현장 기술자를 위한

소형 PC에 의한 시퀀스 제어

현·장 기·술 3

역 / 박 한 종(협회 교육홍보위원)

제3장 프로그램 작성의 기초로서의 논리회로

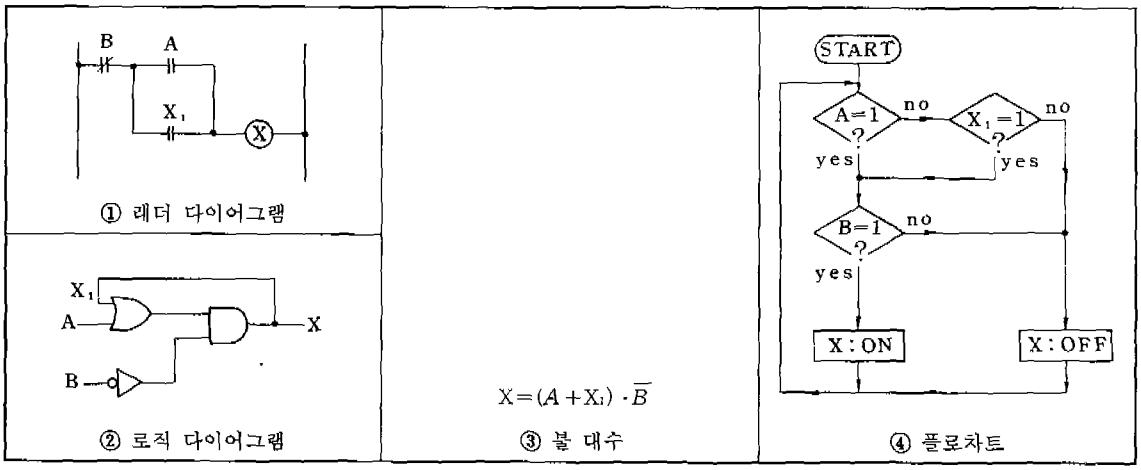
1. 시퀀스 제어 동작의 표현과 프로그램 작성

제2장에서 PC를 사용하여 모터를 제어해 보았다. 그 제어내용은 자기유지회로라는 것이었다. 그러나 내용 그 자체에 대해서는 거의 언급하지 않았다. 따라서 제3장에서는 이 제어동작 내용이라는 것이 어떻게 표현되는가 하는 것에 대해서 생각해 보기로 한다.

그리고 이것은 제4장이 되겠지만 표현된 제어동작 내용을 PC의 프로그램으로서 키 인할 수 있는 형태로 정리하려면 어떻게 생각하면 되는가 하는 것에 연결되는 것이다.

PC의 프로그램을 작성할 때 프로그램의 원시정보(프로그램의 원안)를 부여하는 시퀀스 제어동작의 표현은 대략적으로 다음 네가지로 분류된다.

① 유접점 심볼(래더 다이어그램)방식
② 무접점 심볼(로직 다이어그램)방식
③ 불식방식



〈그림 1.1〉「자기유지회로」에 대한 각 표현방법의 대응

④ 플로차트방식

「자기유지회로」에 대해서 이 각각의 표현방법이 어떻게 대응되고 있는가를 그림 1.1에 나타냈다.

본고에서 채용한 것은 ②의 무접점 심볼(로직 다이어그램)방식이다. 기타 방식도 소개한 이유는 왜 이 로직 다이오그램방식을 채용했는가를 명확히 하기 위해 약간의 설명을 하고 싶었기 때문이다.

그리고 프로그램 코딩(Program coding : PC의 명령 코드를 사용하여 프로그램을 만드는 작업)에 있어서는 프로그램 언어를 생략형으로 표시한 니모닉 코드(LD : Load, OUT나 AND, OR와 같은 명령의 약호)로 기술하는 것과 기능 심볼(⊣⊢, ⊣/⊢ 등의 접점 심볼이나 +, *, ( ) 등의 연산기호)로 기술하여 CRT 디스플레이(브라운관-CRT, cathode ray tube-상에 여러 가지 정보를 표시하는 것)상에서 편집을 하는 것이 있다.

각 PC 메이커방식은 ①의 래더 다이어그램으로 동작 표현을 하고 니모닉 코드나 또는 유접점 기능 심볼로 코딩하는 것이 주류로 되어 있다. 이것은 래더 다이어그램이 시퀀스 제어회로 표현법으로서는 가장 일반적으로 채용되어 온 경위가 있고 PC의 수요자인 현장 조작자나 안전관리담당자에게 가장 친숙한 회로표현법이라고 생각된다.

그러나 본래 래더 다이어그램은 어느 접점에 어느 코일이 연결되어 있는가를 표시한 것이고 논리의 흐름이나 구조를 표현하는 데는 반드시 적합한 것은 아니다.

이 때문에 래더 다이어그램으로부터의 프로그램 코딩시에 약간의 제약조건이나 언어 사용방법에 불명료성이 남아 이것이 수요자에게는 PC가 알기 쉽고 사용하기 쉬운 것으로 되지 못하는 원인이 된다고 본다.

PC는 시퀀스 제어전용 컴퓨터로서 그 취급이 쉽다고는 하지만 본질적으로는 컴퓨터와 동일한 기본구성으로 되어 있기 때문에 우리들은 제어내용을 어떻게 논리적으로 프로그램할 수 있는가의 기본적 사고방식이 몸에 익숙해 있지 않으면 안된다고 생각된다.

이러한 실력이 있으면 앞으로 출현할 각종 PC에 대해서도 우리들은 그 때마다 교육을 받지 않더라고 PC 메이커에서 발행되는 간단한 취급설명서를 보면 PC를 사용할 수 있게 될 것으로 생각한다.

이상과 같은 점을 감안하여 본고에서는 무접점 로직 심볼방식으로 제어내용을 표현하고 이것을 니모닉 코드로 기술하는 방법을 채용하여 이것으로 설명을 진행해 나가기로 한다.

③ 불식방식, ④ 플로차트방식은 다음과 같은 이유 때문에 채용하지 않았다.

불식방식은 불식으로 시퀀스 제어동작의 전부를 표현하는 것이 불가능하다는 것과(예를 들면 타이머 기능의 표현을 할 수 없다) 또 이 방식에 의한 기종은 각 PC 메이커에서 거의 생산하고 있지 않기 때문이다.

플로차트방식은 컴퓨터의 플로차트(flow chart : 흐름선도라고 하며, 제어의 흐름을 나타내는 도표)와 동일한 것으로서, 동작표현법으로서는 확실히 효과적이다. 문제점은 플로차트라는 것은 본래 제어처리의 흐름(플로)을 표현하는 것으로서, 제어신호가 어떻게 흐르고 있는가는 모르는 것이다.

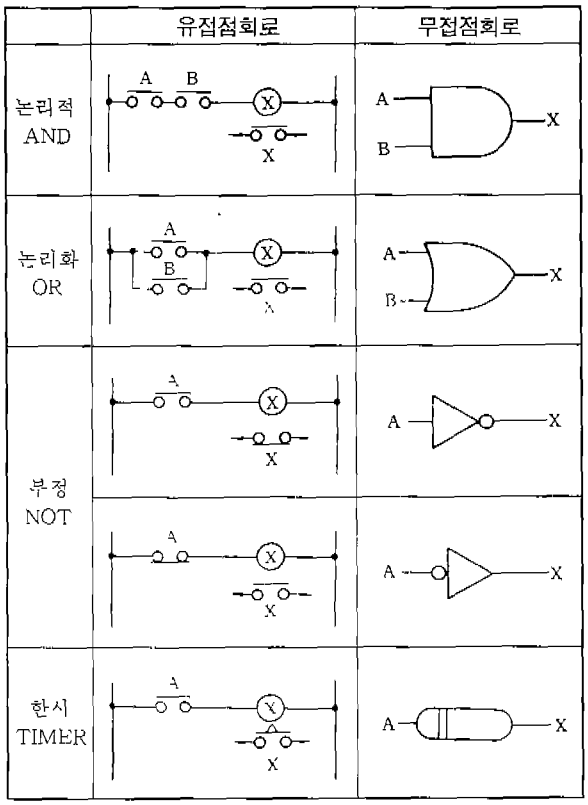
이 때문에 보전, 고장진단시에는 예를 들면 회로의 일목요연성을 바랄 수 없고 입력↔출력간 추적이 곤란하다고 하는 불편이 있다. 그리고 또 시퀀스 제어 현장의 조작자나 안전관리담당자에게 플로차트 그 자체가 친숙하지 않다는 것을 들 수 있다.

2. 무접점 로직 심볼도

전자 릴레이 시퀀스 제어의 경우 실용회로의 대부분은 기본적인 네가지 회로구성 요소의 조합에 의해 표현된다. 네가지 기본요소란 논리적(직렬조건) : AND, 논리합(병렬조건) : OR, 부정 : NOT, 한시 : TIMER이다.

이것을 그림 2.1에 나타냈다. AND, OR 요소에는 2입력인 경우의 예가 표시되어 있지만 당연히 3 이상의 입력도 있다. 또한 유접점회로(전자 릴레이로 표현한 회로)와 등가인 무접점회로(IC 또는 트랜지스터로 표현할 수 있는 회로)가 병기(倂記)되어 있다.

무접점회로에 사용한 기호는 전자회로용 논리기호로서 통일된 미국의 ANSI 규격을 채용하였다. 이것을 채용한 이유는 뒤에 설명하는 정논리(正論



〈그림 2.1〉 기본회로 구성요소

理), 부논리의 논리 레벨상태의 판단이 가능하고 동작상태를 논리회로 기호로 판단할 수 있어 도면을 그리거나 볼 때 논리의 절차가 확실하기 때문이다. 이 때문에 논리의 오차 발견도 용이해진다.

논리기호는 이 밖에 JIS 규격(JIS C 6271 : 한국은 KS C 5603)이 있다. 그러나 전자기기나 컴퓨터에 사용되는 논리기호는 미국의 ANSI 규격에 의해 표현되는 기회가 많은 것 같다. 그러므로 IC 등으로 회로를 만든 경험이 있고 미국의 구 MIL 규격에 길들여진 사람들이 PC를 사용하는 경우에 혼란이 없도록 하기 위한 고려도 하였다.

3. 무접점회로와 유접점회로

전술한 바와 같이 본고는 주로 시퀀스 동작의 원시정보를 무접점회로에서 취하고 이것을 PC의 프로그램으로 만드는(코딩하는) 방법에 대해서 기술하고 있다. 그러므로 무접점회로에 대한 요소 심볼과 기능에 대해서 정리된 이해가 없으면 안된다.

한편, 전자 릴레이 시퀀스 제어계를 전자 릴레이로 바꾸어 PC로 제어코자 할 때 PC 취급자는 여러분들과 같이 지금까지 현장에서 릴레이 시퀀스를 취급하고 있던 사람이 대부분이라고 예상된다. 이와 같은 무접점회로에 소원한 사람들에게는 무접점 요소의 심볼에 빨리 친숙해지고 기능에 대한 이해를 돕기 위한 교육적인 뒷받침이 필요하다.

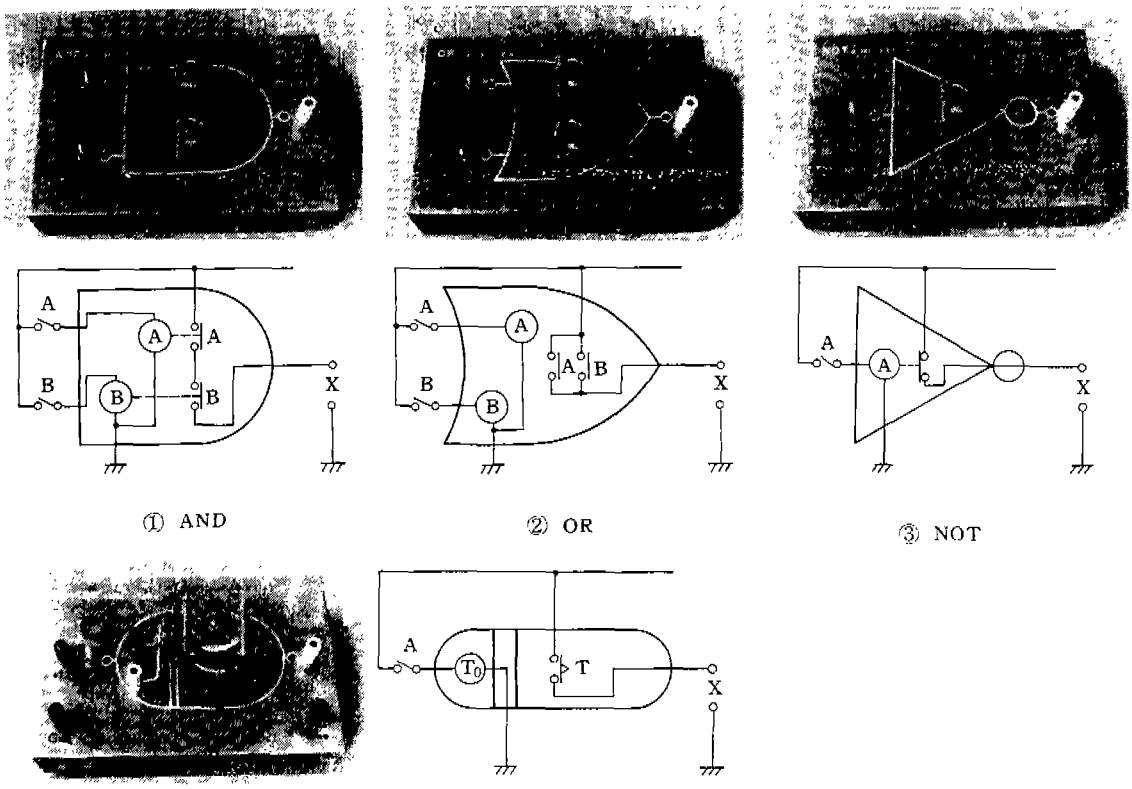
그래서 종래의 릴레이회로와 비교하면서 무접점회로를 이해하는 교구(敎具)를 생각해 보았다. 그림 3.1이 그것이다. 릴레이 시퀀스를 취급한 사람들에게는 친숙한 전자 릴레이의 유접점으로 AND, OR, NOT, TIMER의 등가회로를 구성한 것이다.

그림 2.1에 나타낸 유접점회로와는 다음과 같은 점이 다른 것에 착안하고 보기 바란다.

1. 입력과 출력이 심볼 기호를 끼고 좌우로 나뉘어져 있는 것.
2. 입력, 출력 모두 전압의 고저(전압이 높다는 것은 상대적으로 높은 전압 레벨을 말하며 이것을 「H」(high), 전압이 낮다는 것은 상대적으로 낮은 전압 레벨을 말하며 이것을 「L」(low)로 표현한다)로 동작이 정해져 있는 것.
3. 전원, 어스를 잡는 방법이 다른 것.
4. 전류의 흐름만이 아니고 논리(신호)의 흐름으로 되어 있는 것.
5. 유접점회로가 직렬의 입력구성으로 되어 있어도 무접점회로는 신호전압을 주는 방법이 병렬구성으로 되어 있는 것.

특히 2의 전압의 고저로 동작이 정해지는 것은 주의하여야 한다. 구체적으로는 ① AND에서는 입력 A, B 양 쪽 모두 「H」일 때만 출력 X가 「H」가 되는 것을 나타내고 있다. ②의 OR에서는 입력의 한 쪽이 「H」면 출력은 「H」가 되는 것, ③의 NOT에서는 입력이 「H」일 때 출력은 「L」, 입력이 「L」일 때 출력은 「H」가 되는 것을 각각 나타내고 있다.

그리고 입력상태가 어떻게 조합되어 출력상태가 정해지는가를 표로 나타낸 것을 진리값표(眞理値表)라고 하는데 그림 3.2의 각 논리기호 우측에 표시한 것이 그것이다. 단, (d)의 시한(타이머)만은 시간요소를 포함하고 있어 이 상태를 진리값표로



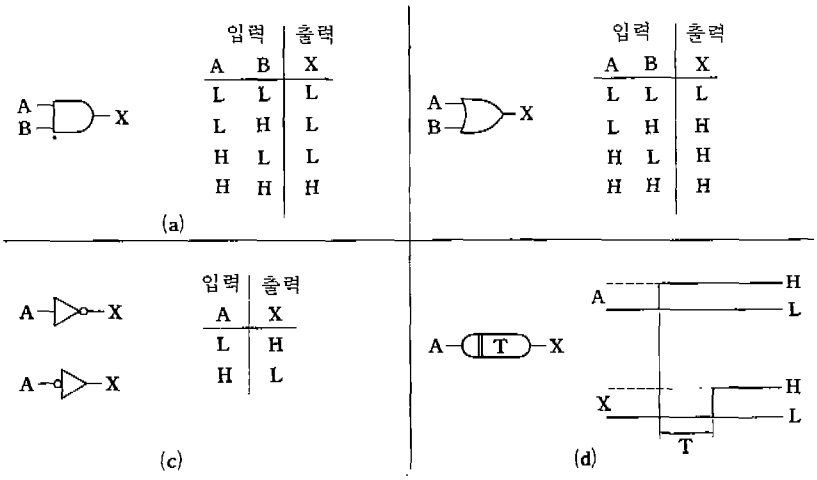
① AND ② OR ③ NOT

④ TIMER

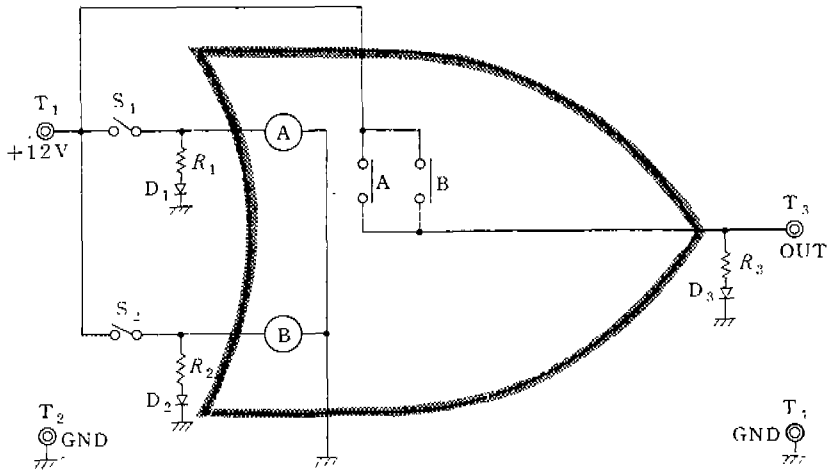
〈그림 3.1〉 릴레이에 의한 무접점 요소의 표현

표시할 수 없기 때문에 (d) 아래 쪽에 나타내는 종축에 「H」, 「L」 레벨, 횡축에 시간을 취한 도표로 표현한다. 이것을 타임차트(Time chart)라고 부르고 있다. 입력 A가 「H」가 되어도 출력 X는 즉시 「H」가 되지는 않으므로 타이머 설정시간 T를 경과하고 비로서 「H」가 되는 것이다.

이와 같이 무접점회로에서는 전압의 고저(「H」, 「L」)로 회로동작을 표현하지만 유접점회로에서는 일반적으로 이것과는 다른 식으로 보는 것이다. 유접점회로에서는 접점의 개폐와 전자 코일에의



〈그림 3.2〉 각론에 있어서의 진리값표(d는 타임차트)



Ⓐ, Ⓑ : 미니 파워 · 릴레이, MY1 1P DC12V 5A 160Ω
D_1~D_4 : 발광 다이오드, DC 순전압 2V, 순전류 15mA
R_1~R_4 : 솔리드 저항, 1/4W 680Ω
S_1~S_2 : 미니 토글스위치, 2P 125V 6A
T_1~T_4 : 존손형 터미널, 125V 10A 12mφ

〈그림 3.3〉 릴레이에 의한 OR회로 배선도

연결방식을 주로 보고 입력접점의 조합과 접점 개폐상태에서 전자 코일에 전류가 흐르고 있는가 아닌가로 전자 코일의 여자, 비여자가 정해지는데 이것으로 출력접점의 개폐상태를 조사하는 것이 일반적이었다.

유접점회로에서는 접점 개폐에 수반한 전류의 유무로 주로 회로를 보는 데 비해서 무접점회로에서는 전압의 유무(「H」, 「L」)로 주로 회로동작을 보는 것이 특징으로 되어 있다. 이것은 시간적 여유가 있으면 그림 3.1의 사진에 나타낸 것을 자신이 만들어 보면 잘 알 수 있을 것으로 생각된다. 이것이 안되는 경우라도 유접점회로를 취급한 경험이 있으면 그림 3.1을 보면서 요소에 대해서 진리값표, 타임차트를 만들어 보는 것으로도 충분할 것이다.

그림 3.1의 사진에 나타낸 것을 만드는 경우의 참고로서 그림 3.3에 배선도와 부품 · 재료표를 들었다. OR의 경우만이지만 기타도 동일하다. 릴레이는 DC 12V 미니튜어 릴레이를 사용하고 있지만 디지털 IC의 전압 레벨 5V에 맞추어서 5V 릴레이(이것에 가까운 것으로 4V 릴레이가 있다)를 사용해도 될 것이다.

입출력 각각에 「H」, 「L」 레벨을 표시하는 발광 다이오드를 달았다. 입력 스위치 ON, OFF상태를 발광 다이오드의 점멸을 보고 릴레이 접점의 움직임을 눈과 귀로 확인할 수 있으며 유접점회로의 경험을 기본으로 유접점회로를 이해할 수 있다. 동시에 무접점 로직 심볼에 대해서 친숙해질 것이다.

4. 유접점회로를 무접점회로로 표시

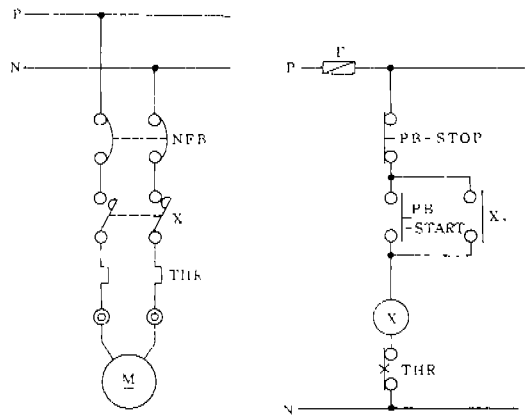
여기서는 유접점 릴레이를 사용한 시퀀스 제어회로를 무접점 로직회로로 바꾸는 요령을 설명한다. 이것으로 충분하지는 않지만 본고에서 취급하는 유접점 릴레이회로는 지금부터 기술하는 것의 응용으로서 무접점회로로 변환할 수 있게 하였다. 그리고 몇가지 예를 연습하여 변환에 관해 숙달되도록 하기 위해 이것을 제4장, 제5장에서 연습할 수 있도록 배려하였다.

본고는 이미 알고 있겠지만 유접점회로를 무접점회로로 변환하는 방법에 주안점을 두고 설명된 것은 아니다. 그러므로 유접점 릴레이 시퀀스 제어계를 무접점화하기 위한 변환방법이나 이 무접점회로를 실제 로직 IC 등으로 실현시키기 위한 구체적인 방법에 대해서는 다른 참고서를 참조하기 바란다.

본고는 전술한 바와 같이 무접점 심볼도가 부여되어 있으면 프로그램 코딩을 알기 쉽고 또한 용이하게 기계적으로 할 수 있게 되는 것과 또한 프로그램의 원시정보를 부여해 주므로 프로그램의 구조를 명확하게 파악할 수 있게 되는 것에 주안점을 두고 기술하고 있다.

그러므로 여기서는 AND, OR, NOT의 3요소를 포함한 가장 간단하고 기본적인 자기유지회로에 대해서 무접점화의 요령을 간단히 기술하는 것으로 끝내고자 한다.

그림 4.1에 전자 릴레이를 사용한 리셋 우선회



〈그림 4.1〉 자기유지(리셋 우선)회로

로를 들었다. 이 그림은 지난호 그림 4.1에 든 것의 일부이다. 이 회로를 예를 들어 무접점회로에의 변환 요령을 설명해 나가기로 한다.

(1) 회로동작

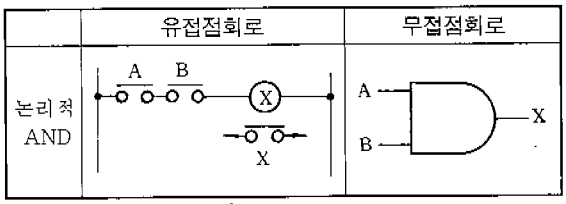
여러분들에게는 기본적인 동작설명이 필요없을 것으로 생각되므로 여기서는 무접점화의 유의점만을 기술하기로 한다.

그림 2.1에 나타낸 바와 같이 시퀀스 제어회로를 구성하기 위한 기본요소는 AND, OR, NOT, TIMER이며, 이것을 유접점회로로 나타내면 그림 좌측의 회로가 된다고 설명하였다.

그러나 유접점 릴레이회로 영역에서는 이 기본회로의 것을 AND, OR, NOT라고는 일반적으로 부르지 않을 것이다. 종래의 유접점 영역에서는 강한 언어를 사용한다고 하면 AND를 직렬조건, OR를 병렬조건이라고 하는 정도이고 주지하는 바와 같이 회로를 논리적으로 보는 일은 없었다.

예를 들면 그림 4.2의 AND회로라면 ⓧ코일에는 A, B의 접점이 직렬로 연결되어 있는 것이다, A, B 양 쪽의 접점이 닫히지 않으면 코일은 여자되지 않는 것이다, 라는 식으로 보았다(사실 이것은 한 마디로 표현하면 논리 AND를 말하고 있는 것이지만).

그런데 무접점 논리회로는 이 회로를 그림 4.2 우측의 AND 심볼로 표현하게 된다. 심볼의 알맹이는 블랙박스로 보이지 않는다. 그러므로 심볼이 어떠한 기능을 가지고 있는가를 자신의 머리 속에 이미지로서 바로 그릴 수 있게 되는 훈련을 하여



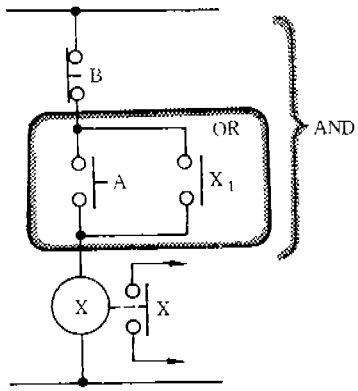
〈그림 4.2〉 유접점, 무접점회로를 보는 방식

두어야 한다. 이 훈련용으로 앞에서 든 교구가 크게 도움이 될 것으로 확신한다.

다음에는 유접점회로를 논리적으로 본다는 것이 중요하다. 그림 4.3으로 이것을 어떻게 보는가를 생각해 보자. 이 그림은 그림 4.1의 일부를 고쳐 그린 것으로서, 푸시버튼 스위치의 부호를 각각 PB−START⇨A, PB−STOP⇨B로 하고 있다. 그리고 퓨즈, 과부하계전기(서멀 릴레이, THR)는 별회로로 생각하기 때문에 제거하였다.

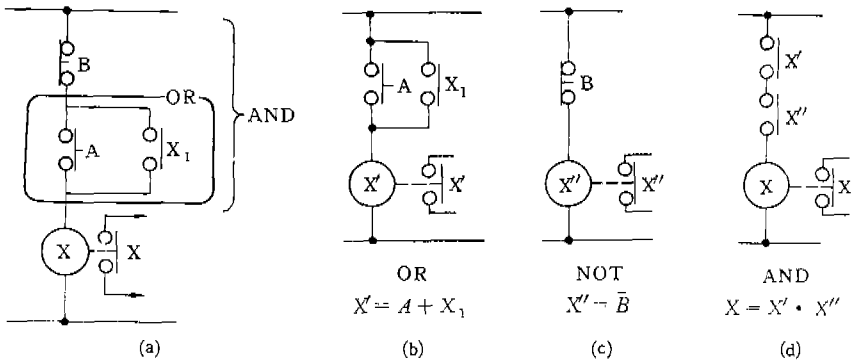
우선 푸시버튼 스위치 A와 X_1 접점이 병렬로 접속되어 있는 부분에 착안하자. 그 이유는 저항의 직병렬회로의 합성저항치를 계산할 때 병렬부분에서 직렬부분으로 보아 나가는 것을 생각하면 될 것이다.

A와 X_1이 병렬조건, 즉 논리회로에서는 OR의 논리를 구성하고 있다. 이 OR에 대해서 푸시버튼 스위치 B가 직렬조건, 즉 AND 구성으로 되어 있다. 정리하면 A와 X_1의 OR 논리에 B가 AND로



A : 운전 푸시버튼 스위치(세트 신호)
B : 정지 푸시버튼 스위치(리셋 신호)
ⓧ : 전자 코일
X : 출력접점
X_1 : 보조접점(유지용)

〈그림 4.3〉 유접점→무접점화를 위한 구성도(1)



〈그림 4.4〉 유접점→무접점화를 위한 구성도(2)

구성되어 있다는 것이다. 이 논리구성으로 출력 X의 동작이 결정되는 것이다.

(2) 불식(논리식)

이상의 것을 말로만 표현하려고 하면 약간 복잡해진다. 이럴 때 수학의 힘을 빌리면 쉽게 표현될 때가 있다. 이 수학을 불 대수(Boolean algebra)라고 하며, 대수식을 불식(불(1815~1864), 영국의 수학자. 그가 만든 대수로, 그 대수식을 불식이라고 한다) 또는 논리식이라고 부르고 있다. 대수식을 사용하면 논리 구성을 정확하게 표현할 수 있고 유접점→무접점 변환의 경우 논리의 틀림을 방지하며 변환후의 회로의 동작을 정하는 논리 구성이 올바른가의 여부를 체크하는 데도 이용된다.

불대수에 대해서 상세한 것은 이에 대한 참고서를 보면 되므로 여기서는 당장 필요한 사항만을 들면 대체로 다음과 같이 된다.

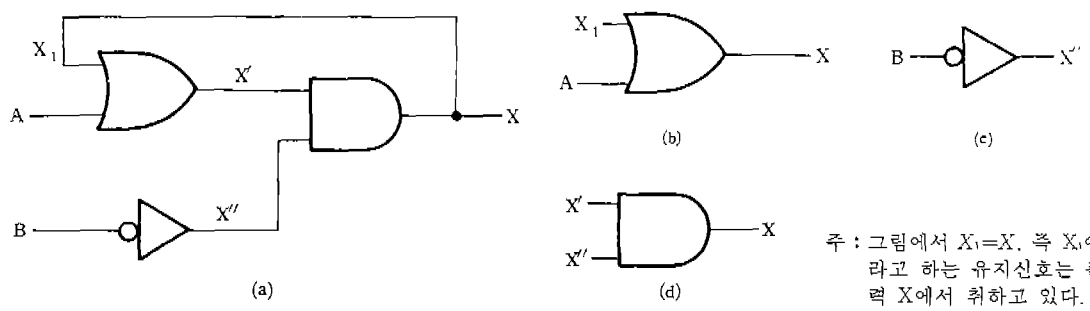
1. AND를 곱셈 기호「·」, OR를 덧셈 기호「+」, NOT를 그 대수문자 위에「−」를 붙여서 각각 표시한다.
2. 변수(신호)의 값은 1인가 0 어느 것이다. 그 이외의 값은 취하지 않는 것으로 한다.
3. $\left\{\begin{array}{lll} 1+1=1 & 1+0=0+1=1 & 0+0=0 \\ 1\cdot 1=1 & 1\cdot 0=0\cdot 1=0 & 0\cdot 0=0 \\ \overline{1}=0 & \overline{0}=1 & \end{array}\right\}$

이 성립한다.

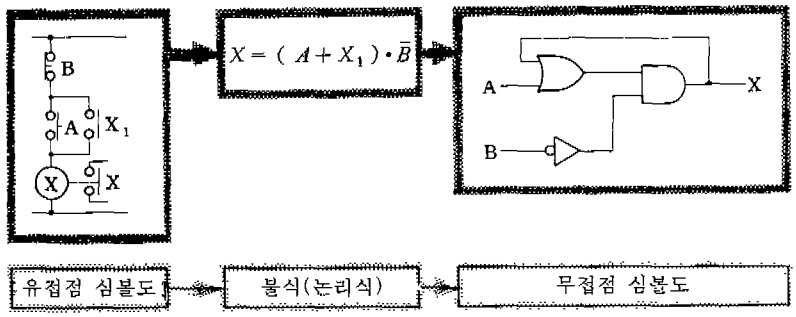
4. TIMER는 그 기능을 불식으로는 표현할 수 없다(이것에 대해서는 뒤에 TIMER 요소를 포함한 간격동작회로의 무접점화시에 구체적으로 처리방법을 기술한다).

위의 3은 무접점 변환시에 직접 필요한 것은 아니지만 무접점회로로 변환된 후의 회로동작을 확인하는 데에 이용된다. 그리고 1+1=1, $\overline{1}=0$, $\overline{0}=1$이 되는 것 이외는 종래의 수학과 동일하며 특별한 주의가 필요없다.

그러면 그림 4.3의 회로를 불식으로 표현하는 것을 생각해 보자. A와 X_1의 병렬 회로만이 있다고 생각한 경우의 출력 X 접점의 동작상태(X 코



〈그림 4.5〉 무접점 회로표현의 변환 코스



〈그림 4.6〉 변환과정

일이 여자하고 X접점이 닫히는 것)를 부여하는 접점 구성 논리는 OR였다. 이것을 불식으로 쓰면 다음과 같이 된다(그림 4.4(b)).

$$X' = A + X_1 \quad \cdots\cdots (3.1)$$

B만이 단독으로 있다고 하면 그림 4.4(c)와 같이

$$X'' = \overline{B} \quad \cdots\cdots (3.2)$$

가 된다. 그리고 X는 X'와 X''와의 AND이므로 결국

$$X = X' \cdot X'' = (A + X_1) \cdot \overline{B} \quad \cdots\cdots (3.3)$$

가 된다. 이 $X = (A + X_1) \cdot \overline{B}$가 이 자기유지회로의 최종적인 불식이 되는 것이다.

이상, 유접점회로를 논리적으로 보고 이것을 논리적 대수식, 즉 불식을 사용하여 쓰는 요령을 알았을 것으로 생각한다.

(3) 무접점회로 표현

다음에 불식(3.5)을 로직 심볼을 사용하여 무접점회로로 표시해 보자. 무접점회로는 앞에서 본 바와 같이 그림 좌측이 입력, 우측이 출력이 되도록 그린다. 그리고 로직 심볼은 그 형상이 정해져 있으며(부록 Ⅰ, 부도 1) 그것에 따라 그리게 된다. 그러나 심볼을 그 때마다 제도하는 것은 손이 가므로 시판되고 있는 템플레이트(부록 Ⅰ, 부도 2) 등을 이용하여 그리면 편리하다.

그림 4.5에 무접점 심볼도로의 변환 코스를 나타냈다. 이 그림은 그림 4.3(a), (b), (c), (d)의 동부호도에 각각 대응하고 있다. 여기서 주의할 점은 유지신호 X_1은 출력 X와 동일하다. 즉, $X_1 = X$인 것이다. 이 출력 X를 앞으로 되돌려 X_1으로 하고 있다(피드백 신호라고 불러도 되는 것이다).

4절에서 기술한 내용을 정리하면 그림 4.6과 같이 된다. 불식을 중계에 넣은 것은 변환에 익숙해지지 않은 경우에 유접점회로를 식을 사용하여 정확히 논리적으로 표현하여 논리의 누락이 생기지 않게 하기 위해서이고 좀 숙달되면 유접점 심볼도에서 직접 무접점 심볼도로의 변환이 가능해진다.

☞ 다음호에 계속 ☜