

GRAF CET에 의한 단일 기관 컴퓨터의 순차제어용 프로그래밍 도구의 개발

국금환*, 김민환**, 최기봉*

* 한국기계연구소 로봇공학실

** 부산대학교 컴퓨터공학과

Development of GRAFCET Programming Tool for Sequential Control of One-Board Computer

Keum-Hoan Kuk*, Min-Hwan Kim**, Kee-Bong Choi*

* Korea Institute of Machinery & Metals

** Pusan National University

ABSTRACT

GRAF CET is the program language for a sequential control and utilized in the description of sequence of a process.

This study is to implement GRAFCET editor and compiler for personal computer. The GRAFCET editor module is developed to use as menu-driven method based on the graphic screen and the graphic symbol. So, users can use it interactively. The GRAFCET compiler module analyzes the already drawn GRAFCET diagram, then translates it to Z80 assembly code. After generating Z80 machine code from assembly code, we can send the outputed machine code file to Z80 one-board computer through RS232C.

We have developed the GRAFCET programming tool to utilize under IBM PC/AT with the VGA monitor, and it may be contribute to low cost automation system.

1. 서론

CIM(Computer Integrated Manufacturing)으로 대표되는 생산공정의 자동화 및 정보통합화에 있어서 PLC(Programmable Logic Controller)는 컴퓨터와 더불어 정보의 흐름 및 제어분야에 많은 비중을 차지하고 있다. PLC는 순차제어를 위한 도구로 자동화 공정에 있어서 중요성이 날로 증가하고 있으며, 이를 위하여 보다 새롭고 사용이 용이한 도구의 개발이 절실히 요구되고 있다.

순차제어를 위한 도구로 자동화 공정에 있어서 중요성이 날로 증가하고 있으며, 이를 위하여 보다 새롭고 사용이 용이한 도구의 개발이 절실히 요구되고 있다.

순차제어 공정을 기술하기 위한 각종 프로그래밍 언어는 IEC(International Electro-Technical Commission)의 규격안에 따르면 그림 1과 같다. 이것은 크게 텍스트와 그래픽

표현 방식으로 분류된다. 그래픽 표현방식에는 현재 많이 사용되는 래더 다이어그램(Ladder Diagram) 및 흐름도의 형식으로 표현하는 SFC(Sequential Function Chart)등이 있다.

SFC는 IEC에서 GRAFCET(Graphe de Commande Etape-Transition)을 규격화 작업의 기초자료로 이용하여 채택한 PLC용 언어이다. GRAFCET은 순차적 수행용 프로그램으로서 실행순서와 처리내용을 시각적으로 표현할 수 있게 한 도구이며, 자동화 분야에서 제어기의 프로그램에 많이 이용되고 있다.

본 연구는 GRAFCET을 개인용 컴퓨터(PC)에서 구현하여 프로그램의 작성 즉 GRAFCET 다이어그램의 작성과 이것을 분석하여 Z80 단일 보드 컴퓨터용 Z80 어셈블러 언어의 프로그램 코드를 자동적으로 생성시킬 수 있는 프로그래밍 도구를 개발하는 것이다.

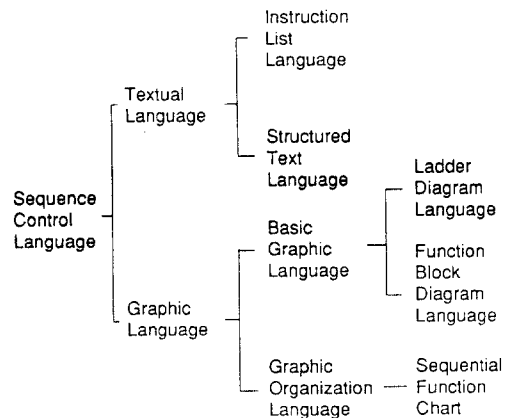


그림 1 순차제어 공정의 기술을 위한 프로그래밍 언어
Fig. 1 Programming languages for description of sequential control processes

2. 기존 GRAFCET의 분석

GRAFCET은 기본적으로 STEP, ACTION, TRANSITION, LINK로 구성된다. STEP은 정사각형으로 표시하며 동작의 단계를 구분하는 데 사용된다. ACTION은 직사각형으로 표시하며, STEP에서 실제로 행하여질 동작을 사각형 안에 기입한다. 임의의 STEP에서 복수의 ACTION은 복수개의 직사각형을 연결하여 나타낸다. TRANSITION은 십자형태로 표시하며, 바로 앞 STEP이 활성화 되었을 때 그 STEP에 상응하는 ACTION의 완료나 상태의 변화를 감지하여 다음 STEP으로 진행시키기 위한 전이조건 감지에 이용된다. LINK는 STEP과 TRANSITION을 연결하는 것으로 루프의 구성, 조건분기, 및 동시진행 등을 나타내는 데 사용된다.

그림 2는 GRAFCET으로 작성된 프로그램의 예를 보여준다. 먼저 초기화 STEP을 이중의 사각형으로 표시했으며, 초기화가 이루어지도록 직사각형의 ACTION에 초기화를 기입하였다. 초기화 STEP은 TRANSITION을 갖지 않았기 때문에 초기화가 종료되면 바로 STEP 1의 동작 A를 실행시킨다. 동작 A가 종료되면 조건 a가 만족되는 가를 조사한다. 조건이 만족되면 두줄의 수평선으로 표시된 동시진행에 따라 STEP 2와 STEP 3이 동시에 수행된다. 이때 STEP 3에서는 동작 C와 D가 모두 수행된다. 그 다음에 있는 두줄의 수평선은 동시진행을 합병시키는 과정으로 조건 b와 c가 모두 만족되었을 때만 그 다음 과정인 STEP 4를 수행시킬 수 있다. STEP 4의 동작 E가 수행된 다음에 조건 d가 만족되었을 때는 STEP 1으로 가서 루프를 형성하고, 만약 조건 e가 만족되었을 때는 STEP 5이하를 수행시킨다.

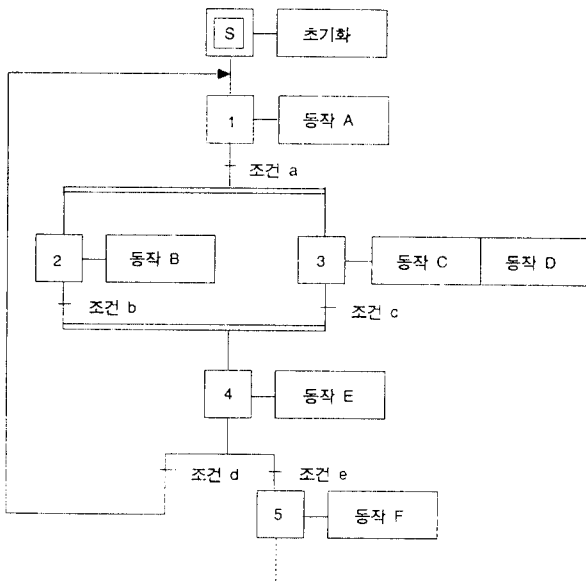


그림 2 GRAFCET으로 표현된 프로그램의 예
Fig. 2 The example of a program described as GRAFCET

3. KGPT(KIMM GRAFCET Programming Tool)의 구조

GRAFCET을 PC 상에서 작성 및 컴파일하기 위하여 PC의 그래픽 환경 하에서 사용할 수 있는 프로그래밍 도구를 개발하였으며, 이 소프트웨어의 명칭을 KGPT(KIMM GRAFCET Programming Tool)이라 명명하였다. 이것의 구조는 그림 3과 같다. 먼저 GRAFCET 에디터를 이용하여 GRAFCET 다이어그램을 작성한 후 GRAFCET 컴파일러를 이용하여 Z80 단일 보드 컴퓨터용 어셈블러 언어의 프로그램 코드를 생성시킨다. 생성된 Z80 어셈블러 코드는 기존의 Z80 용 어셈블러 컴파일러를 통하여 기계어 코드를 생성시킨 후 통신용 소프트웨어를 통하여 Z80 단일보드 컴퓨터에 송신한다.

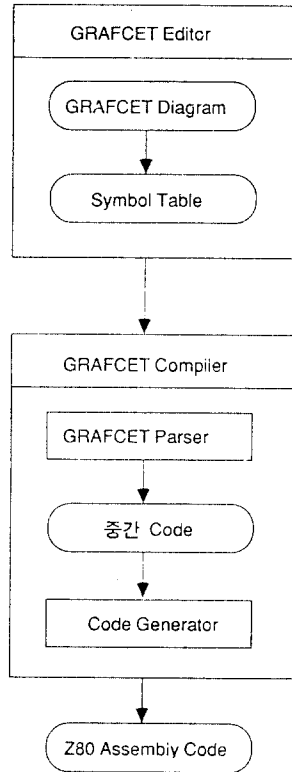


그림 3 KGPT의 구조
Fig. 3 The structure of KGPT

3.1 그래픽 심볼(graphic symbol)

GRAFCET 에디터로 GRAFCET 다이어그램을 용이하게 작성하기 위하여 그림 4와 같이 15개의 기본 심볼(symbol)들을 정의하였다. 이것들은 각각 다음과 같은 목적에 이용된다.

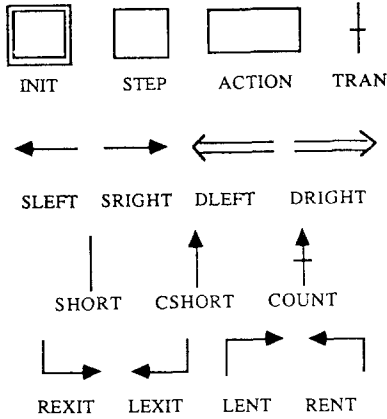


그림 4 GRAFCET 작성을 위해 정의된 그래픽 심볼
Fig. 4 The defined graphic symbols to edit GRAFCET diagram

- INIT : 초기화의 STEP에 이용된다.
- STEP, ACTION, TRAN : 각각 처리단계, 실제의 작업내용, 및 천이조건을 나타내는 데 이용된다.
- SLEFT, SRIGHT : 분기(branch)와 경로(path)를 나타내는 데 이용된다.
- DLEFT, DRIGHT : 동시진행 과정에서 상하의 과정을 연결 하는데 이용된다.
- CSHORT, COUNT : 루프(loop)를 형성하기 위해 위로 계속 연결하는 데 이용된다. 이때 COUNT는 루프의 계속 진행 조건을 나타내는 데 이용된다.
- LEXIT, REXIT : 루프의 진입을 나타내기 위한 곳에 이용된다.
- LENT, RENT : 루프 형성의 목적지를 나타내는 데 이용된다.

이들 심볼들 중 SLEFT, SRIGHT, DLEFT, 및 DRIGHT에는 화살표가 부착되어 있으나 이것은 단지 그 방향을 구분하는 것에 불과한 것으로 실제의 화면에는 나타나지 않는다.

3.2 그래픽 심볼의 자료구조

그래픽 심볼들을 인식하기 위하여 심볼을 노드(node)로 하여 이진 트리(binary tree)를 구성하는 자료구조를 그림 5와 같이 정의하였다. 이 자료구조에서 각 변수의 의미는 다음과 같다.

- TYPE : 심볼의 이름을 나타낸다.
- X, Y : 화면상에서 심볼의 기준점의 위치로서 화면의 절대좌표로 나타낸다.
- PAGE, WIDTH : 심볼이 화면에 나타내어 질 때의 구간 블록으로서 실제의 (PAGE x WIDTH)는 (30x10)으로 한정하였다.

- PRX1, PRY1, PRX2, PRY2 : 오른쪽 자노드(child node)와 왼쪽 자노드의 좌표를 나타내는 것으로 그림 6에는 각 심볼들의 기준점 및 자노드의 위치가 표시되어 있다.
- MESSAGE : 심볼에 기입되는 메시지를 저장하기 위한 것이다.
- LCHILD, RCHILD : 왼쪽과 오른쪽의 자노드를 가리키기 위한 포인터이다.

TYPE	
X	Y
PAGE	WIDTH
PRX1	PRY1
PRX2	PRY2
MESSAGE	
LCHILD	RCHILD

그림 5 심볼의 자료구조
Fig. 5 The data structure of the graphic symbol

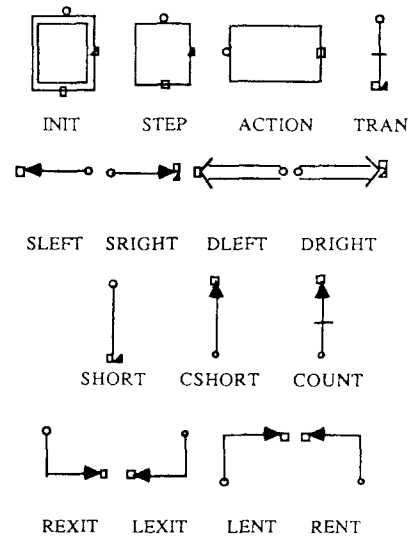


그림 6 심볼의 기준점 및 자노드의 위치
Fig. 6 The reference points of the symbols and the positions of the child nodes

3.3 그래픽 심볼들의 연결 제약조건

각 심볼들을 연결할 때에는 일반적인 컴퓨터 언어의 컴파일러 구성을 위한 문법과 같은 각 심볼들의 연결 문법이 존재해야 한다. 이를 위해 어떤 심볼은 하나의 자노드만을 가지고 있으며, 또 심볼간의 연결관계도 실제의 의미있는 연결만을 위하여 그림 7과같이 연결 관계도를 규정하였다. 이로써 GRAFCET 에디터로 GRAFCET 다이어그램을 작성할 때 연결 가능성이 없는 심볼들 간의 연결은 미리 배제할 수 있다.

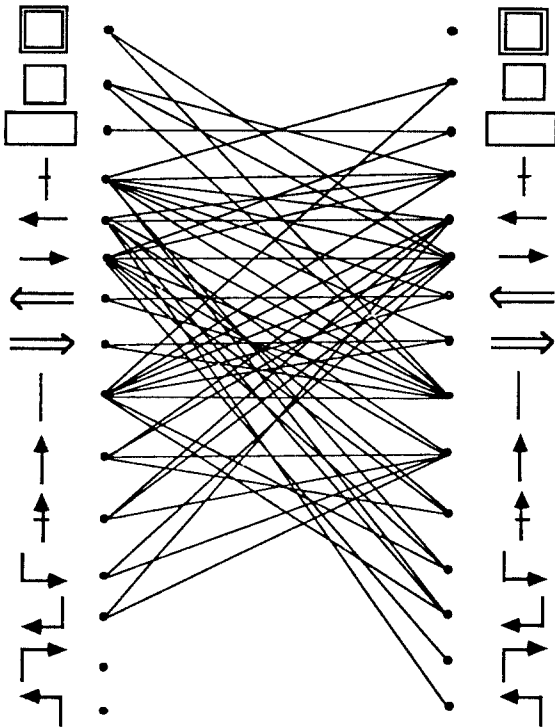


그림 7 심볼들 사이의 연결 가능성도

Fig. 7 The connecting relation diagram between the symbols

4. GRAFCET 에디터

GRAFCET 다이어그램을 그리기 위한 대화형 그래픽 에디터로써 이것의 전체적인 구성은 그림 8과 같다. 이 때 그래픽 에디터의 화면 구성은 그림 9와 같다. DRAW FIELD는 GRAFCET 다이어그램이 그려지는 부분으로 화면상의 크기를 한 페이지라 정의할 때 (30x10) 페이지의 분량을 스크롤(scroll)함에 의해 그림 수 있다. STATUS FIELD는 화면에 나타난 페이지를 나타낸다. SYMBOL FIELD는 심볼들을 그려넣는 부분으로서 사용자가 키조작에 의해 선택할 수 있게 하였으며 이때 선택된 심볼은 SELECTION FIELD에 표시된다.

MENU FIELD에는 사용할 수 있는 명령이 표시되어 있어 사용자가 키조작에 의해 선택하여 그 기능을 수행시킬 수 있다.

에디터로 GRAFCET 다이어그램을 작성한 후에는 심볼 테이블을 작성해야 한다. 심볼 테이블은 GRAFCET의 ACTION에 해당하는 기기의 동작을 Z80 단일기판 컴퓨터에서 각 작동기(actuator)에 신호를 전송이나 TRANSITION에 해당하는 센서의 신호 검출을 할 수 있도록 표를 만들어 참조할 수 있게 한 것이다. 이것을 위하여 먼저 I/O 번호를 정의한 후 I/O 포트(port)와 각 포트의 비트(bit)를 입/출력으로 구분한다. 그 후 각 ACTION과 TRANSITION에 해당하는 동작 내용을 기입하는 데 이 때 그 I/O 비트의 동작 방식, 즉 ON이나 OFF 상태를 "+"나 "-"의 기호로 각 I/O 번호와 함께 기입한다.

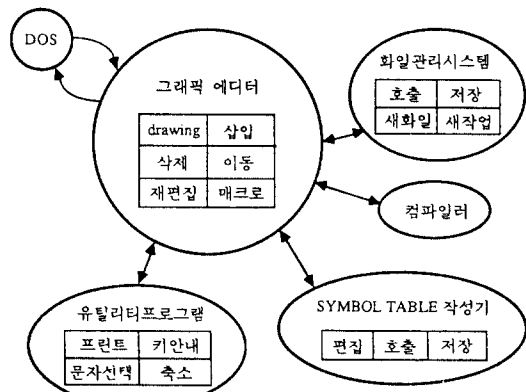


그림 8 GRAFCET 에디터의 전체적 구성

Fig. 8 The overall structure of GRAFCET editor

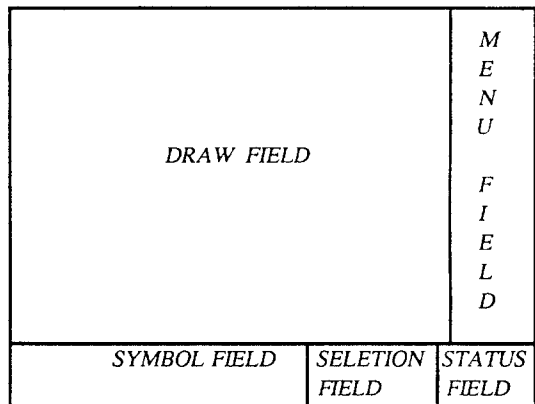


그림 9 그래픽 에디터의 화면구성

Fig. 9 The screen status of the graphic editor

5. GRAFCET 컴파일러

GRAFCET 컴파일러는 그림 10과 같이 전체적 구조 파악 및 에러 검출을 한 후 중간코드를 생성해 내는 파서(parser)와 중간코드로부터 Z80 어셈블러 코드를 발생시키는 코드 발생기(code generator)로 구성되어 있다.

파서는 GRAFCET 다이어그램에 대한 이진 트리를 탐색하여 중간코드 생성시 필요한 분기의 목적지에 대한 레이블(label)생성 및 심볼간의 연결 관계에서 체크되지 않는 의미상의 에러를 검출한다.

코드 발생기는 먼저 시스템 초기화 코드를 생성해 낸 후 중간코드를 차례대로 해석하여 I/O 정의표와 심볼 테이블에 수록된 I/O 포트 번호 및 센서 등의 정보를 이용하여 각 동작 코드에 해당하는 Z80 어셈블러 코드를 생성해 낸다. 또한 동시진행 과정의 시작과 병합을 큐를 이용하여 올바르게 처리하도록 큐를 초기화 시키는 과정 및 큐 작동 루틴 등을 추가한다.

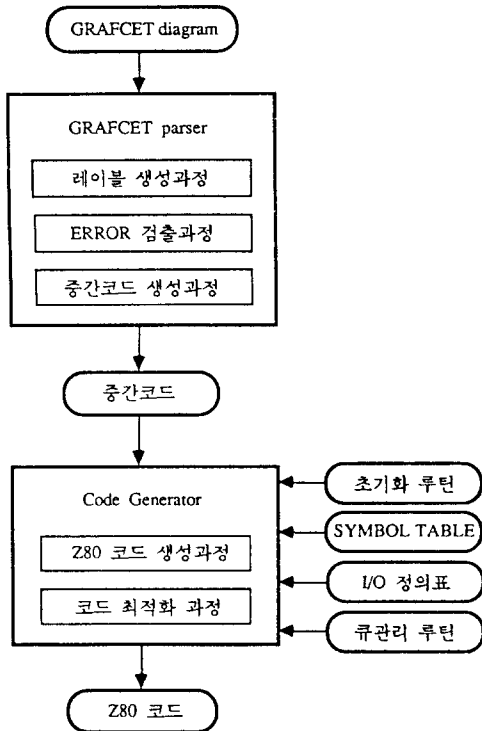


그림 10 GRAFCET 컴파일러의 구조

Fig. 10 The structure of GRAFCET compiler

6. KGPT에 의한 GRAFCET 작성 예

그림 11과 같이 간단한 구조를 갖는 프레스를 순서제어하기 위하여 KGPT로 프로그램하는 예를 고려하자. 이 프레스의 동작순서는 다음과 같다. 먼저 작동준비 램프(PL R)가 켜지면 ST 스위치를 눌러 작동 시작을 프레스 제어기

에 알려 프레스를 작동시킨다. 이어서 리미트 스위치 SLTd가 ON이 될 때까지 프레스의 스탬프(stamp)를 하강시키며 SLTd가 ON이 되면 다시 SLTu가 ON이 될 때까지 상승시킨다. 그 후 다이(die)를 DLTd가 ON이 될 때까지 하강시키며, 이어서 성형완료 램프(PL E)를 타이머가 작동될 동안 ON시켜 성형된 제품을 배출시키도록 한다. 타이머 동작이 완료되면 성형완료 램프를 OFF시키며 DLTu가 ON이 될 때까지 다이들을 상승시켜 처음상태로 되돌아가게한다.

그림 12는 KGPT에 의해 위의 작업을 작성한 GRAFCET 다이어그램이며, 그림 13은 그에 상당하는 심볼 테이블이다.

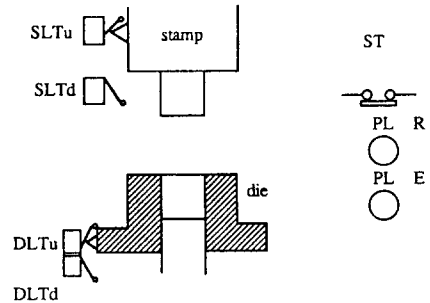


그림 11 예를 위한 프레스 공정

Fig. 11 A press process for example

7. 결론

GRAFCET을 개인용 컴퓨터에 구현하기 위한 GRAFCET 프로그래밍 도구를 개발하였다. 이것은 그래픽 에디터 모듈과 컴파일러 모듈로 이루어져 있으며, 메뉴와 심볼을 사용하여 용이하게 이용할 수 있도록 구현하였다. 이것은 VGA 그래픽 보드가 장착된 IBM PC/AT 이상의 컴퓨터에서 이용될 수 있도록 개발하였으며, KGPT라 명명하였다. 이것을 이용하여 간이자동화 시스템의 공정을 프로그램하는 데 이용하면 많은 시간과 비용을 절감할 수 있다고 여겨진다.

참고문헌

1. "GRAFCET : a functional chart for sequential processes", report of team Automatismes sequentiels, ADEPA, France, 1979.
2. "PC에서 SFC언어란?", 월간 자동화 기술 9월호, pp.140-146, 1989.
3. J.M.Peterson, Petri net theory and the modeling of systems, Prentice-Hall, 1981.
4. E.Horowitz and S.Sahni, Fundamentals of data structures in Pascal, Computer Science Press, 1987.

5. 정연규, "Graphic Sequence Programming 및 자동화 기계 기능 설계법 연구", 한국기계연구소 보고서, pp.29-43, 1988.

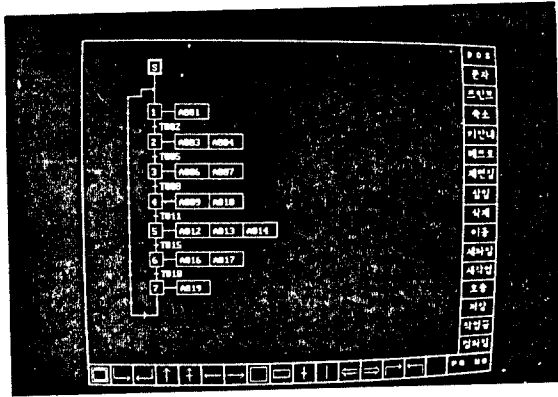


그림 12 프레스 공정을 위해 작성된 GRAFCET 다이어그램
 Fig. 12 The GRAFCET diagram edited for the press process

SYMBOL TABLE					
T001	ON Ready Lamp	:9+	T016	OFF 사출 Lamp	:18-
T002	ST 농밀	:1-	T017	ON Bie 상승	:4+
T003	OFF Ready Lamp	:9-	T018	OFF Bie 상승	:13-
T004	ON Stamp 하강	:22+	T019		
T005	Stamp 하강	:23+	T020		
T006	OFF Stamp 하강	:22-	T021		
T007	ON Stamp 상승	:21+	T022		
T008	Stamp 상승	:24+	T023		
T009	OFF Stamp 상승	:21-	T024		
T010	ON Bie 하강	:14+	T025		
T011	하강단	:5+	T026		
T012	OFF Bie 하강	:14-	T027		
T013	ON 사출 Lamp	:10+	T028		
DN14	하이퍼 구동	:1,1000	T029		
T015	하이퍼 구동	:1,0	T030		

그림 13 프레스 공정 작성을 위한 상당 심볼 테이블
 Fig. 13 The corresponding symbol table for editing the press process