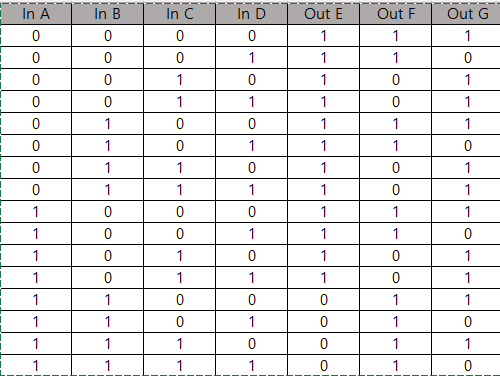
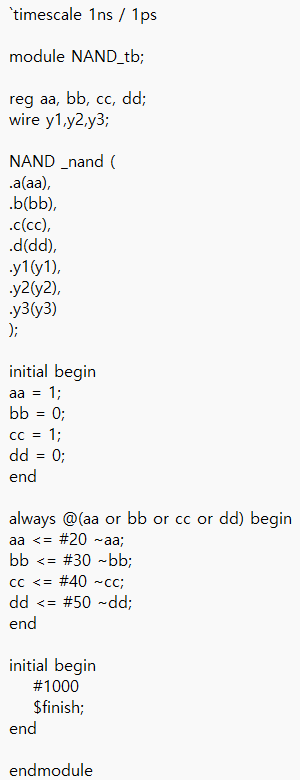
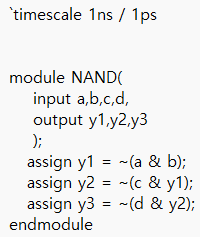
4주차 결과보고서

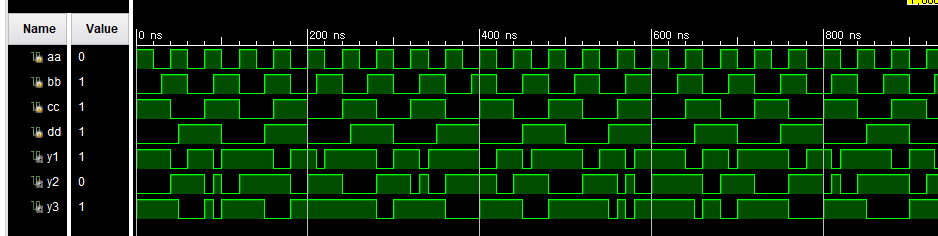
전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

**1. 목적**

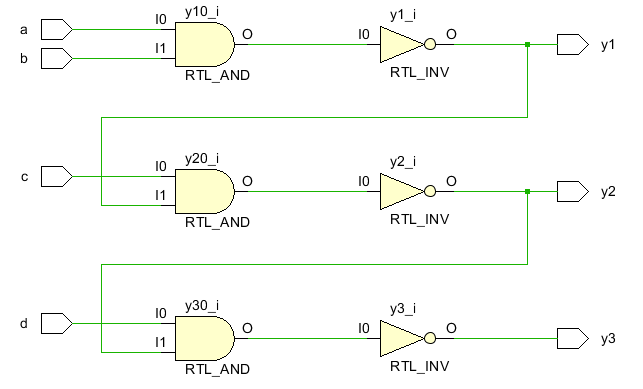
다소 생소할 수 있는 NAND/ NOR/ XOR /AOI 게이트를 직접 코드를 통해 구현해보면서, 어떤 식으로 동작하는지에 대한 이해 및 확인한다. 특히, 시뮬레이션과 직접 비교하여 진리표를 작성하고 FPGA 보드를 통해 회로 동작 결과를 확인한다.

**2. NAND**

****

****

NAND 게이트이 경우 AND게이트에 Not 게이트를 더한 것과 같은 결과를 출력한다. AND게이트의 경우 모든 input이 High(1)이어야만 output을 1로 출려했다. 그러나, NAND의 경우 모든 input이 High라면 0을 출력하며, 나머지 경우에는 1을 출력하게 된다. 이 회로의 경우 aa와 bb가 모두 1일 때, 시뮬레이션에서 y1은 0이 되는 것을 확인 가능하며, 그 외의 경우 모두 1임을 알 수 있다. 마찬가지로, cc와 y1이 input이 되고 y2가 output이 되는 경우도 동일하며, dd와 y2 가 input이고 y3가 output인 경우도 동일한 결과를 나타낸다.



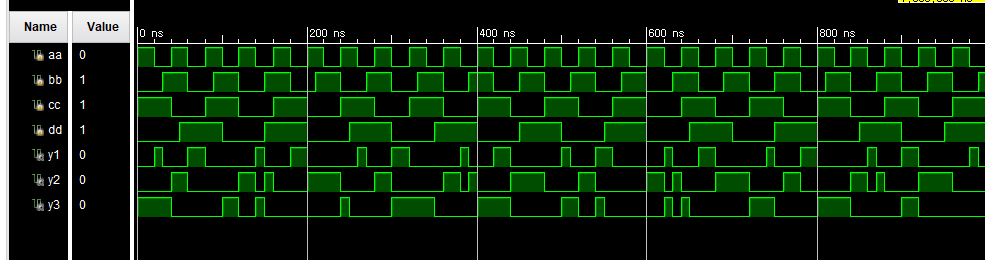
**3. NOR**

**테이블이(가) 표시된 사진

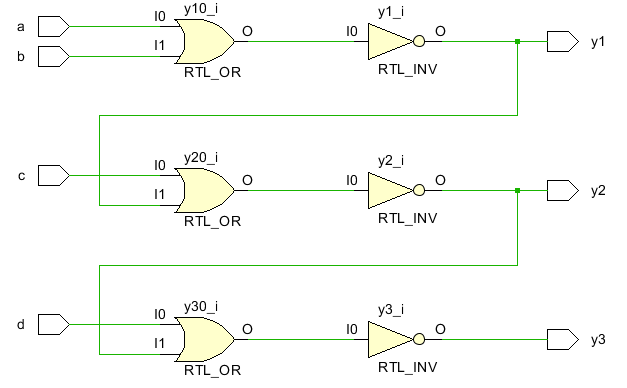
자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

****

NOR게이트의 경우 OR게이트와 NOT게이트를 합친 것과 같은 결과를 출력한다. 따라서 기존 OR게이트의 경우 입력이 모두 0인 경우 0을 출력하며 이를 제외하면 모두 1을 출력했던 반면, NOR은 입력이 모두 0일 때만 1을 출력하고 이를 제외하면 0을 출력한다. 시뮬레이션 결과를 보면, aa와 bb가 모두 0일 때 y1이 1의 값을 가지는 것을 알 수 있다. 또한, y1과 cc가 0일 때 y2가 0을 출력하며, y2와 dd가 0일 때 y3는 1을 출력한다. 이를 바탕으로 진리표를 작성할 수 있다.

****

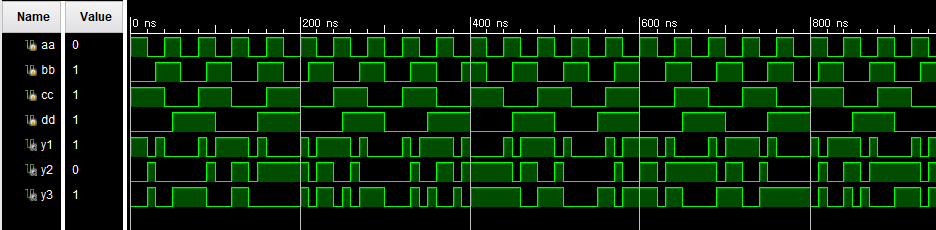
**4. XOR**

**테이블이(가) 표시된 사진

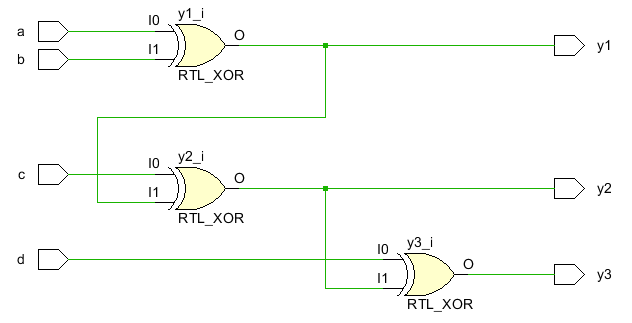
자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명**

****

XOR게이트는 앞선 게이트와 다르게 입력 값에서 1의 개수가 홀수라면 1의 출력을 내보내며, 반대로 짝수라면 0을 출력한다. 또한, 특별한 ⊕와 같은 기호를 사용한다. 시뮬레이션 결과를 살펴보면, aa와 bb가 서로 다른 값을 가질 때, y1이 1을 출력하는 것을 알 수 있다. 또한, y1과 cc의 값 역시 서로 다를 때 1을 출력하며, y2와 dd도 동일한 방식을 따른다.

****

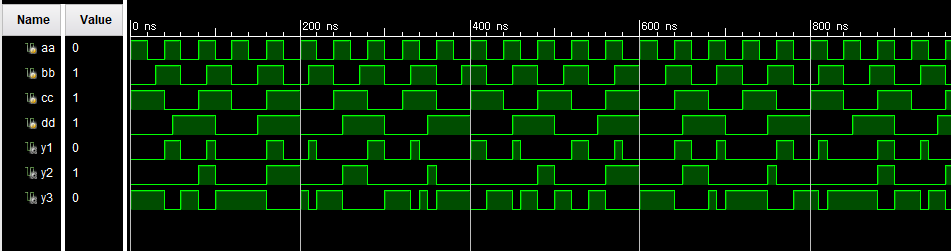
**5. AOI**

**테이블이(가) 표시된 사진

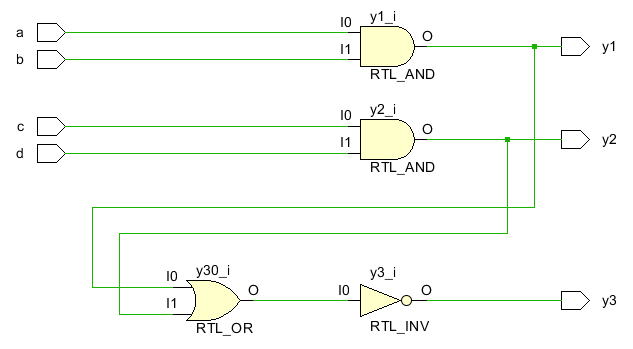
자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명**



앞서 예비보고서에서 조사한 바에 따르면, AOI게이트는 트랜지스터의 절약이나, 공간절약의 이점을 가지기 때문에 사용한다. AND-OR-INVERT라는 명칭처럼 AND와 NOR게이트(OR게이트 그리고 NOT게이트)를 사용한다. 실습에서는 4개의 aa,bb,cc,dd의 input이 있을 때, (aa,bb)가 첫 and 게이트의 input으로 들어가며, (cc,dd)가 두 번째 and 게이트의 input으로 들어간다. 각각의 output (y1,y2)들은 NOR게이트의 input이 되게 되며, 최종적으로 y3의 output이 출력된다. 최종 결과 y3는 y1과 y2의 값에 따라 나뉘며, 당연하게도 y1과 y2는 각각의 input들에 따라 결정된다.

****

**6. 결과 검토 및 논의 사항**

결과의 경우 예상한대로 나오게 되었다. 앞서 말했듯, NAND게이트는 AND게이트에 NOT게이트를 붙인 것과 동일하게 input이 모두 0일 경우 1을 출력했다. 또한, NOR의 경우는 input이 모두 1인 경우 0을 출력했다. XOR게이트는 Boolean 식이 임을 생각해보면, 1의 개수가 홀 수일 때 출력을 1로하며 짝수 일 때 0으로 출력했다. 마지막으로 AOI게이트의 경우 두개의 AND게이트와 하나의 NOR게이트로 이루어져 있다. 이를 통해, 두 개의 AND게이트 중 Output에 하나라도 1이 있다면, AOI게이트의 최종 출력은 0이 됨을 알 수 있다.

실습에서 다루지는 않았지만, NAND게이트 (a)와 (b)를 불 대수를 통해 알아보면 다음과 같다. 우선 (a)의 경우 ~(abcd)가 된다. 반면, (b)의 경우 ~(~(~(ab)c)d)가 된다. (a)는 드모르간 법칙에 의해 a’+b’+c’+d’ 바뀐다. (b)는 (((a’+b’)c)’d)’ 이며, 이는 다시 (a’+b’)c + d 가 된다. 따라서 서로 다른 결과를 도출한다는 점을 알 수 있다! 이때, 주의할 점은 NAND와 NOR게이트에 대해 불 대수를 사용할 땐 교환법칙과 결합법칙이 적용되지 않는다는 점을 알아야한다! (XOR과 XNOR게이트는 가능하다. )

**7. 추가이론**

왜 NAND게이트와 NOR게이트를 사용했는지 기억해보면, 이는 NOR과 NAND를 이용하면 모든 논리게이트를 표현할 수 있기 때문있으며, 비용이나 공간 절감에 유리하기 때문이었다. 그러나, 스케메틱으로 그려보면 이해되지 않는다. 즉 트랜지스터 레벨 상에서 이해해야한다!

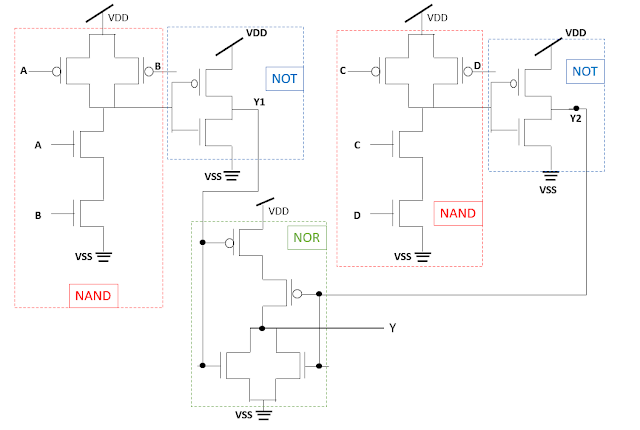
이유는 각 게이트에 사용되는 트랜지스터의 개수 때문이다. 김주호 교수님의 디지털 회로개론 강의자료를 참고하면, 각 게이트의 트랜지스터 수는 다음과 같다. 테이블이(가) 표시된 사진

자동 생성된 설명

NAND 게이트와 NOR게이트의 트랜지스터 수는 4로 적다.

또한, AOI경우 역시 이러한 장점을 가지고 있다. 그러나 아직 CMOS가 뭔지 제대로 이해하지 못했기 때문에, AOI의 CMOS transistor level을 소개한다. 다음 그림에서 트랜지스터는 총 8개가 필요하다. 그러나, 만약 NAND게이트와 NOR게이트를 각각 이용해 AOI를 구성하게 되면 총 14개의 트랜지스터가 필요함을 알 수 있다!

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

**8. 참고문헌**

디지털 회로개론 강의자료. 김주호 교수

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