3주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20191583 이름: 김태곤

**1. 논리게이트 AND/OR/NOT의 구조를 Transistor-Level로 그리시오.**

1) AND

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

2) OR

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

3) NOT

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**2. AND/OR/NOT Logic의 특성에 조사하시오.**

1) AND Logic 특징

AND Logic은 모든 입력이 참(1)일 때만 출력이 참(1)이 되는 논리 연산을 수행한다.

2) OR Logic 특징

OR Logic은 최소한 하나의 입력이 참(1)일 때 출력이 참(1)이 되는 논리 연산을 수행한다.

3) NOT Logic 특징

NOT Logic은 단일 입력의 논리적 반전을 출력한다. 입력이 참(1)이면 거짓(0)을, 거짓(0)이면 참(1)을 출력한다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | AND | OR | NOT |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

**3. Fan-out에 대하여 조사하시오.**

Fan-out은 논리 회로에서 하나의 논리 게이트의 출력이 얼마나 많은 논리 게이트의 입력으로 사용되는지에 대해 서술할 때에 쓰인다. 이는 하나의 디지털 회로 출력이 안정적으로 구동할 수 있는 다른 회로 입력의 최대 수를 나타낸다. 즉 하나의 회로 요소(게이트, 트랜지스터, 또는 칩)의 출력이 연결될 수 있는 입력 단자의 수이다. Fan-out이 크다는 말은 하나의 출력이 많은 논리 게이트의 입력으로 사용될 수 있다는 것을 의미한다. 하나의 출력 신호가 너무 많은 입력에 연결되면, 출력이 제대로 모든 입력을 구동하지 못할 수 있고, signal quality와 timing 문제 때문에 Fan-out은 중요하다. 이를 해결하기 위해 버퍼를 사용하거나 로직 게이트의 종류 변경, 분산 시스템 디자인 등 여러 방법으로 Fan-out을 줄여 신호의 정확성과 신뢰도를 보장한다.

**4. 전파지연에 대하여 조사하시오.**

전파지연(propagation delay)란 신호값의 변화가 입력에서 출력까지 전달되는 데 걸리는 시간을 의미한다. 이는 신호가 한 지점에서 다른 지점으로 전달되는 데 걸리는 시간을 의미하며, 회로의 속도와 성능을 결정하는 중요한 요소 중 하나이다. 전파 지연은 다양한 요인에 의해 영향을 받으며, 이에 따라 신호가 게이트, 트랜지스터, 또는 회로의 다른 구성 요소를 통과하는 데 필요한 시간이 달라질 수 있다. propagation delay 식은 다음과 같이 표현할 수 있다.

*TPD = (TPHL+TPLH)/2*

전파 지연 시간 = 입력 신호에 의해 출력이 1에서 0으로 변하는 시간+ 입력 신호에 의해 출력이 0에서 1로 변하는 시간의 평균이다. 여러 게이트를 통과할수록 더 많은 지연 시간이 걸리며, 전선의 길이, 온도 등 여러 요인에 의해 달라질 수 있다. 이를 줄이기 위해 회로 최적화, 소자 선택, 회로 설계 등 효율적으로 설계를 해야한다.

**5. Verilog 의 task 및 function 에 대해 조사하시오.**

Verilog의 task와 function은 코드의 재사용성을 높이고, 복잡한 설계를 더 쉽게 관리할 수 있도록 도와주는 중요한 문법이다. C언어의 function처럼 코드를 간결하게 만들 때 사용한다. task와 function 모두 특정 작업을 수행하는 코드 블록을 정의하는 데 사용되지만, 사용 방법과 제한 사항에 있어서 차이점이 있다.

1) task

task는 시작과 끝을 task와 endtask로 구성하여 선언한다. 값의 반환 없이 호출될 수 있으며, 여러 출력 값을 가질 수 있습니다. 출력 및 inout 인자를 통해 호출한 쪽에 여러 값을 전달할 수 있다. task는 function과 달리 외부 변수의 값을 변경하거나, 시스템의 다른 부분과 상호작용할 수 있다. 즉 task가 보다 복잡한 작업을 수행하는데 적합함을 의미한다. 또한, 시간 지연 delay 기능을 사용할 수 있으며, 사용하지 않는 경우 합성이 가능하다.

2) function

function은 시작과 끝을 function과 endfunction으로 구성하여 선언한다. 반드시 하나의 값을 반환해야 하며, 이 값의 데이터 타입은 함수를 정의할 때 지정해야 한다. 함수는 입력 인자만을 사용해 계산을 수행하며, 함수 외부의 변수를 변경하거나, 출력 및 inout 포트에 접근할 수 없다. 이는 함수가 순수 계산 목적으로만 사용됨을 의미한다. function은 항상 같은 시뮬레이션 타임에서 실행되며, 호출될 때 추가적인 시뮬레이션 시간을 소비하지 않는다.

간단하고 순수한 계산을 위해서는 function을 사용하고, 보다 복잡하거나 여러 출력 값을 요구하며, 시간이 소요되는 작업을 위해서는 task를 사용하는 것이 좋다.