
4주차 예비보고서

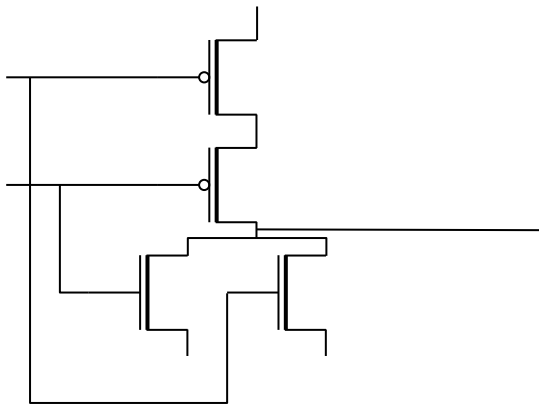
전공: 컴퓨터공학

학년: 2학년

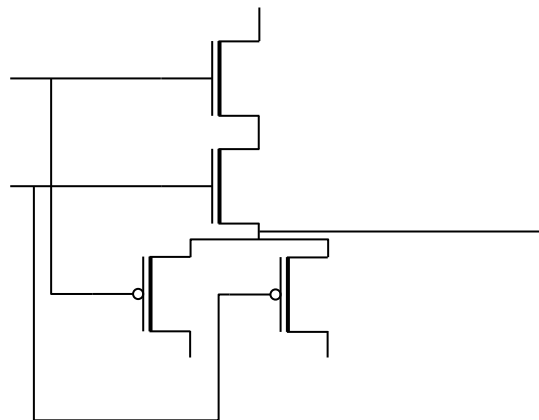
학번: 20191629

이름: 이주현

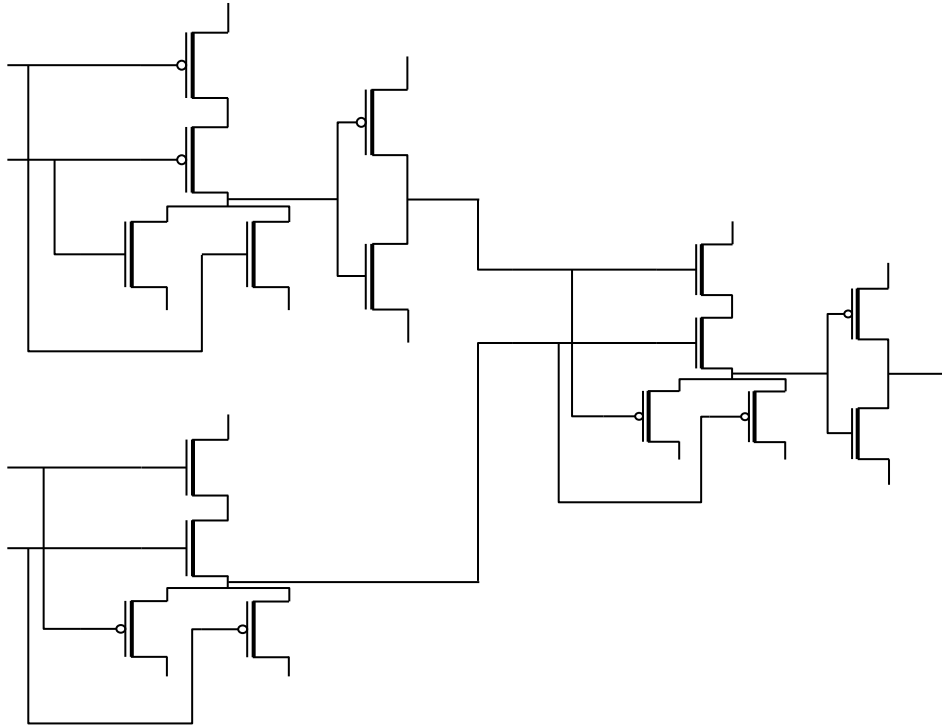
1. 논리게이트 AND/OR/NOT의 구조를 transistor-level로 그리시오.



NOR 게이트 (OR 게이트에 인버터를 삭제한 모양)



NAND 게이트 (AND 게이트에 인버터를 삭제한 모양)



XOR 게이트 (NAND, OR, AND 게이트를 조합한 모양)

2. NAND/NOR logic의 특성을 조사하시오.

NAND 게이트와 NOR 게이트는 모두 두 개의 입력을 받아 하나의 값을 출력하는 논리 게이트이다. NAND 게이트는 AND 게이트의 출력에 하나의 인버터를 덧붙인 것과 같은 논리 특성을 지니며, NOR 게이트는 OR 게이트의 출력에 하나의 인버터를 덧붙인 것과 같은 논리 특성을 지닌다.

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

왼쪽: NAND 게이트의 진리표, 오른쪽: OR 게이트의 진리표

XOR 게이트 역시 두 개의 입력을 받아 하나의 값을 출력하는 논리 게이트이지만, AND, OR과 달리 두 입력이 "서로 다를 때만" 논리적 참을 반환하는 논리 게이트이다. 즉, A의 입력이 1, B의 입력이 0이거나 A의 입력이 0, B의 입력이 1일 때만 논리적 참을 반환하고, A와 B 모두 0 또는 1인 경우는 거짓을 반환하게 된다.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

NOT 게이트의 진리표

3. 기본 논리게이트 (AND/OR/NOT)와 변환 관계를 조사하시오. (NAND, NOR)

NAND는 기본 논리 게이트인 AND의 출력에 NOT 인버터를 추가한 모양이다. 또는, NAND는 OR 게이트의 모든 입력에 인버터를 추가한 결과라고 볼 수 있다. NOR은 기본 논리 게이트인 OR의 출력에 NOT인버터를 추가한 모양이다. 또는, NOR은 AND 게이트의 모든 입력에 인버터를 추가한 결과라고 볼 수 있다.

트랜지스터 레벨로 생각해 보았을 때, NAND는 AND 게이트의 출력 NOT 인버터를 제거한 모양이고, NOR은 OR 게이트의 출력 NOT 인버터를 제거한 모양이라고 생각하는 것이 조금 더 현실적인 구현 방법에 가깝다.

4. AND-OR-INVERT logic의 응용

AND-OR-INVERT logic은 보통 3개 또는 4개의 입력을 받아 하나의 출력을 가지는 논리 게이트의 묶음이다. 4개의 입력을 가지는 게이트는 2-2 AOI 게이트라고 불리며, 네 입력 신호가 두 개씩 각각 AND 게이트를 지난 후 그 결과물이 OR으로 연결되는 구조를 가지고 있다. 3개의 입력을 가지는 게이트는 2-1 AOI 게이트라고 불리는데, 세 입력 중 2개만 AND 게이트에 연결되고 그 출력과 세 번째의 입력이 OR으로 연결된 구조이다.

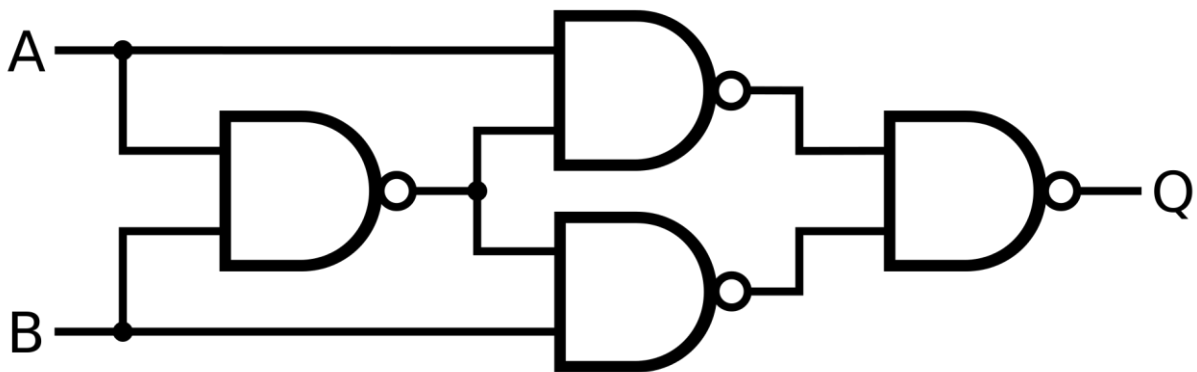
AND-OR-INVERT 게이트는 트랜지스터의 특성상 각각의 기본 논리 게이트를 구현하는 것보다 효율적으로 논리 회로를 구성할 수 있다는 장점이 있다. 이는 실제로 논리 회로를 기판에 구현할 때 전파 지연 시간을 단축시키고, 트랜지스터를 더 적게 사용하여 전력을 더 적게 사용할 수 있을 뿐만 아니라 제작 비용을 더욱 떨어뜨릴 수 있기 때문에 유용하다.

5. XOR logic의 구현 방법

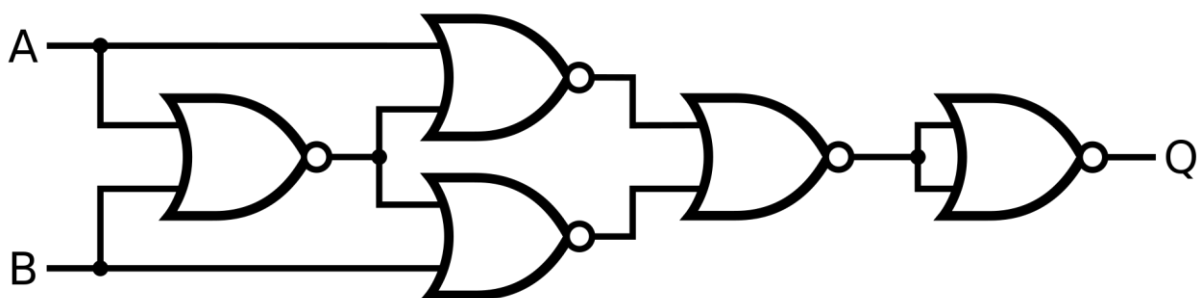
배타적 논리합 논리 식 $A \oplus B$ 는 논리곱과 논리합만을 사용해서 다음과 같이 구현할 수 있다.

$$A \oplus B = A \cdot \underline{B} + \underline{A} \cdot B = (A + B) \cdot (\underline{A} + \underline{B})$$

따라서, XOR 게이트를 구현하는 한 가지 방법은 위와 같이 OR 게이트, NAND 게이트, 그리고 AND 게이트를 조합하는 방법이 있다. 그러나 이 방법은 세 가지의 서로 다른 게이트를 사용하여 XOR 게이트를 구현하는 것이기 때문에 설계할 게이트의 양이 많아지게 된다. 사실 NAND와 NOR만을 이용해서도 XOR 게이트를 설계할 수 있다.



NAND 게이트만을 이용한 XOR 게이트



NOR 게이트만을 이용한 XOR 게이트

이렇게 하나의 게이트를 구현하는 데 여러 방법이 사용될 수 있다.

6. 기타이론

위에서 구현하였듯이, XOR 게이트는 NAND 또는 NOR 게이트만을 이용하여 구현할 수 있다. 사실, NAND와 NOR 게이트는 모두 “범용 논리 게이트”로서, 하나의 종류만 가지고 모든 기본적인 논리 연산을 구현할 수 있는 게이트이다. 이러한 방식의 논리 게이트는 우리가 자주 사용하는 저장장치인 플래시 스토리지에 사용되는 경우가 많다.

비록 NAND/NOR 게이트만 가지고 모든 연산을 할 수 있지만, 우리가 사용하는 FPGA에서는 NAND가 아니라 LUT를 가지고 연산을 구현한다. 그 이유는 일반 NAND/NOR 게이트를 사용하게 되면 실제로 프로그래밍하여 재사용하기 어렵기 때문이다. FPGA가 대두되기 이전에는 NAND Gate Array가 사용되는 경우가 있었으나 낭비되는 게이트의 수가 너무 많고, 그만큼 값이 비쌌기에 결국 사장되고 FPGA를 주로 사용하게 되었다.