

효율적 코드변환 알고리즘에 기반한 PLC의 체계적 설계

Systematic Design of Programmable Logic Controllers Based on Efficient Code Conversion Algorithm

차 종 호, 조 광 현

(Jong-Ho Cha and Kwang-Hyun Cho)

Abstract : The ladder diagram (LD) for programmable logic controllers (PLCs) are responsible for much important roles in advanced industrial automation. As automated systems become more complex, the design procedures of the systems become more difficult as well. Hence, the design automation issues based on discrete event models (DEMs) are receiving more attention. One of the popular ways of tackling these problems is employing Petri nets. In this paper, we use the modified automation Petri net (MAPN) to model the manufacturing system and the modified token passing logic (MTPL) method to convert the Petri net model into the LD. Especially, in order to design a supervisory controller for PLCs, this paper proposes the efficient code conversion (ECC) algorithm based on the MAPN and the MTPL. Finally, an example of the manufacturing systems is provided to illustrate the proposed ECC algorithm.

Keywords : manufacturing systems, PLC, Petri net, LD program, MAPN, MTPL, ECC algorithm

I. 서론

PLC(programmable logic controller)에 기반한 플랜트 제어를 하기 위해서는 엔지니어의 숙련된 경험에 의존하는 LD (ladder diagram) 언어를 이용한 프로그램 작성을 필요로 한다. 이 때문에 LD 언어를 사용한 프로그램 작성은 체계적이고 분석적인 면에서 부족한 점이 있다[1]. 따라서 보다 체계적으로 플랜트를 모델링하고 분석하기 위해서 페트리네트(Petri net)와 같은 그래픽 특성이 우수한 도구를 사용한 연구가 많이 이루어져 왔다[2][3]. 특히 Grafset과 같은 페트리네트를 이용한 제어툴(control tool)은 LD 등의 기존의 프로그램에 있어서 이러한 단점을 보완하여 플랜트를 제어하는 것을 가능하게 한다[4]. 하지만 페트리네트를 이용한 플랜트 제어 연구는 실제 적용에 있어서 장치제어와 차이를 보이기 때문에 또 다른 문제에 봉착했고 이를 해결하기 위해 시스템을 페트리네트로 모델링한 후, 플랜트를 제어하는 것이 가능하도록 LD로 변환하는 연구가 최근에 와서 이루어지고 있다. 특히 Uzam은 일반적인 페트리네트(ordinary Petri net)가 제어기의 센서 입력신호와 액츄에이터 동작신호를 다루어 줄 수 없었기 때문에 이 문제를 해결하기 위해 자동 페트리네트(automation Petri net, 이하 APN)알고리즘을 제안하였다[3]. APN 알고리즘은 페트리네트 모델을 LD로 변환하는 방법인 토큰전달(token passing logic, 이하 TPL) 방법과 관리제어(supervisory control)가 가능하도록 하는 금지아크(inhibitor arc) 방법을 통해서 플랜트를 체계적으로 모델링하고 분석하기 쉽게 하면서 LD 언어를 제어프로그램으로 하는 PLC에도 적용이 용이하도록 하였다[5]. 하지만 APN

알고리즘은 실제 사용되고 있는 PLC 제어프로그램에 비해 프로그램의 라인 수와 용량이 큰 결과물을 가진다. 따라서 실제 현장에서 APN 알고리즘에 의해 생성된 제어프로그램을 사용할 경우 플랜트의 제어장치 설치비용이 증가될 우려가 있다. 본 논문은 APN 알고리즘이 가지는 이러한 문제점을 해결하고 실제 현장에서 사용가능한 프로그램 생성을 목적으로 효율적 코드변환(efficient code conversion, 이하 ECC) 알고리즘을 제안한다. ECC 알고리즘은 APN 알고리즘에 비해 제어기를 중심으로 시스템을 모델링하고 제어기의 동작을 중심으로 LD 프로그램을 생성하기 때문에 산업 현장에서 사용되고 있는 실제 LD 프로그램과 유사한 형태의 결과물을 가진다. 또한, 최근 시스템은 확장성과 유연성을 목적으로 객체지향적 특성을 가지도록 구현되고 있는데 [6] ECC 알고리즘에 의해 모델링된 시스템의 제어기는 시스템을 구성하는 각각의 요소로서 의미를 가지므로 이러한 객체지향적 특성을 가지게 되는 장점이 있다.

본 논문의 구성은 다음과 같다: 2절에서는 플랜트를 모델링하는 방법들에 관해서 설명한다. 특히, 2.1절에서는 일반적인 페트리네트와 APN을 소개하고, 2.2절에서는 APN 알고리즘과 TPL 방법을 소개하며 이들의 문제점을 해결하기 위한 MAPN(modified APN)과 MTPL(modified TPL)을 새로이 도입한다. 3절에서는 이들을 토대로 ECC 알고리즘을 제안하고 그 특징에 대해서 언급한다. 4절에서는 실제 현장에서 쓰이고 있는 생산시스템에 ECC 알고리즘을 모의 적용시킨 후 그 결과와 APN 알고리즘의 결과를 비교해 본다. 마지막으로 5절에서는 결론을 맺고 향후 연구과제를 논한다.

II. 플랜트 모델링

1. 페트리네트

페트리네트는 이산사건시스템(discrete event system)을 대상으로 시각적 표현이 가능한 그래프형의 모델링 기법으

접수일자 : 2001. 4. 9., 수정완료 : 2001. 8. 30.

차종호, 조광현 : 울산대학교 전기전자정보시스템공학부(cjh@sys.ulsan.ac.kr, ckh@mail.ulsan.ac.kr)

※ 본 연구는 2001년도 한국학술진흥재단의 지원(KRF-2001-041-E00279)에 의하여 수행되었습니다.

로서 대상시스템의 논리적 특성표현이 용이한 모델링 도구이다. 일반적인 페트리네트는 플레이스(place)와 트랜지션(transition), 아크(arc)로 구성되어 있으며 플레이스 상의 토큰 이동으로 상태 변화를 표시한다. 페트리네트는 일반적으로 다음 5개의 원소로 구성되어 있다:

$N = \{ P, T, I, O, M_0 \}$, 여기서

$P = \{ P_1, P_2, \dots, P_n \}$: 유한한 n개의 플레이스의 집합,

$T = \{ T_1, T_2, \dots, T_m \}$: 유한한 m개의 트랜지션의 집합,

$I \subseteq (P \times T)$: 플레이스로부터 트랜지션으로 향하는 아크 집합을 정의하는 입력함수,

$O \subseteq (T \times P)$: 트랜지션으로부터 플레이스로 향하는 아크 집합을 정의하는 출력함수, 그리고

M_0 : 초기마킹(initial marking)을 각각 의미한다. 페트리네트 모델의 실행 규칙은 다음과 같이 두 가지가 있다:

1) 활성화 규칙(enabling rule): 트랜지션의 입력 플레이스에 토큰이 있고 아크의 토큰 숫자보다 입력 플레이스의 토큰 숫자가 같거나 크면 트랜지션은 활성화된다.

2) 점화 규칙(firing rule): 트랜지션이 활성화되면 트랜지션이 점화되어 입력 플레이스의 토큰이 1개 감소되고 출력 플레이스에 토큰이 1개 증가한다.

2 MAPN & MTPL

APN 알고리즘은 APN을 사용해서 시스템을 설계한다. APN은 일반적인 페트리네트가 제어기의 센서 입력신호와 액추에이터 동작신호를 다루어 줄 수 없었기 때문에 이를 보완한 형태의 페트리네트이다. APN은 다음과 같이 9개의 구성요소로 이루어져 있다:

$$APN = \{ P, T, Pre, Post, In, En, \chi, Q, M_0 \} [4].$$

APN 알고리즘은 금지아크방법, TPL과 함께 시스템을 페트리네트로 모델링하여 PLC의 제어프로그램을 생성한다. 금지아크방법은 실제 현장에서 발생가능한 문제들을 방지하는 설계방법으로 시스템에서 발생가능한 상태를 모두 표시해 준 후 오류상태를 제거해서 오류상태에 빠지지 않도록 제어한다. 이 방법은 실제 시스템의 하위장치 부분에 해당하는 액추에이터와 생산물 이동라인을 연결시켜 줌으로써 전체 시스템을 원활히 제어한다. TPL 방법은 페트리네트로 모델링된 시스템 내의 토큰 이동을 실제 시스템의 자원 혹은 액추에이터 동작으로 표현하는 방법이다. APN 알고리즘에 의해 생성된 PLC 제어프로그램은 시스템 내의 모든 발생가능한 상태를 제어하도록 설계되어 있다. 하지만 이 때문에 다음과 같은 문제점을 가진다. 첫째, 생성된 프로그램의 라인 수가 실제 현장에서 쓰이는 프로그램의 라인 수에 비해 많다. 둘째, 페트리네트로 모델링된 시스템 설계가 복잡하다. 생성된 프로그램이 실제 현장에서 사용되는 프로그램에 비해 라인이 많다면 제어기의 점접수가 늘어나고 프로그램의 용량도 증가시키는 결과를 가져온다. 이러한 문제점을 해결하기 위해서 본 논문에서는 ECC 알고리즘을 제안한다. ECC 알고리즘은 APN 알고리즘에서 사용되는 APN이나 TPL을 수정한 MAPN과 MTPL을 이용한다.

MAPN의 기본 구성요소는 APN과 동일하다[3]. MAPN에서 시스템은 생산물 이동라인과 액추에이터 동작라인으

로 구분되어 표현된다. 생산물 이동라인은 라인 내부에 존재하는 토큰 수에 따라 다르게 모델링된다. 예를 들어 라인 내부에 토큰이 3개 존재하는 생산물 이동라인은 토큰을 공급하는 소스 플레이스 내에 토큰을 3개 표시한다. 그리고 토큰이 플레이스당 하나씩만 존재할 수 있도록 (즉, 토큰이 하나씩만 전달될 수 있도록) 각 출력 플레이스에 토큰이 존재하지 않는다는 신호를 트랜지션에 보내어 출력 플레이스에 2개 이상의 토큰이 존재하지 못하도록 한다. MAPN을 이용한 생산물 이동라인의 기본 설계 모형은 그림 1의 TYPE F1과 TYPE F2와 같다. MAPN에서 액추에이터 동작라인은 on, off 상태를 나타내는 상태 플레이스에 의해 타이머, 더블실린더, 싱글실린더에 따라 다른 액추에이터 모형을 가지도록 표현된다. 이는 각 액추에이터를 다르게 설계함으로써 각 액추에이터의 동작 특성을 나타낼 수 있도록 한다. 특히 액추에이터 동작라인의 설계에 있어서 자기유지가 필요한 동작에 대해서는 다른 플레이스와 달리 자기유지 플레이스로 표시해준다. 자기유지 플레이스는 보통의 플레이스가 트랜지션이 활성화된 후에 즉시 토큰을 전달하는데 반해 삭제 트랜지션이 활성화되기 전까지는 토큰을 플레이스 내에 유지한다. 자기유지 플레이스는 그림 1에서 2개의 원이 겹쳐 있는 플레이스로 표시한다. 삭제 트랜지션은 보통의 트랜지션과는 다르게 자기유지 플레이스의 토큰을 제거시키면서 토큰을 전달하는 역할을 하며 그림 1에서 회색사각형 모양으로 표시되어 있다. 예를 들어 싱글 실린더가 사용될 액추에이터 동작라인 모델은 동작 1개만 자기유지를 걸어주면 되기 때문에 자기유지 플레이스를 1개만 사용하고 더블실린더의 경우에는 모든 동작에 대해 자기유지를 걸어주어야 하므로 2개의 자기유지 플레이스를 사용하며, 타이머의 경우에는 시간지연 트랜지션을 사용해서 토큰전달을 원하는 시간만큼 지연시킨 후 전달하도록 설계한다. 따라서 액추에이터 모형은 그림 1의 TYPE 1, TYPE 2, TYPE 3과 같이 주어진다.

MTPL은 TPL과 같이 토큰의 이동으로 시스템의 흐름을 표현한다. MTPL은 MAPN으로 표현된 액추에이터 모델과 생산물 이동라인모델을 LD로 변환하기 위한 방법이다. MTPL은 자기유지 플레이스가 하나 있는 경우 하나의 출력 접점을 가지도록 이전 플레이스와 다음 플레이스, 이들 중간에 존재하는 트랜지션을 묶어서 한 라인으로 표시한다. MTPL은 액추에이터의 출력접점을 자기유지 플레이스 수에

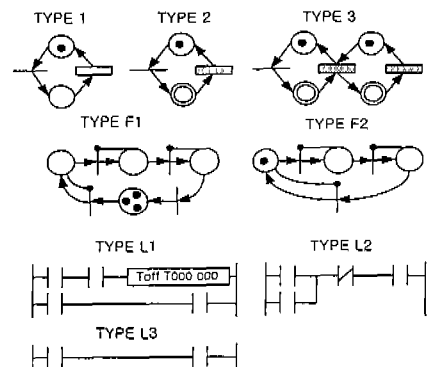


그림 1. ECC 알고리즘의 타입변수에 대한 정의.
Fig. 1. Definition of the type variables in ECC algorithm.

따라 다르게 표현한다. 그러므로 자기유지 플레이스가 있는 경우에는 그림 1의 TYPE L2와 같이 변환하고 자기유지 플레이스가 없는 경우에는 TYPE L3와 같이 변환하며, 액츄에이터가 타이머인 경우에는 TYPE L1과 같이 변환한다.

III. ECC 알고리즘

ECC 알고리즘은 2절에서 설명한 APN 알고리즘의 문제점을 해결하는데 목적을 두고 있다. ECC 알고리즘을 이용한 시스템 모델링 및 LD 프로그램으로의 변환은 다음과 같은 단계로 구성되어 있다:

- 단계 1. 플랜트를 MAPN으로 모델링한다.
 - 단계 2. 도달가능한 상태표(table of reachable states)를 만든다.
 - 단계 3. 금지 접점(inhibitor contact), 동작 접점(action contact)을 찾아서 이에 따른 관리제어기를 구현한다.
 - 단계 4. 관리제어기와 초기 플랜트 모델을 조합하여 전체 시스템 모델을 만든다.
 - 단계 5. MTPL 방법을 사용하여 설계된 시스템 모델을 LD로 변환한다.
- 그림 2는 ECC 알고리즘의 pseudo 코드를 나타낸 것이다.

```

Begin
if( SM=2 in Actuator Route)
    use type 3
else
    if(TM ≠ 0)
        use type 2
    else
        use type 1
if( TN >1 in Product Route)
    use type F1
else
    use type F2
Ri={ s0, ..., sn-1, p0, ..., pn-1, a0, ..., an-1, t0, ..., tn-1}
RS={ R1, R2, R3, ..., Ri} where i=0, 1, ..., n-1
for(j=0 ; j=i ; j++)
    if( Rj ≠ Rj+1)
        for(k=0 ; k= δ ; k++)
            find s, p, t, a having relation with ak
        for(m=0 ; m= θ ; m++)
            find s, p, t, a having relation with Om
construct{ s0, ..., sx, p0, ..., px, t0, ..., tx, a0, ..., ax}
replace{ s0, ..., sx, p0, ..., px, t0, ..., tx, a0, ..., ax}
with I={ I0, I1, ..., I(x+1)×4}
for(j=0; i=j; j++)
    if( Ij=0)
        b contact
    else if( Ij=1)
        a contact
if( SM>0)
    use type L1
else
    if(TM >1)
        use type L2
    else
        use type L3
End
    
```

그림 2. 효율적 코드변환 알고리즘의 pseudo 코드.
Fig. 2. Pseudo code of the ECC algorithm.

표 1. 그림 2에 대한 변수명 정의.

Table 1. Definition of the variable in Fig. 2.

변수명	정 의
SM	자기유지 플레이스의 수
TN	생산물 이동라인에 있는 내부 토큰의 수
TM	타이머의 수
p _g	생산물 이동라인에 있는 g번째 플레이스
R _i	도달가능한 i번째 상태그룹(reachable state group)
t _g	액츄에이터 동작라인에 있는 g번째 타이머
a _g	g번째 액츄에이터
s _g	g번째 스위치
θ	토큰 전달 명령
δ	액츄에이터의 수
O _m	사양에 명시된 m번째 기본동작
a _k	시스템 내에 필요한 k번째 액츄에이터
RS	Table of reachable states

그림 2에 사용되는 각각의 변수들은 표 1에 정의되어 있다. 시스템 내의 액츄에이터는 주어진 사양(specification)에 의해 기본 동작들을 명시할 수 있고 이러한 기본동작들을 수행하기 위해서는 필요한 접점 혹은 신호가 존재한다. 본 논문에서는 이 필요한 접점 혹은 신호를 찾기 위해 오류상태의 발생을 막는 제어방법을 사용한다. 이때 사용되는 오류상태란 다음과 같이 정의된다:

- 1) 실제 생산물의 이동에 위배되는 액츄에이터의 동작 혹은 센서의 입력,
- 2) 액츄에이터의 동작 후에 발생하는 잘못된 생산물의 이동.

따라서 그림 2에 사용된 'relation'은 이 오류상태에 도달하는 것을 방지하는 접점 혹은 액츄에이터의 동작 신호를 의미한다. 여기서 'relation'은 ECC 알고리즘에 정의된 상태표를 통해 도달가능한 상태그룹을 정의한 후 상태그룹에서 값 변화가 없는 접점을 선택한다. 하지만 상태표에서 접점값의 변화가 없을지라도 상태그룹의 액츄에이터 동작이나 생산물 이동라인과 직접적 관련이 없는 부분은 제외시킨다. 이것은 상태그룹에 표현된 동작 내에 있는 트랜지션을 중심으로 관련 접점을 선택하기 때문이다. 예를 들어 시동 접점은 초기 토큰 주입에만 관여할 뿐 이를 제외한 다른 상태그룹에는 영향을 미치지 않는다. 그림 2의 pseudo 코드는 이와 같이 MAPN과 MTPL에 의해 표현된 타입변수를 선택하고 'relation'을 통해 접점을 구하는 과정을 나타낸다.

IV. 시스템 모의 적용 사례

본 절에서는 앞에서 제시한 ECC 알고리즘을 생산 시스템에 모의 적용하여 각 단계별로 기술하고 APN 알고리즘의 결과와 비교한다.

그림 3의 생산시스템은 페그(peg)와 링(ring)이 혼합되어 입력라인으로 유입되면 분류영역에서 페그와 링을 구분하

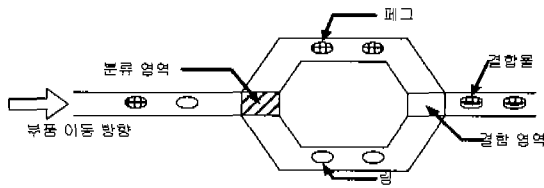


그림 3. 생산시스템 예의 구성도.
Fig. 3. Configuration of the example manufacturing system.

여 이동시키고 결합영역에서 각각의 폐그와 링을 결합시켜 원하는 물품을 만드는 시스템이다. 이때 분류영역과 결합영역에는 각각 분류 혹은 결합동작을 수행하는 솔레노이드(solenoid)가 존재한다.

그림 3의 생산시스템을 ECC 알고리즘에 따라 다음과 같이 페트리네트로 모델링하고 LD로 변환할 수 있다:

단계 1: 플랜트를 MAPN으로 모델링한다. 그림 3의 생산물 이동라인은 3개의 플레이스와 3개의 내부 토큰으로 구성할 수 있다. 액츄에이터 동작라인은 타이머로 구동되는 2개의 솔레노이드로 구성되어 진다. 따라서 그림 4와 같이 생산시스템의 기초모델을 만들 수 있다.

단계 2: 도달가능한 상태표를 만든다. 그림 3의 경우는 내부 토큰이 3개이므로 실제로 주어진 사양을 위해 중요한

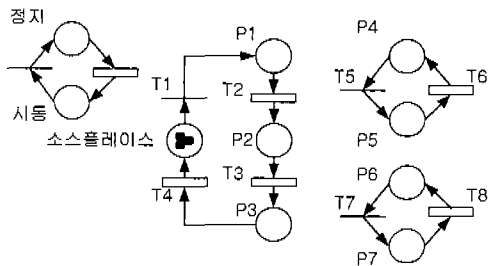


그림 4. MAPN에 의해 생성된 생산시스템의 기초 모델.
Fig. 4. Basic model of the manufacturing system via MAPN.

표 2. 그림 3의 도달가능한 상태표.
Table 2. Table of reachable states in Fig. 3.

동작	접점	P1	P2	P3	시동	A	B	L1	L2
부품 주입 ①	0	0	0	1	0	0	0	0	0
	0	1	0	1	0	0	0	0	0
	0	1	1	1	0	0	0	0	0
분류 ②	1	0	0	1	0	0	0	0	0
	1	0	1	1	0	0	0	0	0
	1	0	1	1	0	0	0	0	0
결합 ③	1	1	0	1	0	0	0	0	0
	1	1	0	1	0	0	0	0	0
	0	1	0	1	0	0	0	0	0
결합물 출력 ④	1	1	1	1	0	0	0	0	0
	0	1	1	1	0	0	0	0	0
	0	0	1	1	0	0	0	0	0

동작에 관해서만 도달가능한 상태표를 만들면 표 2와 같다. 이때 표 2는 그림 4와 주어진 사양으로부터 생산물 이동라인의 P1, P2, P3 접점, 액츄에이터 동작라인의 A, B 접점, 시동과 L1, L2 접점의 상태를 각 동작별로 1(on)과 0(off)으로 표시해준다. A, B는 액츄에이터 1의 P4와 P5, 액츄에이터 2의 P6와 P7의 접점을 각 하나의 접점이 on, off 되는 것으로 묶어서 표시해 준 것이다. 표 2에서의 동작은 생산시스템의 사양으로부터 얻을 수 있다.

단계 3: 표 2가 만들어지면 이 표로부터 ECC 알고리즘에 따라 액츄에이터 동작라인과 생산물 이동라인의 모델사이의 'relation'를 찾을 수 있다. 이 'relation'은 금지접점과 동작접점으로 이루어지며, 금지접점은 동작수행을 금지하는 역할을 하고 동작접점은 동작을 가능하게 하는 역할을 한다. 각 동작에 따른 금지접점과 동작접점은 다음과 같다. 부품주입은 ECC 알고리즘에 따라 P1 플레이스에 토큰이 존재하지 않아야만 동작한다. 분류는 P2플레이스에 토큰이 존재하지 않고 액츄에이터의 1과 2의 접점이 off 상태에 있어야만 동작한다. 결합은 토큰이 P2 플레이스에는 존재하고 P3 플레이스에는 존재하지 않으며, 액츄에이터 1과 2의 접점이 off 상태에 있어야만 동작이 가능하다. 결합물 출력은 P3 플레이스에 토큰이 존재하면 동작한다. 따라서 다음과 같이 접점을 표시할 수 있다. ①에서 $\overline{sP1}$, ②에서 $\overline{sP3}$, $sP2$, \overline{A} , \overline{B} , ③에서 $\overline{sP2}$, $sP1$, \overline{A} , \overline{B} , ④에서 $sP3$ 의 접점을 구할 수 있다. 이러한 접점들을 바탕으로 관리제어기를 구성한다. 이때 sP_n 은 P_n 에 토큰이 존재한다는 신호를 나타내고, $\overline{sP_n}$ 은 P_n 에 토큰이 존재하지 않음을 의미한다.

단계 4: 단계 3에서 구성한 관리제어기를 포함시켜 전체 시스템을 모델링하면 그림 5와 같다. 이때 단계 3에서 구한 접점을 각 트랜지션에 연결하여 전체 시스템을 모델링한다. 예를 들어 부품주입이란 동작을 수행하기 위해서는 P1 플레이스에 토큰이 존재하지 않도록 P1으로부터 T1으로 금지아크를 표시해 준다. 다른 동작들에 대해서도 이런 방법으로 접점들을 연결한다.

단계 5: 그림 5의 페트리네트모델을 MTPL 방법에 의해서 LD로 변환하면 그림 6과 같다.

한편, 위 생산시스템에 대해 APN 알고리즘을 적용시키면 그림 7의 LD가 생성된다. 그림 6의 LD와 그림 7의 LD를 비교 분석한 결과는 표 3에 정리되어 있다.

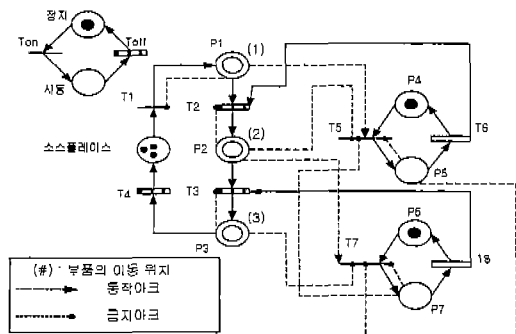


그림 5. 생산시스템의 페트리네트 모델.
Fig. 5. Petri net model of the manufacturing system.

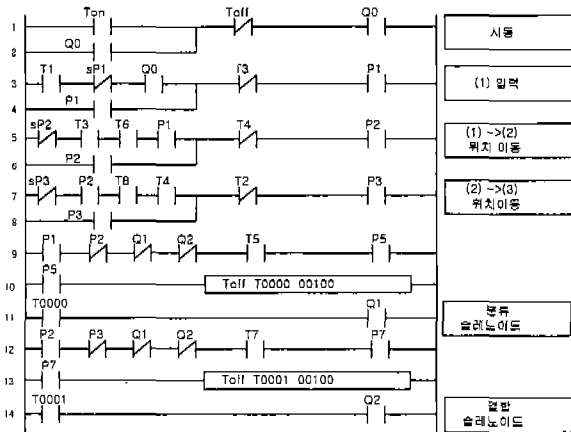


그림 6. ECC 알고리즘에 의해 생성된 생산시스템의 LD.
Fig. 6. LD of the example manufacturing system via the ECC algorithm.

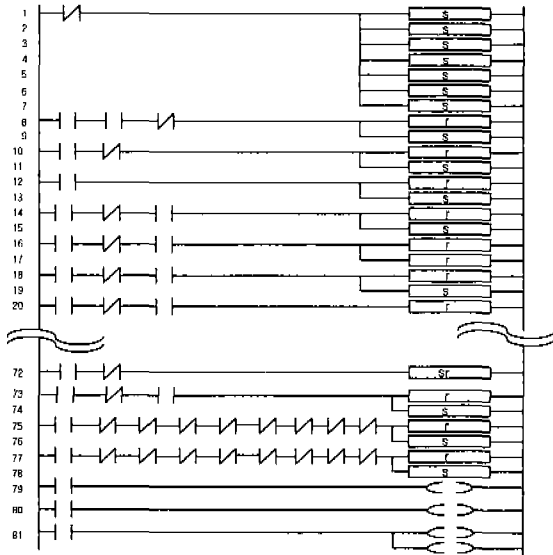


그림 7. APN 알고리즘에 의해 생성된 생산시스템의 LD.
Fig. 7. LD of the example manufacturing system via the APN algorithm.

표 3. APN과 ECC 알고리즘에 의해 각각 생성된 LD의 비교.

Table 3. Comparison of LDs by the APN and the ECC algorithm.

비교항목 알고리즘	생성된 LD의 라인 수	LD의 용량 [byte]	LD 프로그램의 스캔 타임 [μ s]
ECC 알고리즘	14	44	70
APN 알고리즘	81	113	405

표 3은 APN 알고리즘과 ECC 알고리즘에 의해 생성된 LD의 라인 수와 용량, 스캔 타임(scan time)을 기준으로 비교한 것이다. 특히 스캔 타임은 PLC에서 한 라인을 읽어들이는데 걸리는 시간으로 본 논문에서는 한 라인당 5 μ s로 스캔 타임을 설정했다. 표 3에서 보인 것과 같이 ECC 알고리즘에 의해 생성된 LD가 APN 알고리즘에 의해 생성된 LD에 비해

라인 수와 용량, 스캔 타임 모든 항목에 있어서 적은 결과를 가지므로 더 효율적임을 알 수 있다. 이와 같이 APN 알고리즘과 ECC 알고리즘에 의한 결과가 차이가 나는 이유는 APN 알고리즘은 시스템 전체의 토큰 흐름을 중심으로 모델링하는데 반해 ECC 알고리즘은 제어기 중심으로 시스템을 모델링하기 때문이다. 요컨대 ECC 알고리즘의 장점은 APN 알고리즘에 비해 보다 체계적이며 결과적으로 생성된 LD의 라인 수가 적기 때문에 점점의 낭비와 프로그램의 대용량화를 막을 수 있다. 이런 ECC 알고리즘의 장점은 실제 현장에서 사용되고 있는 LD 프로그램과의 차이를 줄여 적용 면에서 효율성을 증대시킬 것으로 기대된다.

V. 결론

페트리네트를 통해 시스템을 모델링하는 것은 LD로 시스템을 직접 기술하는 것보다 분석적인 면에서 상당히 우수하다. 하지만 페트리네트로 모델링한 시스템을 직접 PLC에 적용시켜 사용할 수는 없었다. 이에 APN 알고리즘은 시스템을 페트리네트로 모델링한 후 다시 LD로 바꾸어 PLC에 적용시킬 수 있도록 하였지만 본 논문에서 언급한 바와 같이 몇 가지 문제점을 가지고 있었다. 본 논문에서는 이러한 APN 알고리즘의 문제점을 해결하기 위하여 ECC 알고리즘을 제안하였고, APN 알고리즘과의 비교분석결과 향상된 LD를 생성할 수 있음을 확인하였다. 향후 연구과제로는 실제 플랜트에서 사용되는 보다 많은 장치들을 포함한 플랜트 모델링과 LD 프로그램으로의 변환에 관한 연구 등이 있다.

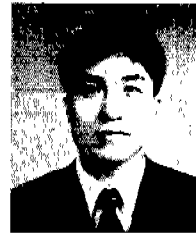
참고문헌

- [1] 이기범, 이진수, "이산사건 시스템 모델링과 구동 PLC 모델링을 결합한 통합 페트리 네트," 제어·자동화·시스템 공학회 논문지, 제4권, 제5호, pp. 29-40, 1999.
- [2] M. C. Zhou and E. Twiss, "Design of industrial automated systems via relay ladder logic programming and Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, pp. 137-150, Feb., 1998.
- [3] M. Uzam and A. H. Jones, "Discrete event control system design using automation Petri nets and their ladder diagram implementation," *International Journal of Advanced manufacturing Technology*, vol. 14, no. 10, pp. 716-728, Oct., 1998.
- [4] V. Carre-Menetrier, C. Ndjab, F. Gellot, and J. Zaytoon, "A CASE tool for the synthesis of optimal control implementation of Grafset," *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, Reims, France, vol. 1, pp. 796-801, Oct., 1999.
- [5] M. Uzam, A. H. Jones, and I. Yücel, "Using a Petri-net-based approach for the real-time supervisory control of experimental manufacturing system," *International Journal of Advanced manufacturing Technology*, vol. 16, pp. 498-515, July, 2000.
- [6] R. J. Machado, J. M. Fernandes, and A. J. Proenca, "Specification of industrial digital controllers with object-oriented Petri nets," *Proc. of IEEE International Symposium on Industrial Electronics*, Guimaraes, Portugal, vol. 1, pp. 78-83, July, 1997.



차 종 호

2001년 울산대학교 전자공학과 졸업. 2001년~현재 울산대학교 전기전자정보시스템공학부 석사과정. 관심분야는 이산사건시스템의 관리제어, 유연생산시스템의 설계 및 분석, 생산시스템의 자동화 등.



조 광 현

1993년 한국과학기술원 전기 및 전자공학과 졸업. 동대학원 석사(1995). 동대학원 박사(1998). 1998~1999. 동대학원 위촉연구원 및 연수연구원. 1999. 3.~현재 울산대학교 전기전자정보시스템공학부 조교수. 관심분야는 이산사건시스템의 해석 및 관리제어, 생산시스템 자동화(반도체 생산시스템 자동화), 통신망 분석 및 제어, 광통신망 시스템(스위칭 및 라우팅), 바이오정보공학 응용 등.