SFC에서 이벤트 발생에 관한 연구

유정봉
공주대학교 전기전자제어공학부
e-mail: jbyou@kongju.ac.kr

Study On The Generation of Event by SFC

Jeong-Bong, You
Dept of Electrical & Electronics & Control Eng. Kongju University

요 약

본 논문에서는 SFC로 기술된 PLC 프로그램에서 외부 이벤트가 발생하는데 엣지동작에 관해 고찰해 본다. SFC 프로그램에서는 이벤트가 발생하는데 상승엣지와 하강엣지일 때의 이진 신호 이론을 고찰해보고, 수학적인 정의는 어떻게 정리되는지를 고찰해본다.

1. 서론

SFC(Sequential Function Chart)는 IEC 848과 IEC 1131-3 표준을 통해 로컴 레벨에서 순차 제어 논리를 위한 표현 형태로 산업에서 널리 사용되고 있다. 1972년에 프랑스에서 GRAFCET 알고리즘을 처음 개발하여 사용하면서 PLC(Programmable Logic Controller)에서 사용하도록 SFC로 진화되면서 현대 제어에 사용빈도가 점점 높아가고 있다.

PLC는 복잡하고 다양한 기능을 요구하는 공정제어에서 필수 제어기로 사용되고 있으며, 조건처리를 하는데 간단하게 사용이 되며, 인터록을 구현하는데 편리하게 사용되고 있다.

PLC는 대부분 LD(Ladder Diagram)언어를 사용하여 프로그램하고 있다. 이 것은 조건을 처리하기가 수월하고, 인터록 구현하기가 용이하여 대부분의 프로그래머들은 LD 언어를 사용하고 있다. 그러나 LD 언어는 조합된 논리를 사용하여 조건을 처리하기는 수월하지만, 한 곳에서 에러가 발생하면 각각의 입력조건을 일일이 찾아내어 에러를 수정해야 한다. 따라서 LD 언어는 코딩은 오래걸리지 않지만, 디버깅하고 에러처리에 상당한 시간이 소요된다. 반면에 SFC는 순차논리로 기술되기 때문에 전체 공정의 흐름을 파악하기가 상당히 수월하다. 전체 공정이 현재 어느 공정을 처리하고 있는지를 모니터링할 수 있고, 전 공정과 다음 공정등 각각의 공정 처리를 용이하게 파악할 수 있는 것이다.

LD언어에서 이벤트(Event)를 처리할 때는 펄스 입력 조건을 부여해주면 되는 것이다.

SFC에서는 트랜지션(Transition)을 처리할 때 조건과 이벤트를 명확히 구분해주어야 하고, 이 이벤트를 수학적 개념으로 고찰해보는 것이 중요하다.

본 논문에서는 SFC로 설계할 때 트랜지션에서 이벤트의 상승엣지와 하강엣지에 대한 수학적 개념을 정리하고, 고찰해본다.

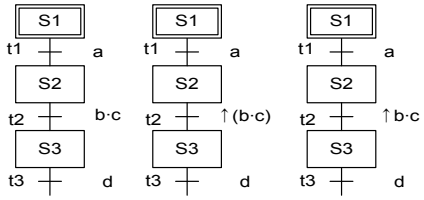
2. 이벤트의 처리

LD언어에서 이벤트(Event)를 처리할 때는 펄스 입력 조건을 부여해주면 되는 것이다.

예를 들어 그림 1에서 이벤트는 부여만 해주면 된다.

[그림 1] 입력 조건 예

그림 1에서 외부입력(P)은 상승엣지에서 입력 조건을 받아들이는 이벤트를 동작을 하고, 외부입력2(N)는 하강엣지에서 입력 조건을 받아들이는 이벤트 동작을 한다. 그러나 SFC에서는 다음 그림 2와 같이 트랜지션이 3가지 형태로 기술되게 된다.

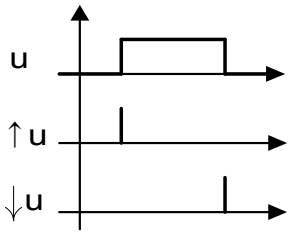


(a)조건 (b)이벤트 (c)혼합

[그림 2] SFC에서 트랜지션 조건

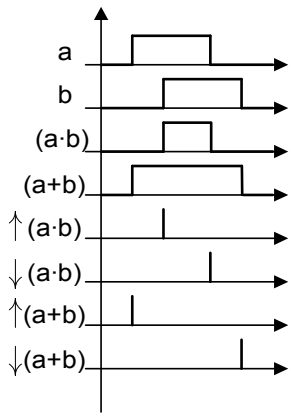
그림 2에서 (a)는 조건을 나타내며, (b)는 이벤트를 나타낸다. 그리고 (c)는 조건과 이벤트의 혼합을 나타낸다.

이벤트는 그림 3과 같이 상승 엣지와 하강 엣지로 나누어진다.

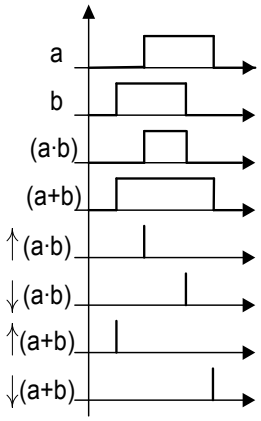


[그림 3] 이벤트의 상승 엣지와 하강 엣지

그림 3은 단일 이벤트에 대한 엣지동작을 나타낸다. 그러나 결합된 이벤트의 동작은 그림 4의 경우와 그림 5의 경우로 나누어질 수 있다.



[그림 4] a가 앞서는 엣지



[그림 5] b가 앞서는 엣지

(1) End 상승 엣지

$\uparrow(a \bullet b)$는 그림 4에서 a가 on되어 있는 상태에서 b의 펄스가 상승 엣지되는 신호가 되어 $\uparrow b \bullet a$ 로 되고. 그림 5에서 b가 on되어 있는 상태에서 a의 펄스가 상승 엣지되는 신호가 되어 $\uparrow a \bullet b$가 된다. 따라서 이 두 경우가 각각의 경우 발생할 수 있기 때문에 $\uparrow(a \bullet b)$는 그림 4 또는 그림 5의 경우가 식(1)과 같이 된다.

$$\uparrow(a \bullet b) = \uparrow a \bullet b + \uparrow b \bullet a \qquad (1)$$

(2) End 하강 엣지

$\downarrow(a \bullet b)$는 그림 4에서 a가 on되어 있는 상태에서 b의 펄스가 하강 엣지되는 신호가 되어 $\downarrow b \bullet a$ 가 되고. 그림 5에서 b가 on되어 있는 상태에서 a의 펄스가 하강 엣지되는 신호가 되어 $\downarrow a \bullet b$가 된다. 이 두 경우가 각각의 경우 발생할 수 있기 때문에 $\downarrow(a \bullet b)$는 그림 4 또는 그림 5의 경우가 식(2)와 같이 된다.

$$\downarrow(a \bullet b) = \downarrow a \bullet b + \downarrow b \bullet a \qquad (2)$$

(3) Or 상승 엣지

$\uparrow(a+b)$는 그림 4에서 b가 off되어 있는 상태에서는 b의 상태가 반전되고 a의 펄스가 상승 엣지되는 신호가 되어 $\uparrow a \bullet \overline{b}$ 가 되고. 그림 5에서 a가 off되어 있는 상태에서 a의 상태가 반전되고 b의 펄스가 상승 엣지되는 신호가 되어 $\uparrow b \bullet \overline{a}$가 된다. 이 두 경우가 각각의 경우 발생할 수 있기 때문에 $\uparrow(a+b)$는 그림 4 또는 그림 5의 경우가 식 (3)과 같이 된다.

$$\uparrow(a+b) = \uparrow a \bullet \overline{b} + \uparrow b \bullet \overline{a} \qquad (3)$$

(4) Or 하강 엣지

$\downarrow(a+b)$는 그림 4에서 a가 off되어 있는 상태에서는 a의 상태가 반전되고 b의 펄스가 하강 엣지되는 신호가 되어 $\downarrow b \bullet \overline{a}$ 가 되고. 그림 5에서 b가 off되어 있는 상태에서 b의 상태가 반전되고 a의 펄스가 하강 엣지되는 신호가 되어 $\downarrow a \bullet \overline{b}$가 된다. 이 두 경우가 각각의 경우 발생할 수 있기 때문에 $\downarrow(a+b)$는 그림 4 또는 그림 5의 경우가 식 (4)와 같이 된다.

$$\downarrow(a+b) = \downarrow a \bullet \overline{b} + \downarrow b \bullet \overline{a} \qquad (4)$$

3. 이벤트의 부울대수

2절에서 기술한 이벤트에 대해 식(1)에서 식(4)까지를 부울 대수식으로 표현하면 식(5)에서 식(12)와 같다.

$$\uparrow(a \cdot b) = \uparrow a \cdot b + \uparrow b \cdot a = \uparrow \prod_{i=1}^{n} u_i \quad (5)$$

$$\downarrow(a \cdot b) = \downarrow a \cdot b + \downarrow b \cdot a = \downarrow \prod_{i=1}^{n} u_i \quad (6)$$

$$\uparrow(a+b) = \uparrow a \cdot \bar{b} + \uparrow b \cdot \bar{a} = \uparrow \sum_{i=1}^{n} u_i \quad (7)$$

$$\downarrow(a+b) = \downarrow a \cdot \bar{b} + \downarrow b \cdot \bar{a} = \downarrow \sum_{i=1}^{n} u_i \quad (8)$$

4. 결론

본 논문에서는 SFC로 프로그래밍할 때 외부 이벤트에 대한 상승 엣지동작과 하강 엣지동작에 대해 부울대수로 어떻게 표현하는지를 나타냈다. 이러한 엣지동작에 대해 이벤트를 표현함으로써 래더다이어그램에서 사용되는 외부입력(P, N)에 대해 SFC에서 완벽하게 대응할 수 있게 된다. 현대의 공정제어가 복잡할수록 SFC 프로그램의 가치를 인정하게 될 것이고, SFC 프로그래밍시 트랜지션과 외부 이벤트의 완벽한 처리가 완벽한 공정제어 설계를 가능하게 할 수 있게 되었다.

참고문헌

[1] R.W.Lewis, "Programming Industrial Control Systems Using IEC1131-3", The Institution of Electrical Engineers, 1992.

[2] M. Zhou and E Twiss, "Design of Industrial automated systems via relay ladder logic programming and Petrinets", IEEE Trans on Systems, Man and Cybernetics -part C ; Applications and Reviews, Vol 28, No 1, pp137- 150, 1998.

[3] Jeong-Bong You, "Improvement Implementation of Interlock Using Management Step Described by SFC", 조명·전기 설비학회 논문지, Vol. 19, No. 3, May 2005.

[4] Jean-Jacques LESAGE, Jean-Marc ROUSSEL and Christophe THIERRY, "A theory of Binary Signal", IMACS-IEEE Multiconference on Computational, Engineering in Systems Appilications(CESA '96), Lille, France, July 9-12,1996