 2. 기존의 EMFG

이산 시스템을 기존의 EMFG[9,10,15]로 설계하는 경우, EMFG의 구성 요소들이 모두 포함되는 예는 찾아보기 어렵다. 따라서 EMFG의 모든 요소가 포함된 가상적인 EMFG를 그림 1과 같이 구성하여 기존의 EMFG의 정의와 표기 및 성질 등을 설명하기로 한다. 그리고 본 장에서의 EMFG는 기존의 EMFG를 뜻한다.

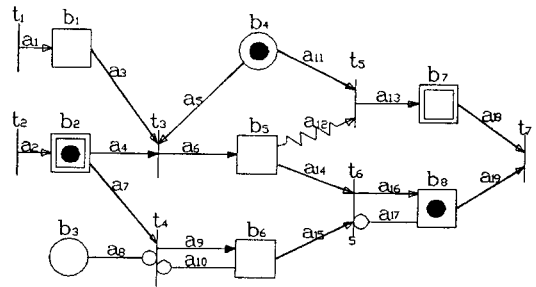


그림 1. EMFG의 한 예

### 2.1 EMFG의 정의 및 표기

EMFG는 박스(box), 트랜지션(transition), 아크(arc) 들로 구성되는 마크(mark)를 갖는 방향성 선도로 정의된다.

박스는 상태(개념적인 상태, 제어 상태, 신호 상태 등)를 나타내며, 상태의 만족 여부는 박스 내에 마크를 두어 표시한다. 즉 박스의 상태가 만족되면 마크를 두고, 만족되지 않으면 마크를 두지 않는다.

박스에는 세 종류의 박스[9,10]가 있다. 첫째, 레벨 박스(level box)는 자신의 조건에 의해 또는 자신과 다른 상태(들)의 조건(들)에 의해서 변화되는 상태를 나타내고, 사각형으로 표기된다. 둘째, 펄스 박스(pulse box)는 상태가 만족되는 즉시 다른 상태 변화에 기여할 수 있고, 만약 기여되지 못하면 즉시 만족되지 않는 상태로 변화되며, 이중 사각형으로 표기된다. 셋째, 게이트 박스(gate box)는 기계적인 또는 수동적인 변화에 의해서만 변화되는 상태를 나타내고, 원으로 표기된다. 그림 1에서 b1, b5, b6, b8은 레벨

박스이고,  $b_2$ ,  $b_7$ 은 펄스 박스이며,  $b_3$ ,  $b_4$ 는 게이트 박스이다.

트랜지션은 박스(들)의 상태가 조합되어 박스 자신의 상태가 변화하거나 다른 박스(들)의 상태를 변화시키는 곳, 즉 전이가 일어나는 곳이다. 그리고 트랜지션에서 전이가 일어나는 과정을 트랜지션이 점화(fire)한다고 한다.

트랜지션에는 두 종류[9,10]가 있다. 첫째, 일반 트랜지션(general transition)은 점화시간을 요하지 않거나 특히 시간에 무관하게 점화하는 경우로서 막대(bar)로 표기한다. 둘째, 시간 트랜지션(time transition)은 점화시간을 요하는 트랜지션으로서 막대 위 또는 아래에 점화시간을 명기하여 표기한다. 그림 1에서  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ ,  $t_5$ ,  $t_7$ 은 일반 트랜지션이고,  $t_6$ 은 시간 트랜지션이다.

아크(arc)는 박스와 트랜지션 사이에서 트랜지션의 점화조건을 결정하고, 트랜지션이 점화될 때의 박스의 마크 상태를 결정한다.

아크에는 세 종류[9,10]가 있다. 첫째, 일반 아크(general arc)는 트랜지션의 입력 또는 출력으로 사용되며, 화살표로 표기한다. 둘째, 보존 아크(self arc)는 트랜지션의 입력으로만 사용되며, 물결 모양의 화살표로 표기한다. 셋째, 역 아크(inhibit arc)는 트랜지션의 입력으로만 사용되며, 선과 작은 원으로 된 아크로 표기한다. 그림 1에서  $a_{12}$ 는 보존 아크이고,  $a_8$ ,  $a_{10}$ ,  $a_{17}$ 은 역 아크이며, 그 외의 아크는 모두 일반 아크이다.

## 2.2 EMFG의 성질

EMFG는 Petri Net[1]에서 파생되었으므로 병렬성이나 동시성 등 Petri Net가 가지고 있는 성질 대부분을 그대로 가지고 있다. 따라서 Petri Net와 성질이 다른 부분만을 나타내기로 한다.

### 2.2.1 트랜지션의 점화조건

트랜지션의 점화조건은 트랜지션의 모든 입력 박스(들)의 마크 상태가 모두 조건에 부합되어야 한다. 이 조건들은 연결된 아크의 종류에 따라 달라진다[9,10,15]. 즉 일반 아크 또는 보존 아크로 연결된 입력 박스에는 마크가 있어야, 역 아크로 연결된 입력 박스에는 마크가 없어야 조건에 부합된다.

그리고 소스 트랜지션(source transition: 입력 박

스가 없는 트랜지션)은 점화조건을 부합시킬 어떤 입력 박스도 없으므로 항상 점화조건이 만족되어 있다. 그림 1에서 점화조건이 만족되어 있는 트랜지션은  $t_1$ 과  $t_2$  및  $t_4$  뿐이다.

### 2.2.2 트랜지션의 점화 동작

트랜지션은 점화조건이 만족되면 즉시 점화를 개시하게 된다. 일반 트랜지션은 점화시간을 요하지 않지만, 시간 트랜지션은 점화시간을 요구한다. 그리고 시간 트랜지션은 점화 도중에 어떤 원인에 의해 점화조건이 만족되지 않으면 즉시 점화를 중지하게 된다.

점화조건이 만족된 트랜지션은 점화시간이 경과되면 입력 박스의 마크 상태가 먼저 변화하고, 이어서 출력 박스의 마크 상태가 변화하여 점화가 완료된다. 그리고 박스의 마크 상태는 연결된 아크의 종류에 따라 다르게 변화한다. 즉 일반 아크로 연결된 입력 박스의 마크는 소멸되고, 보존 아크 또는 역 아크로 연결된 입력 박스의 마크는 그대로 보존되며, 출력 박스에는 마크가 없을 경우에만 마크를 생성시킨다. 그러나 게이트 박스의 마크 상태는 점화와 관계없이 대응되는 소자의 기계적인 또는 수동적인 동작 등에 의해 결정되고, 펄스 박스의 마크는 생성되는 즉시 일반 트랜지션의 점화에 기여하지 못하면 소멸되며, 싱크 트랜지션(sink transition: 출력 박스가 없는 트랜지션)은 입력 박스의 마크 상태만 변화되어 점화가 완료된다. 그림 1에서  $t_4$ 의 점화가 완료되면,  $b_2$ 의 마크는 소멸되고  $b_3$ 의 마크는 그대로 유지되며,  $b_6$ 에는 마크가 생성된다.

그리고 EMFG는 점화동작에서 Petri Net와 다른 점들이 있다. 첫째, 트랜지션의 점화가 완료될 때, Petri Net는 출력 박스의 마크 유무에 관계없이 마크를 생성시키지만, EMFG는 출력 박스에 마크가 없을 때만 생성시킨다. 둘째, 충돌 박스(conflict box: 두 개 이상의 출력 아크를 가진 박스)[4-8]의 마크가 다수의 출력 트랜지션의 점화조건을 동시에 만족시키는 경우, Petri Net에서는 점화조건이 만족된 하나의 트랜지션만이 임의로 선택되어 점화되지만, EMFG에서는 모든 트랜지션이 동시에 점화된다.

이외에도 펄스 박스 및 게이트 박스에 관련된 성질들이 다르고, 보존 아크에 관련된 성질이 서로 다르다.

### 2.3 EMFG의 회로변환

개념적으로 설계된 EMFG[9,10]는 시스템이 동작하는 추상적인 개념을 도식화한 것이므로 시스템의 동작 분석이나 성능 향상 등에 이용되고 회로변환에는 사용되지 않는다. 그러나 상세하게 설계된 EMFG [9,10]는 시스템의 동작을 도식화한 것이므로 회로로 변환하여 주변장치를 붙이던 구현하고자 하는 시스템이 된다. 따라서 다음에 설명되는 회로변환 방법 [9,10]은 상세하게 설계된 EMFG에 관한 것이다.

펄스 박스는 그림 2(a)와 같이 하나의 단자로 대체시키고, 게이트 박스는 그림 2(b)와 같이 대응되는 스위치나 푸시 버튼으로 대체시키며, 비동기 시스템의 레벨 박스는 그림 2(c)와 같이 클럭이 없는 플립플롭(RS 플립플롭 또는 JK 플립플롭)으로 대체시키고, 동기 시스템의 레벨 박스는 그림 2(d)와 같이 클럭 펄스 단자(CP)가 있는 플립플롭으로 대체시킨다.

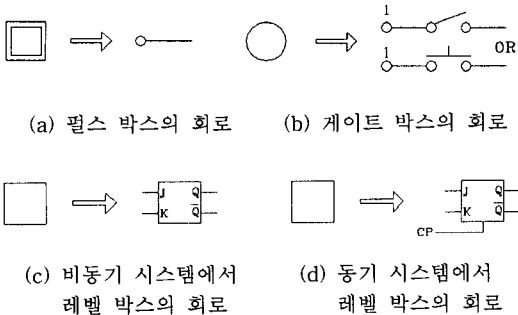


그림 2. 박스의 회로

비동기 시스템의 소스 트랜지션은 그림 3(a)와 같이 논리 레벨 1인 단자로 대체시키고, 동기 시스템의 소스 트랜지션은 그림 3(b)와 같이 클럭 펄스 단자로 대체시킨다. 일반 트랜지션은 그림 3(c)와 같이 AND 게이트로 대체시키며, AND 게이트의 입출력은 트랜지션에 연결된 입출력 아크에 대응된다. 시간 트랜지션은 그림 3(d)와 같이 점화조건이 만족되었을 때, 타이머가 동작하여 점화시간이 경과한 후 출력이 1이 되는 회로를 포함시킨다.

펄스 박스나 게이트 박스에서 트랜지션으로 연결된 일반 아크는 그림 4(a)와 같이 박스에 대응된 소자 출력에서 트랜지션에 대응된 AND 게이트 입력으로 연결시킨다. 레벨 박스에서 트랜지션으로 연결된 일반 아크는 그림 4(b)와 같이 박스에 대응된 플

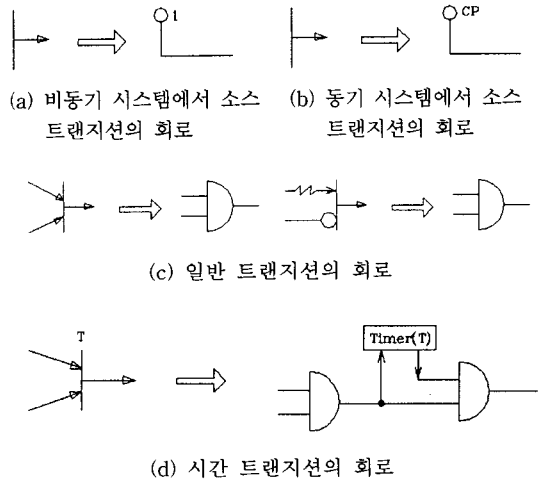


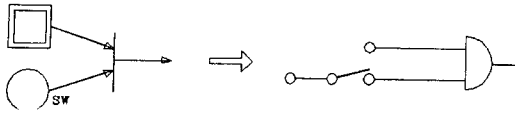
그림 3. 트랜지션의 회로

립플롭의 정상출력에서 트랜지션에 대응된 AND 게이트 입력으로 연결시킴과 동시에 AND 게이트의 출력에서 플립플롭의 R단자(또는 K단자)로 연결시킨다. 물론 R단자(또는 K단자)로 다수가 연결될 때는 OR 게이트를 사용한다. 레벨 박스에서 트랜지션으로 연결된 보존 아크는 그림 4(c)와 같이 입력 박스에 대응되는 소자 출력에서 트랜지션에 대응되는 AND 게이트 입력으로 연결시킨다. 펄스 박스나 게이트 박스에서 트랜지션으로 연결된 역 아크는 그림 4(d)와 같이 대응되는 소자 출력에서 NOT 게이트를 거쳐 트랜지션에 대응되는 AND 게이트 입력으로 연결시키고, 레벨 박스에서 트랜지션으로 연결된 역 아크는 그림 4(e)와 같이 레벨 박스에 대응되는 플립플롭의 보수출력에서 트랜지션에 대응되는 AND 게이트 입력으로 연결시킨다.

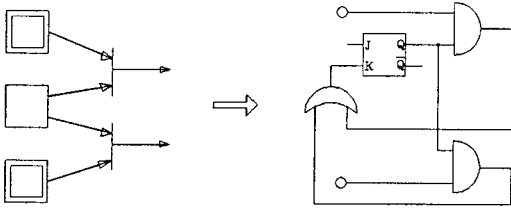
트랜지션에서 펄스 박스로 연결된 일반 아크는 그림 5와 같이 트랜지션에 대응되는 AND 게이트 출력에서 박스에 대응되는 단자로 연결시키고, 트랜지션에서 레벨 박스로 연결된 일반 아크는 그림 5와 같이 트랜지션에 대응되는 AND 게이트 출력에서 박스에 대응되는 플립플롭의 S단자(또는 J단자)로 연결시킨다. 물론 S단자(또는 J단자)로 다수가 연결될 때는 OR 게이트를 사용한다.

### 3. EMFG의 재정의 및 그 성질

EMFG는 이산 시스템을 상세하게 설계하여 회로



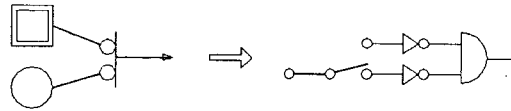
(a) 입력 펄스 박스나 입력 게이트 박스에서 트랜지션으로 연결된 일반 아크의 회로



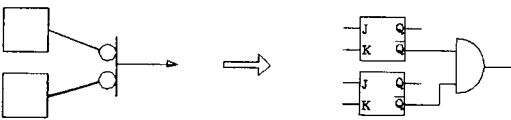
(b) 입력 레벨 박스에서 트랜지션으로 연결된 일반 아크의 회로



(c) 입력 레벨 박스에서 트랜지션으로 연결된 보존 아크의 회로



(d) 입력 펄스 박스나 입력 게이트 박스에서 트랜지션으로 연결된 역 아크의 회로



(e) 입력 레벨 박스에서 트랜지션으로 연결된 역 아크의 회로

그림 4. 트랜지션의 입력 아크의 회로

로 구현하기 용이하다. 그러나 기존의 EMFG에서는 박스 및 아크가 여러 가지 형태로 표현되어 기호화하기 어려울 뿐만 아니라 상호간의 관계 표현이 어려워 수학적 표현이 불가능하다. 이로 인해 EMFG의 각 마크들의 상태 변화를 수동적으로 파악해야 할뿐만 아니라 컴퓨터 시뮬레이션이 어려워져 시스템의 분석과 평가 및 설계에 많은 시간이 소요되고 어려움이

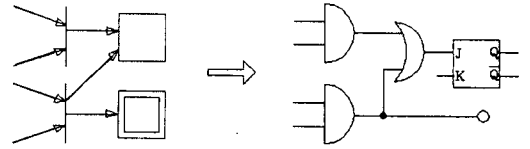


그림 5. 트랜지션의 출력 아크의 회로

따른다. 따라서 본 연구에서는 수학적 해석이 가능하도록 EMFG를 재정의한다.

### 3.1 EMFG의 재정의 및 표현

우선 EMFG부터 재정의하기로 한다.

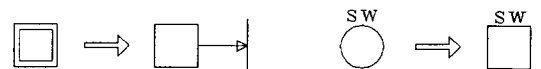
정의 1) EMFG G에서 박스의 집합을 B, 트랜지션의 집합을 T, 트랜지션에 대한 입력함수를 I, 트랜지션에 대한 출력함수를 O, 각 박스의 마크 상태를 나타내는 마크 벡터를 **M**이라 할 때, EMFG G는 다음 5개의 순서쌍으로 정의되는 선도[24]이다.

$$G = (B, T, I, O, M)$$

다음은 박스의 성질 및 간소화 표현에 대하여 설명하고, 박스를 기호화하여 수학적으로 표현한다.

펄스 박스에 마크가 생성되었을 때, 자신에 연결된 어떤 트랜지션도 점화되지 못하면 마크는 자동 소멸된다. 그리고 레벨 박스와 자신에만 연결된 싱크 트랜지션이 있는 경우, 이 레벨 박스에 마크가 생성되었을 때도 자신에 연결된 다른 트랜지션이 점화되지 못하면 싱크 트랜지션이 점화되어 마크는 자동 소멸된다. 따라서 펄스 박스는 그림 6(a)와 같이 레벨 박스와 자신에만 연결된 싱크 트랜지션으로 대체시킬 수 있다.

그리고 게이트 박스를 레벨 박스로 대체하는 경우, 트랜지션의 점화조건 만족에는 영향을 주지 않지만, 트랜지션이 점화할 때의 마크 상태는 달라진다. 그러나 게이트 역할을 하는 레벨 박스는 별도 표기하여 관리하고, 회로로 변환할 때에도 달리 변환한다면 문제는 발생하지 않는다. 따라서 게이트 박스는 그림 6(b)와 같이 레벨 박스로 대체시킬 수 있다.



(a) 펄스 박스의 대체 (b) 게이트 박스의 대체

그림 6. 펄스 박스와 게이트 박스의 대체

결국 기존의 EMFG의 펄스 박스, 레벨 박스 및 게이트 박스는 하나의 레벨 박스로 표현될 수 있으므로 기호화할 수 있다. 그리고 한 종류의 레벨 박스만 있으므로 레벨 박스를 박스라 칭하기로 한다.

정의 2) EMFG G에서 B는 다음과 같이 정의되는 박스의 유한집합이다.

$$B = \{b_i \mid 1 \leq i \leq n\}, \text{ 단 } n \text{은 양의 정수}$$

여기서  $b_i$ 는 EMFG에 있는 임의의 박스이며, 실선의 사각형으로 표기한다.

다음에는 트랜지션을 수학적으로 표현한다. 트랜지션은 간소화되지 않으므로 단지 트랜지션의 집합에 대해서만 정의한다.

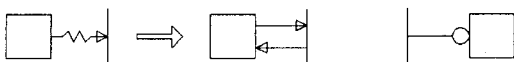
정의 3) EMFG G에서 T는 다음과 같이 정의되는 트랜지션의 유한집합이다.

$$T = \{t_i, t_j(s) \mid 1 \leq i \leq m, 1 \leq j \leq n, s \text{는 시간}\}, \text{ 단 } m, n \text{은 양의 정수}$$

여기서  $t_i$ 는 EMFG에서 시간이 명기되지 않은 임의의 일반 트랜지션이고,  $t_j(s)$ 는 EMFG에서 시간  $s$ 가 명기된 임의의 시간 트랜지션이며, 막대로 표기한다.

다음은 아크의 성질 및 간소화 표현에 대하여 설명하고, 출력 역 아크를 도입하여 EMFG의 수학적 표현이 가능하도록 한다.

기존의 EMFG에서 보존 아크에 연결된 트랜지션의 점화가 완료되면 보존 아크에 연결된 입력 박스의 마크는 변화하지 않는다. 이 동작은 보존 아크를 입력의 일반 아크와 출력의 일반 아크로 대치하여 동작시킨 결과와 동일하다. 따라서 보존 아크는 그림 7(a)과 같이 대치할 수 있다.



(a) 보존 아크의 대치 (b) 출력 역 아크

그림 7. 보존 아크의 대치 및 출력 역 아크

결국 기존의 EMFG의 아크들은 일반 아크와 역 아크의 두 종류로 표현될 수 있다.

그러나 두 종류의 아크는 상호간의 역할이 명백하지 않다. 즉 일반/역 아크로 연결된 트랜지션의 입력 박스에는 마크가 있/없어야 점화조건을 만족시키고, 점화가 완료되면 입력 박스의 마크를 소멸/유지시키므로 서로 상반되는 역할을 하지만, 트랜지션에서 박스로 연결된 역 아크는 존재하지 않으므로 서로 상반되는 역할을 하지 않는다.

뿐만 아니라 트랜지션이 점화될 때 점화조건에 관련되지 않는 어떤 박스에 마크를 생성시킬 수 있지만 소멸시킬 수 없다. 이러한 현상은 실제 시스템에서 충분히 발생 가능하나 기존의 EMFG에서는 표현할 방법이 전혀 없다.

따라서 그림 7(b)와 같이 트랜지션의 출력 역 아크를 사용할 수 있도록 한다면, 위 두 가지 문제점은 해결된다.

정의 4) 트랜지션에서 박스로 역 아크를 연결하여 사용할 수 있도록 하며, 이 역 아크를 트랜지션의 출력 역 아크라 정의한다.

트랜지션의 출력 역 아크가 새로이 도입되었으므로 트랜지션의 점화조건이 만족되었을 때의 점화 과정을 다시 정의할 필요가 있다.

정의 5) 일반 트랜지션의 점화조건이 만족되었을 때(시간 트랜지션은 점화조건이 만족된 후, 점화시간이 경과되었을 때), 트랜지션의 점화가 완료되는 과정을 다음과 같이 정의한다:

- (1) 트랜지션에 일반/역 아크로 연결된 입력 박스의 마크는 소멸/유지된다.
- (2) 트랜지션에서 일반/역 아크로 연결된 출력 박스에 마크가 없/있으면 생성/소멸시키고, 마크가 있/없으면 그대로 유지시킨다.

일반 아크와 역 아크는 모두 트랜지션의 입력 아크 또는 출력 아크로 사용될 수 있으며, 트랜지션의 점화조건을 만족시킬 때에도 두 아크는 서로 상반적인 역할을 하고, 트랜지션이 점화하는 과정에서도 정의 5와 같이 두 아크는 서로 상반적인 역할을 한다. 따라서 일반 아크와 역 아크는 개념 및 동작에 있어서 항상 서로 상반되는 역할을 함으로써 EMFG의 설계 및 해석이 용이해진다.

그리고 기존의 EMFG에서는 표현 불가능한 부분

도 트랜지션의 출력 역 아크를 사용한다면 표현 가능하다. 예를 들어 트랜지션  $t_4$ 가 점화하였을 때  $t_4$ 에 연결되어 있지 않은 어떤 박스  $b$ 의 마크를 소멸시키고자 하는 경우, 기존의 EMFG에서는 표현이 불가능하다. 그러나 재정의된 EMFG의 경우,  $t_4$ 에서  $b$ 로 역 아크를 연결시킨다면 이 동작을 행할 수 있으므로 표현이 가능해진다.

이제 EMFG를 구성하는 원소들이 모두 재정의되었다. 재정의된 EMFG를 사용하여 그림 1을 그림 8과 같이 다시 표현하였다.

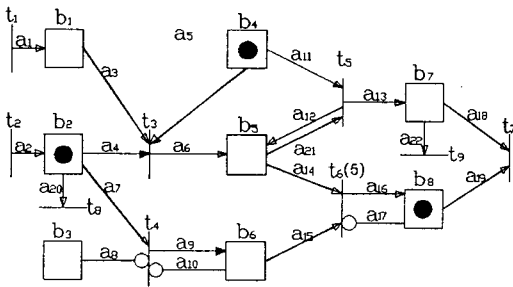


그림 8. 재정의한 EMFG로 표현한 그림 1의 EMFG

그림 8에서 박스의 집합 B와 트랜지션의 집합 T를 구하면 다음과 같다.

$$B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\} \quad (1)$$

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6(5), t_7, t_8, t_9\} \quad (2)$$

식 (1)과 (2)와 같이 박스와 트랜지션을 기호화하고 수학적으로 표현할 수 있다. 그러나 박스와 아크 및 트랜지션의 관계에 있어서는 아직 미흡하므로 이들 관계를 수학적으로 표현하기로 한다.

기존의 EMFG에서는 박스와 아크 및 트랜지션의 관계를 EMFG 상에서만 확인해야 하므로 트랜지션의 점화조건을 검사하거나 점화가 완료된 후의 마크 상태를 확인할 때 어려운 점이 많다. 따라서 서로의 관계를 트랜지션의 입출력함수를 도입하여 해결하기로 한다.

정의 6) EMFG G에서 트랜지션의 집합 T의 어떤 원소  $t$ 에 대한 입력함수  $I(t)$ 는 박스의 집합을 B라 할 때, 다음과 같이 정의한다:

$$I(t) = \{b_i, b_j' \mid b_i \in B, b_j \in B, b_i/b_j \text{는 } t \text{에 일반/역 아크로 연결된 출력 박스}\}$$

아크로 연결된 입력 박스}

정의 7) EMFG G에서 트랜지션의 집합 T의 어떤 원소  $t$ 에 대한 출력함수  $O(t)$ 는 박스의 집합을 B라 할 때, 다음과 같이 정의한다:

$$O(t) = \{b_i, b_j' \mid b_i \in B, b_j \in B, b_i/b_j \text{는 } t \text{에 일반/역 아크로 연결된 출력 박스}\}$$

그림 8에서 각 트랜지션의 입력함수와 출력함수를 구하면 다음과 같다.

$I(t_1) = \emptyset$	$O(t_1) = \{b_1\}$
$I(t_2) = \emptyset$	$O(t_2) = \{b_2\}$
$I(t_3) = \{b_1, b_2, b_4\}$	$O(t_3) = \{b_5\}$
$I(t_4) = \{b_2, b_3', b_6'\}$	$O(t_4) = \{b_6\}$
$I(t_5) = \{b_4, b_5\}$	$O(t_5) = \{b_5, b_7\}$
$I(t_6(5)) = \{b_5, b_6, b_8'\}$	$O(t_6(5)) = \{b_8\}$
$I(t_7) = \{b_7, b_8\}$	$O(t_7) = \emptyset$
$I(t_8) = \{b_2\}$	$O(t_8) = \emptyset$
$I(t_9) = \{b_7\}$	$O(t_9) = \emptyset$

표현된 바와 같이 트랜지션의 입출력함수는 트랜지션에 입력 또는 출력으로 연결된 아크 및 박스를 판별할 수 있으므로 상호간의 관계를 표현한 식이 된다. 따라서 트랜지션의 입출력함수로써 EMFG로 도식화할 수 있으며, 또한 EMFG 상에서 각 트랜지션의 입출력함수를 구할 수 있다.

그런데 박스들의 마크 상태, 즉 시스템의 동작 상태는 수학적으로 표현되지 않았다. 박스들의 마크 상태는 EMFG의 각 박스들과 대응되고 마크 수는 0이거나 1이므로 박스들의 마크 상태를 벡터로 표현할 수 있다.

정의 8) EMFG G에서 마크 벡터(mark vector)  $\mathbf{M}$ 은 다음과 같이 정의되는 열벡터 또는 행벡터이다.

$$\mathbf{M} = (\mu_1, \mu_2, \dots, \mu_i, \dots, \mu_n)$$

여기서  $\mu_i$ 는 박스  $b_i$ 의 마크 수이며,  $n$ 은 EMFG내에 존재하는 박스의 수이다.

예를 들어 그림 8에서 마크 벡터  $\mathbf{M}$ 은 (0, 1, 0, 1, 0, 0, 0, 1)이다.

이제 EMFG의 모든 원소들이 기호화되고 수학적

으로 표현되었다. 그리고 박스와 아크 및 트랜지션의 관계도 수학적인 함수로 표현되었다.

따라서 이들을 이용하면 EMFG 동작, 즉 시스템 동작을 수학적으로 해석할 수 있다. 다시 말해 마크 벡터와 트랜지션의 입력함수로서 트랜지션의 점화 조건을 검사할 수 있으며, 트랜지션의 점화가 완료한 후의 마크 상태는 마크 벡터와 트랜지션의 입출력 함수를 사용하여 구할 수 있게 되었으므로 이산 시스템의 설계 및 분석이 용이해진다.

### 3.2 재정의된 EMFG의 회로변환

EMFG가 재정의되었으므로 회로변환 방법도 달라져야 한다. 그러나 기존의 EMFG의 성질을 대부분 그대로 가지고 있으므로 회로변환도 많은 부분에서 동일하다. 재정의된 EMFG의 회로변환 방법은 다음과 같다.

EMFG의 박스는 속성에 따라 두 부류로 나눌 수 있다. 한 부류는 게이트 역할을 하는 소스박스(입력 아크를 가지지 않는 박스)로서 스위치나 푸시 버튼 등의 기계적이거나 수동적인 동작을 표현하고, 다른 한 부류는 게이트 역할이 아닌 일반적인 박스로서 시스템의 제어상태나 제어신호 등을 표현한다.

그러므로 EMFG에서 게이트 역할을 하는 소스박스는 EMFG 상에서 그 역할이 표기되어 있으므로 표기된 해당 소자로 변환된다. 예를 들어 게이트 역할을 하는 소스박스가 스위치로 표기되었다면, 스위치로 변환되어야 한다. 즉 소스박스에 마크가 존재하면 스위치가 닫혀 있는 상태이고, 존재하지 않으면 열려 있는 상태이다.

그리고 게이트 역할을 하지 않는 박스는 자신에 연결된 트랜지션(들)의 점화에 의해 마크가 생성되거나 소멸되거나 또는 그대로 유지될 것이다. 따라서 플립플롭은 이와 같은 기능을 수행할 수 있으므로 게이트 역할을 하지 않는 박스는 플립플롭으로 변환될 수 있다.

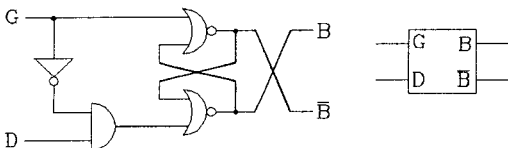


그림 9. GD 플립플롭의 회로도 및 블록도

정리 1) EMFG에서 게이트 역할을 하지 않는 박스는 그림 9와 같은 GD 플립플롭으로 변환될 수 있다. 단 EMFG가 동기 시스템을 표현한 경우는 클럭이 있는 GD 플립플롭을 사용하여야 한다.

증명) 게이트 역할을 하지 않는 박스를 b라 하자. 그리고 b에 마크를 생성시키는 경우를 g, b의 마크를 소멸시키는 경우를 d, b의 현재의 마크 상태를 b(t), b에 연결된 트랜지션(들)이 점화하여 변환된 b의 다음 마크 상태를 b(t+1)이라고 한다면, 이들 사이의 관계는 표 1과 같은 특성표로 표현된다.

표 1. 게이트 역할을 하지 않는 박스의 특성표

g	d	b(t)	b(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

b(t+1)을 g와 d 및 b(t)로 표현한 식, 즉 b의 특성 방정식을 표 1에서 카르노 맵을 사용하여 구하면 다음과 같다.

$$b(t+1) = g + b(t) * d' \quad (3)$$

그리고 그림 9의 GD 플립플롭에서 B의 현재상태를 B(t), B의 다음상태를 B(t+1)이라 하는 경우, B의 특성표는 표 1과 같은 모양이 되며, B(t+1)을 G와 D 및 B(t)로 표현한 식, 즉 B의 특성방정식을 구하면 다음과 같다.

$$B(t+1) = G + B(t) * D' \quad (4)$$

식 (4)과 식 (3)을 비교하여 보면, G는 g에, D는 d에, B(t)는 b(t)에, B(t+1)은 b(t+1)에 각각 대응된다는 것을 알 수 있다. 즉 b의 특성은 B의 특성과 동일하므로 게이트 역할을 하지 않는 박스의 회로는 그림 9와 같은 GD 플립플롭이 된다. 그리고 동기 시스템은 클럭에 의하여 플립플롭의 상태가 바뀌므로 회로변환시 클럭이 있는 플립플롭이 사용되어야 한다. 따라서 정리는 타당하다. □



표 1에서 (g, d)가 (0, 0)인 경우는 마크를 유지하는 경우이며, (0, 1)인 경우는 마크를 소멸시키는 경우이며, (1, 0)인 경우는 마크를 생성시키는 경우이며, (1, 1)인 경우는 박스의 현재 마크 상태에 따라 두 가지 경우를 생각할 수 있다. 첫째, 마크가 존재하지 않는 경우, 마크를 생성시키는 기능과 소멸시키는 기능은 동시에 동작하는데, 소멸시킬 마크가 존재하지 않으므로 소멸 기능은 작용하지 않고, 마크를 생성시키는 기능만이 동작하여 결국 박스에는 마크가 생성된다. 둘째, 마크가 존재하는 경우, 정의 5에 의하여 마크를 소멸시키는 기능이 먼저 동작하여 마크가 소멸된 후에 마크를 생성시키는 기능이 동작하여 결국 마크가 생성된다.

그리고 그림 9의 G는 출력을 1로 하는(박스의 마크를 생성시키는(Generate)) 역할을 하고, D는 출력을 0으로 하는(박스의 마크를 소멸시키는(Degenerate)) 역할을 한다.

그런데 기존의 EMFG에서는 박스를 JK 플립플롭이나 RS 플립플롭으로 변환하였기 때문에 문제가 발생하는 경우가 있다.

예를 들어 그림 10(a)와 같은 EMFG에서 트랜지션  $t_1$ 과  $t_2$ 가 동시에 점화조건이 만족되면 동시에 점화한다. 두 트랜지션의 점화가 완료되면, 정의 5에 의하여 두 트랜지션의 입력 박스의 마크 상태가 먼저 변화하고, 다음에 두 트랜지션의 출력 박스의 마크 상태가 변화한다. 따라서 점화가 완료된 후의 마크 상태는 (0, 1, 1)에서 (0, 0, 1)로 변한다. 그러나 그림 10(b)의 변환된 회로에서는 FF3의 J와 K에 동시에 1이 가해져 FF3의 출력을 0으로 만들어  $b_3$ 의 마크가 0인 효과를 가져오게 되므로 잘못된 회로변환이 된다. RS 플립플롭을 사용하는 경우에도 플립플롭의 출력은 미정인 상태가 되어 대응되는 박스의 마크 상태를 판단할 수 없으므로 잘못된 회로변환이 된다.

따라서 정리 1과 같이 박스를 GD 플립플롭으로

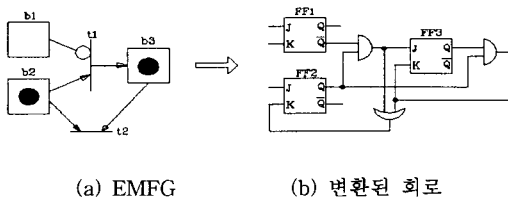


그림 10. 회로변환이 잘못되는 경우 1

변환하면 이 문제는 해결된다.

다음은 트랜지션의 회로변환에 대하여 알아본다.

소스 트랜지션은 입력 박스를 가지지 않으므로 항상 점화조건이 만족된다. 따라서 소스 트랜지션은 출력박스의 마크 상태가 항상 변화되도록 변환되어야 하므로 논리 레벨 1을 가진 단자로 변환되어야 한다.

그리고 소스 트랜지션이 아닌 트랜지션은 점화조건이 만족되어야 점화 가능하고, 이 트랜지션에 연결된 박스(들)의 마크를 생성시키거나 소멸시켜 마크 상태가 변화한다. 그러므로 소스 트랜지션이 아닌 트랜지션을 회로변환할 때는 트랜지션의 점화조건에 부합되도록 하여야 한다.

트랜지션의 점화조건은 트랜지션에 연결된 모든 입력 박스(들)의 마크 상태가 동시에 만족되어야 만족된다. 이는 논리 AND의 의미와 동일하므로 트랜지션은 AND 게이트로 변환될 수 있다. 이 때, 트랜지션에 연결된 입력 박스의 마크 상태는 변환된 AND 게이트의 입력에 대응되므로, 입력 박스에서 일반/역 아크로 연결된 경우에는 입력 박스가 변환된 소자의 정상/보수 출력이 AND 게이트의 입력이 된다.

그런데 트랜지션이 일반 트랜지션이면 점화를 완료하는데 점화시간을 요하지 않으므로 이와 같이 변환하면 충분하다. 그러나 트랜지션이 시간 트랜지션인 경우는 점화조건이 만족된 후 점화시간이 경과하여야 점화가 완료된다. 만약 점화 중에 점화조건이 만족되지 않으면 점화가 중지될 것이다. 그러므로 시간 트랜지션은 이와 같이 AND 게이트로 변환한 후, AND 게이트의 출력이 타이머를 동작시키는 입력이 되도록 하고, 그림 3(d)와 같이 이 AND 게이트의 출력과 타이머의 출력을 입력으로 하는 새로운 AND 게이트로 구성하면 된다.

표 2. GD 플립플롭의 여기표

B(t)	B(t+1)	G	D
0	0	0	x
0	1	1	x
1	0	0	1
1	1	x	0
		1	x
0	-	x	x
1	-	x	x

다음은 박스 b가 플립플롭 B로 변환되었을 때, 플립플롭의 입력 G 및 D를 구하는 방법에 대하여 알아 본다.

b의 상태 변화는 B의 상태 변화와 동일하다. 따라서 B의 현재상태 B(t)가 다음상태 B(t+1)로 변하기 위한 플립플롭의 입력조건 즉 여기표를 표 2와 같이 표현할 수 있다. 표 2에서 x는 무관(don't care)항이며, - 표시는 현재상태에서 다음상태로 전이할 수 없는 경우를 나타낸다.

정리 2) EMFG에서 일반/역 아크로 연결된 입력 트랜지션 m/n 개와 일반/역 아크로 연결된 출력 트랜지션 p/q 개를 가지는 박스 b<sub>2</sub>를 플립플롭 B<sub>2</sub>로 변환하는 경우, 일반/역 아크로 연결된 임의의 입력 트랜지션 t<sub>i</sub>/t<sub>j</sub>가 T<sub>i</sub>/T<sub>j</sub>로 변환되고, 일반/역 아크로 연결된 임의의 출력 트랜지션 t<sub>x</sub>/t<sub>y</sub>가 T<sub>x</sub>/T<sub>y</sub>로 변환되었다고 하였을 때, 플립플롭의 입력 G<sub>2</sub>와 D<sub>2</sub>는 다음과 같다.

$$G_z = \sum_{i=1}^m T_i \quad (5)$$

$$D_z = \sum_{x=1}^p T_x + \sum_{j=1}^n T_j \quad (6)$$

증명) EMFG의 동작과 변환된 회로의 동작은 일치해야 하므로 변환된 회로의 해석은 EMFG의 동작으로 해석될 수 있다.

즉 t<sub>i</sub>, t<sub>j</sub>, t<sub>x</sub>, t<sub>y</sub>의 점화조건을 각각 F<sub>i</sub>, F<sub>j</sub>, F<sub>x</sub>, F<sub>y</sub>라 하면, 이들은 변환된 회로에서 각각 T<sub>i</sub>, T<sub>j</sub>, T<sub>x</sub>, T<sub>y</sub>에 대응된다. 박스에 일반 아크로 연결된 입력 트랜지션 중 하나 이상이 점화 가능하면 박스에는 마크가 존재하게 될 것이다. 이는 박스에 일반 아크로 연결된 모든 입력 트랜지션의 점화조건을 각각 논리 OR한 결과(ΣF<sub>i</sub>로 표현)가 되며, 변환된 회로에서는 T<sub>i</sub>를 논리 OR한 결과(ΣT<sub>i</sub>로 표현)가 된다. 다른 트랜지션들에 대해서도 마찬가지로 방법으로 해석된다.

그러면 ΣF<sub>i</sub>, ΣF<sub>j</sub>, ΣF<sub>x</sub>, ΣF<sub>y</sub> 들의 상태에 따라 b<sub>2</sub>의 현재상태가 다음상태로 변화하기 위한 조건, 즉 ΣT<sub>i</sub>, ΣT<sub>j</sub>, ΣT<sub>x</sub>, ΣT<sub>y</sub> 들의 상태에 따라 B<sub>2</sub>의 현재상태가 다음상태로 변화하기 위한 플립플롭의 입력조건을 구해보자.

그런데 표 2에서 현재상태 1에서 다음상태 1로 가기 위한 (G<sub>2</sub>, D<sub>2</sub>)의 값이 (x, 0) 또는 (1, x)의 두 가지가 있기 때문에 플립플롭의 여기표를 포함하는 상태

변화표는 32가지가 있을 수 있다. 이 중에서 가장 간단한 표를 표 3에 나타내었다.

표 3에서 G<sub>2</sub>와 D<sub>2</sub>를 카르노 맵을 사용하여 구하면 다음과 같다.

$$G_z = \sum T_i \quad (7)$$

$$D_z = \sum T_x + \sum T_j \quad (8)$$

표 3. 박스를 표현한 플립플롭의 상태 여기표

현재 상태					다음상태	G <sub>z</sub>	D <sub>z</sub>
ΣT <sub>i</sub>	ΣT <sub>j</sub>	ΣT <sub>x</sub>	ΣT <sub>y</sub>	B <sub>2</sub> (t)	B <sub>2</sub> (t+1)		
0	0	0	0	0	0	0	x
0	0	0	0	1	1	x	0
0	0	0	1	0	0	0	x
0	0	0	1	1	-	x	x
0	0	1	0	0	-	x	x
0	0	1	0	1	0	0	1
0	0	1	1	x	-	x	x
0	1	0	0	0	0	0	x
0	1	0	0	1	0	0	1
0	1	0	1	0	0	0	x
0	1	0	1	1	-	x	x
0	1	1	0	0	-	x	x
0	1	1	0	1	0	0	1
0	1	1	1	x	-	x	x
1	0	0	0	0	1	1	x
1	0	0	0	1	1	1	x
1	0	0	1	0	1	1	x
1	0	0	1	1	-	x	x
1	0	1	0	0	-	x	x
1	0	1	0	1	1	1	x
1	0	1	1	x	-	x	x
1	1	0	0	0	1	1	x
1	1	0	0	1	1	1	x
1	1	0	1	0	1	1	x
1	1	0	1	1	-	x	x
1	1	1	0	0	-	x	x
1	1	1	0	1	1	1	x
1	1	1	1	x	-	x	x

식 (7)과 (8)는 식 (5)과 (6)의 다른 표현이므로 정리는 타당하다. □

그리고 기존의 EMFG 회로변환에서 잘못 변환되는 경우가 또 있다. 그림 11과 같은 EMFG의 경우, 박스 b<sub>2</sub>의 마크는 그대로 유지되어야 하지만, 변환된 회로에서는 마크가 소멸되는 상태가 된다. 재정의된 EMFG에서는 보존 아크가 임출력 아크로 대체되었으므로 이와 같은 경우도 정리 2를 사용하여 회로로 변환된다면 전혀 문제없이 변환된다.

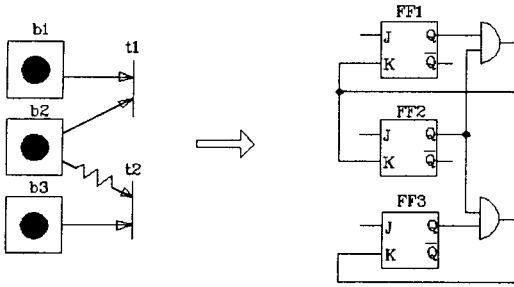


그림 11. 회로변환이 잘못되는 경우 2

정리 3) 박스 b가 자신에만 연결된 싱크 트랜지션  $t_k$ 를 가지고 있을 때, b와  $t_k$ 를 회로변환한 P는 다음과 같다.

$$P = \sum_{i=1}^m T_i \quad (9)$$

여기서  $T_i$ 는 b에 일반 아크로 연결된 임의의 입력 트랜지션이 변환된 소자이며, m은 그 수이다.

증명) 박스 b가 자신에만 연결된 싱크 트랜지션을 가지고 있을 때, 마크를 생성시키는 경우를 g, 마크를 소멸시키는 경우를 d, b의 현재의 마크 상태를  $b(t)$ , b의 다음 마크 상태를  $b(t+1)$ 로 나타내면, 표 4와 같다.

표 4. 자신에만 연결된 싱크 트랜지션을 가지는 박스의 특성표

g	d	b(t)	b(t+1)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

표 4에서 b의 특성방정식  $b(t+1)$ 을 카르노 맵을 사용하여 구하면 다음과 같다.

$$b(t+1) = g \quad (10)$$

식 (10)은 P의 특성방정식과 동일한 의미를 가지게 되고, g는 b에 일반 아크로 연결된 모든 입력 트랜

지션이 변환된 소자들을 논리 OR한 결과, 즉 식 (9)의 우변과 같은 의미를 가지므로 정리는 타당하다. □

이제 재정의된 EMFG의 회로변환 방법에 대하여 모두 설명되었으므로, 그림 8의 EMFG를 회로로 변환해 보자.

그림 8에서  $b_3$ 가 스위치이고  $b_4$ 가 푸시 버튼이라고 할 때, 각 정리들을 적용하여 회로변환 수식들을 구하면 다음과 같다. 여기서  $(B_5B_6B_8')_5$ 의 표기는  $B_5B_6B_8'$ 의 AND 게이트 출력에 시간 5에서 동작하는 타이머를 부가한 표시이다.

$$\begin{aligned}
 T_1 &= 1 & T_2 &= 1 & T_3 &= B_1B_2B_4 \\
 T_4 &= B_2B_3'B_6' & T_5 &= B_4B_5 \\
 T_6(5) &= (B_5B_6B_8')_5 \\
 T_7 &= B_7B_8 & T_8 &= B_2 & T_9 &= B_7 \\
 G_1 &= T_1 & D_1 &= T_3 \\
 G_2 &= T_2 & D_2 &= T_3 + T_4 + T_8 \\
 B_3 &= \text{스위치 출력} & B_4 &= \text{푸시 버튼의 출력} \\
 G_5 &= T_3 + T_5 & D_5 &= T_5 + T_6(5) \\
 G_6 &= T_4 & D_6 &= T_6(5) \\
 G_7 &= T_5 & D_7 &= T_7 + T_9 \\
 G_8 &= T_6(5) & D_8 &= T_7
 \end{aligned}$$

위의 수식들을 사용하여 실제 회로로 그린 것이 그림 12이다.

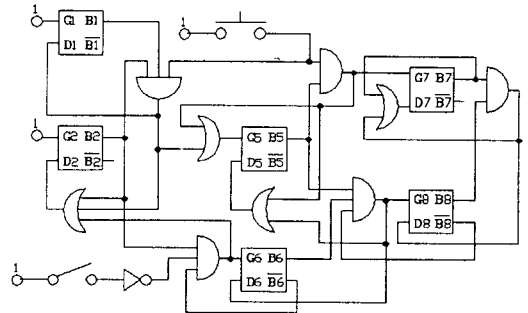


그림 12. 그림 8에 대응되는 회로

따라서 이산 시스템을 재정의된 EMFG로 상세하게 설계한 경우, 설계된 EMFG를 쉽게 수학적으로 표현할 수 있을 뿐만 아니라 이를 이용하여 용이하게 회로로 변환할 수 있으므로 이산 시스템의 설계와 구현 및 분석을 쉽게 해준다.

### 4. 기존의 EMFG와 재정의된 EMFG와의 비교 및 설계 예

#### 4.1 기존의 EMFG와 재정의된 EMFG와의 비교

표 5는 기존의 EMFG와 재정의된 EMFG를 비교한 표이다.

표 5에서 보면 재정의된 EMFG가 기존의 EMFG에 비해 여러 부분에서 개선되었다는 것을 알 수 있다. 그리고 재정의된 EMFG를 사용하는 것이 기존의 EMFG를 사용하는 것보다 시스템의 설계와 구현 및 분석에 유리할 뿐만 아니라, 컴퓨터 프로그램을 사용할 수 있으므로 시스템 설계 및 분석의 CAD화도 가능하다.

#### 4.2 재정의된 EMFG를 사용한 설계 예

입력 x에서 0101의 신호열이 감지되면 출력 y가

1이 되는 0101 감지기 회로[25]를 설계한다. 이 회로는 비교적 간단한 시스템이므로 개념적인 설계는 생략하고, 직접 재정의된 EMFG를 사용하여 상세하게 설계하면, 그림 13과 같은 EMFG가 된다.

그림 13을 보면 시스템의 동작을 쉽게 파악할 수 있다. 즉 초기에 박스 b1에 마크가 주어지면 시스템

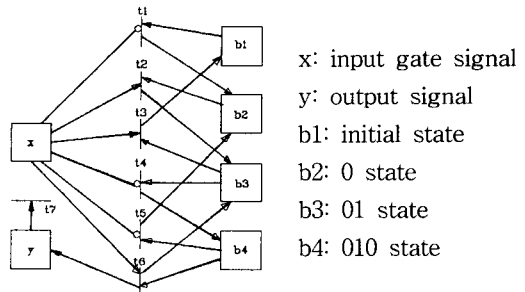


그림 13. 0101 감지기 회로의 EMFG

표 5. 기존의 EMFG와 재정의된 EMFG를 비교한 표

번호	비 교 항 목	기존의 EMFG	재정의된 EMFG	재정의된 EMFG가 기존의 EMFG보다 개선된 점
1	박스	레벨 박스, 펄스 박스, 게이트 박스	박스	간소화
2	아크	일반 아크, 보존 아크, 역 아크	일반 아크, 역 아크	간소화
3	EMFG의 수학적 표현	불가능	가능	시스템의 설계 및 분석이 용이하고, 프로그램에 의한 분석 가능
4	점화조건 검사 및 박스의 마크 상태 변화추이	EMFG 상에서 가능	EMFG 상에서 가능, 표현된 수식에 의해서도 가능	프로그램에 의한 검사 가능
5	트랜지션의 출력 역 아크	없음	있음	도입
6	일반 아크와 역 아크와의 관계	없음	상호 상반적인 관계	시스템 설계용이
7	점화조건에 관련되지 않은 박스의 마크 소멸 가능성	없음	있음	표현 불가능한 경우를 표현 가능하게 함
8	회로변환 방법	일대일 변환	일대일 변환, 수식적인 변환	프로그램에 의한 변환 가능
9	회로변환시 사용 플립플롭	JK 플립플롭 또는 RS 플립플롭	GD 플립플롭	개선
10	EMFG가 회로로 잘못 변환되는 경우	존재함	해결	개선
11	시스템의 개념적인 설계	가능	가능	유사
12	시스템의 상세한 설계	가능	가능	시스템 설계용이

이 동작한다. 이때 입력신호 x가 0이면 0 상태인 b2로 마크가 이동하고, 1이면 초기상태 그대로 유지한다. 0 상태에서 x가 0이면 상태가 그대로 유지되며, 1이면 01 상태인 b3으로 마크가 이동한다. 01 상태에서 x가 0이면 010 상태인 b4로 마크가 이동하고, 1이면 초기상태인 b1로 마크가 이동한다. 010 상태에서 x가 0이면 0 상태인 b2로 마크가 이동하고, 1이면 출력을 1로 만들면서 01 상태인 b3으로 마크가 이동한다. 이와 같은 방법으로 시스템은 동작한다.

그런데 그림 13을 기존 방식(기존의 EMFG)으로 설계하는 경우에는 그림 13의 x가 게이트 박스로 표현되고, y가 펄스 박스로 표현되어 그림 13과 유사하고 그 동작도 동일하지만, 수학적 표현이 불가능하다. 그러나 본 방식(제정된 EMFG)으로 설계된 그림 13은 다음과 같이 수학적으로 표현된다. 여기서 초기에 마크는 b1에만 존재한다고 가정한다.

$$\begin{aligned}
 \text{EMFG } G &= (B, T, I, O, M) \\
 B &= \{x, y, b1, b2, b3, b4\} \\
 T &= \{t1, t2, t3, t4, t5, t6, t7\} \\
 I(t1) &= \{x', b1\} & O(t1) &= \{b2\} \\
 I(t2) &= \{x, b2\} & O(t2) &= \{b3\} \\
 I(t3) &= \{x, b3\} & O(t3) &= \{b1\} \\
 I(t4) &= \{x', b3\} & O(t4) &= \{b4\} \\
 I(t5) &= \{x', b4\} & O(t5) &= \{b2\} \\
 I(t6) &= \{x, b4\} & O(t6) &= \{y, b3\} \\
 I(t7) &= \{y\} & O(t7) &= \emptyset \\
 M &= (0, 0, 1, 0, 0, 0)
 \end{aligned}$$

그리고 각 트랜지션들의 점화조건은 트랜지션들의 입력함수의 원소들을 AND한 값이고, 박스들의 다음 마크 상태는 점화조건과 트랜지션들의 출력함수에 의해 결정되므로 프로그램에 의해서 계산될 수 있다. 따라서 펜티엄 컴퓨터 상에서 프로그램을 통하여 박스들의 마크들이 이동되는 과정, 즉 EMFG가 동작되는 과정을 확인하였다.

다음에는 설계된 EMFG를 회로로 변환하여 보자. 기존 방식으로 설계된 경우에는 EMFG 원소들을 일대일 변환을 사용하여 회로로 변환하여야 한다. 그러나 본 방식으로 설계된 경우에는 기존의 방법을 사용하여 회로로 변환할 수도 있지만, 표현된 수식과 변

환 정리들을 사용하여 변환할 수 있다. 즉 정리 1을 사용하여 박스들을 변환하고, 트랜지션들은 트랜지션의 입력함수들을 사용하여 변환하며, 변환된 플립플롭의 입력 G와 D는 정리 2와 트랜지션의 입출력함수를 사용하여 변환하면 된다. 다시 말해 트랜지션에 대응되는 AND 게이트의 입력들은 트랜지션의 입력함수로 구할 수 있으며, 박스에 대응되는 플립플롭의 입력들은 트랜지션의 입출력함수들을 사용하여 구할 수 있으므로 변환 수식들은 프로그램에 의해서 구할 수 있다. 따라서 펜티엄 컴퓨터에서 프로그램을 사용하여 회로변환 수식들을 구한 결과, 다음과 같은 수식이 되었다. 여기서 입력 게이트 신호와 출력신호의 기호는 EMFG의 기호와 동일하게 사용하였다.

$$\begin{aligned}
 T_1 &= x'B_1 & T_2 &= xB_2 & T_3 &= xB_3 \\
 T_4 &= x'B_3 & T_5 &= x'B_4 & T_6 &= xB_4 \\
 T_7 &= y \\
 G_1 &= T_3 & D_1 &= T_1 \\
 G_2 &= T_1 + T_5 & D_2 &= T_2 \\
 G_3 &= T_2 + T_6 & D_3 &= T_3 + T_4 \\
 G_4 &= T_4 & D_4 &= T_5 + T_6 \\
 y &= T_6
 \end{aligned}$$

그리고 이 수식들을 사용하여 회로로 변환하면 그림 14가 된다. 물론 회로변환 수식들을 사용하여 컴퓨터 프로그램으로 회로까지 도식화할 수 있을 것이다.

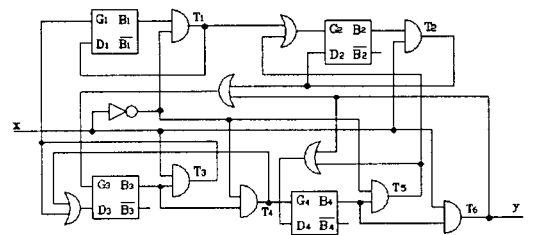


그림 14. 0101 감지기 회로

이 설계 예에서 보면, 비교적 설계 대상이 간단하므로 설계의 용이성은 기존 방식을 사용한 것과 본 방식을 사용한 것이 서로 유사하다. 그러나 설계하고자 하는 시스템이 복잡하게 되면, 복잡한 기호들로 구성되어 있고 출력 역 아크를 사용할 수 없는 기존 방식의 설계가 더 어려워진다는 것은 분명하다.

그리고 기존 방식의 설계에서는 수학적 표현이 불

가능하던 것을 본 방식을 사용하게 되면 수학적 표현이 가능할 뿐 아니라 프로그램을 통하여 EMFG의 동작 즉 시스템의 동작을 분석할 수 있다는 것이 본 설계 예에서 입증되었다.

또한 기존 방식의 설계에서는 EMFG 상에서만 회로변환이 가능하던 것을 설계 예에서와 같이 본 방식을 사용하여 설계하게 되는 경우에는 EMFG의 수학적 표현 및 회로변환 수식을 사용하여 컴퓨터 프로그램에 의해 회로변환이 가능하다는 것이 입증되었다.

따라서 본 방식을 사용하여 설계하는 것이 기존 방식을 사용하는 것보다 여러 가지 면에서 유리하다는 것이 설계 예에서 입증되었다.

## 5. 결 론

EMFG는 Petri Net의 성질 대부분을 그대로 가지고 있으므로 시스템의 동시성이나 병렬성을 잘 표현할 수 있으며 동기 및 비동기 시스템에도 그 적용이 쉽다. 뿐만 아니라 시스템을 EMFG를 사용하여 설계하면 회로로 쉽게 변환될 수 있으므로 이산 제어 시스템 및 공장 자동화의 제어회로 등을 모델링하고 설계할 때 강력한 도구로 사용될 수 있을 것이다.

본 연구에서는 다음과 같은 결론을 얻었다.

첫째, 본 연구에서 제시된 재정의된 EMFG는 기존의 EMFG의 성질이 그대로 존속되면서 표현이 간소화되었다. 이는 이산 시스템의 설계 및 분석이 쉽게 되도록 한다.

둘째, 기존의 EMFG에서는 박스, 트랜지션, 아크, 상호간의 관계 등의 수학적 표현이 불가능하였으나, 본 연구에서는 이것을 가능하게 하여 EMFG의 수학적 해석이 가능하게 되었다. 이는 EMFG의 동작 즉 시스템의 동작을 컴퓨터 프로그램에 의해 분석 가능하게 해준다.

셋째, 본 연구에서 트랜지션의 출력 역 아크가 도입되어 기존의 EMFG에서 표현 불가능한 현상들이 표현 가능하게 되었다. 또한 기존의 EMFG에서는 일반 아크와 역 아크의 역할이 불명확하였으나, 본 연구에서는 출력 역 아크로 인하여 서로 상반적인 관계가 됨으로써 상호간의 역할이 명확하게 되었다. 이는 이산 시스템의 설계를 더욱 쉽게 되도록 해준다.

넷째, 기존의 EMFG에서는 EMFG 원소들이 일대

일로 회로변환되었지만, 본 연구에서는 재정의된 EMFG의 수식적인 회로변환 방법이 제안되었다. 이는 컴퓨터 프로그램을 사용한 회로변환을 가능하게 해준다.

다섯째, 기존의 EMFG에서는 회로로 잘못 변환되는 경우들이 있었지만, 본 연구에서는 GD 플립플롭을 사용함으로써 이들이 문제없이 변환되었다.

결론적으로 재정의된 EMFG는 기존의 EMFG보다 여러 가지 면에서 개선되었으며, 더욱이 이산 시스템의 설계와 구현 및 분석을 용이하게 해 준다.

## 참 고 문 헌

- [1] J. L. Peterson, "Petri Nets," ACM Computing Surveys, Vol. 9, No. 3, pp.223-252, 1977.
- [2] R. Karp and R. Miller, "Properties of a Model for Parallel Computations: Determinacy, Terminations, Queueing," SIAM J. Appl. Math. Vol.14, pp.1300-1411, 1966.
- [3] F. Commoner, A. Holt, S. Han and A. Pnueli, "Marked Directed Graph," Journal of Computer and System Sciences, Vol.5, No.5, pp.511-523, Oct. 1971.
- [4] T. Murata, "A Method for Synthesizing Marked Graphs from Given Marking," Proceedings of the Tenth Annual Asilomar Conference on Circuits, and Computers, pp.202-206, Nov. 1976.
- [5] Tadao Murata, "Petri Nets, Marked Graph and Circuit System Theory," IEEE, Vol. 11, No. 3, pp.1-12, 1977.
- [6] T. Murata, "Circuit Theoretic Analysis and Synthesis of Marked Graph," IEEE Transactions on Circuit and Systems, Vol.CAS-24, No.7, pp. 400-405, July 1977.
- [7] T. Murata, "Circuit Theoretic Analysis and Synthesis of Marked Graph," IEEE Transactions on Circuit and Systems, Vol.CAS-24, No.7, pp. 400-405, July 1977.
- [8] 전순미, "마크흐름선도의 Deadlock 해석," 부산대학교 대학원 석사학위 논문, 1978. 2.
- [9] 여정모, "마크흐름선도의 확장," 부산대학교 대

학원 석사학위 논문, 1982. 2.

- [10] 여정모, 황창선, "확장된 마크흐름선도와 시퀀셜 제어시스템에의 응용," 부산대학교 공과대학 연구보고, 제25집, 1983. 6.
- [11] K. Hasegawa, "Mark Flow Graph and Its Application to FA," Journal of the SICE, Vol.22, No.11, pp.946-951, Nov. 1983.
- [12] K. Hasegawa, K. Takahashi, R. Masuda and H. Ohno, "Proposal of Mark Flow Graph for Discrete System Control," Trans. of SICE, Vol.20, No.2, pp.122-129, Feb. 1984.
- [13] T. Murata, N. Komoda, K. Matsumoto and K. Haruna, "A Petri Net-based Controller for Flexible and Maintainable Sequence Control and its Applications in Factory Automation," IEEE Transaction on Industrial Electronics, Vol.33, No.1, pp. 1-8, Jan. 1986
- [14] A. Merabet, "Synchronization of operations in a flexible manufacturing cell: the Petri Net approach," Journal of Manufacturing Systems, Vol.5, No.3, pp.161-169, 1986.
- [15] 여정모, "EMFG 회로의 간략화에 관한 연구," 부산개방대학 연구보고, 제29집, 1987. 12.
- [16] K. Hasegawa, K. Takahashi and P. E. Miyagi, "Application of the Mark Flow Graph to Represent Discrete Event Production Systems and System Control," Trans. of SICE, Vol.24, No.1, pp.69-75, Jan. 1988.
- [17] P. E. Miyagi, K. Hasegawa and K. Takahashi, "A Programming Language for Discrete Event Production Systems Based on Production Flow Schema and Mark Flow Graph," Trans. of SICE, Vol.24, No.2, pp.183-190, Feb. 1988.
- [18] R. Willson and B. Krogh, "Petri net tools for the specification and analysis of discrete controllers," IEEE transaction on Software Engineering, Vol.16, No.1, pp.39-50, Jan. 1990.
- [19] Ferrarini, L., "An incremental approach to logic controller design with Petri Nets," IEEE Transaction on System, Man, and Cybernetics, Vol.22, No.3, pp.461-473, 1992.
- [20] M. Zhou, DiCesare and F. Desrochers, "A Hybrid Methodology for Synthesis of Petri Net Models for Manufacturing Systems," IEEE Trans. on Robotics and Automation, Vol.8, No.3, pp.350-361, 1992.
- [21] M. Jeng and F. DiCesare, "A review of synthesis techniques for Petri Nets with applications to automated manufacturing systems," IEEE transaction on Systems, Man and Cybernetics, Vol.23, No.1, pp.301-312, Jan. 1993.
- [22] A. Falcione and B. Krogh, "Design Recovery for Relay Ladder Logic," IEEE Control Systems Magazine, pp.90-98, April 1993.
- [23] D. Y. Lee and F. DiCesare, "Integrated models for scheduling flexible manufacturing systems," Proceeding of the 1993 IEEE International Conference on Robotics and Automation, Atlanta, GA, pp.827-832, May 1993.
- [24] 여정모, 하재목, "확장된 마크흐름선도의 재구성 과 회로변환," 한국멀티미디어학회, 1998년도 춘계 학술발표논문집, pp.423-431, 1998.6.
- [25] ZVI KOHAVI, "Switching and Finite Automata Theory," McGraw-Hill Book Company.



여 정 모 (余定模)

1980년 2월 동아대학교 공과대 전자공학과 졸업(공학사)  
 1982년 2월 부산대학교 대학원 전자공학과 전자공학 전공(공학석사)  
 1993년 2월 울산대학교 대학원 전자및전산기공학과 전자

공학 전공(공학박사)

1986년 8월~현재 부경대학교 전자계산학과 부교수  
 관심분야: 마이크로프로세서, CAD