

4주차 예비보고서

전공: 경영학과

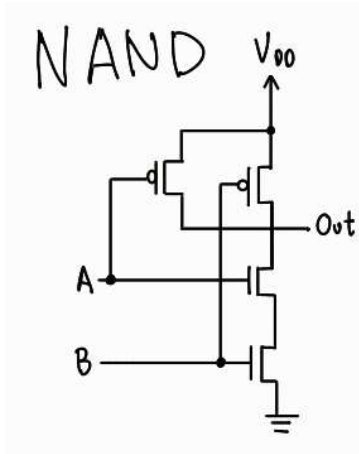
학년: 4학년

학번: 20190808

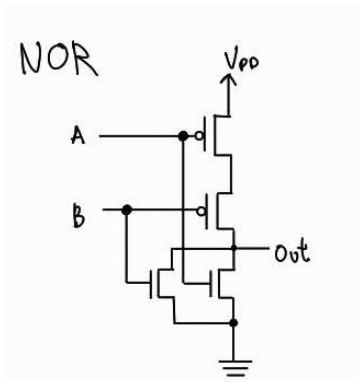
이름: 방지혁

1.

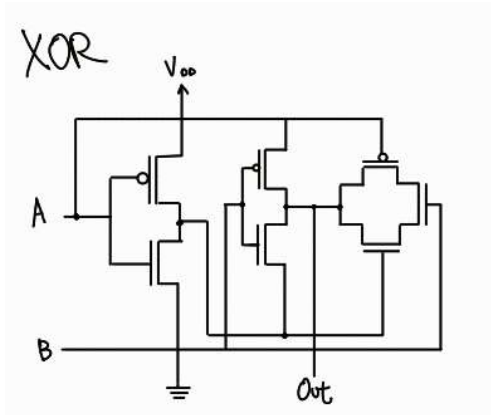
NAND



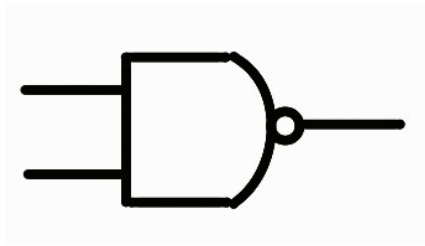
NOR



XOR



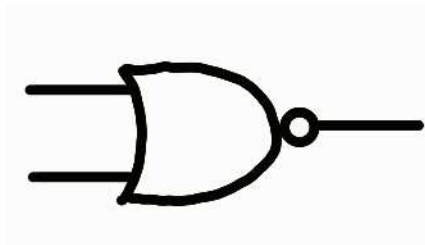
2.



A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

1) NAND gate

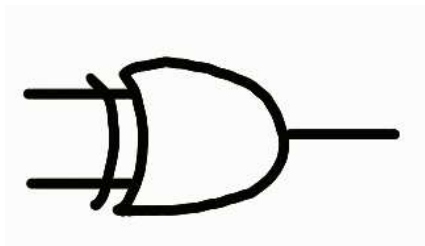
우선, 위 그림에 보드시피 NAND gate는 AND gate에 NOT(invert)를 취해준 것과 같습니다. input 값이 모두 1이 되면, output이 0이 나옵니다.



A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

2) NOR gate

우선, 위 그림에 보드시피 NOR gate는 OR gate에 NOT(invert)를 취해준 것과 같습니다. input 값이 모두 0이 되면, output이 1이 나옵니다.



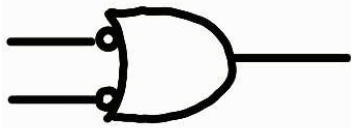
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

3) XOR gate

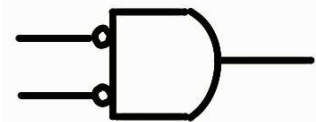
XOR gate는 input의 입력 값 중 1의 개수가 홀수이면 1, 양수(0개 포함)이면 0을 반환한다고 보면 됩니다.

3.

위의 서술 항목에서 보드시피 NAND 게이트는 AND게이트에 not을 취한 형태, OR gate에 not 을 취한 형태로 볼 수 있습니다. 앞에서는 게이트를 지난 후 부정형을 취한 것이지만, 미리 input에 부정형을 취한 후 and 및 or 게이트를 통과시켜 NAND 와 NOR 게이트를 구현할 수도 있습니다.



NAND 게이트는 기존의 경우처럼 AND 게이트에 부정형을 취하는 것이 아니라, 각각의 input에 부정형을 취한 후 OR gate를 통과시켜 구현할 수 있습니다.



NOR 게이트는 기존의 경우처럼 OR 게이트에 부정형을 취하는 것이 아니라, 각각의 input에 부정형을 취한 후 AND gate를 통과시켜 구현할 수 있습니다.

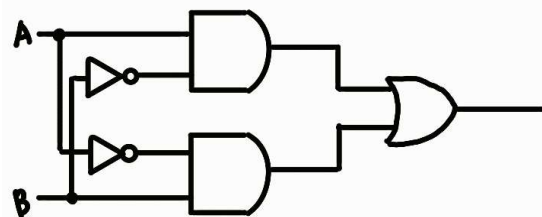
4.

AND-OR-INVERT logic은 논리 연산을 간단하게 하여, 하드웨어 설계를 더더욱 효율적으로 할 수 있도록 합니다. 예를 들어, 위의 회로처럼 input 값을 AND gate에 들어가게 하고, 그 이후 OR gate를 거쳐, not gate를 거치게 하는 것입니다. 예를 들어 여러 logic을 조합하여 $((A * B) + (C * D))'$ 라는 논리식으로 표현할 수도 있습니다. 이러한 AND-OR-INVERT logic은 논리 연산을 단순화하여 고속 디지털 회로에서 사용될 수 있습니다. 게이트 지연이 감소하고 데이터 처리 속도가 빨라집니다. 디지털 신호처리가 그 예시가 될 수 있겠습니다. 또한, ASIC과 FPGA에서 공간을 절약하고 전력 소비를 줄이는데 이바지합니다.

5.

XOR logic의 표현식은 다음과 같습니다.

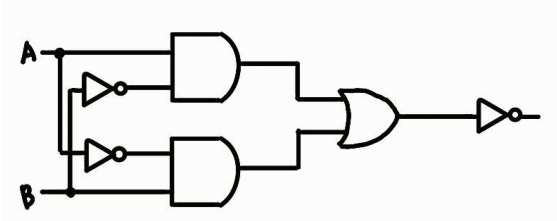
$A \oplus B = (A \cdot B') + (A' \cdot B)$. 쉽게 말하자면 A와 B input의 출력 값이 다를 때만 1을 출력한다는 것입니다. 아니면, input 값 중 1의 개수가 홀수이면 1, 양수(0개 포함)이면 0을 반환한다고 보면 됩니다. 이를 구현하기 위해서 우선 NOT gate를 통해 각 input에 대해 부정형을 취해줍니다. 그리고 각각 다른 input과 AND 연산을 해줍니다. AND gate의 두 개의 결과들을 OR 게이트에 넣어서 최종적으로 XOR을 다음과 같이 구현할 수 있습니다.



6.

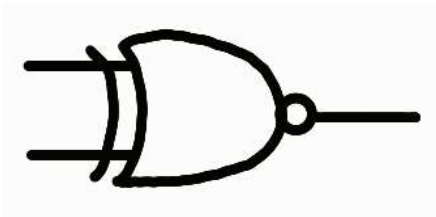
XNOR gate는 XOR gate에 inverter를 합친 것입니다.

input의 두 값이 같으면 1이 출력되고, 다르면 0이 출력됩니다.



A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

다음과 같이 구현할 수 있으며, 진리표는 위와 같습니다.



다음과 같이 표현합니다.