

현장 기술자를 위한

소형 PC에 의한 시퀀스 제어

현·장
기·술

6

역 / 박 한 중 (협회 교육홍보위원)

제5장 기본 시퀀스 제어회로의 프로그래밍과 실행

5.2 간격 동작회로

(1) 유접점 회로도

그림 5.5에 회로도를 나타낸다. 이 회로는 푸시 버튼 스위치 A를 눌러 출력을 ON하여 자기 유지를 건다. 타이머 설정시간을 경과하면 타이머의 한시동작 b접점이 작동, 출력이 OFF가 된다. B는 비상정지(리셋)용 푸시버튼 스위치이다.

(2) 무접점 회로도

그림 5.5의 유접점 회로를 불식으로 표현하면 다음과 같이 된다.

$$T = (A + X_1) \cdot \bar{B} \dots\dots\dots (5.2)$$

$$X = (A + X_1) \cdot \bar{B} \cdot \bar{T}_1 = T \cdot \bar{T}_1 \dots\dots\dots (5.3)$$

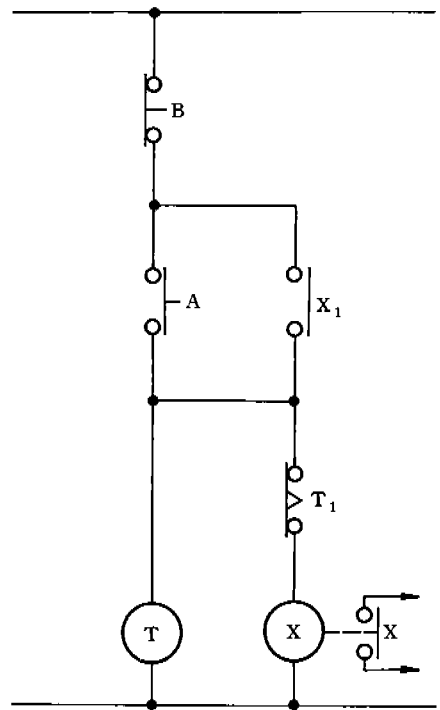
여기서 T₁은 한시동작.

위의 식을 중개로 하여 무접점 회로로 변환한 것을 그림 5.6에 나타낸다. 불식의 T와 T₁의 관계가 무접점 회로의 타이머 심볼 및 입력 T, 출력 T₁과의 관계를 보면 확실해질 것으로 생각한다.

(3) 번지의 할당

그림 5.7과 같이 할당한다. 일시 기억을 분기개소 10과 11번지에 설정한다. 10번지는 연산결과의

분기인 동시에 연산처리 대기 개소이기도 하다.



<그림 5.5> 유접점 간격 동작회로

(4) 코딩

03 스텝 STO 10

분기 개소인 동시에 04 스텝으로부터의 타이머 연산처리 대기 개소이기도 하다. 이것은 메모리블이 전자계산기로 말하면 가감승제의 복합계산을 할 때 다음 계산결과 대기를 위해 일단 도중의 계산결과를 메모리에 기억시키는 것에 상당한다.

04 스텝 TIM 80

4초의 타이머를 일시 기억 30번지에 만든다.

07 스텝 AND 10

04~06 스텝으로 타이머 연산처리가 종료했으므로 이 결과와 앞의 03 스텝에서 연산처리 대기로 STORE해 둔 일시 기억 10번지와 AND를 취한다.

메모리블이 전자계산기 계산으로 말하면 전에 메모리에 넣은 결과와 이번에 계산을 끝낸 결과에서 다음 계산을 실행하는 것에 상당한다.

08 스텝 STO 11

출력 X를 유지신호 X_1 으로서 앞으로 되돌리고 있고 분기 개소로 되어 있으므로 STORE하여 두어야 한다.

<표 5.2> 간격 동작 프로그램 리스트

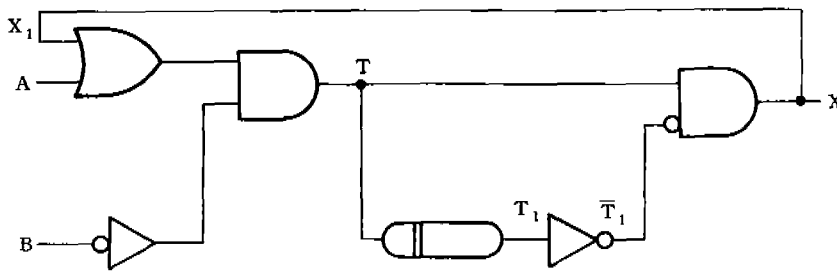
스텝 No.	명령		비고
	OP	ADD	
00	LD	00	4초
01	OR	11	
02	AND	01	
03	STO	10	
04	TIM	80	
05	LD	30	
06	INV	00	
07	AND	10	
08	STO	11	
09	OUT	E0	

이상의 코딩 상태를 그림 5.8에 나타낸다. 메모리블이 전자계산기를 사용하여 계산을 진행하는 요령으로 프로그램 코딩되는 것을 보기 바란다. 10번지의 내용을 STORE해 두는 이유가 확실해졌으므로 생각한다.

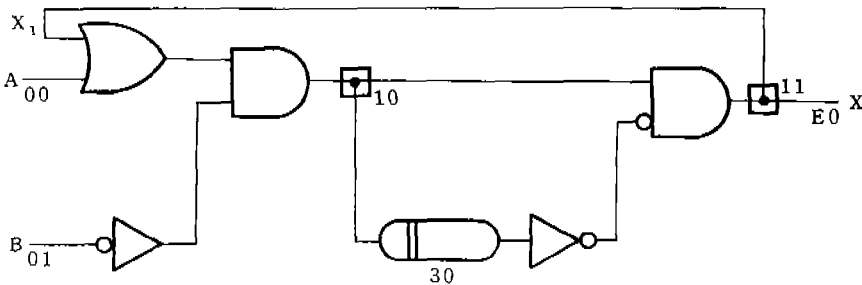
코딩 작업시 펠트펜 등으로 무점점 회로상에 작업 종료부분을 마크해 둔다. 이렇게 하면 코딩의 진척상태를 파악할 수 있어 코딩 누락을 없애는 데도 도움이 된다.

(5) 동작 확인

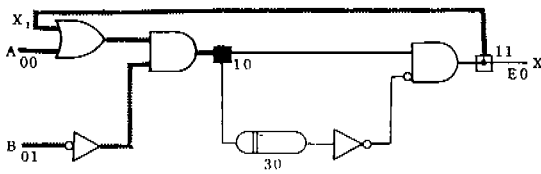
프로그램을 실행하고 입력 B의 조작 스위치를 ON으로 하여 둔다. 이 상태로 입력 A의 조작 스위치를 순시 ON하면 출력이 ON하여 자기유지가



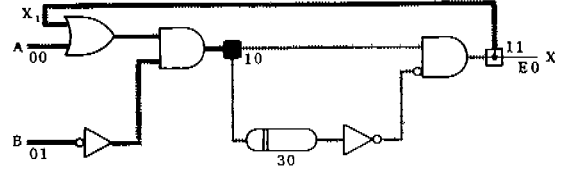
<그림 5.6> 무점점 간격 동작회로



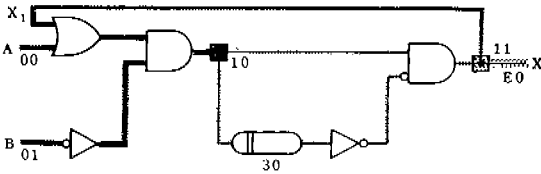
<그림 5.7> 번지의 할당



(a) 코딩 상황(03 스텝 STO 10까지)



(b) 코딩 상황(07 스텝 AND 10까지)



(c) 코딩 상황(09 스텝 OUT E0까지)

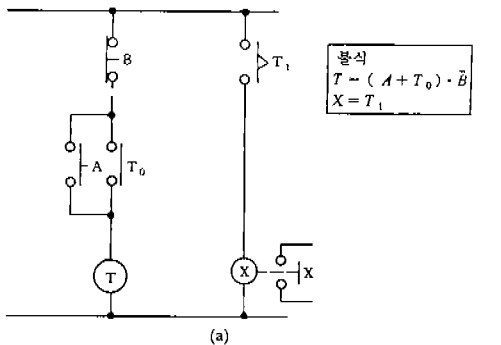
<그림 5.8> 프로그램 코딩 상황

되고 출력표시 LED가 ON을 계속한다. 그리고 4초가 경과하면 자동적으로 출력이 OFF가 되면 되는 것이다.

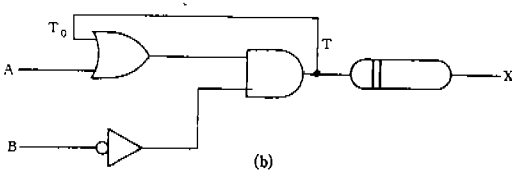
4초 이전에 비상정지를 하는 사태가 일어난 경우는 입력 B의 조작 스위치를 순시 OFF 조작하면 출력은 즉시 OFF가 되어야 한다. 이것도 확인해 두도록 하자.

연습문제 5.2

그림 5.9에 지연동작의 유접점 회로도화 이것과 동일 내용의 무접점 회로도화 있다. 그리고 참고



(a)



(b)

<그림 5.9> 지연동작회로

를 위해 불식도 표시하고 있다.

이 무접점 회로도상에 2번지를 할당하고 이 회로와 동일 동작을 하는 프로그램을 만들어라. 단, 타이머의 시간 설정은 4초로 한다.

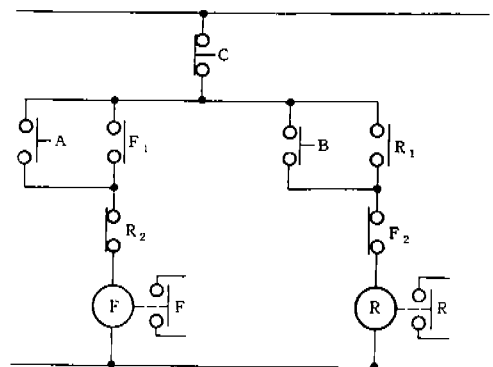
5.3 인터록 회로

(1) 유접점 회로도

그림 5.10에 유접점 회로도화 나타낸다.

푸시버튼 스위치 A를 눌러 출력 F를 ON시키고 자기 유지를 건다. 이 상태에서 푸시버튼 스위치 B를 눌러도 출력 R는 ON하지 않는다. 즉, 이것은 F가 ON하고 있는 동안은 R는 ON이 되지 않도록 코일 F의 보조 접점 F2가 코일 R와 직렬로 접속되어 있기 때문이다.

F를 일단 OFF하고나서가 아니면 R를 ON할 수는 없다. 이 리셋을 거는 것이 푸시버튼 C이다. 또



<그림 5.10> 유접점 인터록 회로

<표 5.3> 인터록 · 프로그램 리스트

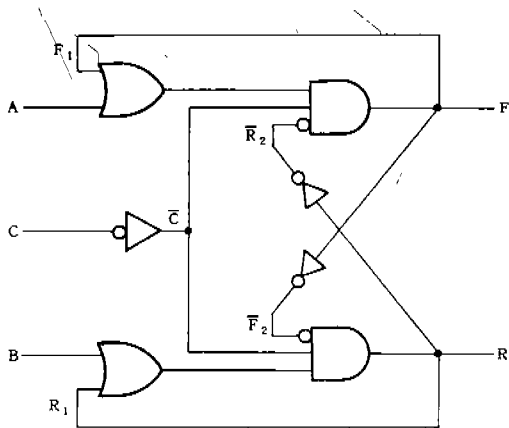
스텝 No.	명령		비고
	OP	ADD	
00	LD	00	
01	OR	20	
02	STO	10	
03	LD	21	
04	INV	00	
05	AND	10	
06	AND	02	
07	STO	20	
08	OUT	E0	
09	LD	01	
0A	OR	21	
0B	STO	11	
0C	LD	20	
0D	INV	00	
0E	AND	11	
0F	AND	02	
10	STO	21	
11	OUT	E1	

한 반대로 R가 ON하고 있을 때는 F를 ON으로 할 수가 없다.

어느 조건이 성립될 때까지 상대의 동작을 저지하고 상호 자물쇠를 걸고 있는 것같은 회로구성으로 되어 있기 때문에 이것을 인터록 회로라고 한다. 또한 이 회로는 모터의 정역운전회로 등의 제어회로로서 응용되는 것이다.

(2) 무접점 회로도

그림 5.10의 유접점 회로를 불식으로 표현하면



<그림 5.11> 무접점 인터록 회로

다음과 같이 된다

$$F = (A + F_1) \cdot \bar{R}_2 \cdot \bar{C} \dots\dots\dots (5.4)$$

$$R = (B + R_1) \cdot \bar{F}_2 \cdot \bar{C} \dots\dots\dots (5.5)$$

위의 식을 중개로 하여 무접점화한 회로를 그림 5.11에 나타낸다.

(3) 번지의 할당

그림 5.12와 같이 할당한다. 일시기억 10, 11번지는 연산처리 대기, 20, 21번지는 분기개소에 있기 때문에 설정한 것이다. 단, 입력 C의 부정 \bar{C} 개소에 분기가 있지만 여기에는 일시기역을 설정할 필요가 없다.

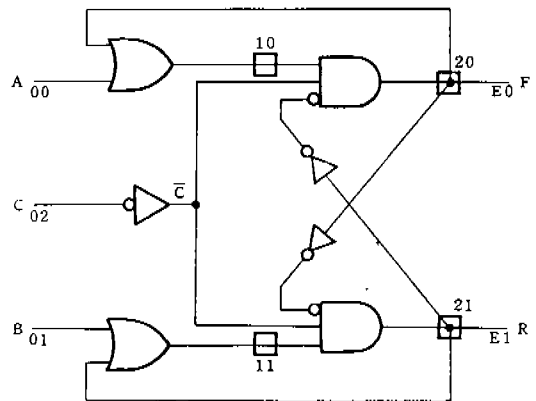
(4) 코딩

표 5.3과 같다. 지금까지의 설명으로 충분히 이해될 것으로 특별한 설명은 필요가 없다.

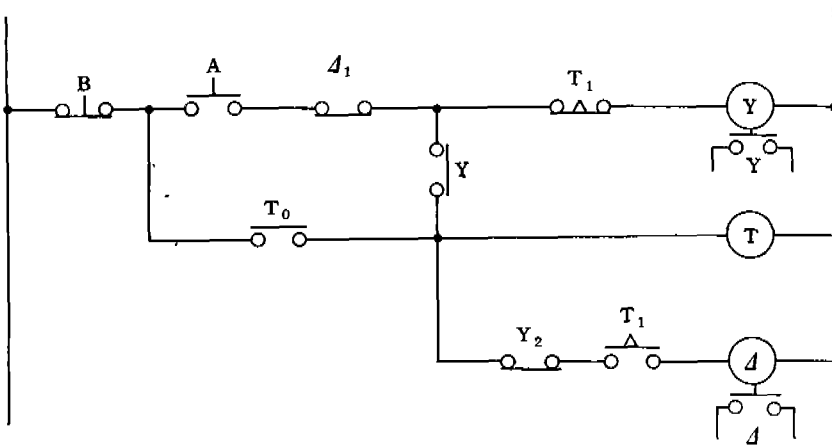
(5) 동작 확인

프로그램을 실행하여 입력용 조작 스위치 C를 ON으로 하여 두고 입력 A를 순시 ON 조작한다. 그렇게 하면 출력 F가 ON하여 자기 유지가 걸리며 LED가 계속 점등한다. 이 상태에서 입력 B를 조작(ON)하더라도 출력 R는 ON이 되지 않으면 되는 것이다.

출력 F를 OFF로 하는 데는 입력 C를 순시 OFF 조작한다. 그 후 입력 B를 순시 ON 조작한다. 이 때 출력 R가 ON하여 자기 유지가 걸리고 R 상태를 표시하는 LED가 점등을 계속한다. 이것을 확인해 두자.



<그림 5.12> 번지의 할당



<그림 5.13> 유접점 한시동작의 이행회로

5.4 한시동작의 이행회로

(1) 유접점 회로도

그림 5.13에 유접점 회로도를 나타낸다. 푸시버튼 스위치 A를 누르면 코일 Y가 여자되고 그 보조 a접점 Y₁이 ON하며 타이머 코일 T가 여자된다. 그러면 타이머 T의 순시동작 a접점 T₀가 ON하여 자기유지되고 출력 Y가 ON한다. 이 때 출력 Δ는 ON하지 않게 코일 Y의 보조 b접점 Y₂에서 코일 Δ가 여자되지 않도록 인터록을 하고 있다.

일정시간 경과하면 타이머의 한시동작 b접점 T₁이 OFF가 되어 Y코일의 여자가 풀리고 출력 Y를 OFF로 한다. 이어서 Y코일의 보조 a접점 Y₁이 OFF가 되고 b접점 Y₂가 ON한다. 이 때 타이머의 한시동작 a접점이 ON으로 되어 있으므로 코일 Δ가 여자되고 출력 Δ가 ON이 된다. 또한 코일 Δ의 보조 b접점 Δ₁에서 코일 Y가 여자되지 않도록 인터록이 걸려 있다.

리셋은 푸시버튼 B를 누르면 모든 코일의 여자가 풀리고 출력 Y도 Δ도 OFF가 된다.

이상과 같이 한시동작이 Y부터 Δ로 이행하고 있으므로 이 회로를 본고에서는 한시동작의 이행회로라고 부르기로 하였다. 또한 이 제어회로는 그림 5.14의 주회로에 표시하는 것 같은 3상 유도전동기의 자동 Y-Δ 기동에 응용되는 것이다.

(2) 무접점 회로도

유접점 회로를 불식으로 표현하면 다음과 같다.

$$Y = \bar{B} \cdot \bar{T}_1 \cdot (A \cdot \bar{\Delta}_1 + T_0 \cdot Y_1) \dots\dots\dots (5.6)$$

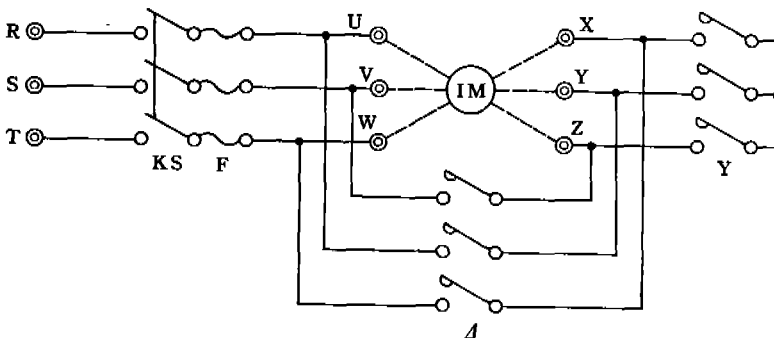
$$T = \bar{B} \cdot (A \cdot \bar{\Delta}_1 \cdot Y_1 + T_0) \dots\dots\dots (5.7)$$

$$\begin{aligned} \Delta &= \bar{Y}_2 \cdot T_1 \cdot \bar{B} (A \cdot \bar{\Delta}_1 \cdot Y_1 + T_0) \\ &= \bar{Y}_2 \cdot T_1 \cdot T \dots\dots\dots (5.8) \end{aligned}$$

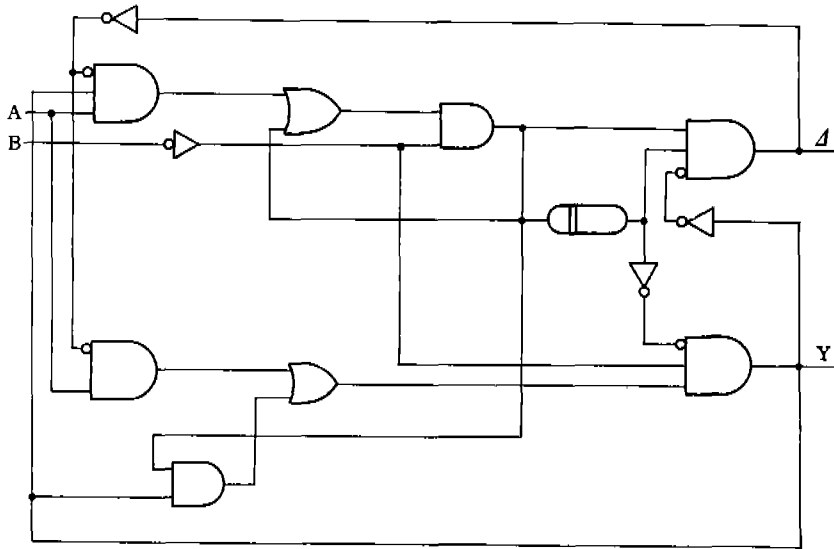
위의 식을 중개료 하여 무접점 회로로 변환한 것을 그림 5.15에 나타낸다.

(3) 번지의 할당

그림 5.16과 같이 할당한다. 일시기억 번지를 분



<그림 5.14> 3상 유도전동기 Y-Δ 기동 주회로



〈그림 5.15〉
무접점 한시동작의 이행회로

기개소에 설치하는 것은 물론이고 연산처리 대기를 위해 수 개소에 일시기억 번지를 설정할 필요가 있다.

연산처리 대기를 위한 일시기억 번지는 코딩을 어디부터 하여 나가는가에 따라 설정 개소가 달라지게 된다. 이 그림의 할당은 몇가지 있는 중의 하나의 예로서 보기 바란다.

(4) 코딩

표 5.4와 같다. 지금까지의 설명에 의해 그림 5.16을 기본으로 코딩해 나가면 특별히 문제가 없다.

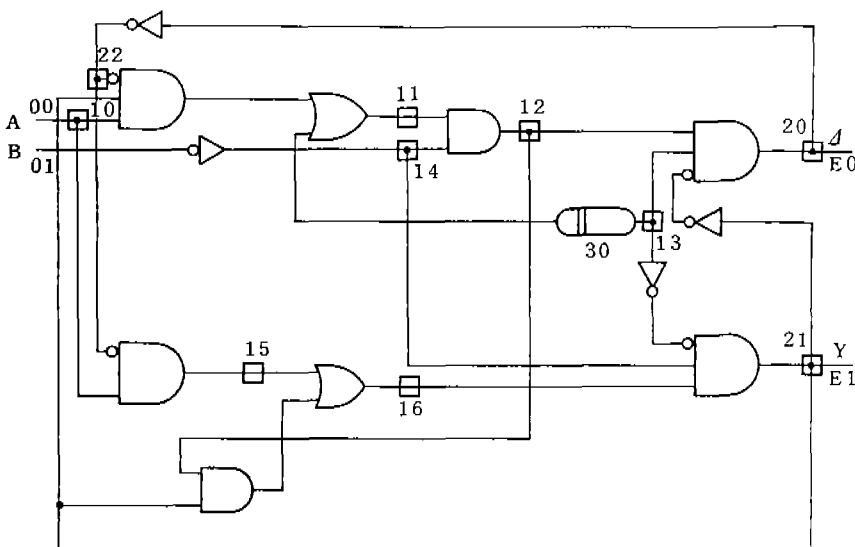
타이머 설정에 관해서 OA 스텝에 주의하기 바란다.

OA 스텝 TIM 40

2초의 타이머를 설정한 것이다.

(5) 동작 확인

프로그램을 실행하여 입력용 조작 스위치의 B를 ON으로 하여 두고 입력 A를 순시 ON 조작한다. 그렇게 하면 출력 Y가 ON하여 자기유지가 걸리고 출력 표시 LED가 점등한다. 2초 경과하면 출력 Y가 OFF하고 출력 A가 ON하여 LED가 점등을 계속한다.



〈그림 5.16〉 번지의 할당

〈표 5.4〉 한시동작의 이행 프로그램 리스트

스텝 No.	명령		비고	스텝 No.	명령		비고
	OP	ADD			OP	ADD	
00	LD	00		10	AND	13	
01	STO	10		11	STO	20	
02	AND	21		12	OUT	E0	
03	AND	22		13	LD	20	
04	OR	12		14	INV	00	
05	STO	11		15	STO	22	
06	LD	01		16	AND	10	
07	STO	14		17	STO	15	
08	AND	11		18	LD	12	
09	STO	12		19	AND	21	
0A	TIM	40	2초	1A	OR	15	
0B	LD	30		1B	STO	16	
0C	STO	13		1C	LD	13	
0D	LD	21		1D	INV	00	
0E	INV	00		1E	AND	14	
0F	AND	12		1F	AND	16	

정지하는 경우는 입력 B를 순시 OFF 조작하면

모든 출력은 OFF가 된다.

지금까지의 명령의 정리①

명령	OP	ADD	기능
LOAD	LD	입력 00-0F번지 출력 E0-EF번지 일시기역 10-3F번지	ADD에서 지정하는 내용의 판독입력
STORE	STO	일시기역 10-3F번지	일시기역
OUT	OUT	출력 E0-EF번지	ADD에서 지정하는 출력단자への 출력
AND	AND	입력 00-0F번지 출력 E0-EF번지 일시기역 10-3F번지	논리적
OR	OR	입력 00-0F번지 출력 E0-EF번지 일시기역 10-3F번지	논리합
NOT	INV	00	부정
TIMER	TIM	시간설정* 1~F 일시기역 (3)0-(3)F번지	한시동작지정, 시간설정
NOP	NOP	00	아무것도 실행하지 않고 다음 스텝으로 진행한다

* 타이머의 지정, 시간설정

ADD의 처음 숫자는 시간설정(1~F)

설정시간 T(초)는 다음 식으로 계산 : $T = \text{설정값}(1\sim F) \times 0.5$

ADD의 다음 숫자는 타이머의 일시기역 번지의 지정(30번지의 1자째의 0~F만 쓴다)

☞ 다음호에 계속 ☞