4주차 예비보고서

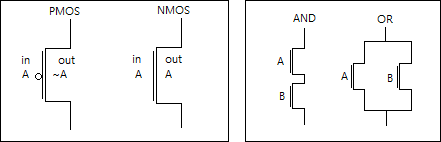
|  |
| --- |
| - NAND/NOR/XOR의 구조(transistor level)  - NAND/NOR/XOR logic 특성  - AND/OR/NOT과의 변환 관계  - AND/OR/NOT의 응용  - XOR의 logic 구현  -기타 이론 |

20141196 김성희

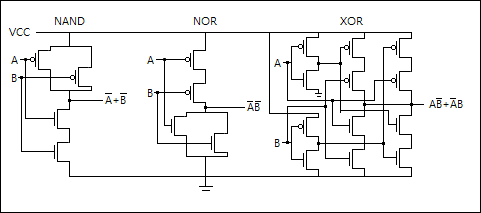
1. NAND/NOR/XOR 구조 (Transistor-level)

MOS 트랜지스터를 주로 쓴다. MOS 트랜지스터에는 n채널과 p채널을 이용하는 2가지 방법이 있다. n채널 트랜지스터는 전압을 줘야만 전류가 흐르고, p채널 트랜지스터는 전압을 주지 않았을 때(평상시에) 전류가 흐른다. (즉 p채널은 전압을 가하면 전류가 끊긴다.) p채널과 n채널 트랜지스터는 각각 장단점이 있고 이를 합쳐서 안정성을 높은 기술을 CMOS 기술이라 한다. 그리고 주로 CMOS를 이용해서 논리 게이트를 만든다.

**<PMOS NMOS 그리고 트랜지스터의 AND OR 표현>**



**<CMOS를 적용한 NAND NOR XOR 게이트>**

****

\* 선이 겹친 경우에 점이 찍혀 있으면 서로 연결 o, 찍혀 있지 않았다면 연결x

\* 가로 막대 두 개가 위아래로 있는 것은 접지를 표현

\* VCC는 전압을 의미한다.

**2. NAND NOR XOR 특성**

NAND, NOR, XOR는 NOT과 함께 기본적인 게이트로 쓰인다. CMOS 기술을 적용했을 때 AND와 OR를 따로 만들지 않고 NAND와 NOR 그리고 NOT의 조합으로 만들었을 때 트랜지스터의 수가 가장 적다. XOR는 NAND, NOR, NOT의 조합으로 만든 것보다 위 1번의 XOR구조의 그림처럼 직접 디자인 하는 것이 더 적은 트랜지스터 수가 쓰인다.

**다음은 각각 NAND, NOR, XOR 진리표다.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **NAND** | **NOR** | **XOR** |
| **0** | **0** | **1** | **1** | **0** |
| **0** | **1** | **0** | **1** | **1** |
| **1** | **0** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** | **1** |

**3. AND OR NOT과의 변환 관계**

위 2번에서 언급했듯이 AND, OR는 NAND, NOR, NOT으로 만든다.

AND는 NAND + NOT의 조합으로, OR는 NOR + NOT의 조합으로 게이트를 만든다. NAND가 NOR가 각각 AND와 OR에 NOT을 씌운 결과임을 역으로 이용하였다.

**4. AND OR NOT의 응용과 XOR logic의 구현**

A XOR B는 (A AND ~B) OR (~A AND B)와 동치이기 때문에 Verilog에서 (~B \* A) | (~A \* B)와 같이 AND와 OR 그리고 NOT을 응용하여 XOR를 구현할 수 있다.

**5. 기타 이론**

AND 게이트 2개의 출력 값을 입력 값으로 사용하는 OR 게이트의 경우 AND 게이트 2개와 OR 게이트 1개를 모두 NAND 게이트로 바꿀 수 있다. 드모르간 법칙을 사용하면 된다. (편의상 X AND Y를 XY로 X OR Y를 X+Y로 쓰겠다. NOT X를 X**`**로 쓰겠다.) (X+Y)**`** = X**`**Y**`**를 (X+Y) = (X**`**Y**`** )**`**로 바꾼 뒤 X를 AB로 Y를 CD로 바꾼다면 앞서 말한 3 NAND 게이트를 사용한 logic으로 바뀐다. 회로를 구성할 때 이를 많이 이용하는 데 그 이유는 AND(OR)가 NAND에 비해 트랜지스터 수를 더 많이 쓰기 때문이다. 마찬가지로 OR 게이트 2개의 출력 값을 입력으로 사용하는 AND 게이트 logic은 NOR 게이트 3개로 대체할 수 있다. 여기서는 (XY)**`** = (X**`**+Y**`** )를 XY = (X**`**+Y**`** )**`**로 바꾼 드모르간 법칙을 이용한다. 마찬가지로 트랜지스터 개수를 절약할 수 있다.

XOR가 XY**`**+X**`** Y logic과 동치인 것을 이용해 AND, OR, NOT 게이트의 조합 또는 NAND게이트들의 조합으로 쓴다면 트랜지스터가 낭비될 것이다. 따라서 위 1번의 XOR 구조처럼 단일 게이트를 만든 다면 트랜지스터를 더 줄일 수 있다. 이처럼 트랜지스터의 개수를 줄이는 것이 별것 아닌 것처럼 보일 수 있으나 실제 반도체를 만들 때는 직접회로에 억 단위의 트랜지스터를 사용하기 때문에 상당히 많은 공간과 돈, 재료를 절약할 수 있다.