

제2장 기초지식에 대하여

우리는 지금 어떤 시대에 살고 있는가? 우리에게 편리함을 가져다주는 장치의 내부에는 무엇이 들어있는가? 복잡한 동작을 항상 반복적으로, 그러면서도 가변적으로 하기 위해서는 어떤 기능들이 있어야만 가능한가?

이런 질문의 끝에는 인간의 두뇌와 비슷한(?) 기능을 갖는 장치나 소자가 있어야 됨을 알 수 있다. 즉, 기억하고 외부자극을 받아드리고, 기억된 내용(경험)으로 비교 판단하여 행동할 수 있는 신호를 발생시키는 장치가 필요하다. 우리는 이것을 컴퓨터라고 부른다. 컴퓨터의 기본 기능은 계산할 수 있다는 것이다.

그러나 컴퓨터가 정보의 저장능력을 갖지 못한다면 경험을 축적하지 못하므로 항상 1회성 동작만 가능할 뿐이다. 그러면, 정보를 손실없이 또는 파괴로부터 안전하게 기억시키기 위한 가장 좋은 방법은 무엇인가?

이것이 바로 디지털 값이다. 즉 on/off, 있다/없다, 5V/0V 등의 확실히 구분되는 2가지 상태만으로 정보를 표현하면 정보의 파괴를 최소화시킬 수 있으며, 이런 이유가 현재를 디지털 시대로 만든 주 이유이다.

▷▷ 여기서 정보의 저장은 시간, 공간을 모두 포함한다. 따라서 정보의 단순한 저장뿐만 아니라 전송, 공유 및 활용까지를 의미한다.

그럼 디지털 신호란 어떻게 만들어지는가?

여러분이 집에서 전등을 켜다고(on) 또는 껐다고(off)할 때 2가지 상태가 나오며, 이것이 바로 디지털 값이다. 따라서 스위치의 on/off 동작으로 얻어지는 신호가 디지털 신호이다.

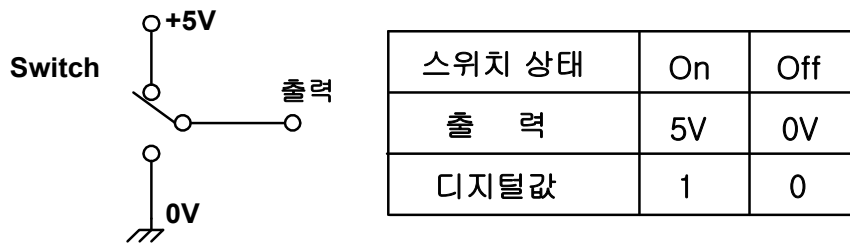


그림 2-1. 디지털 신호란

디지털 신호를 전압값으로 표현할 때 어떤 의미에서는 전압의 크기는 중요한 것은 아니다. 다시 말하면, 디지털 값 '1'은 항상 5V가 아니라는 것이다. 다만, 표준전압으로 사용하는 것이 5V일 뿐이며, 원칙적으로 낮은 전압을 사용하는 것이 더 좋은 특성을 가진다. 따라서, 최근의 디지털 소자들은 전원을 3.3V로 하는 것이 표준적으로 사용되고 있다.

그리고, 비록 디지털 신호도 여러 단계를 거치게 되면 외부 잡음에 의해서 신호에 혼돈이 생기게 되므로 전자소자를 사용하여 신호를 확실한 '1', '0'로 다시 만들어 주어야 하며, 또 디지털 신호 사이의 함수관계를 만들어 주는데 이것을 논리소자(gate)라 한다.

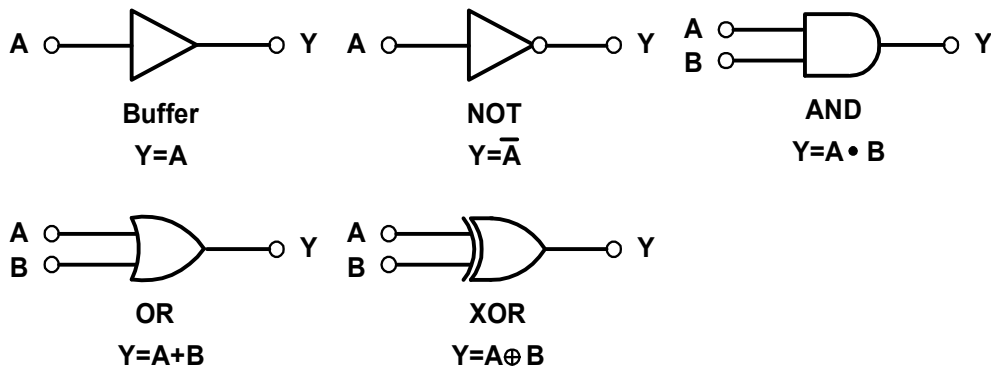


그림 2-2. 논리소자들

우리는 논리소자들을 결합하여 다양한 형태의 논리기능을 만들 수 있다. 한 예로 s/w1이 on, s/w2가 off 시에 만 출력이 '1'이 나오는 회로를 만들면, 회로는 아래와 같다.

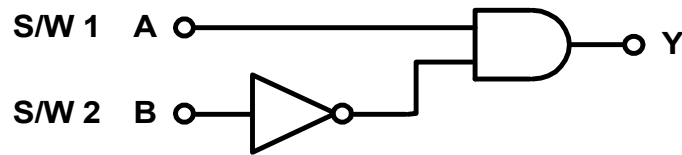


그림 2-3. 논리회로도

이것을 논리식으로 나타내면, $Y = A \cdot \bar{B}$ 가 된다. 자세한 것은 논리회로 과목을 참고하기 바란다.

그러면 디지털 값을 저장하는 소자는 어떻게 만들어지겠는가?

기본적으로는 출력을 입력으로 feedback시키는 회로를 구성하면 된다.

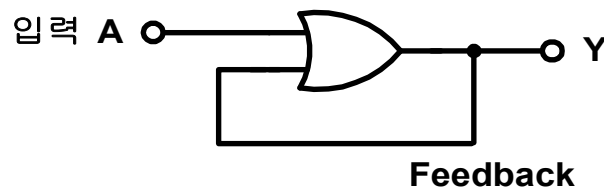


그림 2-4. 기억소자

위 회로에서 출력의 초기값을 '0'이라고 하고, 입력이 '0'이면 출력은 '0'을 유지한다. 즉 $Y=A$ 가 된다. 그러나, 입력을 '1'로 만들면 출력은 '1' ($Y=1$)로 되며, 다시 입력을 '0'으로 만들어도 출력은 '1'을 계속 유지한다. 즉 위 회로는 출력이 '1'인 상태를 저장할 수 있는 기능을 가지는 회로가 된다. 마찬가지로 소자를 AND gate로 변경하면, '0'을 기억시킬 수 있는 소자가 된다.

따라서, 이런 기능을 확장하면 '1', '0'을 필요에 따라서 전부 기억시킬 수 있는 소자를 만들 수 있으며, 이런 소자를 기본적으로는 latch라고 부른다.

각 소자의 내부회로는 여러분이 다른 과목에서 다룰 것이므로, 여기서는 기호만을 설명한다.

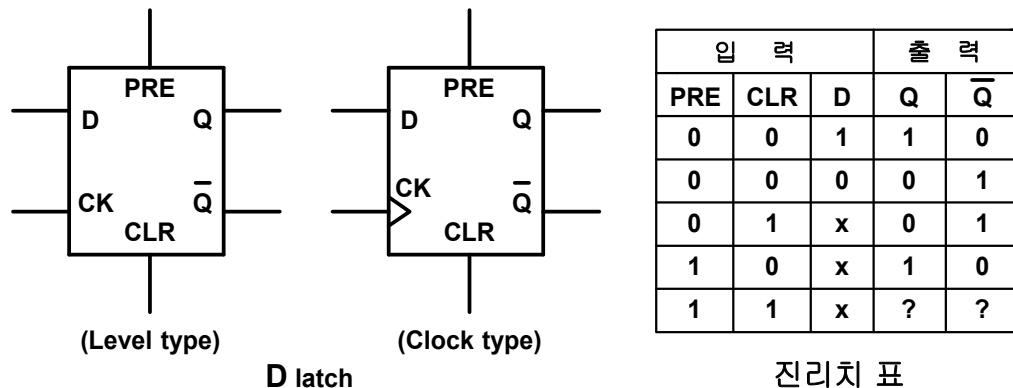


그림 2-5. D latch

그림에서 PRE(preset), CLR(clear)단자는 기억된 내용을 초기 설정하는 입력 신호이며, D(data)나 CK(clock) 입력 신호보다 우선적으로 동작된다.

진리표에서 'X' 표시는 '1', '0' 모두를 나타내며, '?' 표시는 어떻게 결정될지 모른다는 의미이다. 그리고, 기억은 PRE, CLR 입력이 '0' 인 상태에서, 입력을 D에 인가하고 기억시키라는 제어신호를 CK에 인가하면 된다. Level type에서는 CK가 '1'인 상태일 때 D의 내용이 출력으로 전달되어 기억되어지고 (최종기억은 CK가 '1'에서 '0'으로 변화될 때이므로 이때의 D의 내용이 출력 Q에 기억되어 유지됨), clock type인 경우는 CK이 '0'에서 '1'로 천이할 때의 D의 내용이 출력Q에 기억된다. 그리고, CK의 상태를 NOT시키기 위해서 NOT gate가 추가된 것이 있는데, 기호는 CK단자에 작은 o를 표시한다.

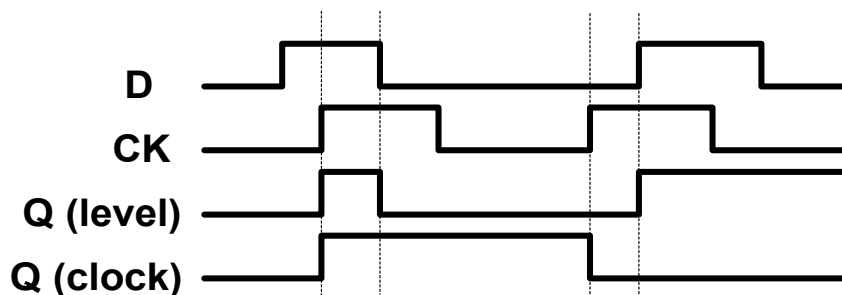


그림 2-6. D latch의 출력파형

디지털 정보를 여러개 동시에 기억시키려고 하며, 각각의 D latch에서 D는 개별로 하고 CK는 공통으로 연결하면 된다. 컴퓨터 내에서 사용하는 기억소자는 전부 이런 구조이며, 따라서 회로를 간단하게 표현하기 위해 입력D와 출력Q는 여러 개를 묶어서 다발로 만들어 표시한다. 이러한 다발은 선 위에 숫자로 개수를 표시하며, 통상적으로 bus line이라고 칭한다.

만약 여러 개의 기억소자가 필요하다면 어떻게 만들면 되겠는가?

입력 D는 공통으로, 출력Q도 공통으로 하여야 회로의 신호선을 최소화 할 수 있다. 그러나, 이때의 문제점은 여러개 중에서 특정한 기억소자를 선택할 수 있어야 한다는 것이다.

즉, 각 기억소자의 CK를 개별적으로 선택할 수 있어야 하며, 따라서 각각의 기억소자에 고유번호를 부여한다. 이것을 주소라고 하며, 주소의 개수가 바로 메모리의 용량을 나타낸다.

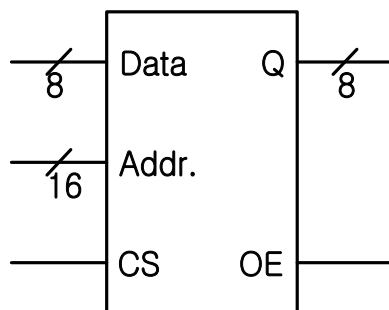


그림 2-7. 기억소자(8bit×64K)

위 그림은 8bit(8개가 병렬 연결된 구성)data를 갖는 것이 2^{16} (64K ; K=1024)개 존재하는 기억소자이다. Bit라는 단위는 디지털 신호 1개를 의미하며, K는 kilo 단위와 같은 단위이나 실제로는 2의 10승으로 1024(1000이 아님)이다. 그리고 기억소자를 부를 때는 기억용량의 크기로 부르며, 8bit×64K, 64Kbyte 또는 512Kbit로 표현한다. 8bit×64K는 디지털 데이터 8개를 단위로 하여 총 64K(65536) 개의 기억소자가 있다는 의미이고, 64Kbyte는 8bit를 새로운 단위인 1byte로 정의하여 표현한 것이며, 마지막으로 512Kbit는

512×1024개의 디지털 신호를 저장할 수 있다는 표현이다. 비슷해 보이지만 의미가 많이 다르므로 명확히 구분할 수 있어야 한다.

앞 설명에서 이런 메모리 소자를 만들기 위해서는 각각의 단위 기억소자들을 입력D는 서로 연결하고, 출력Q도 서로 연결하면 된다고 하였는데.....문제점은 없을까요? 문제점이 보이지 않으면 논리소자를 아직 이해하지 못한 것입니다. 예로 2개 논리소자의 출력을 서로 연결하면 어떻게 될까요? 머리에 떠오르지 않으면 그림을 그려보세요. 그리고 모든 입출력 경우에 대하여 따라가 보면 알 수 있지요.

해 보았나요? 문제가 되는 경우를 구체적으로 설명하면, “한 논리소자의 출력은 ‘1’이고, 한 논리소자의 출력은 ‘0’ ”이라고 하면 ‘1’과 ‘0’을 연결하면 ‘1’, ‘0’ 또는 ‘0.5’가 나올까요? 아니면 다른 값이... 우선 ‘0.5’는 논리값이 아니지요. 출력을 따지기 전에 심각한 문제일까요? 보통의 문제일까요? 아직 구분이 되지 않지요. 왜냐하면 숫자 놀음은 눈에 바로 들어오지 않으니까요. 그러면 그림 2-1을 다시 보도록 합시다. 디지털값 ‘1’는 전자회로에서 전압으로는 5V이고, ‘0’은 0V입니다. 즉 ‘1’과 ‘0’를 연결하면 실제로는 5V와 0V를 서로 연결하는 것이지요. 이렇게 되면 일명 합선되는 것이며 화로가 망가지게 됩니다. 우리는 이런 상태를 단락(short circuit)이라고 합니다.

따라서 논리소자의 출력(일반적으로 모든 것의 출력도 해당됨)을 서로 바로 연결시키면 안됩니다. 그러므로 논리소자의 출력을 연결시키기 위해서는 새로운 기능을 갖는 소자가 더 필요하다는 것을 의미하지요. 바로 3-state gate입니다. Gate 내부에 스위치가 달려있어 출력이 최종 출력단자로 나가지 못하게 하는 구조를 가집니다. 당연히 출력이 나가도록 할 때만 스위치를 on 해주는 기능을 가지는 제어신호가 더 필요하지요.

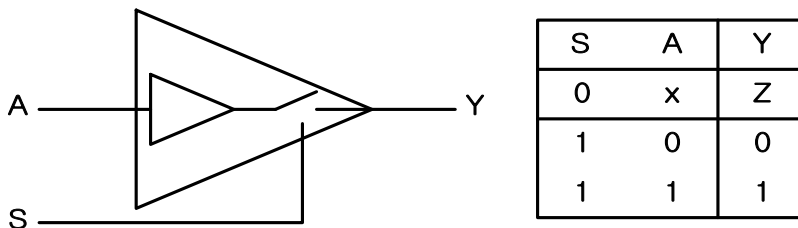


그림 2-8. 3-state gate

그림 2-8에서 보면, 스위치 제어신호 S가 ‘0’이면 스위치가 off되며 따라서 출력 Y는 open된다. 그러나 S가 ‘1’이면 스위치가 on되어 일반 gate와 같이

동작하게 된다. 이는 출력의 상태가 '1', '0', open(high impedance)으로 3가지를 가지는 gate가 되어 이를 3-state gate라고 한다. 기억소자에서의 출력도 이와 같이 만들어져 있으며, 기억소자의 pin 중에서 OE (output enable: 이름은 중요하게 생각하지 말 것)는 S와 같은 의미를 갖는 제어신호이다.

우리는 마이크로프로세서 자체를 설계하고 만들려고 하는 것이 아니므로 이 정도의 기초지식만 있으면 본 강의를 수용하는데 지장이 없을 것이다.

■ HOME WORK B

- 1) 디지털 신호의 전압크기가 낮을수록 더 좋다고 했는데, 그 이유가 무엇이라고 생각합니까?

 - 2) 기본 논리소자(AND, OR, NAND, NOR, XOR)에 대한 입/출력 관계를 모든 경우를 고려한 표(truth table)로 만들어 보시오.

 - 3) 일반 컴퓨터와 PC와의 차이점을 설명해보시오.

 - 4) 74HC573이라는 기억소자(latch)가 있습니다. Internet에서 자료를 찾아 내부회로를 그리고 핀의 기능 및 동작을 설명해 보시오.
-