

현장 기술자를 위한

소형 PC에 의한 시퀀스 제어

현·장
기·술

5

역 / 박 한 중 (협회 교육홍보위원)

제4장 논리 연산과 프로그래밍

4.2 논리 연산 기능의 프로그램 코딩

(1) 번지 할당

프로그램 코딩에 있어서 우선 입출력 및 일시 기억 번지의 할당을 한다.

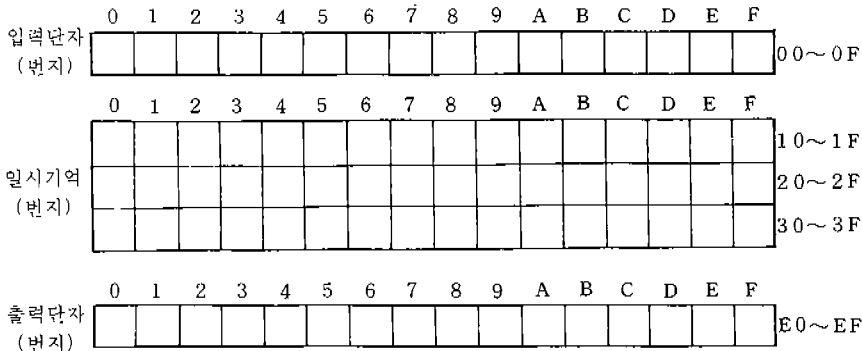
입력 번지는 00~0F간, 출력 번지는 E0~EF간 각각 16점 중에서 할당한다.* 할당한 입출력 번지에는 이것과 대응한 입출력 단자가 설치되어 있으므로 이 단자와 입출력 기기를 접속하는 것이 된다.

일시 기억 번지의 할당은 10~3F간의 48점 중에서 한다. 단, 이 중의 30~3F의 16점은 타이머의 일시 기억 번지로서 사용하는 곳이다. 그 때문에 타이머로서 사용한 번지는 그 이외의 일시 기억 번지로서는 사용할 수 없게 된다.

일시 기억 번지의 설정 개소는 무접점 회로도의 다음과 같은 곳으로 한다.

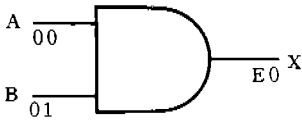
1. 연산결과가 분기하는 개소
2. 연산처리 대기가 생기는 개소

구체적인 것에 대해서는 설정 개소가 문제가 되는 제5장에서 기술한다.



<그림 4.1>
입출력 및 일시 기억의
설정 번지군

* 기종에 따라서는 번지를 10진수로 표시하는 것도 있다. 그리고 입출력 점수는 여기서 들은 기종이 위에 표시한 것과 같이 되어 있어 그 예를 표시한 것이고, 입출력 점수는 그 PC의 규모에 따라 변화하는 것이다.



<그림 4.2> AND회로

(2) AND

번지의 할당

프로그램 코딩에 앞서서 그림 4.2와 같이 회로도에 번지를 할당한다. 위에서 기술한 입출력의 소정 번지군에서 선정하여 입력 A를 00, B를 01, 출력 X를 E0로 한다.

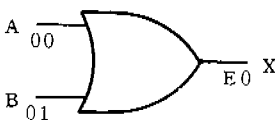
코딩

표 4.1에 AND회로의 프로그램 리스트를 나타낸다. 명령 사용방법, 코딩 순서 모두 거의 문제없다고 생각되지만 최초이기도 하여 확실히 다짐하기 위해 설명하여 둔다.

- 00 스텝 LD 00
어드레스에서 지정하는 번지(00)의 내용을 판독 입력한다.
- 01 스텝 AND 01
어드레스에서 지정하는 번지(01)의 내용과 앞에서 판독 입력한 00번지 내용의 AND를 취한다.
- 02 스텝 OUT E0
출력번지(E0)에 위의 결과를 출력한다.

동작 확인

프로그램을 실행하고 입력 A, B의 신호를 조작용 스위치에서 부여한다. 입력 A, B 모두 ON(「H」레벨)으로 했을 때만 출력 LED가 점등(출력이 「H」레벨)하면 AND 동작이다.



<그림 4.3> OR회로

<표 4.1> AND회로 프로그램 리스트

스텝 No.	명령		비고
	OP	ADD	
00	LD	00	
01	AND	01	
02	OUT	E0	

PC 조작방법은 이미 제2장에서 기술한 바 있지만 잊었다면 다시 한번 제2장을 보기 바란다. 그리고 후술하는 조작방법도 참고하기 바란다.

(3) OR

번지의 할당

앞의 AND와 동일하게 그림 4.3과 같이 할당한다.

코딩

표 4.2에 프로그램 리스트를 나타낸다. AND의 경우와 다른 것은 01 스텝의 OR 01만이다.

동작 확인

입력 A, B 어느 한 쪽을 ON하였을 때 출력 LED가 점등하면 OR 동작이다.

(4) NOT

번지의 할당

그림 4.4와 같이 할당한다.

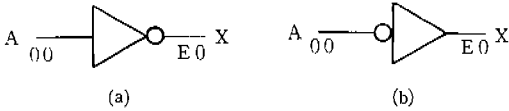
코딩

표 4.3에 프로그램 리스트를 나타낸다. 여기서 주의할 것은 표 4.3(a)의 01 스텝이다.

- 01 스텝 INV 00
먼저 00 스텝에서 판독 입력한 내용의 INV를 취한다. INV 00에서의 어드레스 00은 이 경우 00번지라는 의미가 아니고 부정을 취하는 경우의 「결정」

<표 4.2> OR회로의 프로그램 리스트

스텝 No.	명령		비고
	OP	ADD	
00	LD	00	
01	AND	01	
02	OUT	E0	



〈그림 4.4〉 NOT회로

으로 생각하기 바뀐다.

동작 확인

그림 4.4(a)에서는 입력 A는 a접점 푸시버튼 스위치로 부여되므로 조작용 스위치를 ON하는 것은 푸시버튼 스위치를 조작한 것(누른 것, 능동 「1」)에 대응한다. 이 때, 즉 능동 「1」에서 출력 LED가 꺼지면 NOT 동작이다.

A를 OFF하면 출력 LED는 점등한다.

한편, 그림 4.4(b)에서는 입력 A는 b접점 푸시버튼 스위치로 부여되므로 조작용 스위치를 ON하는 것은 푸시버튼 스위치를 조작하지 않는 것(누르지 않는 것, 비능동 「0」)에 대응한다. 이 때, 즉 비능동 「0」으로 출력 LED가 점등하면 이것도 역시 NOT 동작이다.

A를 OFF하면 출력 LED는 꺼진다.

노트 ①

2개의 NOT회로와 프로그램의 상위

동일한 NOT인데 왜 프로그램이 다른가를 생각

〈표 4.3〉 NOT회로의 프로그램 리스트

스텝 No.	명령		비고
	OP	ADD	
00	LD	00	
01	INV	00	
02	OUT	E0	

(a)

스텝 No.	명령		비고
	OP	ADD	
00	LD	00	
01	OUT	E0	

(b)

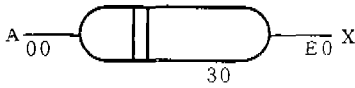
해 보자.

이 상위는 앞의 제3장 「3.5 정논리와 부논리」에서 기술한 바와 같이 입력 A의 푸시버튼 스위치의 조작(누름 「1」, 휴지 「0」)과 전압 레벨의 관계에 의해 생기는 것이다. 이 관계를 표 4.4에 들었다. (a), (b)의 두 회로도, 진리값표에서 프로그램이 왜 달라지는가를 알 수 있을 것이다.

보충해서 설명하면 다음과 같이 된다. 동작 확인에서 안 것과 같이 푸시버튼 스위치의 조작 (A)와 출력 X간에는 확실히 (a), (b) 모두 NOT가 성립한다. 그러나 진리값표를 보면 알 수 있듯이 (a)에서는 입력-출력간에 전압 레벨의 반전이 있으므로 프로그램에 INV 00이 필요하지만 (b)에서는 반전이 없으므로 이 명령이 불필요한 것이다.

〈표 4.4〉 두가지 NOT회로의 상위

	(a)	(b)																								
회로도																										
진리값표	<table border="1"> <thead> <tr> <th colspan="2">입력 A</th> <th>출력 X</th> </tr> <tr> <th>푸시버튼 조작</th> <th>전압 레벨</th> <th>전압 레벨</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>L</td> <td>H</td> </tr> <tr> <td>1</td> <td>H</td> <td>L</td> </tr> </tbody> </table>	입력 A		출력 X	푸시버튼 조작	전압 레벨	전압 레벨	0	L	H	1	H	L	<table border="1"> <thead> <tr> <th colspan="2">입력 A</th> <th>출력 X</th> </tr> <tr> <th>푸시버튼 조작</th> <th>전압 레벨</th> <th>전압 레벨</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>H</td> <td>H</td> </tr> <tr> <td>1</td> <td>L</td> <td>L</td> </tr> </tbody> </table>	입력 A		출력 X	푸시버튼 조작	전압 레벨	전압 레벨	0	H	H	1	L	L
입력 A		출력 X																								
푸시버튼 조작	전압 레벨	전압 레벨																								
0	L	H																								
1	H	L																								
입력 A		출력 X																								
푸시버튼 조작	전압 레벨	전압 레벨																								
0	H	H																								
1	L	L																								
불식	$X = \bar{A}$	$X = \bar{A}$																								



〈그림 4.5〉 타이머

노 트 ②

무접점 회로도에 있어서의 입력 유닛의 생략

표 4.4의 회로도에 표시되어 있듯이 입력 푸시 버튼 스위치는 입력 유닛 **IU**를 거쳐 접속하게 되어 있다.

그러나 이제부터의 회로도에서는 이 입력 유닛은 특별하게 필요가 없는 한 회로도에는 표시하지 않는 것으로 한다. 이 때문에 입력 유닛에 접속하는 푸시버튼 스위치는 입력측에 상대표시 기호(○)가 있는가 없는가로 a접점을 접속하는가 b접점을 접속하는가를 판단하게 된다.

(5) 타이머

번지의 할당

그림 4.5와 같이 할당한다. 유의할 것은 타이머 번지의 설정이다. 타이머는 일시 기억 번지군의 30~3F간의 16점에서 선택한다. 여기서는 30번지로 한다.

코 딩

표 4.5에 나타낸다. 01과 02 스텝에 주목한다.

01 스텝 TIM 40

2초의 타이머로서 일시 기억 메모리 30번지를 사용하는 것이다. 어드레스 처음의 4는 시간 설정(2초)이다. 다음의 0은 30번지의 1자리째의 0을 지칭한다. 즉, 여기에는 16진수 30번대의 1자리째의 0~F만을 쓰고 2자리째의 3은 타이머의 경우 이 30번대에 고정을 위해 생략한다. 타이머의 설정시간 T[초]는 다음 식에 의해 계산한다.

$$T = \text{설정값}(1 \sim F) \times 0.5$$

02 스텝 LD 30

타이머로서 사용한 일시 기억 메모

〈표 4.5〉 타이머의 프로그램 리스트

스텝 No.	명 령		비 고
	OP	ADD	
00	LD	00	
01	TIM	40	
02	LD	30	
03	OUT	E0	

〈표 4.6〉 프로그램 리스트

스텝 No.	명 령		비 고
	OP	ADD	
00	LD	00	
01	TIM	F0	
02	LD	30	
03	TIM	51	
04	LD	31	
05	OUT	E0	

리 30번지를 판독 입력한다. 이 LD 30의 1자리째와 위의 TIM 40의 1자리째의 숫자를 동일하게 한다. 만일 일시 기억 번지를 31로 하면 LD 31에 대해서 TIM 41로 한다.

이상에서 알 수 있듯이 01 스텝의 TIM 40과 02 스텝의 LD 30은 1세트로 되어 있다. 타이머의 경우는 이와 같이 TIM과 LD는 항상 세트로 사용하게 되어 있다.

동작확인

입력 A를 ON으로 하고 2초 후에 출력 LED가 점등하면 타이머 동작이다.

노 트

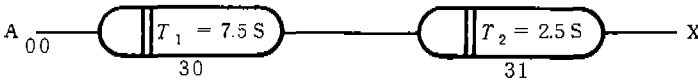
타이머의 설정시간을 연장하는 방법

하나의 TIM 명령으로 만들 수 있는 타이머 설정시간 T는 이 기종의 PC는 최대 $F \times 0.5 = 7.5$ 초이다. 이것으로는 짧은 경우는 다음과 같이 해서 시간을 연장한다.

예를 들면 10초의 타이머를 만들고 싶을 때는 7.5초까지는 30번지의 타이머로 설정하고 나머지 2.5초를 다음의 31번 타이머로 만들면 되므로

$$\text{설정값} = 2.5 / 0.5 = 5$$

로 하면 된다.



<그림 4.6> 타이머의 설정시간을 연장하는 방법

그림 4.6 및 표 4.6에 그 결과를 나타낸다.

제5장 기본 시퀀스 제어회로의 프로그래밍과 실행

이 장에서는 드디어 본격적인 시퀀스 제어회로인 자기 유지, 한시 동작, 인터록, 한시 동작의 이행과 같은 기본회로의 프로그래밍과 그 실행에 대해서 기술한다. 유접점회로가 이미 부여되어 있으므로 이것을 어떻게 프로그래밍하는가를 생각해 보자.

제3장에서는 무접점 논리회로와 유접점회로를 무접점회로에 변환하는 방법에 대해서 기술하였다. 여기에는 제3장의 성과에 입각하여 무접점회로로 변환한 것이 표시되어 있다. 그러나 각자 유접점회로를 논리적으로 보는 연습이라는 의미에서 각각의 회로에 대해서 유접점→무접점회로에의 변환을 하여 보면 좋을 것이다.

입력기기 사용방법은 다음과 같이 하였다. 기동(세트) 푸시버튼 스위치와 같이 유접점회로에 있어서 a접점으로 사용하고 있으면 무접점회로라도 원칙적으로 a접점을 사용하고 정지(리셋)와 같이 b접점의 것은 동일하게 무접점회로라도 b접점을 사용하고 있다.

그리고 이제부터 나오는 회로는 제어회로만이 표시되고 있고 주회로에 대해서는 상세히 기술하지 않는다. 주회로의 접점용량은 부하의 크기에 따라 선택된다. 부하가 작은 경우는 직접 부하를 출력 유닛에 접속할 수 있다. 그러나 부하가 커지면 부하를 전자접촉기로 개폐하므로 출력 유닛에 전자접촉기를 접속하게 된다.

상세한 것에 대해서는 제7장에서 기술한다. 그러므로 여기서는 입력은 조작용 스위치로 부여하고 출력상태를 출력 유닛의 LED로 관찰한다. 관찰 결과를 바탕으로 제어회로의 동작이 정상인가의 여부를 확인하게 하고 있다.

5.1 자기유지회로

(1) 유접점 회로도

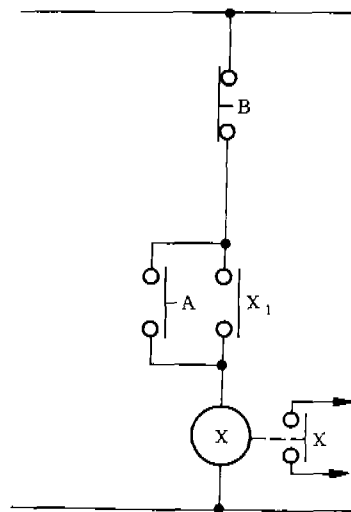
그림 5.1은 제3장에서 취급한 회로이므로 특별한 설명이 필요없는 것이다.

(2) 무접점 회로도

그림 5.1을 불(Boolean)식으로 표현하면 다음과 같이 된다.

$$X = (A + X_1) \cdot \bar{B} \quad \dots \dots \dots (5.1)$$

위의 식을 중개로 하여 무접점회로로 변환한 것을 그림 5.2에 나타낸다. 정지용 푸시버튼 스위치에 b접점을 사용하므로 입력측에 상태표시 기호(○)를 달아야 한다.



<그림 5.1> 유접점 자기유지회로

<표 5.1> 자기유지 프로그램 리스트

스텝 No.	명령		비고
	OP	ADD	
00	LD	00	
01	OR	10	
02	AND	01	
03	STO	10	
04	OUT	E0	

(3) 번지의 할당

입출력 번지는 전술한 요령으로 지정 번지군 중에서 각각 그림 5.3과 같이 할당한다.

일시 기억 번지에 대해서는 다음과 같이 한다. 우선 설정개소는 분기개소가 1개소 있으므로 그곳에 설정한다. 연산처리 대기는 여기에는 없다. 다음에 번지의 할당은 소정의 10~3F 번지내에서 선정하여 그림과 같이 10으로 할당한다.

(4) 코딩

01 스텝 OR 10

분기개소의 일시 기억 번지 10과의 OR를 취한다. 이 10번지는 뒤에 이 번지까지의 연산처리가 종료한 시점에서 STO 명령으로 STORE하여 두어야 한다.

02 스텝 AND 01

입력 B : 01번지는 b접점의 푸시버튼 스위치로 부여되는 것이므로 입력 측이 부논리(負論理)로 상태표시 기호

(○)가 달려 있다. 이 때문에 01 스텝의 OR 출력과 이 01 번지와 AND를 취하면 되게 된다. 부정 기호가 있다고 해서 부정(INV 00)으로 하면 틀리게 된다.

이 점에 주의가 필요하다.

03 스텝 STO 10

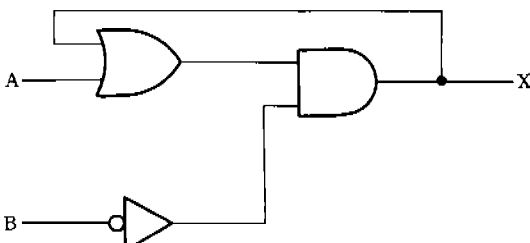
10번지가 분기개소로 되어 있으므로 여기에 일시 기억 번지 10을 할당한다. 일시 기억 번지를 할당했을 때는 이와 같이 연산처리가 끝난 데서 결과를 STORE하여 두지 않으면 안된다. 이 명령이 없으면 소정의 회로 동작이 되지 않는다."

(5) 동작 확인

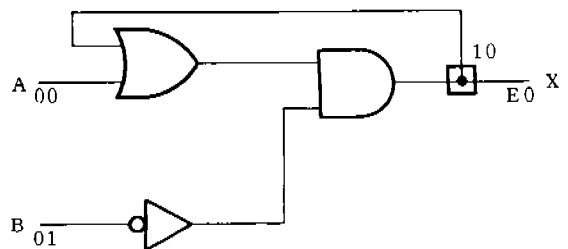
프로그램을 실행하고 입력 A, B를 조작용 스위치로 부여하기 바란다. 입력 A는 수동조작 자동복귀의 a접점, 입력 B는 수동조작 자동복귀의 b접점 이므로 이와 같이 조작용 스위치를 조작한다. 즉, 입력 B는 처음부터 ON하여 두고 입력 A를 순시 ON 조작하도록 한다.

이 조작을 하였을 때 출력이 ON하고 LED가 계속 점등하는 자기유지 동작을 하고 있는가의 여부를 확인한다. 그리고 입력 B를 순시 OFF 조작하면 출력은 OFF가 되고 출력 LED가 꺼진다.

다음에 이 회로는 리셋 우선 자기유지회로이므로 리셋 우선 동작도 확인해 두어야 한다. 이것은 입력 A, B가 동시에 조작되면 리셋 푸시버튼 스위치 B가 우선해서 동작 정지가 되는 것이다.



<그림 5.2> 무접점 자기유지회로



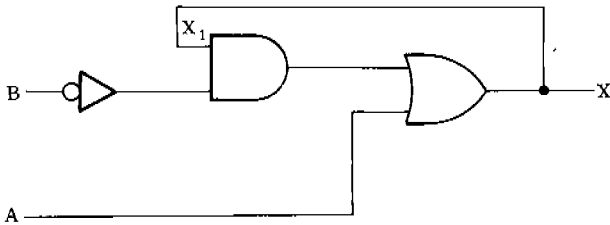
<그림 5.3> 번지 할당

** 일부 STO 명령을 사용하지 않고 프로그램 코딩하는 방법이 없는 것은 아니다. 이것에 대해서는 후술한다.

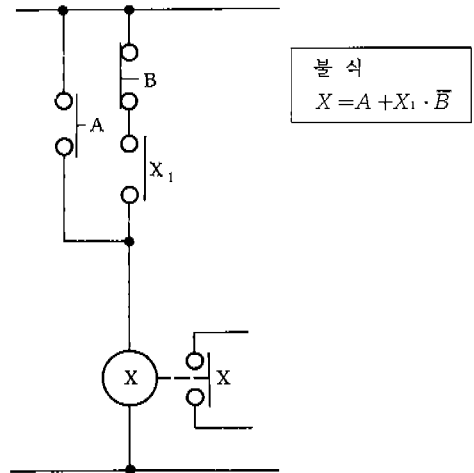
연습문제 5.1

그림 5.4에 세트 우선 자기유지회로의 유접점 회로도라 이것과 동일 내용의 무접점 회로도가 있다. 그리고 참고를 위해 불식도 표시하였다.

이 무접점 회로상에 번지를 할당하여 이 회로와 동일한 동작을 하는 프로그램을 만들 것.



〈그림 5.4(b)〉 세트 우선 자기유지회로(무접점)



불 식
 $X = A + X_1 \cdot \bar{B}$

〈그림 5.4(a)〉 세트 우선 자기유지회로(유접점)

☞ 다음호에 계속 ☞

일본 어투 생활용어 순화

순화대상용어	순화한 용어	순화대상용어	순화한 용어	순화대상용어	순화한 용어
구로(黒)	검정	구루마(車)	수레, 달구지	구립뿌(clip)	클립
쿠사리(腐)	면박, 핀잔	쿠세(癖)	①버릇, ②다트	구인(拘引)	끌어감
구좌(口座)	계좌	구지비끼(籤引)	제비뽑기, 추첨	구찌(口)	몹
구찌베니(口紅)	입술(연지)	구찌뽀찌(punch)	입심, 말숨씨	금반(今般)	이번
금회(今回)	이번	급사(給仕)	사환, 사동	기라성(綺羅星)	빛나는 별
기레빠시(切端)	꼬트러기 자투리	기리카이(切替)	바꾸기, 교체	기마이(氣前)	선심, 호기
기소(基礎)	기초, 밑바탕	기스(傷)	흠(집)	기지(生地)	천
꼬분(子分)	부하	깡깡(金柑)	금귤, 동귤	나가리(流)	① 유찰, ② 깨짐
나래비(並)	줄서기	나라시(均)	고루펴기	나마가시(生菓子)	생과자
나마비류(生麥酒)	생맥주	나베우동	냄비국수	나오시(直)	고침질
나이타(nighter)	야간 경기	나카마(仲間)	한패, 거간, 중간상	나닌구(running shirts)	러닝 셔츠
남바(number)	번호, 넘버	낫또(nut)	암나사, 너트	네지(螺子)	나사(못)
네지마와시(螺子廻)	나사돌리개	네다바이	사기, 야바위	노가다(土方)	(공사판)노동자
노견(路肩)	갓길	노짱(土管)	토관	노리마끼(海苔卷)	김밥
노리카에(乗替)	갈아타기	니꾸사구	배낭	니부(二分)	두푼
니구돈부리(肉井)	고기덮밥	닌징(人蔘)	당근	다라이	(큰)대야
다마(珠, 球)	구슬, 전구	다마네기(玉蔥)	양파	다무시(田蟲)	벼집, 백선