전공: 컴퓨터공학 학년: 2학년 학번: 20161603 이름: 신민준

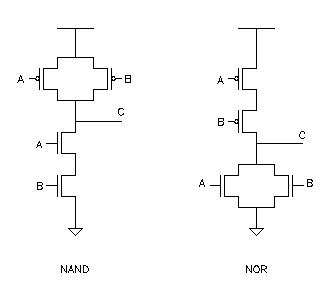
1. 논리 게이트 NAND/NOR/XOR 구조를 Transistor-Level로 그리시오.

* NAND gate

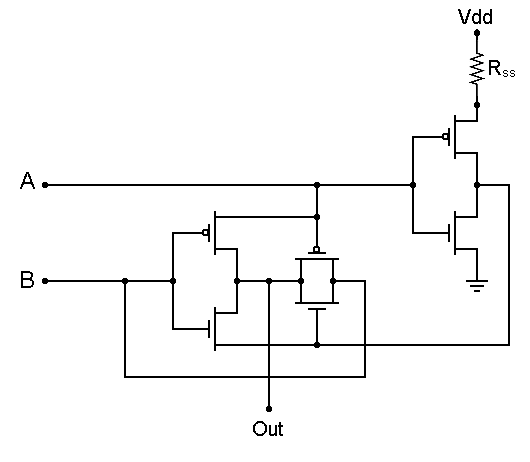
개체이(가) 표시된 사진

자동 생성된 설명

* NOR gate



* XOR gate



1. NAND/NOR/XOR Logic의 특성을 조사하시오.

* NAND logic

“Not AND”의 줄임말로, NAND logic은 다음과 같이 표시한다.



NAND는 AND 의 output에 NOT을 붙인 것과 동일한 결과를 가지며, 따라서 진리표는 다음과 같다.

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output Q |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

* NOR logic

“Not OR”의 줄임말로, NOR logic은 다음과 같이 표시한다.



NOR은 OR logic의 output에 NOT 게이트를 붙힌 것과 동일하며, 이에 대한 진리표는 다음과 같다.

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output Q |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

* XOR logic

“Exclusive OR”의 약자로, XOR은 다음과 같이 표기한다.



식으로 표시할 때는 ‘⊕’ 문자를 사용해 표시한다. 진리표는 다음과 같다.

|  |  |  |
| --- | --- | --- |
| Input A | Input B | A ⊕ B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

1. 기본 논리 게이트(AND/OR/NOT)과의 변환 관계

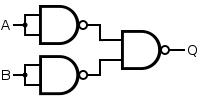
NAND gate의 경우, 두 input을 같은 wire로 연결한다면, NOT gate와 동일한 작동을 한다. 이는 다음과 같다.



또한, 두 NAND gate를 직렬로 연결하면, NAND gate 뒤 NOT gate가 오는 것과 동일한 작동을 하므로, 전체적인 output은 AND gate의 output과 동일해진다. 다음과 같이 연결하면 된다.



OR 게이트는 드 모르간의 법칙에 의해 세 개의 NAND 게이트를 사용해 구현할 수 있다. 두 개의 NAND 게이트는 NOT 게이트처럼 두 Input을 Invert하고, 이 두 input을 다음과 같이 NAND 게이트로 묶어주면 된다.



1. AND-OR-INVERT logic의 응용

AND-OR-Invert logic은 다수의 AND gate의 output들을 NOR gate로 묶어놓는 회로이다. 이 회로의 강점은 CMOS 구현 방식에서 각 게이트들을 따로 구현해서 합쳐놓는 것 보다 전체 회로에서 사용하는 트랜지스터의 수가 적다는 데에 있다. 다음 그림은 AOI 회로를 사용한 2-1 AOI 게이트를 보이고 있다.

개체, 시계이(가) 표시된 사진

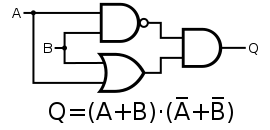
자동 생성된 설명

위 그림에서는 2-1 AOI 게이트를 단 6개의 트랜지스터로 구현한 것을 확인할 수 있다. 만약 이 회로가 AND, OR, Invert로 각각 구성되었다면, 2-input NAND 게이트(트랜지스터 4개), Inverter(트랜지스터 2개), 그리고 2-input NOR 게이트(트랜지스터 4개)로 총 10개의 트랜지스터를 소비했을 것이다.

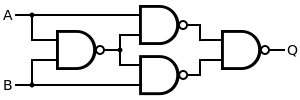
1. XOR logic의 구현 방법

XOR logic의 구현은 기본적으로 의 변형으로 볼 수 있다. 따라서,

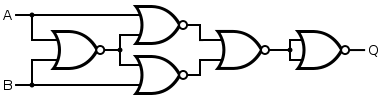
이고, 이를 드 모르간의 법칙을 우항에 적용시켜 처럼 바꾸면 다음과 같이 세 개의 게이트를 사용해 구현할 수 있다.



이를 변형시켜 NAND 게이트 4개로 구현한다면 다음과 같다.



또는 5개의 NOR 게이트로 구현할 수도 있다. 다음에서 이를 보인다.



1. 기타 이론

NOR, NAND 게이트들은 소위 “Universal Gate”, 범용 게이트라고 불린다. 이는 이 두 게이트를 사용하면 모든 논리 함수를 제작하는 것이 가능하기 때문이다. 위의 예제에서도 이를 볼 수 있다.