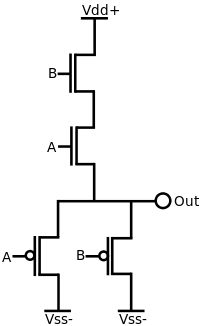
전공: 컴퓨터공학 학년: 2학년 학번: 20161603 이름: 신민준

1. 논리 게이트 AND/OR/NOT의 구조를 Transistor-Level로 그리시오.

* AND gate

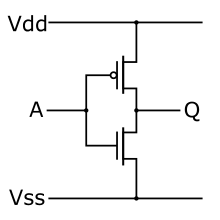


* OR gate

실내, 별, 검은색이(가) 표시된 사진

자동 생성된 설명

* NOT gate



1. AND/OR/NOT Logic의 특성에 대해 조사하시오.

AND Logic의 경우, 두 개의 input을 받으며, 각 input이 둘 다 1로 동일한 값을 가질 때에만 출력으로 1을 내보내고, 나머지의 모든 경우에 대해서는 0을 출력한다. 일반적으로 NAND Gate 2개로 만들거나, NOR Gate 3개로 구현한다.

이와 달리, OR Logic은 두 개의 input 중 어느 하나라고 1이 있을 경우, 1을 출력하고, 만약 두 input 모두 다 0인 경우에만 출력으로 0을 내보낸다. NAND Gate 3개나 NOR Gate 2개로 구현이 가능하다.

NOT Logic의 경우, Inverter라고도 하며, 입력으로 하나의 값을 받고, 입력 받은 값과 반대의 값을 출력으로 보내주는 역할을 한다. Verilog 문법에서는 Y<=!A 처럼 나타낼 수 있다.

1. Fan-out에 대하여 조사하시오.

디지털 회로에서 사용 되는 표준논리소자들 중에서, 1개의 출력신호에 대해 접속 가능한 최대 입력 신호 수가 정해져 있다. 이를 Fan-out이라 한다. 이러한 Fan-out이 정해져 있는 이유는 우선 각 소자에서 Output을 담당하는 부분에는 흐를 수 있는 최대 전류가 정해져 있으므로, 이 Fan-out 제한을 초과한 부하를 가하면 출력 전류가 너무 지나치게 커지기 때문에 회로에 손상이 올 수도 있기 때문에 이를 방지하거나, 입력 신호의 논리 상태를 보전하기 위해서이다.

Fan-out의 값은 output 부분의 output current와 input의 input current의 비율에 의해서 결정되며, 그 식은 다음과 같다.

1. 전파지연에 대하여 조사하시오.

전파지연은 Propagation Delay라고 부르며, 회로 상에 있는 하나의 노드에서 다른 노드로 데이터나 정보가 전파될 때 그 과정의 통신 링크 상에서 걸리는 시간을 의미한다.

전파지연 시간은 거리를 전파속도로 나눈 것을 의미한다. 전파 속도는 사용하는 링크의 매체에 따라 달라지므로, 전파 지연 또한 매체에 따라 달라지게 된다.

1. Verilog의 task 및 function에 대해 조사하시오.

Verilog의 task는 C언어의 함수(function)과 비슷한 면모도 있지만 확실하게 차이가 존재한다. 먼저, task의 경우, 다른 task나 function을 사용할 수 있다. 또한, task는 non-zero의 시뮬레이션 시간에 수행이 가능하고, 지연이나 사건, 또는 타이밍을 제어하는 코드를 포함할 수 있다. 또한, input/output/inout을 다수로 가질 수 있거나, 아니면 필요에 따라 하나도 가지지 않을 수도 있다. 또한 값을 되돌릴 수 없으며, output/input의 값을 이용해 여러 값들을 동시에 전달하는 것이 가능하다.

반대로, Function의 경우, 다른 함수 사용은 가능하지만 다른 task의 사용은 불가능하다. 또한, 항상 시간 0에 시뮬레이션을 수행할 수 있다. Task와 달리 지연, 사건, 타이밍 제어 코드를 사용할 수 없고, 최소 하나의 input 인수를 가져야 작동한다. 또한, 언제나 하나의 값을 리턴하므로, output과 input 인수를 가지지 않는다.

이러한 task나 function을 활용하면, 작성한 모델의 이식성, 가독성을 크게 향상시켜주며, 유지보수가 용이하게 만든다.

Verilog의 function은 대체적으로 FORTRAN언어의 function과 유사하다. C언어와는 다르게, Verilog에서의 함수는 function 키워드를 사용해 함수의 시작 줄을 나타내고, 같은 함수에서, 그 끝에는 endfunction 키워드를 반드시 명시해주어 함수가 해당 키워드에서 끝남을 알려야 한다.

Task는 FORTRAN의 subroutine과 유사한데, 이 또한 function과 비슷하게, task의 시작을 task라는 키워드로 알리고, task의 끝을 endtask라는 키워드를 사용해 표시한다.

위에서 소개한 task와 function을 사용해 AND 연산을 수행하는 프로그램은 다음의 Figure 1과 같다.

|  |
| --- |
| **function** foo;  **input** a, b, c;  **begin**  y = a & b;  **end**  **endfunction**  **function** bar;  **output** y;  **input** a, b;  **begin**  y = a & b;  **end**  **endfunction** |

Figure