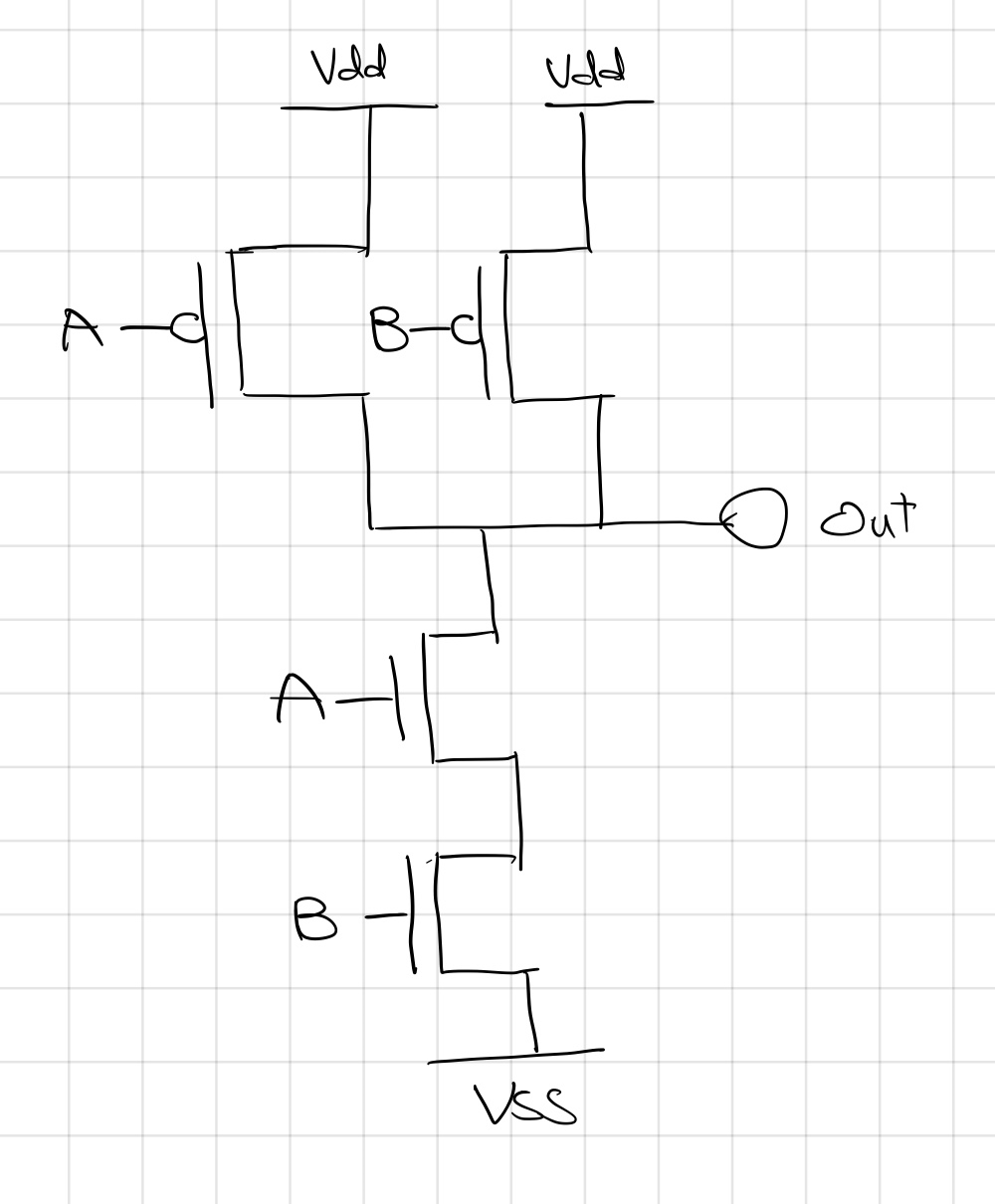
4주차 예비보고서

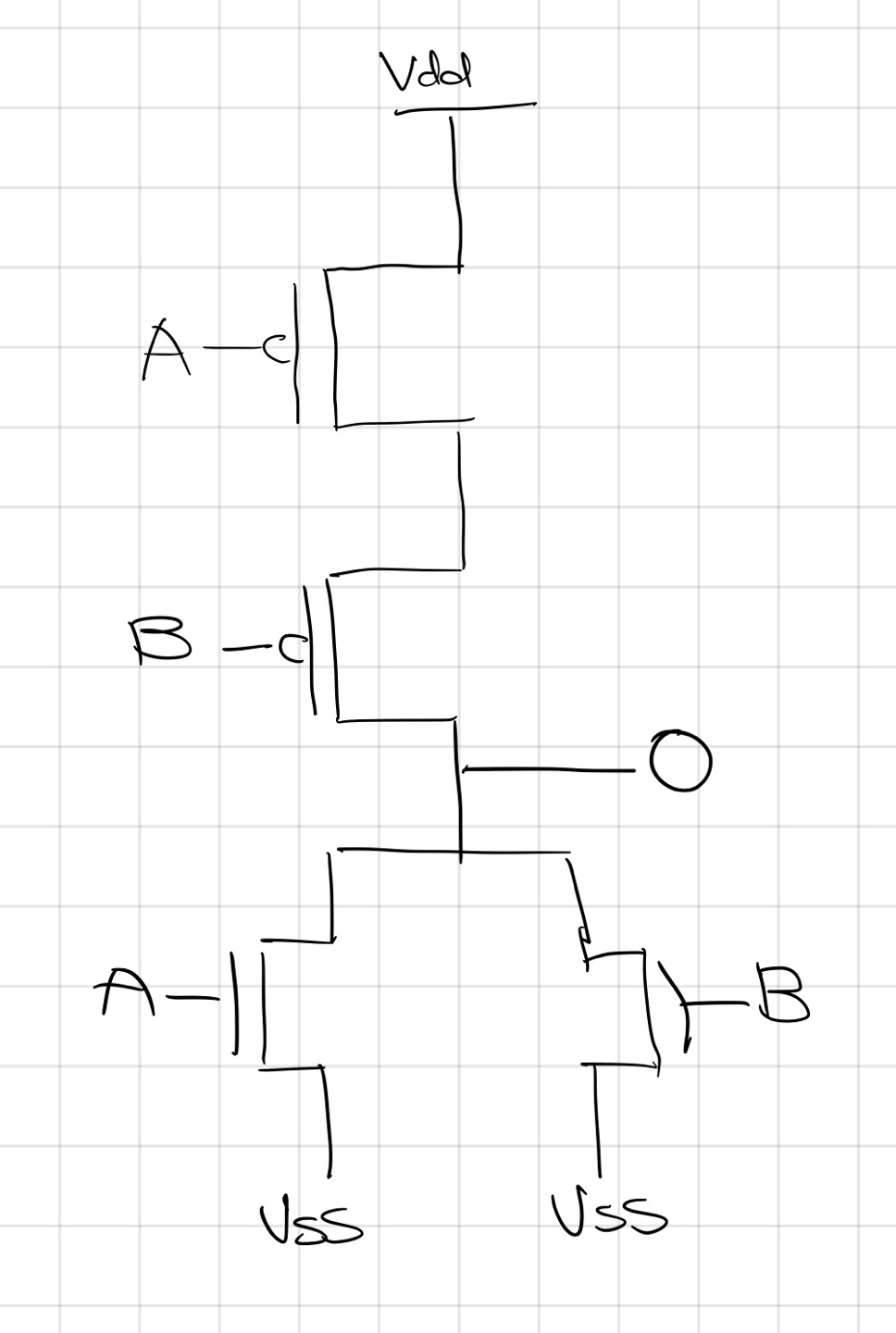
전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

**1. NAND/ NOR/ XOR Transistor – Level**

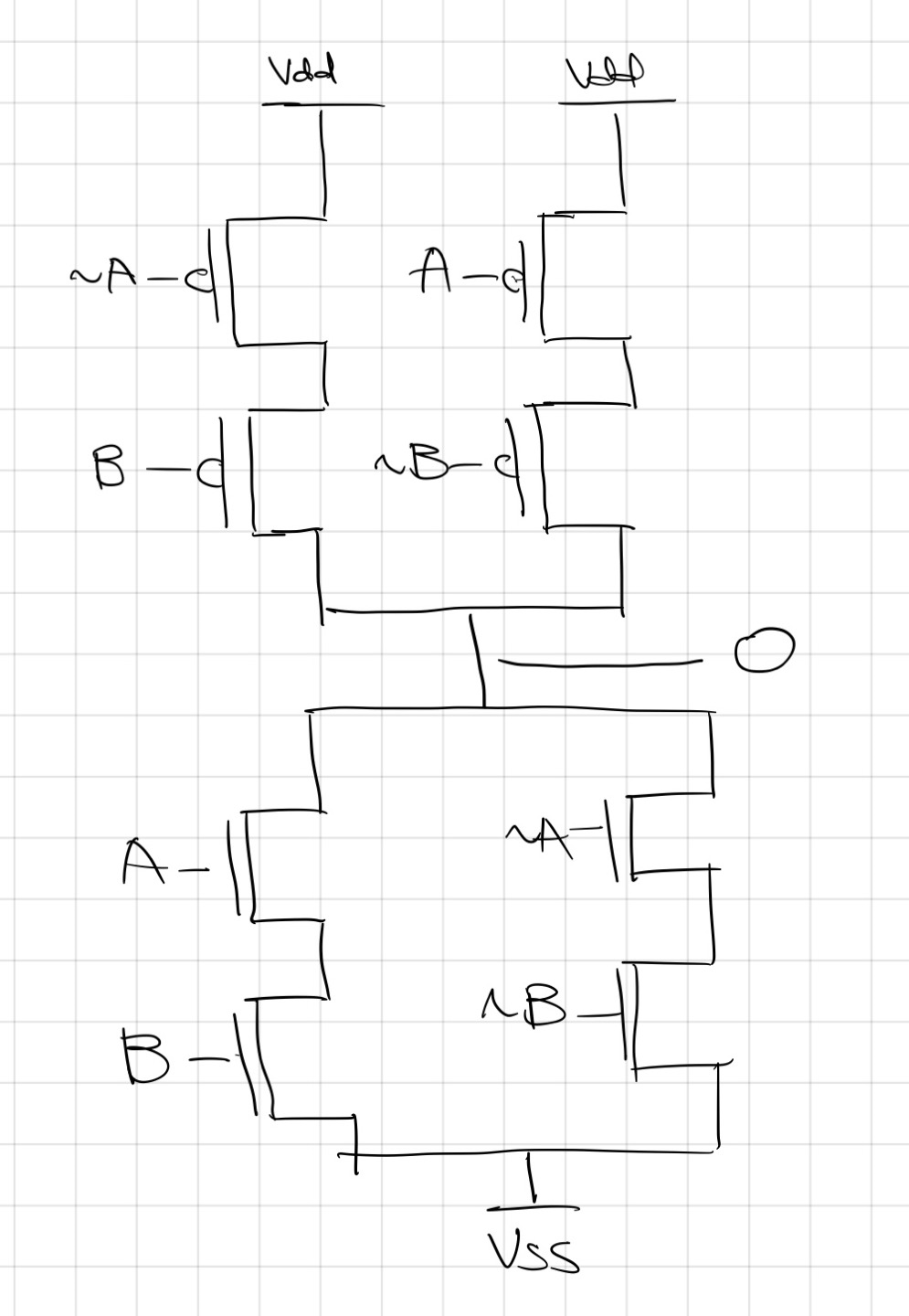
NAND의 CMOS는 다음과 같이 그려진다.



NOR의 CMOS는 다음과 같이 그려진다.



XOR의 CMOS는 다음과 같이 그려진다.

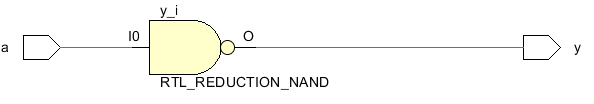


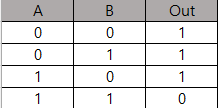
**2. 각 Logic 특성**

2.1 NAND

이 게이트는 이름에서 유추할 수 있듯, AND게이트와 NOT게이트를 합친 것이다. 기존 AND게이트의 경우 모든 논리 값이 참(1)일 경우에 참을 출력으로 내보낸다. 이 결과값을 NOT 게이트의 입력값으로 사용한다면, 거짓이 나오게 된다. 즉, 모든 입력이 참일 경우 항상 거짓을 출력으로 한다.

미국 국립 표준 협회에서 지정한 기호와 진리표 다음과 같으며 Boolean 식으로 쓴다. ( 진리표는 두개의 input을 기준으로 작성하였다.)

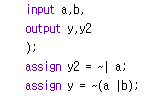


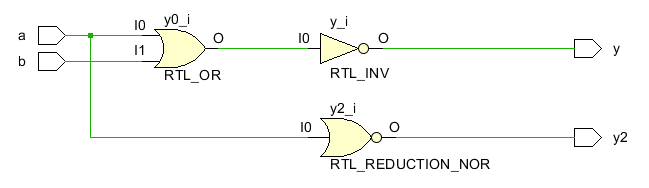


2.2 NOR

이 게이트는 NAND처럼 OR게이트와 NOT게이트를 합친 게이트이다. 기존 OR게이트의 논리 결과를 NOT게이트의 입력으로 한다. 따라서, 입력 받는 논리 값에 1이 포함되어있다면, 항상 0을 출력하며 모두 0이라면 1을 출력한다.

테이블이(가) 표시된 사진

자동 생성된 설명 마찬가지로 기호와 진리표는 다음과 같다. Boolean 식으로 쓴다.



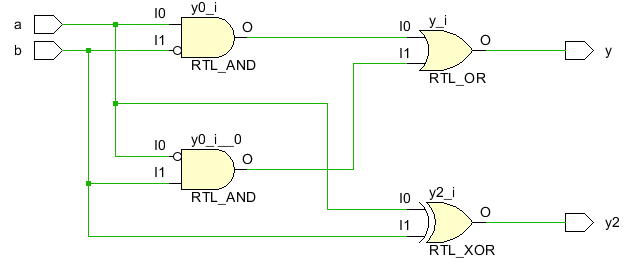
2.3 XOR

이 게이트는 조금 특이한 연산을 한다. 입력 값에서 1의 개수가 홀수라면 1의 출력을 내보내며, 반대로 짝수라면 0을 출력한다. 입력 값이 두개일 경우 Boolean 식은 으로 표기한다. (A⊕B로 표기하기도 한다.) 다음은, 진리표와 기호이다.

테이블이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

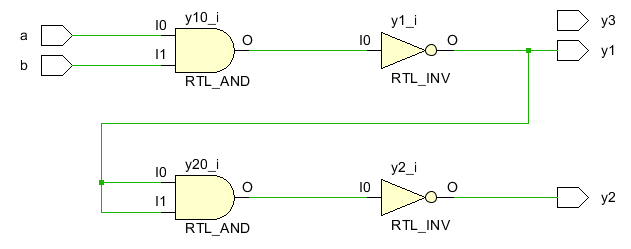
자동 생성된 설명



**3. 기본 논리게이트와의 변환 관계**

우선 두개의 input만을 가진다는 가정을 한다. 또한, 주로 실제 현업에서 AND, OR과 같은 게이트 보다 NAND, NOR게이트를 주로 사용한다고 한다. 이는 NOR과 NAND를 이용하면 모든 논리게이트를 표현할 수 있기 때문이며, Universal Gate라 부른다. 또한, CMOS의 구조상 OR이나 AND게이트에 비해 공간절약이나 게이트의 딜레이를 줄일 수 있기 때문이다.

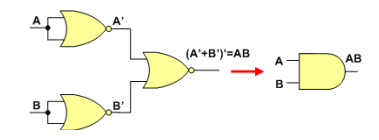
AND 게이트의 구현은 다음으로 이루어질 수 있다. Verilog를 통해 구현하려 했지만, NAND게이트가 AND + NOT의 형태로만 나와 참고자료로 대체한다.



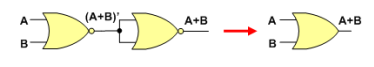
예로, Boolean 식으로 살펴보면, 이다. 드모르간 법칙에 의해 이는 {(A`+B`)\*(A`+B`)}` 이 되며, 기본법칙 X \* X = X 에 의해 (A`+B`)` 이 된다. 이는 다시 드모르간에 의해 A\*B 가 된다.

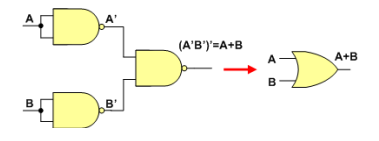
첫번째는 NAND만을 이용한 AND게이트이며, 두번째는 NOR만을 이용한 AND게이트이다.



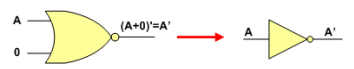
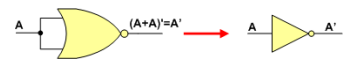
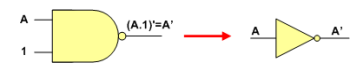
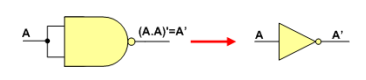


OR 게이트 역시 다음과 같이 이루어진다.





마지막으로 NOT 역시 다음과 같다.

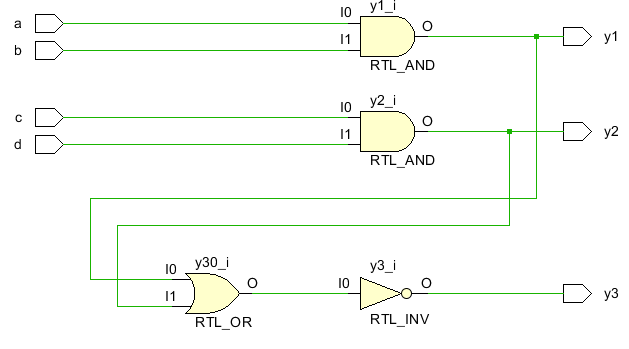


모두 앞선 예시에서 설명한 것 처럼 Boolean 식을 통해 설명이 가능하며, Verilog로 구현할 수 있다.

**4. AND – OR – INVERT logic 응용**

AOI라고 부르는 이 게이트는 AND 게이트와 NOR 게이트의 조합으로 이루어진 조합 논리 회로를 의미한다. 앞서 잠깐 말한 CMOS 구현 방식에서의 장점과 유사하다. 각 게이트를 각각 구현해서 합쳐놓은 것에 비해서 트랜지스터가 절약되며, 공간절약이 되기 때문이다. 트랜지스터의 절약과 공간의 절약은 면적과 비용 절감 등의 효과로 이어진다.

Boolean 식은 와 같이 쓴다.



**5. XOR logic 구현**

앞서 이미 XOR(Exclusive OR) 게이트에 대해서 대략적으로 설명했다. 앞선 예시에서 AND와 OR게이트를 이용해 XOR게이트를 구현했지만, 마찬가지로 NAND와 NOR게이트를 사용해서 구현할 수 있다. AND와 OR 게이트를 NAND와 NOR게이트로 그대로 바꾸어 적용하면 된다.

**6. 기타이론**

CMOS란, Complementary metal-oxide-semiconductor의 줄임말로, 오늘날 가장 널리 사용되는 집적회로의 한 종류이다. SRAM과 마이크로프로세서 같은 디지털 회로를 구성한다. MOSFET 기술을 기반으로 한다. CMOS는 기존 PMOS와 NMOS 고유의 문제를 두 유형을 통합하여 해결했다. 이에 회로설계의 유연성이 높아졌고, 전자 노이즈에 대한 복잡성과 민감성이 줄어들었다.

**7. 참고문헌**

Mitigation and Predictive Assessment of SET Immunity of Digital Logic Circuits for Space Missions - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Gate-level-and-transistor-level-representation-of-NAND2-X1-and-its-truth-table\_fig5\_339075490 [accessed 22 Sep, 2022]

<https://www.quora.com/Why-do-we-use-NAND-and-NOR-Gate-for-implementing-any-logic-design>

<http://www.uop.edu.pk/ocontents/Lec-10-universal%20gates.pdf>

<http://www.vlsi-expert.com/2018/01/aoi-and-or-inverter-cell.html>

https://www.techtarget.com/whatis/definition/CMOS-complementary-metal-oxide-semiconductor