

확장된 마크흐름선도의 재구성과 회로변환

○

여 정 모, 하 재 목

부경대학교 전자계산학과

The Reconstruction and the Circuit Translation of An Extended Mark Flow Graph

○

Jeong Mo Yeo and Jae Mok Ha

Dept. of Computer Science, Pukyong National University

요 약

이진값을 가지는 이진 시스템을 모델화하고 설계하여 구현할 수 있도록 Petri Net의 한 부류인 박스의 마크수가 최대 1인 안전(safe)한 확장된 마크흐름선도(EMFG : Extended Mark Flow Graph)를 재구성하였다. 레벨박스와 펄스박스 및 게이트박스를 하나의 박스로 통일하였으며, 보존아크를 제거하고 서로 상반되는 개념을 가지는 일반아크와 역아크만을 사용하였고, 특히 트랜지션의 출력아크로 새로이 역아크를 도입함으로써 EMFG의 표현을 명확하게 하였다. 또한 EMFG에서 자기루프를 정의하였다. 그리고 이진 시스템을 모델링한 EMFG를 실제회로로 구현하는 방법이 새로이 제시되었다.

1. 서 론

마크선도(Mark Graph)[1]는 Karp와 Miller에 의해 제안되었으며 Petri Net에서 제한을 가한 그래프이고, 방향성마크선도(Marked Directed Graph)[2]는 마크선도에서 충돌(conflict)하지 않고 동시성을 표현할 수 있었으며, 그래프의 안전성(safeness), 활성(liveness) 및 도달가능성(reachability)을 해결하는 알고리즘을 제시하였다. 마크흐름선도(MFG: Mark Flow Graph)[3]는 Petri Net에서 모든 place의 마크가 안전(safe)하도록 제한을 가한 그래프이다. 이러한 마크선도와 방향성마크선도 및 마크흐름선도는 비동기 처리[4]나 순서회로[5] 또는 여러 가지 회로[6-9] 등의 설계나 분석에 사용되었다.

순서제어시스템의 모델링과 설계 및 구현에 적합하도록 마크흐름선도를 확장하여 확장된 마크흐름선도(EMFG : Extended Mark Flow Graph)가 [10]에서 제안되었다. 확장된 마크흐름선도로서 시스템을 모델링하면 Petri Net나 마크선도로서 시스템을 설계하는 것과 마찬가지로 시스템의 동시성이나 병렬

성을 잘 표현할 수 있다. 뿐만 아니라 EMFG는 안전한 그래프이므로 모델링된 EMFG를 일대일로 변환하면 직접 제어회로를 얻을 수 있어 그 구현이 아주 쉬웠다. EMFG 사용으로 [10]에서는 4층 엘리베이터 제어회로가 설계되어 직접 구현되었다. 또한 EMFG는 순서제어회로에도 그 적용[11]이 쉬우며 설계된 EMFG 회로도 간략화[12]될 수 있었다.

하지만 [10]에서는 박스를 게이트박스와 레벨박스 및 펄스박스로 분할하였을 뿐만 아니라 아크도 일반아크와 보존아크 및 역아크로 분할하여 그 표현이 복잡하였으며, 역아크가 트랜지션의 출력아크로 사용되지 못함으로 인해 일반아크와 역아크간의 상관관계가 거의 존재하지 못했다. 아울러 EMFG를 회로로 구현할 때도 수식적으로 구현하지 못하고 박스, 트랜지션 및 아크들이 하나씩 일대일로 구현하여 전체회로를 구성하였다.

따라서 본 논문에서는 게이트박스와 레벨박스 및 펄스박스를 하나의 박스로 통일하였으며 아크도 보존아크를 제거하여 일반아크와 역아크로만 구분하였다. 그리고 트랜지션의 출력아크로 역아크를 사용하

수 있도록 함으로써 일반아크와 역아크가 서로 상반되는 개념을 가지고 동작되도록 하였다. 다중아크의 동작을 명확하게 하였으며 자기루프를 정의하여 그 동작도 명확하게 하였다. 또한 박스를 새로이 구성된 플립플롭인 GD 플립플롭으로 변환하고 트랜지션은 AND 게이트로 변환한 후, GD 플립플롭의 입력 G와 D를 수식으로 표현하고 이를 간략화하여 회로를 구성함으로써 EMFG를 쉽게 회로로 구현할 수 있게 하였다. EMFG의 회로변환 예로 동기식 3비트 증가카운터를 EMFG로 설계하여 실제회로로 구현하였다.

마크흐름선도(Mark Flow Graph)가 Hasegawa에 의하여 새로이 제안[14]되었고 공장자동화[13]와 전기 스윗칭시스템[15] 및 생산시스템[16, 17]에 적용되었다. Petri Net도 마찬가지로 논리제어기(logic controller)[18, 20]나 릴레이 라다(ladder)[19]에 적용되었으며, 공장자동화[21-25]에 이용되었다. 이외에도 공장자동화나 논리제어기에 적용된 연구는 수없이 많았다. 본 연구에서 제시된 EMFG와 회로변환을 사용하면 공장자동화나 논리제어기 등에 쉽게 적용되리라 사료된다.

2. EMFG의 정의 및 표현

이전값으로 동작하는 이산시스템의 제어회로를 모델링하고 설계하는데 적합한 Petri Net의 한 부류인 확장된 마크흐름선도(EMFG : Extended Mark Flow Graph)를 정의한다.

정의 1) EMFG G는 다음 5개의 순서쌍으로 정의되는 그래프이다.

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

여기서 B는 박스(box)의 집합이고 T는 트랜지션(transition)의 집합이다. I는 트랜지션에 대한 입력함수이고 O는 트랜지션에 대한 출력함수이며 M은 각 박스의 마크상태를 나타내는 마크벡터이다.

그림 1은 EMFG의 한 예이다. 그림 1에서 b₁, b₂, b₃, b₄, b₅, b₆, b₇는 박스이며 t₁, t₂, t₃, t₄(5), t₅는 트랜지션이다.

EMFG에서 박스는 여러 가지 상태들을 표현하고, 트랜지션은 이런 상태들의 변화조건을 표현한다.

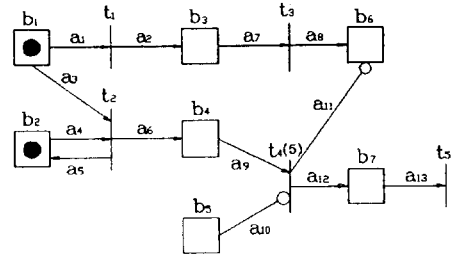


그림 1. EMFG의 한 예

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

$$B = \{b_i \mid i \text{는 양의 정수}\}$$

여기서 b_i는 상태를 나타내며 실선의 사각박스로 표시한다.

그림 1의 EMFG에서 박스의 집합 B를 구하면 다음과 같다.

$$B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$$

시스템의 현재상태에서 다음상태로 상태변화가 일어날 때 특정 시간이 요구되는 경우가 있고 그렇지 않은 경우가 있다. 또한 모든 상태변화의 시간이 공통적으로 동일한 경우가 있으며 그렇지 못한 경우가 있다. 이러한 상태변화의 조건을 트랜지션(transition)으로 표현한다.

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

$$T = \{t_i, t_j(s) \mid i \text{와 } j \text{는 양의 정수, } s \text{는 시간}\}$$

여기서 t_i는 상태변화 시간이 명기되지 않은 트랜지션으로 일반트랜지션(general transition)이고, t_j(s)는 상태변화 시간 s가 명기된 트랜지션으로 시간트랜지션(time transition)이며, 트랜지션은 막대로 표시한다.

그림 1의 EMFG에서 t₁, t₂, t₃, t₅는 일반 트랜지션

이고, $t_4(5)$ 는 5의 상태변화시간을 요하는 시간트랜지션이며, 트랜지션의 집합 T 는 다음과 같다.

$$T = \{t_1, t_2, t_3, t_4(5), t_5\}$$

보통 일반트랜지션은 상태변화시간을 요하지 않는 경우나 상태변화시간을 요구하더라도 시스템의 동작에 영향을 주지 않거나 무시해도 좋은 경우 또는 시스템의 대표적인 상태변화시간을 생략해도 좋은 경우 등에 사용된다. 시간트랜지션은 상태변화시간이 시스템에 영향을 주는 경우나 특별히 시간을 명기하여야 하는 경우 등에 사용된다.

박스과 트랜지션과의 대응관계를 입력함수와 출력함수로 나타내고 아크로 표시한다. 즉 아크를 사용하여 박스에서 트랜지션으로, 트랜지션에서 박스로 연결한다.

정의 4) EMFG에서 실선의 화살표를 일반아크(general arc)로, 실선과 실선의 끝에 붙어 있는 작은 원을 역아크(inhibit arc)로 정의한다.

아크는 상태변화시 상태변화조건을 만족시키는 역할에 사용되거나 어떤 상태에게 영향을 주는 역할을 한다. 즉 아크는 상태가 변화하고 이동되는 경로 역할을 하는 셈이다. 그림 1의 EMFG에서 $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{12}, a_{13}$ 들은 일반아크이고 a_{10}, a_{11} 들은 역아크이다.

정의 5) EMFG G에서 어떤 트랜지션 t 에 대한 입력함수 $I(t)$ 는 다음과 같이 정의된다.

$$I(t) = \{b_i, b_j' \mid i, j \text{는 양의 정수}\}$$

여기서 $b_i(b_j')$ 는 일반아크(역아크)로서 연결된 트랜지션 t 의 입력박스이다.

그림 1의 EMFG에서 각 트랜지션의 입력함수들을 구하면 다음과 같다.

$$\begin{aligned} I(t_1) &= \{b_1\} \\ I(t_2) &= \{b_1, b_2\} \\ I(t_3) &= \{b_3\} \\ I(t_4(5)) &= \{b_4, b_5'\} \\ I(t_5) &= \{b_7\} \end{aligned}$$

정의 6) EMFG G에서 어떤 트랜지션 t 에 대한 출력함수 $O(t)$ 는 다음과 같이 정의된다.

$$O(t) = \{b_i, b_j' \mid i, j \text{는 양의 정수}\}$$

여기서 $b_i(b_j')$ 는 일반아크(역아크)로서 연결된 트랜지션 t 의 출력박스이다.

그림 1의 EMFG에서 출력함수들을 구하면 다음과 같다.

$$\begin{aligned} O(t_1) &= \{b_3\} \\ O(t_2) &= \{b_4\} \\ O(t_3) &= \{b_6\} \\ O(t_4(5)) &= \{b_6', b_7\} \\ O(t_5) &= \emptyset \end{aligned}$$

이산시스템에서의 상태는 여러 가지가 있을 수 있다. 어떤 장치의 동작이나 준비가 하나의 상태가 될 수 있고, 계수기의 동작상태, 플립플롭의 값, 펄스, 스위치, 릴레이 동작 등등이 상태가 될 수 있다. 이 상태들은 각각 하나의 박스로 표현될 수 있으며, 각 박스에서 자신의 상태가 만족되면 박스내에 점을 찍어 표시하고 이를 마크라 한다. 물론 박스에 마크가 없으면 박스자신의 상태가 만족되지 않은 것이다. 따라서 박스내의 마크수는 0이거나 1이다. 즉 박스의 마크수는 1을 초과하지 못한다. 예를 들어 동작된 상태에서 다시 동작시킨다 하더라도 동작 그 자체는 그대로 유지되며 신호가 가해진 상태에서 다시 신호를 가한다 하더라도 그 신호는 그대로 유지될 것이기 때문이다.

정의 7) EMFG G에서 마크벡터(mark vector) M 은 다음과 같이 정의되는 열벡터 또는 행벡터이다.

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

여기서 μ_i 는 박스 b_i 의 마크수이며 0이거나 1이다. n 은 EMFG내에 존재하는 박스의 수이다.

그림 1의 EMFG에서 마크벡터 M 을 구하면 다음과 같다.

$$M = (1, 1, 0, 0, 0, 0, 0)$$

정의 8) EMFG에서 출력트랜지션만 있는 박스를 생성박스(source box)라 하고 입력트랜지션만 있는 박스를 소멸박스(sink box)라 하며, 출력박스만 있는 트랜지션을 생성트랜지션(source transition)이라 하고 입력박스만 있는 트랜지션을 소멸트랜지션(sink transition)이라 한다.

그림 1의 EMFG에서 b_1, b_2, b_5 는 생성박스이고 b_6 는 소멸박스이며, t_5 는 소멸트랜지션이다.

상태변화가 일어나는 현상을 트랜지션이 점화한다고 하고, 상태변화가 일어날 수 있는 조건을 트랜지션의 점화조건이라 한다.

정의 9) EMFG에서 어떤 트랜지션 t 의 모든 입력박스들의 마크상태가 다음을 만족할 때 t 의 점화조건이 만족되었다고 한다. 그리고 한 박스의 마크는 이에 연결된 여러 트랜지션의 점화조건을 동시에 만족시킬 수 있다.

$$\mu_{ig} = 1, \mu_{ii} = 0$$

여기서 $\mu_{ig}(\mu_{ii})$ 는 일반아크(역아크)로서 연결된 트랜지션 t 의 입력박스의 마크수이다.

일반아크로 연결된 입력박스에는 마크가 존재하여야 점화조건이 만족되고 역아크로 연결된 입력박스에는 마크가 존재하지 않아야 점화조건이 만족되므로 일반아크와 역아크는 상반되는 개념을 가진다.

그림 1의 EMFG에서 트랜지션 t_1 과 t_2 만이 점화조건이 만족되어 있으며 다른 어떤 트랜지션도 점화조건이 만족되어 있지 않다. 그리고 b_1 의 마크는 트랜지션 t_1 과 t_2 의 점화조건을 동시에 만족시킨다.

여러 트랜지션이 하나의 입력박스를 공유할 때 이 입력박스를 충돌박스(conflict box)라 한다. 충돌박스에 마크가 존재하여 여러 트랜지션의 점화조건을 동시에 만족시킬 때, Petri Net에서는 어느 하나만이 유효하게 작용되나 정의 9와 같이 EMFG에서는 만족되는 모든 트랜지션에 유효하게 작용한다.

트랜지션의 점화조건은 입력박스(들)의 마크상태에 의해서만 결정되므로 입력박스(들)의 마크유무가 점화조건을 만족시키는 셈이다. 트랜지션의 점화조건이 만족되면 트랜지션은 점화를 시작하고 점화시간이 경과하면 트랜지션에 연결된 입력박스들과 출력박스들의 마크상태가 변화하여 상태변화가 일어남

으로서 점화가 완료된다. 점화중에는 어떠한 상태변화도 일어나지 않기 때문에 박스들의 마크상태에도 변화가 없다. 그리고 트랜지션이 점화완료된 후 입력박스의 마크가 그대로 유지되는 경우도 있고 제거되는 경우도 있다. 즉 실제 회로에서 상태변화가 일어나면 상태변화를 시킨 상태는 그대로 유지되는 경우도 있을 수 있으며 또는 변하는 경우도 있을 수 있다는 것이다.

정의 10) EMFG에서 어떤 트랜지션 t 의 점화조건이 만족되어 점화시간만큼 계속 점화조건이 만족되어 있으면 트랜지션 t 는 점화완료되어 t 에 연결된 입력박스들과 출력박스들의 마크상태는 다음과 같은 순서로 변화한다. 그리고 어떤 하나의 박스에 입력된 여러 트랜지션들이 동시에 점화완료될 때, 이 박스의 마크수는 점화완료된 트랜지션들의 영향이 논리 OR되어 결정된다.

$$(1) \mu_{ig} = 0, \mu_{ii} = 0$$

$$(2) \mu_{og} = 1, \mu_{oi} = 0$$

여기서 $\mu_{ig}(\mu_{ii})$ 는 일반아크(역아크)로서 트랜지션 t 에 연결된 입력박스의 마크수이고, $\mu_{og}(\mu_{oi})$ 는 일반아크(역아크)로서 트랜지션 t 에 연결된 출력박스의 마크수이다.

트랜지션이 점화완료된 후, 일반아크로 연결된 입력박스의 마크는 소멸되고 역아크로 연결된 입력박스의 마크는 없는 상태 그대로 유지된다. 다음에 일반아크로 연결된 출력박스에는 마크가 생성되어 마크가 존재하게 되고, 역아크로 연결된 출력박스에 마크가 있는 경우에 그 마크가 소멸되어 마크가 존재하지 않게 된다. 즉 일반아크와 역아크는 상반되는 개념을 가지고 있는 셈이다.

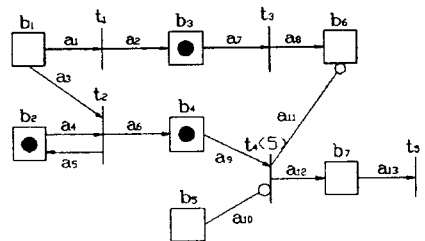


그림 2. 그림 1에서 t_1 과 t_2 가 점화한 후의 EMFG

그림 1의 EMFG에서 트랜지션 t_1 과 t_2 가 점화완료 하면 그림 2와 같이 되어 마크벡터는 (0, 1, 1, 1, 0, 0, 0)이 된다. 만약 b_4 에 마크가 존재하여 t_2 가 점화 완료되더라도 $t_4(5)$ 의 점화에 관계없이 b_4 의 마크수는 1이 된다.

박스 b와 트랜지션 t 사이에 여러개의 같은 방향의 아크를 다중아크(multiple arc)라 하며, 그림 3과 같이 6가지 종류가 있을 수 있다.

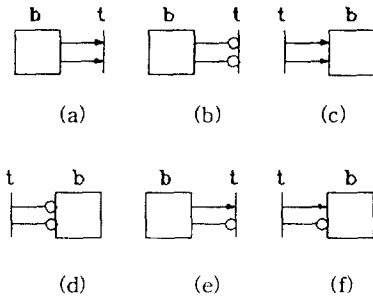


그림 3. 다중아크들

그림 3(a)와 그림 3(b)의 두 아크는 각각 정의 9에 의하여 박스의 마크가 트랜지션에 동일하게 영향을 주므로 하나의 아크를 제거하여 그림 4(a)와 그림 4(b)와 같이 각각 치환할 수 있다. 그리고 그림 3(c)와 그림 3(d)의 두 아크도 각각 정의 10에 의하여 트랜지션의 영향이 박스에 동일한 효과를 가져다 주므로 하나의 아크를 제거하여 그림 4(c)와 그림 4(d)와 같이 치환할 수 있다. 따라서 두 개 이상의 동일한 아크로 구성되는 다중아크는 하나의 아크로 치환할 수 있다.

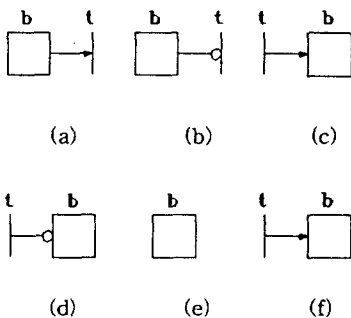


그림 4. 그림 3과 동치인 아크

그림 3(e)의 다중아크는 박스의 마크유무에 관계

없이 트랜지션의 점화조건을 만족시킬 수 없으므로 트랜지션은 결코 점화할 수 없다. 따라서 그림 3(e)와 같은 다중아크를 가지는 트랜지션은 제거되어도 무방하다. 그리고 그림 3(f)의 다중아크는 그림 4(f)와 같이 하나의 역아크가 제거되어 구성되어도 EMFG의 동작에는 영향을 주지 않는다.

정의 11) 박스 b와 트랜지션 t 사이에 방향이 서로 다른 아크가 연결되어 있으면 자기루프(self loop)가 형성되어 있다고 한다. 그림 5(a)의 자기루프를 보존 자기루프(preserved self loop)라 하고, 그림 5(b)의 자기루프를 생성 자기루프(create self loop)라 하며, 그림 5(c)와 그림 5(d)의 자기루프를 의사 자기루프(pseudo self loop)라 한다.

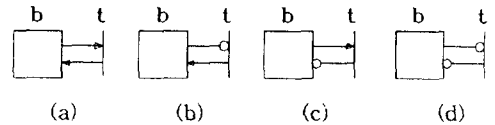


그림 5. 자기루프들

보존 자기루프는 트랜지션이 점화한다 하더라도 박스의 마크는 소멸된 후 다시 생성되므로 그대로 보존되며, 생성 자기루프는 트랜지션이 점화하면 박스에 새로운 마크를 생성시킨다. 그리고 의사 자기루프는 트랜지션의 출력 역아크를 제거하여도 EMFG의 동작에는 영향을 주지 않으므로 출력역아크를 제거할 수 있다. 따라서 의사 자기루프는 출력 역아크가 제거되면 자기루프를 형성하지 않게 된다.

3. EMFG의 회로변환

EMFG는 시스템의 동작을 개념적으로 설계하는데도 사용될 수 있고, 각 개념에 해당하는 실제적인 동작을 설계하는데도 사용될 수 있다. 즉 EMFG가 추상적인 개념을 표현한 경우에는 시스템의 도식적인 표현이기 때문에 시스템의 동작을 분석하고 이해하여 설계하는데 이용될 것이므로 회로변환한다는 것이 무의미하다. 하지만 EMFG가 시스템의 실제적인 동작을 표현한 경우에는 실제회로로 구현되어야 하므로 EMFG의 회로변환이 요구된다. 이와같은 실제회로 변환이 필요한 경우에 EMFG가 어떻게 회로 변환되는지 그 방법을 제시한다.

EMFG의 박스는 어떤 스위치의 동작이나 모터의

동작과 같은 구체적인 소자의 동작상태를 표현하기도 하고, 제어회로에서의 어떤 상태를 나타내기 위해서 사용될 수도 있을 것이다. 따라서 구체적인 소자의 동작상태를 나타내는 박스는 원하는 상태를 만족시킬 수 있도록 구현해 주어야 한다. 그리고 제어회로에서의 어떤 상태를 나타내는 박스는 플립플롭으로 구현해준다.

EMFG의 회로변환에 사용되는 플립플롭은 EMFG의 성질을 만족시키기 위하여 그림 6과 같은 GD 플립플롭을 사용한다. 그리고 GD 플립플롭의 진리표를 표 1에 나타내었다.

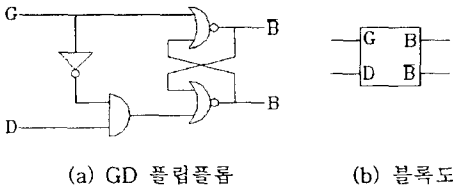


그림 6. EMFG 회로변환에 사용되는 플립플롭과 진리표

표 1. GD 플립플롭의 진리표

G	D	Q(t+1)	Q'(t+1)
0	0	Q(t)	Q'(t)
0	1	0	1
1	0	1	0
1	1	1	0

만약 클럭펄스에 동기되어 동작하는 이진시스템을 EMFG로 설계하여 이를 회로로 변환할 때는 클럭이 있는 GD 플립플롭을 사용하여야 할 것이다.

EMFG의 트랜지션은 상태변화가 일어나는 곳으로서 모든 입력박스의 상태가 만족되어야 점화하므로 논리 AND 게이트로 구현할 수 있을 것이다. 그리고 EMFG의 회로변환에 사용되는 시간 s를 가진 시간트랜지션은 AND 게이트내에 s를 표기하여 나타내도록 하며, AND 게이트의 입력조건이 모두 만족되었을 때 타이머가 동작하여 정해진 시간이 지나면 출력한다고 가정한다. 물론 타이머가 동작중에 AND 게이트의 입력조건이 만족되지 않으면 즉시 타이머의 동작은 중지된다.

정리 1) EMFG의 회로변환은 다음과 같은 방법으로 행한다.

1) 모든 트랜지션을 논리 AND 게이트로 변환한다. 어떤 트랜지션 t_i 를 논리 AND 게이트로 변환했을 때, AND 게이트의 출력 t_i 는 다음의 AND 함수로 표현된다.

$$t_i = \text{AND}(b_x, b_y') \quad (1)$$

여기서 $b_x(b_y')$ 는 t_i 의 입력박스가 변환된 GD 플립플롭 $b_x(b_y)$ 의 정상(보수)출력이다.

2) 구체적인 소자의 동작상태를 나타내는 박스의 회로는 요구한 대로 변환한다.

3) 2)에서 제외된 모든 박스들을 GD 플립플롭으로 바꾸고 각 GD 플립플롭의 입력을 구한다. 어떤 박스 b_2 에 대응되는 GD 플립플롭의 입력 G_2 와 D_2 는 다음의 OR 함수로 표현된다.

$$G_z = \text{OR}(t_i) \quad (2)$$

$$D_z = \text{OR}(t_i, t_k) \quad (3)$$

여기서 $t_i(t_j)$ 는 트랜지션 $t_i(t_j)$ 에서 일반(역)아크로 박스 b_2 에 연결되어 있을 때 트랜지션 $t_i(t_j)$ 에 대응되는 AND 게이트 출력이다. 그리고 t_k 는 박스 b_2 에서 일반아크로 트랜지션 t_k 에 연결되어 있을 때 트랜지션 t_k 에 대응되는 AND 게이트 출력이다.

증명) 박스 b_2 에 존재하는 마크의 모든 변화에 대하여 변환된 회로가 올바르게 동작되는지만 증명하면 되므로 다음 4가지 경우만 타당하면 정리는 성립한다.

1) 마크가 존재하지 않는 그대로 유지되는 경우 이 경우는 일반아크로 연결된 어떠한 트랜지션도 점화조건이 만족되지 않으며, 역아크로 연결된 트랜지션들은 점화조건이 만족될 수도 있는 경우이다. 따라서 (2)식은 0이 되고 (3)식은 0이나 1이 될 수 있으나 b_2 의 상태는 0이 되어 그대로 유지된다.

2) 존재하지 않는 마크가 생성되는 경우 이 경우는 일반아크를 통하여 박스로 입력되는 하나 이상의 트랜지션의 점화조건이 만족되어 있어야 하며, 역아크로서 박스에 연결된 트랜지션들은 점화조건이 만족될 수도 있는 경우이다. 따라서 (2)식은 1이 되고 (3)식은 0이나 1이 될 수 있다. 즉 (G_z, D_z)는 (1, 0/1)이 되므로 표 1의 진리표에서 b_2 는 1이 되어 마크가 생성된다.

3) 존재하는 마크가 소멸되는 경우

이 경우는 일반아크를 통하여 박스에서 출력되는 하나 이상의 트랜지션의 점화조건이 만족되거나 역아크를 통하여 박스로 입력되는 하나 이상의 트랜지션의 점화조건이 만족되어야 하며, 일반아크를 통하여 박스로 입력되는 어떠한 트랜지션도 점화조건이 만족되지 않는 경우이다. 따라서 (2)식은 0이 되고 (3)식은 1이 되어 b_2 는 0이 되어 존재하던 마크는 소멸되어 버린다.

4) 존재하는 마크가 그대로 유지되는 경우

이 경우에는 세가지 경우가 있을 수 있다. 첫 번째 경우는 박스에 연결된 어떠한 트랜지션도 점화조건이 만족되지 않는 경우로서 (2)식과 (3)식이 모두 0이 되어 마크가 존재하는 그대로 유지하게 된다. 두 번째 경우는 보존 자기루프로 구성된 하나 이상의 트랜지션의 점화조건이 만족되어 있으며 다른 트랜지션은 점화조건이 만족될 수도 있는 경우이다. 이 경우는 (2)식과 (3)식이 동시에 1이 되어 역시 마크가 존재하는 그대로 유지된다. 세 번째 경우는 일반아크를 통하여 박스에 입력되는 하나 이상의 트랜지션의 점화조건이 만족되어 있고, 박스에 연결된 다른 트랜지션들은 점화조건이 만족될 수도 있는 경우이다. 이 경우는 (2)식이 1이 되고 (3)식은 0이거나 1이 된다. 즉 (G_2, D_2)는 (1, 0/1)이 되므로 표 1의 진리표에서 b_2 는 1이 되어 마크는 그대로 유지된다.

그림 7은 동기 3비트 증가 카운터를 EMFG로 표현한 것이다. 그림 7에서 $C_2C_1C_0$ 는 카운터의 출력을 나타내는 박스이며 y 는 캐리를 나타내는 박스이다. 그림 7의 EMFG를 회로로 변환한 수식을 표 2에 나타내었으며 이를 회로로 그리면 그림 8과 같이 된다. 그림 8에서 CP는 클럭펄스 입력단자이다.

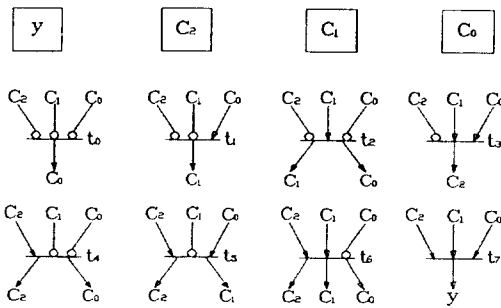


그림 7. 동기 3비트 증가카운터의 EMFG

표 2. 그림 7의 회로변환식

	회로변환식	간소화된 식
t_0	$C_2' C_1' C_0'$	$C_2' C_1' C_0'$
t_1	$C_2' C_1' C_0$	$C_2' C_1' C_0$
t_2	$C_2' C_1 C_0'$	$C_2' C_1 C_0'$
t_3	$C_2' C_1 C_0$	$C_2' C_1 C_0$
t_4	$C_2 C_1' C_0'$	$C_2 C_1' C_0'$
t_5	$C_2 C_1' C_0$	$C_2 C_1' C_0$
t_6	$C_2 C_1 C_0'$	$C_2 C_1 C_0'$
t_7	$C_2 C_1 C_0$	$C_2 C_1 C_0$
G_0	$t_0 + t_2 + t_4 + t_6$	C_0'
D_0	$t_1 + t_3 + t_5 + t_7$	C_0
G_1	$t_1 + t_2 + t_5 + t_6$	$C_1 \oplus C_0$
D_1	$t_2 + t_3 + t_6 + t_7$	C_1
G_2	$t_3 + t_4 + t_5 + t_6$	$C_2 \oplus (C_1 C_0)$
D_2	$t_4 + t_5 + t_6 + t_7$	C_2
y	t_7	$C_2 C_1 C_0$

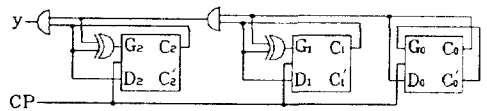


그림 8. 그림 7을 변환한 회로

EMFG는 Petri Net의 성질 대부분을 그대로 가지고 있으므로 시스템의 동시성이나 병렬성을 잘 표현할 수 있으며 동기 및 비동기 시스템에도 그 적용이 쉬울 뿐만 아니라 각 박스의 마크수가 항상 0이나 1인 안전한 그래프이므로 이진값을 가지는 이진시스템 및 공장자동화의 제어회로 등을 모델링하고 설계할 때 강력한 도구로 사용될 수 있다.

4. 결론

본 논문에서는 얻은 결론은 다음과 같다.

(1) 레벨박스과 펄스박스 및 게이트박스가 하나의 박스로 재정의되었고 일반아크와 보존아크 및 역아크가 일반아크 및 역아크로 재정의되었으므로 EMFG가 간소화되어 표현되었다.

(2) 트랜지션의 출력아크로서 역아크가 새로이 정의되어 도입되었을 뿐만 아니라 일반아크와 역아크의 개념이 명확하게 되었다.

(3) 다중아크 및 자기루프의 동작을 명확하게 하였다.

(4) EMFG의 회로변환을 수식으로 해결함으로써 쉽게 EMFG를 실제회로로 변환할 수 있게 하였다.

앞으로 더욱 더 연구하여야 할 과제는 다음과 같은 분야가 있을 수 있다.

(1) EMFG의 동작이 부울함수로 해석될 수 있다면, 시스템의 모델링과 설계가 쉬워지고 설계시간이 단축되며 설계된 EMFG의 분석이 쉬워지고 명확해짐에 따라 성능평가 및 동작특성평가 등이 쉽게 행해질 수 있을 것이다. 따라서 EMFG의 동작과 분석을 부울함수로 해석하는 연구가 필요하다.

(2) 어떤 시스템을 EMFG를 사용하여 모델링하거나 설계하였을 때, EMFG가 간소화될 수 있으면 시스템의 분석이나 설계에 많은 도움을 줄 수 있을 뿐만 아니라 구현되는 실제회로도 간소화될 수 있을 것이므로 EMFG의 간소화에 대한 연구가 필요하다.

(3) EMFG의 동작이 부울함수로 해석될 수 있으므로 EMFG 언어를 사용한다면 EMFG 설계를 컴퓨터 프로그램에 의해 자동설계할 수 있을 것이다. 따라서 EMFG 언어에 관한 연구가 필요하다.

(4) 데드록(deadlock)이나 활성(liveness) 또는 도달가능성(reachability) 등을 연구하여 EMFG의 이론을 더욱 더 진보시킬 필요가 있다.

(5) 컴퓨터 시스템, 동기나 비동기 시스템, 논리 제어기나 릴레이 라다(ladder) 등등 이진값으로 동작하는 실제 시스템을 EMFG로 설계하여 컴퓨터 산업 분야나 공장자동화 분야에 기여할 필요가 있다.

참고문헌

[1] 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.
 [2] 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.
 [3] 전순미, "마크흐름선도의 Deadlock 해석," 부산대학교 대학원 석사학위 논문, 1978. 2.
 [4] T. Agerwala, "A Complete Model for Representing the Coordination of Asynchronous Processes," Hopkins Computer Research Report No. 32, Computer Science Program, Johns Hopkins University, Baltimore, Maryland, pp.58, July 1974.

[5] M. Stucki, "Synthesis of Level Sequential Circuits: Further Development of a Procedure Based on a Petri-Net Type of Behavioral Description," Proceedings of the 13th Annual Allerton Conference on Circuit and System Theory, pp. 896-904, Oct. 1975.
 [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, "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.
 [8] T. Murata, "Petri Nets, Marked Graphs, and Circuit System Theory," IEEE Circuit and Systems Society Newsletter, Vol.11, No.3, pp.2-12, June 1977.
 [9] 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.
 [10] 여정모, "마크흐름선도의 확장," 부산대학교 대학원 석사학위 논문, 1982. 2.
 [11] 여정모, 황창선, "확장된 마크흐름선도와 시퀀셜 제어시스템에의 응용," 부산대학교 공과대학 연구보고, 제25집, 1983. 6.
 [12] 여정모, "EMFG 회로의 간략화에 관한 연구," 부산개방대학 연구보고, 제29집, 1987. 12.
 [13] K. Hasegawa, "Mark Flow Graph and Its Application to FA," Journal of the SICE, Vol.22, No.11, pp.946-951, Nov. 1983.
 [14] 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.
 [15] J. Ayache, M. Diaz and R. Valette, "A Methodology for Specifying Control in Electronic Switching System," Proceedings of the International Switching Symposium, Paris, pp.1049-1056,

May 1979.

[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] 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.

[19] A. Falcione and B. Krogh, "Design Recovery for Relay Ladder Logic," IEEE Control Systems Magazine, pp.90-98, April 1993.

[20] 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.

[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. 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.

[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] 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

[25] 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.