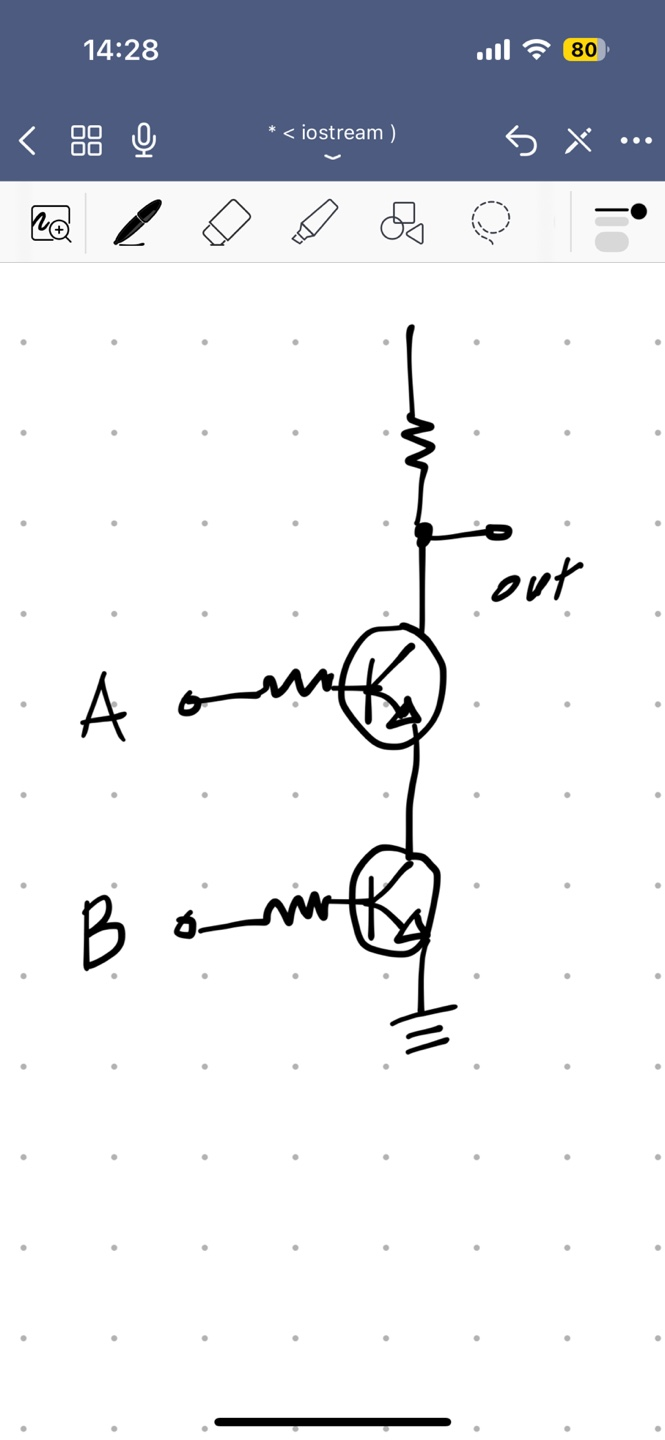
4주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231609 이름: 정희선

**1.**



* NAND gate

텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* NOR gate

텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* XOR gate

**2.**

1) **NAND gate**

Not And gate로, And gate의 출력에 not 연산을 한 논리회로이다. 따라서 and 연산의 반대 값을 결과로 출력한다. 입력이 모두 1인 경우 출력이 0이 된다는 특징을 가진다.

2) **NOR gate**

Not Or gate로, Or gate의 출력에 not 연산을 한 논리회로이다. 따라서 Or 연산의 반대 값을 결과로 출력한다. 입력이 모두 0일 때만 1을 출력값으로 가진다.

3) **XOR gate**

두 입력이 서로 다를 때 1을 출력하는 논리회로이다. 두 값을 비교할 때 주로 사용하는 게이트이다.

**3.**

NAND와 NOR 게이트는 AND, OR, NOT 게이트의 유니버설 게이트다. 즉, NAND와 NOR 게이트 만으로 모든 논리회로를 구현할 수 있다.

1. **AND**

* NAND로 AND 구현하기: NAND는 AND의 부정이므로, NAND 게이트를 NOT으로 다시 부정하면 AND를 구현할 수 있다.
* NOR로 AND 구현하기: 드 모르간 법칙을 사용한다. 먼저 두 입력에 NOT 연산을 취한 후, 그 결과를 NOR 연산해 구현한다.

1. **OR**

* NAND로 OR 구현하기: 드 모르간 법칙을 통해 구현할 수 있다. 먼저 두 입력을 각각 NOT한 후, 결과 값을 NAND 연산을 취한다
* NOR로 OR 구현하기: NOR는 OR의 부정이므로, NOR 게이트를 NOT으로 다시 부정하면 OR를 구현할 수 있다.

1. **NOT**

- NAND로 NOT 구현하기: NAND 게이트에 동일한 입력을 두 번 넣으면 NOT 게이트로 동작한다.

- NOR으로 NOT 구현하기: NAND와 마찬가지로 같은 입력을 두 번 NOR 게이트에 넣으면 NOT 연산을 구현할 수 있다.

**4.**

**AND-OR-INVERT**(AOI) 로직이란, AND와 OR연산을 결합하여 NOT연산을 적용하는 방식이다. 이 방식은 논리식을 효율적으로 구현하고, 복잡한 회로를 간단하게 표현하는데 사용된다.

- 최적의 논리 회로: 게이트 수를 줄여 회로의 복잡도를 줄이는 데 사용된다.

- 전력의 효율적인 소모: 불필요한 게이트를 줄여 전력 소모를 줄이는 데 사용된다.

유사한 예로, **OR-AND-INVERT**(OAI) 로직이 있다. 이는 OR연산 수행 후 AND연산을 수행하고 최종적으로 NOT 연산을 적용하는 방식이다. 이 방법 역시 복잡한 논리식을 단순화하는 데 사용된다.

**5.**

**XOR 로직은** 앞서 이야기했듯 입력이 서로 다를 때 출력이 1이 되는 연산이다. 이를 구현하기 위해 AND, OR, NOT 게이트를 사용한다. 논리식은 다음과 같다.

A⊕B=(A⋅~B)+(~A⋅B)

- 먼저, 첫번째 AND 게이트에서 A와 NOT B를 입력으로 받는다.

- 다음으로 두번째 AND 게이트에서 NOT A와 B를 입력으로 받는다.

- 두 게이트의 출력을 OR 게이트로 결합해 출력을 얻는다.

**6.**

**드 모르간(De Morgan) 법칙**: 논리 연산에서 AND와 OR 연산의 보수 관계를 설명하는 이론이다. 이 법칙은 복잡한 회로를 단순화해 효율적인 회로를 만드는데 기여한다.

* ~(A⋅B)=~A+~B: AND 연산의 NOT은 각 입력의 NOT들을 OR연산으로 변환한 것과 같다.
* ~(A+B)​=~A⋅~B: OR 연산의 NOT은 각 입력의 NOT들을 AND연산으로 변환한 것과 같다.