현장 기술자를 위한

소형 PC에 의한 시퀀스 제어

현·장
기·술
8

출판홍보과

제7장 응용예

이 장에서는 PC를 사용한 제어 시스템의 구성방법과 PC 제어의 응용예에 대해서 설명한다. 응용예는 PC에 의한 프로그램이 지금까지 설명한 내용으로 만들 수 있는 것을 구체적으로 보고 알기 위해서 든 것이다. 그러므로 각 제어내용에 대해서는 간단히 설명하고 상세한 것은 기술하지 않는다. 각 제어에 대해 시퀀스 제어에 대한 전문서를 참고하기 바란다.

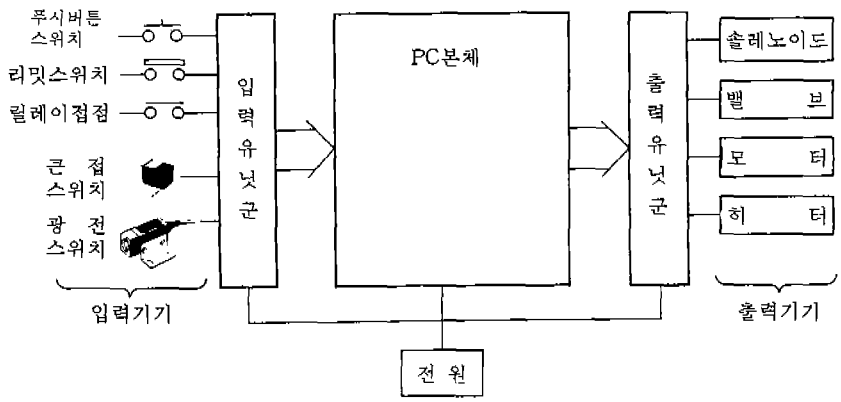
여기서는 시스템이 개략적으로 어떻게 구성되어 있는가를 주로 그림으로 표시한다. 아울러서 제어회로와 이것을 프로그램화한 프로그램 리스트와 그 해설을 들었다.

7.1 PC 제어 시스템의 구성방법

입력, 출력기기의 접속과 I/O 유닛군 구성방법의 기본은 그림 7.1에 나타내는 것과 같다.

전원은 PC 본체 및 입출력 유닛에 안정된 직류전압을 공급하기 위한 것이다. 그 밖에 입력기기에서 PC 본체에 신호를 입력하지 않으면 안된다. 입력기기를 작동하여 입력 유닛에 신호전압을 보내기 위해서 외부 전원이 필요하다. 또한 출력기기에도 이것을 운전하기 위한 동력원이 되는 전원이 필요하다는 것은 말할 것도 없다.

다음에 입력 유닛을 어떻게 선정하고 제어 시스



〈그림 7.1〉
PC 제어 시스템의 구성

템을 구성하는가는 입력기기로부터의 신호상태에 따라 결정된다. 그러므로 입력 유닛의 선정에 관해서는 다음과 같은 사항을 고려하여야 한다.

(1) 입력기기의 종류

푸시버튼 스위치, 리밋 스위치 등의 유접점 기기인가 또는 근접 스위치, 광전 스위치 등의 무접점 기기인가?

(2) 입력기기가 내는 신호전압의 종류와 크기

직류인가 교류인가? 그리고 그 전압의 크기.

출력 유닛 선정시에는 다음 사항을 고려한다. 이 출력 유닛에 의해 제어코자 하는 출력기기의 부하전류 개폐능력이 결정되는 것이다.

(1) 출력기기의 종류와 그 정격전류
(2) 출력기기의 운전용 전원의 종류와 그 전압의 크기

입력 및 출력 유닛 모두 이상의 사항에 대한 검토에 입각해서 적절한 유닛을 선정한다. 구체적으로는 각 PC 메이커의 입출력 유닛에 대한 카탈로그, 취급설명서에 적당한 기능단위로 표준화된 유닛군이 표시되어 있으므로 이들 중에서 선택하게 된다.

7.2 입출력 기기의 접속방법

입출력 기기를 PC에 접속하는 방법에 대한 구체적 예를 그림 7.2에 들었다. 이들 예는 입력을 유접점의 푸시버튼 스위치에 부여하여 푸시버튼 스위치를 조작함으로써 출력측에 접속한 출력기기 모터를 제어하려는 것이다.

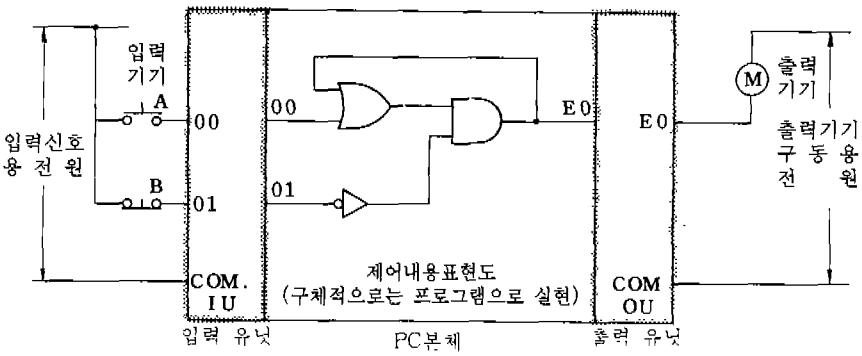
우선 입력기기의 접속은 입력 유닛에 신호전압을 송입한다는 생각으로 한다. 입력 유닛에는 입력기기를 조작함으로써 그 시스템의 입력신호 전압이 인가된다. 이 인가전압이 적당한 입력신호 전압으로 변환되고 그리고 필요하면 전기적인 절연을 하여 PC 본체가 받아들이는 입력신호가 되어 주고 받게 된다.

입력 유닛에서는 신호전압의 변환 등이 행하여지지만 본질적으로는 입력신호를 PC 본체에 송입하는 것으로 입력신호와 PC 본체 내부에서 실현되고 있는 논리회로가 논리적으로 접속되어 있는 점에 착안하기 바란다. 그 이유는 종래의 전자 릴레이 시퀀스 제어회로의 입력 취급은 반드시 이렇지는 않고 전류 통로 중의 1점점으로 취급되어 왔기 때문이다.

다음에 출력 유닛과 외부 출력기기의 접속은 그림 7.2의 출력부에 표시하는 것과 같이 한다. PC 본체로부터의 출력신호로 출력 유닛의 출력개폐기가 구동되고 출력기기의 전원회로가 개폐된다.

이 출력 유닛의 출력개폐기에는 전자 릴레이를 사용한 유접점 방식의 것과 파워 트랜지스터나 트라이액을 사용한 무접점 방식의 것이 있다. 그러나 그림과 같이 출력 유닛이 어느 방식의 경우나 출력기기의 전원회로 개폐부가 되도록 한다.

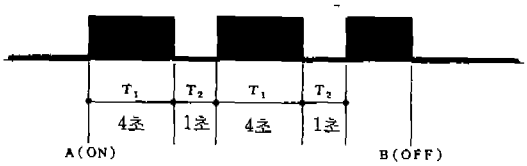
PC 본체로부터의 출력신호가 이 예의 경우는 출력 유닛의 E0번지에 들어가 있지만 이 신호로 직접 출력기기를 구동하는 것이 아니고 출력 유닛(개폐기)의 구동신호로서 사용되고 있는 것에 대한 주의가 필요하다.



〈그림 7.2〉 입출력 기기의 접속 예와 PC 제어내용 표현도와의 관련

7.3 반복운전 제어

그림 7.3의 동작 타임차트에 나타내듯이 타이머로 설정된 시간에 의해 출력기기의 동작을 자동적으로 반복하여 시행하는 제어이다. 이 제어의 응용은 기계의 자동 반복운전 등이다.



〈그림 7.3〉 반복 동작 타임차트

그림 7.4(a), (b)에 유접점 및 무접점의 제어회로를 들었다. 그림 7.5는 번지의 할당이다. 입력 푸시버튼 스위치 A, B를 각각 00, 01번지에, 출력을 E0번지에 할당하고 있다. 이것은 특별히 문제가 없는 것이다. 10, 11, 13번지는 분기개소로 되어 있기 때문에 12번지는 연산처리 대기를 위해 설정한 일시 기억의 번지이다.

그림 7.5에 입각해서 만들어진 프로그램을 표 7.1 프로그램 리스트에 나타낸다. 이것에 대해서 알아보기로 하자.

01 스텝 OR 10

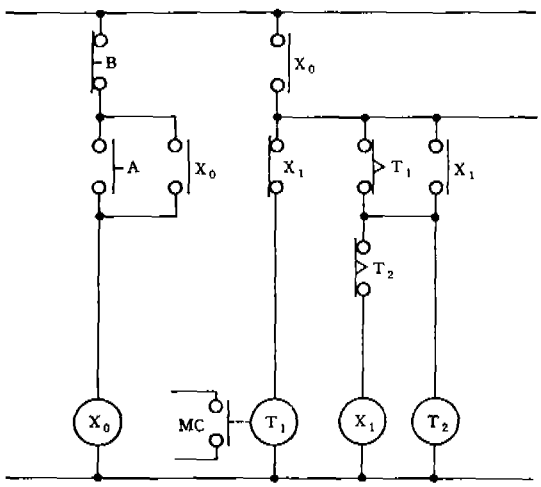
뒤에 STO 명령으로 10번지에 격납되는 내용과 00 스텝으로 판독한 00번지의 내용이 OR 회로를 형성하고 있다.

02 스텝 AND 01

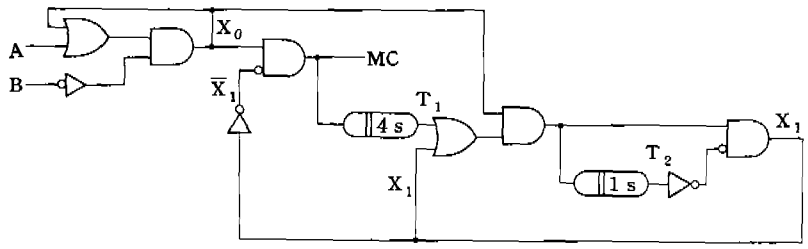
푸시버튼 스위치 B(01번지)는 b접점에서 부여되고 있는데 입력측이 부논리로 상태표시 기호(○부)가 입력측에 달려 있으므로 부정을 취하지 않고 그대로 앞의 결과의 AND가 된다.

03 스텝 STO 10

01 스텝에서 10번지의 내용을 사용

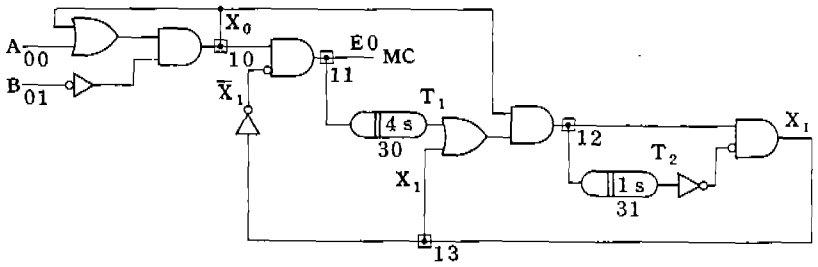


(a) 유접점회로



(b) 반복회로

〈그림 7.4〉 반복회로



〈그림 7.5〉 번지의 할당

하고 있으므로 지금까지의 연산 결과를 10번지에 격납해 둘 필요가 있었다. 그리고 이것은 다음의 연산을 하기 전의 연산처리 대기로도 되어 있다.

04 스텝 LD 13

일시기억 13번지의 내용을 가지고와 다음의 05 스텝에서 그것을 부정하여 반전하고 있다. 02 스텝에서의 입력 01번지와의 취급과 비교하기 바란다.

06 스텝 AND 10

10번지에 격납해 둔 10번지까지의 연산 결과 내용과 04, 05 스텝에서 처리한 13번지의 내용을 반전한 결과와의 AND이다.

09 스텝 TIM 80

타이머 설정시간은 이 경우 4초로 하고 있지만 이것은 기계의 운전상태에 맞추어 적당히 결정되는 것이다. 일례로서 보기 바란다.

08 스텝 OR 13

타이머 처리를 한 연산결과와 일시기억 13번지 내용과의 OR이다. 13번지의 내용은 후에 STO 명령을 사용하여 이 번지에 격납해 두는 것을 잊지 않도록 하지 않으면 안된다.

0D 스텝 STO 12

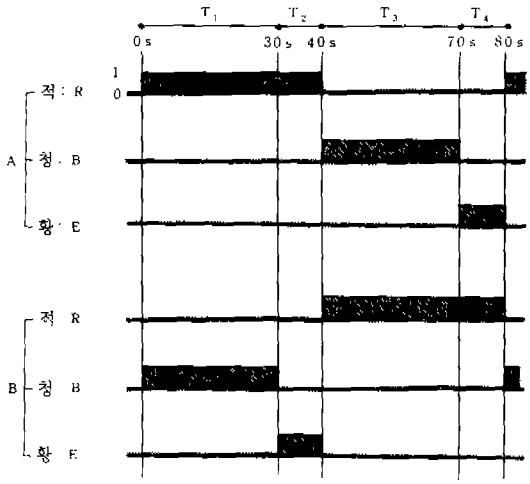
연산처리 대기를 위해 지금까지의 내용을 일단 일시기억 12번지에 격납해 두고 0E 스텝부터 타이머 처리를 하게 된다.

11 스텝 AND 12

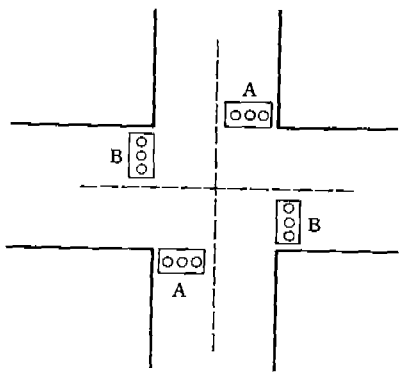
타이머 처리를 한 결과와 전에 일시기억 12번지에 격납한 내용과의 AND가 된다.

〈표 7.1〉 프로그램 리스트

스 텝 No.	명 령 OP	명 령 ADD	비 고
00	LD	00	
01	OR	10	
02	AND	01	
03	STO	10	
04	LD	13	
05	INV	00	
06	AND	10	
07	STO	11	
08	OUT	E0	
09	TIM	80	
0A	LD	30	
0B	OR	13	
0C	AND	10	
0D	STO	12	
0E	TIM	21	
0F	LD	31	
10	INV	00	
11	AND	12	
12	STO	13	



(a) 타임차트

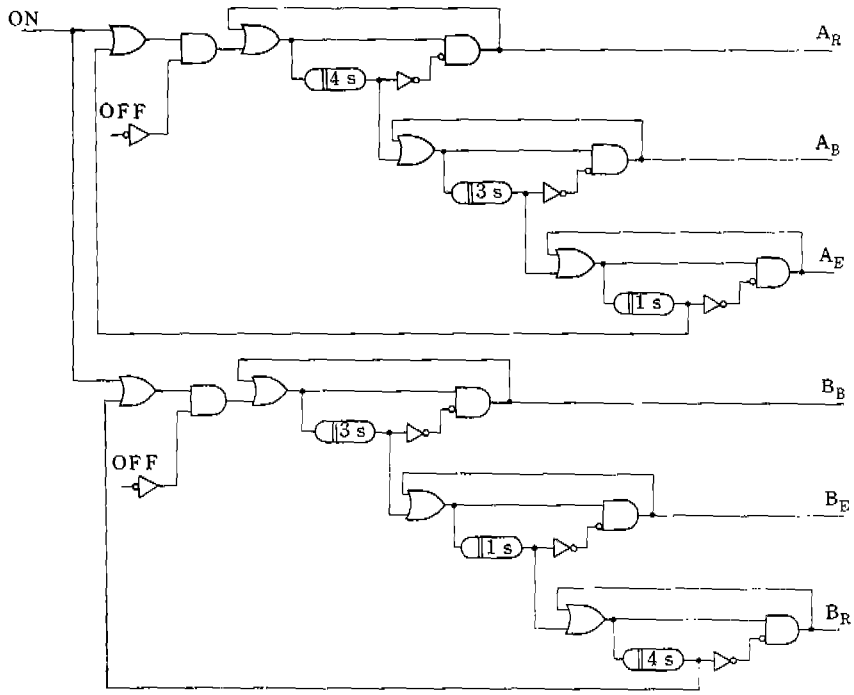


(b) 배치도

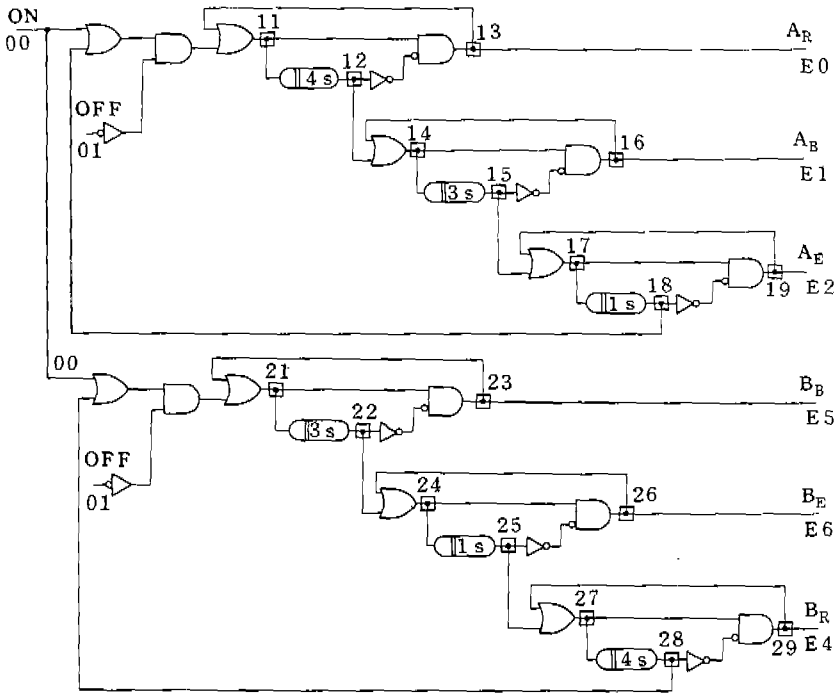
〈그림 7.6〉 교통신호기

12 스텝 STO 13

04 및 0B 스텝에서 일시기억 13번지의 내용을 사용하고 있으므로 STO 명령을 사용하여 소정의 할당번지에 지금까지의 연산결과를 격납하고 있다.



〈그림 7.7〉
교통신호기 제어회로



〈그림 7.8〉 번지의 할당

〈표 7.2〉 프로그램 리스트

스 텝 No.	명 령 OP	명 령 ADD	비 고
00	LD	00	
01	OR	18	
02	NOP	00	
03	NOP	00	
04	NOP	00	
05	AND	01	
06	OR	13	
07	STO	11	
08	TIM	80	4초
09	LD	30	
0A	STO	12	
0B	INV	00	
0C	AND	11	
0D	STO	13	
0E	OUT	E0	
0F	LD	12	
10	OR	16	
11	STO	14	
12	TIM	61	3초
13	LD	31	
14	STO	15	
15	INV	00	
16	AND	14	
17	STO	16	
18	OUT	E1	
19	LD	15	
1A	OR	19	
1B	STO	17	
1C	TIM	22	1초
1D	LD	32	
1E	STO	18	
1F	INV	00	

스 텝 No.	명 령 OP	명 령 ADD	비 고
20	AND	17	
21	STO	19	
22	OUT	E2	
23	LD	00	
24	OR	00	
25	NOP	00	
26	NOP	00	
27	NOP	00	
28	AND	01	
29	OR	23	
2A	STO	21	
2B	TIM	63	3초
2C	LD	33	
2D	STO	22	
2E	INV	00	
2F	AND	21	
30	STO	23	
31	OUT	E5	
32	LD	22	
33	OR	26	
34	STO	24	
35	TIM	24	1초
36	LD	34	
37	STO	25	
38	INV	00	
39	AND	24	
3A	STO	26	
3B	OUT	E6	
3C	LD	25	
3D	OR	29	
3E	STO	27	4초
3F	TIM	85	
40	LD	35	
41	STO	28	
42	INV	00	
43	AND	27	
44	STO	29	
45	OUT	E4	
46			
47			

7.4 모의 교통신호기 제어

그림 7.6에 모의 교통신호기의 동작 타임차트와 배치도가 있다. 이것은 차도의 교통신호기로 간단히 하기 위해 타임차트에 나타낸 바와 같은 T_2~T_4를 1주기로 하여 연속적으로 동작시키게 한 것이다.

동작을 순서적으로 따라가 보자. 우선 T_1 기간에는 신호기 A측 통행이 「정지」 적(:R)으로 되어 있으므로 신호기 B측의 통행은 청(:B)으로 「진행」할 수 있다. 다음의 기간 T_2로 시간을 추이하면 B측의 통행은 「주의」, 황(:E)이 된다. T_2 시간이 경과하면 B측은 「정지」, 적이 되고 그에 따라 A측은 청으로 「진행」 가능이 된다. T_3, T_4 기간은 지금 기술한 동작이 A와 B로 바뀔 뿐이다.

운전에 관해서는 다음과 같이 한다. 운전개시 푸시버튼 ON을 누름으로써 신호기가 동작을 개시하여 운전정지 푸시버튼 OFF를 계속 눌러 타임차트에 표시한 T_4까지의 1주기가 끝나고 T_1 기간의 개시점에 도달했을 때 신호의 동작을 정지하게 한 것이다.

그림 7.7에 무접점의 제어회로를 들었다. 이 회로는 전술한 「반복운전회로」의 무접점회로를 응용해서 만든 것이다. 이 때문에 유접점의 제어회로는 표시하고 있지 않다. 또 이 회로의 동작은 타임차트에 표시한 동작추이 시간을 10분의 1로 단축하고 있다.

그림 7.8에 입출력 및 일시기억 번지의 할당, 표 7.2에 프로그램 리스트가 표시되어 있다. 이 프로그램은 제어회로에서 알 수 있듯이 크게 2개 부분으로 나뉘어져 있다. 하나는 00 스텝에서 22 스텝까지, 또 하나는 23 스텝에서 최후까지이다. 이 두 가지는 각 타이머의 시간설정이 다를 뿐이고 동일 내용이다. 그러므로 프로그램 리스트의 앞부분만 보면 충분하다.

01 스텝 OR 18

운전개시 푸시버튼 스위치(00번지) 내용과 일시기억 18번지 내용과의 OR이다. 일시기억 18번지는 그림 7.8의 A측 황색 램프(A_E) 구동회로부에 있으며 00번지에서 떨어져 있으므로 주의하여야 한다.

02~04 스텝 NOP 00

이 3 스텝은 프로그램을 변경했기 때문에 NOP로 메우고 있다. 특별한 의미를 가지는 것은 아니다.

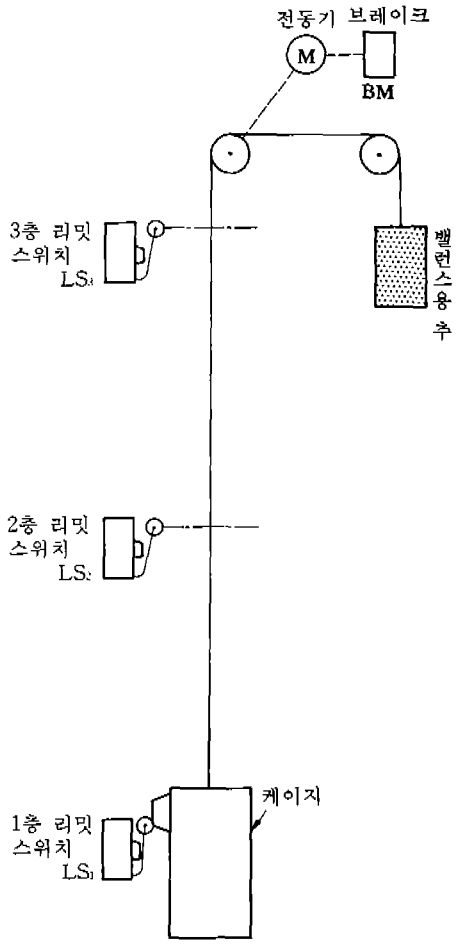
05 스텝 AND 01

운전정지 푸시버튼 스위치를 b 접점에서 부여하고 있으므로 입력측이 부논리이다.

06~0E 스텝 : 그림 7.7의 A측 적색 램프(A_R) 구동회로의 프로그램이다. 번지의 할당, 타이머 시간설정이 잘 되어 있으면 프로그램 코딩 자체는 기계적으로 진행한다.

0F~18 스텝 : A측의 청색 램프(A_B) 구동회로의 프로그램이다. 앞의 06~0E 스텝과 동일하다.

19~22 스텝 : A측의 황색 램프(A_E) 구동회로의 프로그램이다. 여기서 일시기억 18번지에 격납된 데이터가 분기하여 01 스텝에서 사용되고 있다.

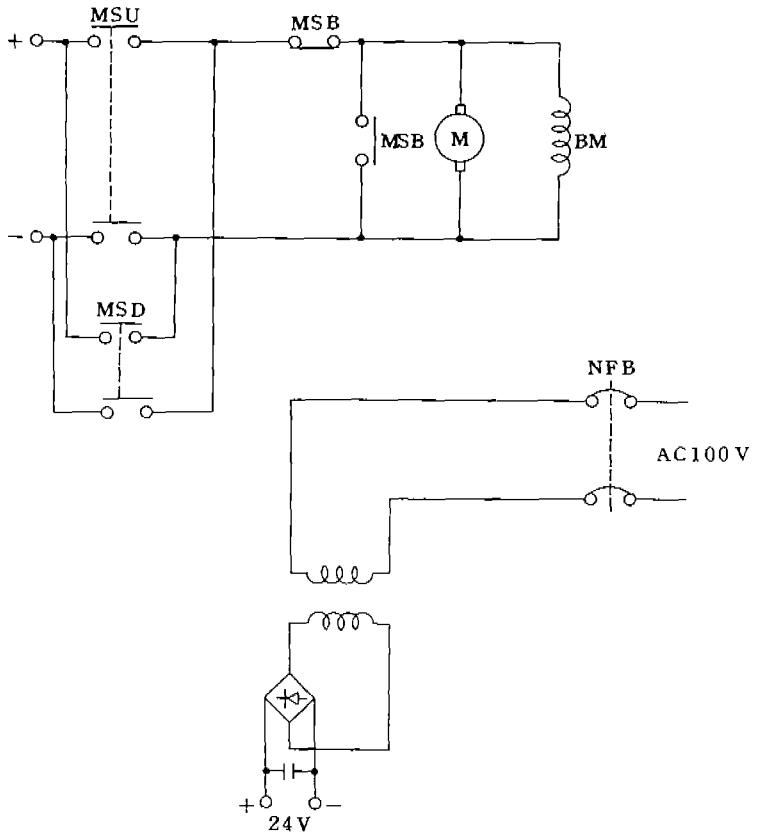


〈그림 7.9〉 리프트 계통도

7.5 간이 리프트 제어

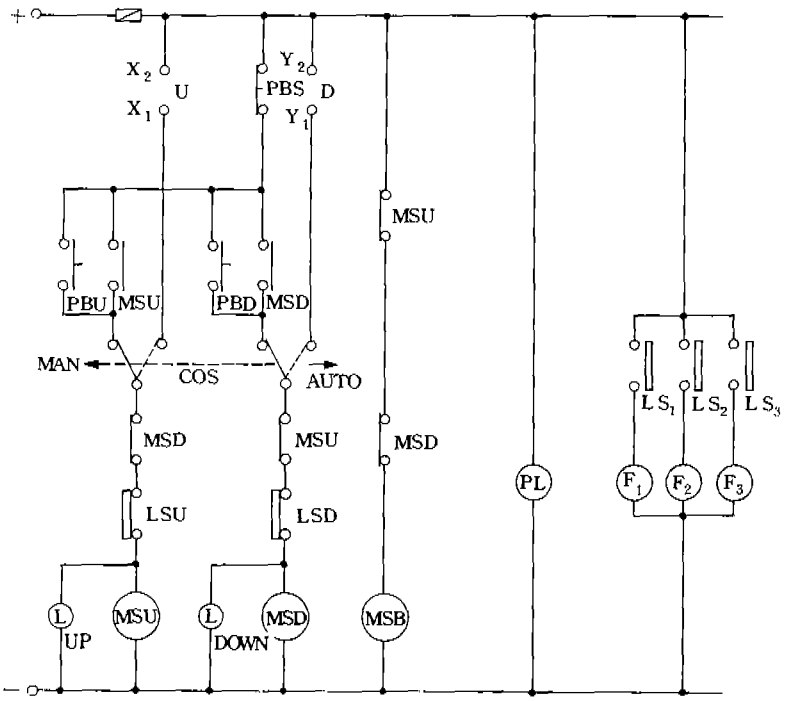
3층의 간이 리프트 제어 예이다. 이 리프트는 그림 7.9와 같이 1층, 2층 및 3층간을 자유롭게 왕복하는 것으로서, 각 층의 케이지 호출 푸시버튼 스위치(그림 7.12의 1C-BS, 2C-BS, 3C-BS)를 조작하면 케이지는 각각 지정된 층에 작용하여 자동 정지하게 되어 있는 것이다.

동력부의 주회로를 그림 7.10, 제어회로(Ⅰ)을



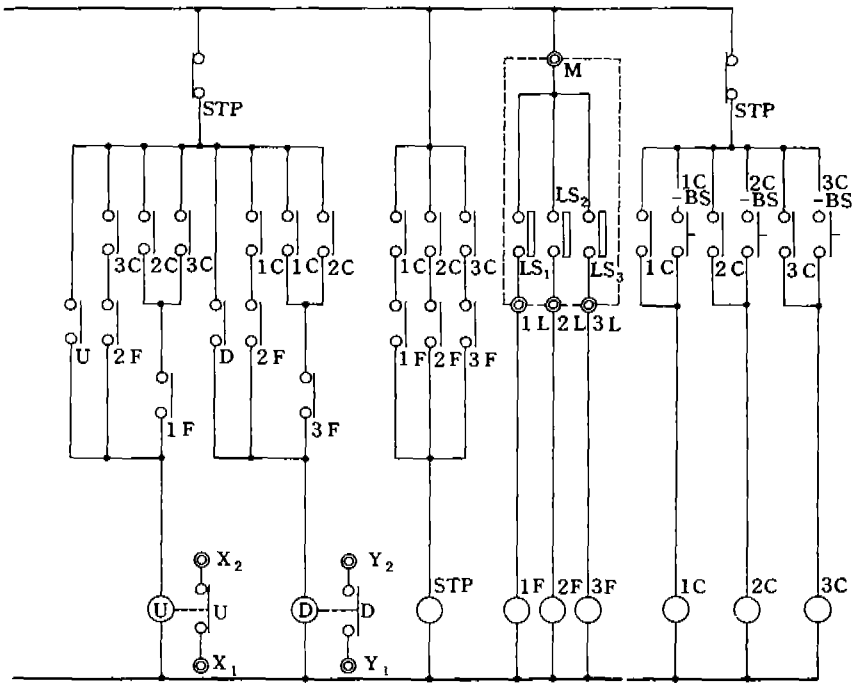
주. MSU : 상승운전용 주전자계전기
MSD : 하강운전용 주전자계전기
MSB : 전자 브레이크
NFB : 배선용 차단기

〈그림 7.10〉 간이 리프트 주회로



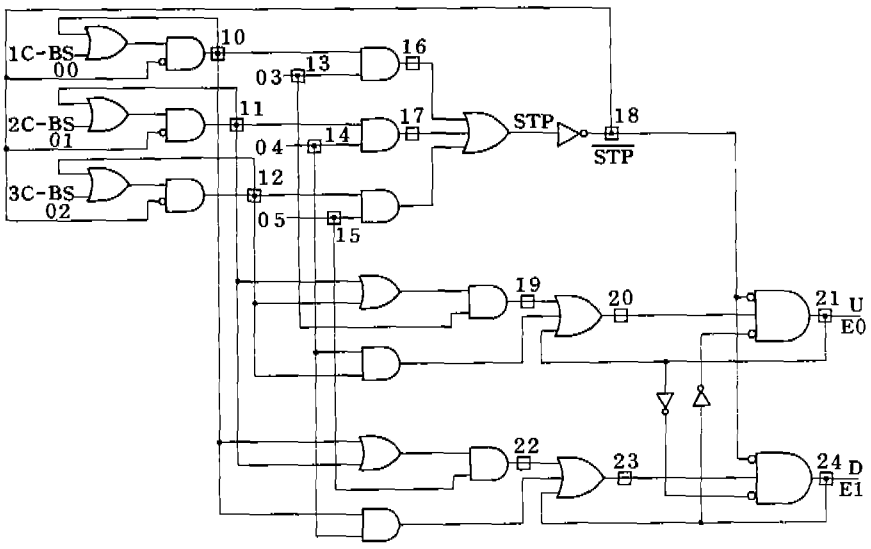
주. MSU : 상승운전용 전자계전기
MSD : 하강운전용 전자계전기
MSB : 전자 브레이크
AUT : 자동
MAN : 수동
COS : 전환스위치
PBS : 정지용 푸시버튼 스위치
PBU : 상승용 푸시버튼 스위치
PBD : 하강용 푸시버튼 스위치
PL : 전원표시등
L_{UP} : 상승표시등
L_{DOWN} : 하강표시등
LS_1 : 1층 위치검출 리밋 스위치
LS_2 : 2층 위치검출 리밋 스위치
LS_3 : 3층 위치검출 리밋 스위치
F_1 : 1층 표시등
F_2 : 2층 표시등
F_3 : 3층 표시등

〈그림 7.11〉 간이 리프트 제어회로[I]

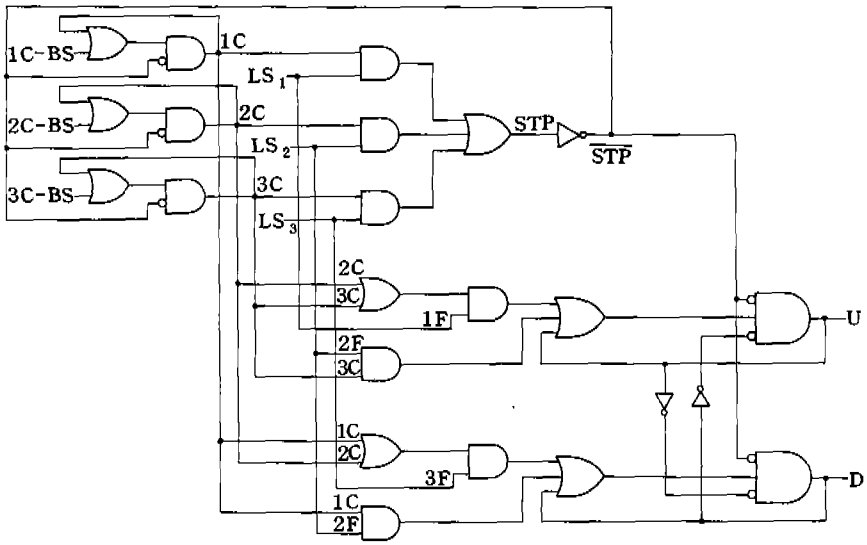


주. 1C : 1층 호출계전기
2C : 2층 호출계전기
3C : 3층 호출계전기
U : 상승용 계전기
D : 하강용 계전기
STP : 정지용 계전기
1F : 1층 위치검출 계전기
2F : 2층 위치검출 계전기
3F : 3층 위치검출 계전기
LS1 : 1층 위치검출 리밋 스위치
LS2 : 2층 위치검출 리밋 스위치
LS3 : 3층 위치검출 리밋 스위치
1C-BS : 1층 표시 푸시버튼 스위치
2C-BS : 2층 표시 푸시버튼 스위치
3C-BS : 3층 표시 푸시버튼 스위치

〈그림 7.12〉 간이 리프트 제어회로[Ⅱ](유접점)



〈그림 7.13〉
간이 리프트 제어회로[Ⅱ]
(무접점)



〈그림 7.14〉 번지의 할당

그림 7.11, 제어회로(Ⅱ)(유접점)를 그림 7.12, 제어회로(Ⅲ)(무접점)를 그림 7.13에 각각 나타낸다. 그림 7.11의 제어회로(Ⅰ)의 U, D를 그림 7.12의 제어회로(Ⅱ)의 상승용 계전기 U와 하강용 계전기 D로 각각 개폐하게 된다.

제어회로(Ⅱ)를 PC의 프로그램으로 실현, 제어한다. 입출력 및 일시기억 번지의 할당을 그림 7.14에, 프로그램 리스트를 표 7.3에 들었다. 그림 7.13, 그림 7.14에 논리도와 번지 설정이 이미 되어 있으므로 프로그램은 거의 기계적으로 만들어진다. 이하, 프로그램을 간단히 보기로 하자.

00~03 스텝 : 1층 지시 푸시버튼 스위치 관련부분회로의 프로그램이다. 동작 정지를 위한 신호는 일시기억 18번지의 내용에서 보내어져 오게 되어 있다.

04 스텝　LD 03

05 스텝　STO 13

04 스텝에서 읽어 넣은 내용을 분기하고 있으므로 05 스텝에서 13번지에 그 내용을 격납하고 있다.

그러나 그와 같은 경우는 STO 없이 지정한 리밋 스위치의 입력번지 그 자체를 분기 각 부분의 번지로서 사용할 수도 있다. 예를 들면

04 스텝　LD 03

05 스텝　NOP 00(불필요*)

로서 분기 데이터를 사용하는 1D 스텝을 다음과 같이 한다.

AND 13을 → AMD 03

06 스텝　AND 10

07 스텝　STO 16

그림 7.14의 일시기억 16번지 설정개소까지의 내용을 연산처리 대기를 위해 16번지에 일단 격납하고 있다.

08~0F 스텝 : 위의 00~07 스텝과 동일 패턴의 프로그램이다.

10~16 스텝 : 이것도 위의 것과 동일한 패턴이지만 최후의 STO가 불필요해진다.

17 스텝　OR 16

18 스텝　OR 17

16 스텝의 연산 결과와 일시기억 16, 17번지에 각각 격납하고 있는 내용과의 OR이다.

19 스텝　INV 00

1A 스텝　STO 18

OR의 결과를 반전하지만 그것이 다음에 몇 개로 분기하고 있으

* 13번지에의 일시기억이 불필요해졌기 때문에 이 스텝의 명령이 필요없고 NOP로 하고 있는 것.

〈표 7.3〉 프로그램 리스트

스 텝 No.	명 령 OP	명 령 ADD	비 고
00	LD	00	
01	OR	10	
02	AND	18	
03	STO	10	
04	LD	03	
05	STO	13	
06	AND	10	
07	STO	16	
08	LD	01	
09	OR	11	
0A	AND	18	
0B	STO	11	
0C	LD	04	
0D	STO	14	
0E	AND	11	
0F	STO	17	
10	LD	02	
11	OR	12	
12	AND	18	
13	STO	12	
14	LD	05	
15	STO	15	
16	AND	12	
17	OR	16	
18	OR	17	
19	INV	00	
1A	STO	18	
1B	LD	11	
1C	OR	12	
1D	AND	13	
1E	STO	19	
1F	LD	12	

스 텝 No.	명 령 OP	명 령 ADD	비 고
20	AND	14	
21	OR	19	
22	OR	21	
23	STO	20	
24	LD	24	
25	INV	00	
26	AND	18	
27	AND	20	
28	STO	21	
29	OUT	E0	
2A	LD	10	
2B	OR	11	
2C	AND	15	
2D	STO	22	
2E	LD	10	
2F	AND	14	
30	OR	22	
31	OR	24	
32	STO	23	
33	LD	21	
34	INV	00	
35	AND	18	
36	AND	23	
37	STO	24	
38	OUT	E1	
39			
3A			
3B			
3C			
3D			
3E			
3F			

므로 일단 이 내용을 일시기억 18번지에 격납한다.

1B~29 스텝 : 상승 모터 구동회로의 프로그램부이다.

1B 스텝 LD 11

에서 일시기억 11번지의 내용을 판독해 넣고 다음의 1C, 1D 스텝에서 OR, AND로 계속하여 그 결과를 연산처리 대기 때문에 0E 스텝에서 19번지에 격납하고 있다.

이 이후는 특별한 문제는 없다. 단, 참조하는 번지가 도면상에서 떨어진 위치에 있으므로 돌리지 않도록 번지를 선택하는 주의가 필요하다.

2A~38 스텝 : 하강 모터 구동회로의 프로그램으로 위의 1B~29 스텝의 패턴과 동일하다.

☞ 다음호에 계속 ☜