

3주차 예비보고서

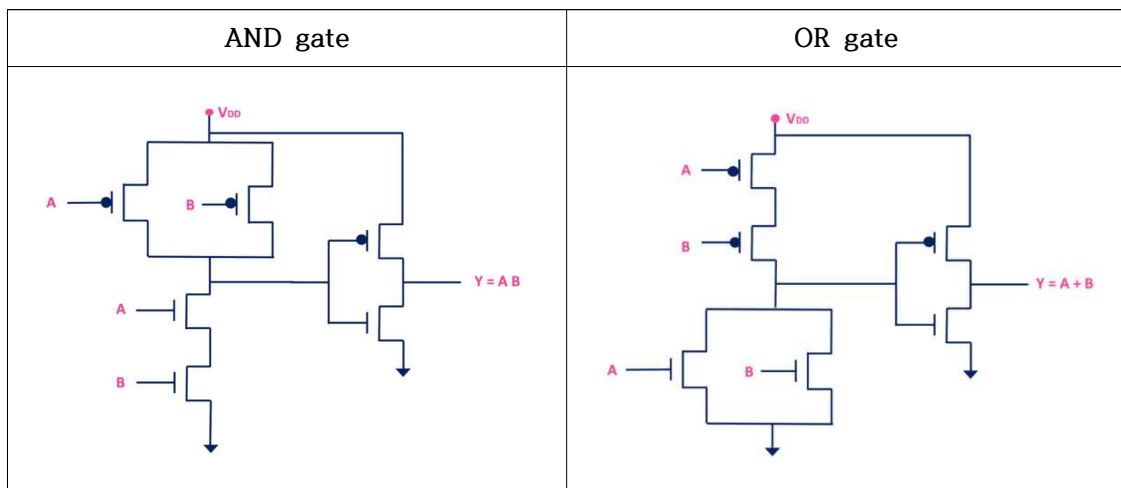
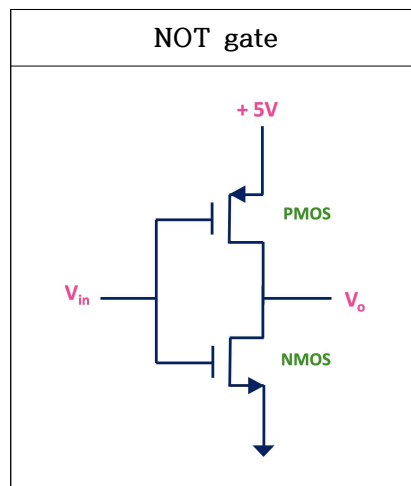
전공: 신문방송학과

학년: 4학년

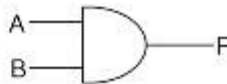
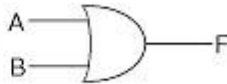
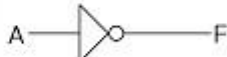
학번: 20191150

이름: 전현길

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



2. AND/OR/NOT Logic의 특성을 조사하시오.

게이트	논리기호	논리식	진리표															
AND		$F = AB$ $= \overline{A+B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = A+B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT (=Inverter)		$F = \overline{A}$	<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0									
A	F																	
0	1																	
1	0																	

AND 게이트는 논리곱을 구현하는 디지털 논리 게이트로, 두 개 이상의 입력 신호를 받고 모든 신호가 high(1)일 때 high(1)를 출력하고 그 외엔 low(0)를 출력한다. Boolean 식으로는 $F = AB$ 로 표현할 수 있으며, 진리표, 논리기호는 위의 그림과 같다.

OR 게이트는 논리합을 구현하는 디지털 논리 게이트로, 두 개 이상의 입력 신호를 받고 신호 중 하나라도 high면 high를 출력하고 그 외엔 low를 출력한다. Boolean 식으로는 $F = A+B$ 로 표현할 수 있으며, 진리표, 논리기호는 위의 그림과 같다.

NOT 게이트는 Inverter라고 불리며, 입력값에 반대되는 값을 출력한다. 입력값이 high라면 low, low라면 high를 출력한다. Boolean 식으로는 $F = \overline{A}$ 로 표현할 수 있다.

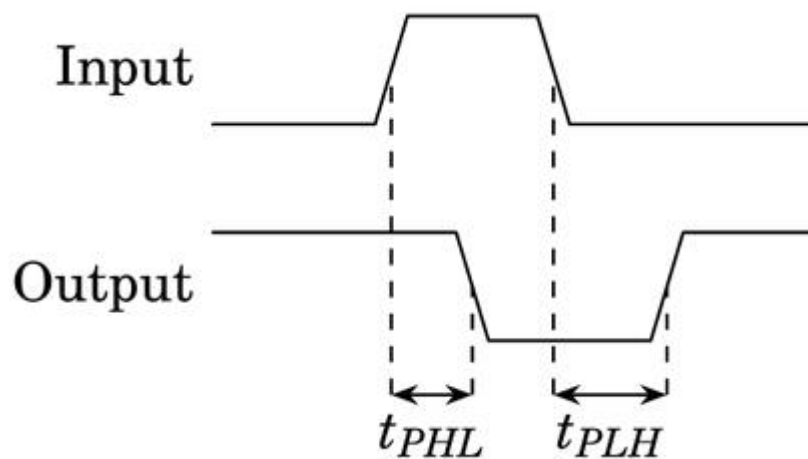
3. Fan-out에 대하여 조사하시오.

Fan-out이란 한 논리 게이트의 출력이 얼마나 많은 다른 논리게이트의 입력으로 사용되는지를 의미한다. 일반적으로 CMOS와 같은 표준 논리 소자들은, 한 개의 출력 신호에서 연결할 수 있는 입력에 대해 제한이 있는데, 이를 fan-out이라고 부른다.

한 게이트의 출력을 fan-out을 초과하여 다른 논리 게이트에 연결하게 되면, 1)신호의 논리 상태를 보장할 수 없는 문제, 2) 출력전류 과다로 회로가 쇼트되는 문제 등이 발생할 수 있다. 따라서, 팬아웃을 초과해 회로를 연결해야 할 경우 중간에 버퍼를 사용하는 것이 좋다.

팬아웃에 대응되는 개념으로 fan-in이 존재한다. 이는 반대로 하나의 게이트가 받을 수 있는 입력의 최대수를 의미한다. 예를 들어, 한 논리 게이트가 5개의 입력을 받을 수 있다면 해당 게이트의 fan-in은 5이다.

4. 전파지연에 대하여 조사하시오.



전파 지연(propagation delay)이란 논리 회로가 입력 신호를 받고 출력 신호를 출력할 때까지 걸리는 시간을 의미한다. 전파 지연 시간은 크게 t_{PLH} 와 t_{PHL} 로 나뉘는데, 각각 입력신호에 반응해 출력이 Low에서 High로 변화하는데 걸리는 시간, High에서 Low로 변화하는데 걸리는 시간을 뜻한다. 여러 개의 게이트를 통과할수록 전체 회로의 전파 지연 시간이 길어지기 때문에, 통과하는 게이트가 불필요하게 길어지지 않도록 설계해야 한다.

5. Verilog의 task 및 function에 대해 조사하시오.

Verilog에서 task와 function은 코드의 중복을 줄이기 위해 재사용 가능한 코드 블록, 서브루틴(subroutine)을 정의하는 방법이다. 이 중 function은 간단한 작업, 동기적인 작업을 처리하는 데에 주로 사용되며, task는 비교적 복잡한 작업 또는 비동기 작업을 처리하는 데에 주로 사용된다.

특징	function	task
입출력	최소 1개 이상의 input이 존재, output/inout이 없음	input, output, inout 제약 없음
반환값	return 값이 존재함	output port를 통해 값을 전달함
호출	다른 function만 호출할 수 있음	task와 function 모두 호출할 수 있음
타이밍 제어	delay, wait 등 타이밍 제어 불가	delay, wait 등 타이밍 제어 가능(#, @)
시뮬레이션 시간	항상 0초에서 시작 (zero simulation time)	0초 이외에도 시작 (non-zero simulation time)
용도	간단한 산술, 논리 연산에 사용됨	비동기 연산, 복잡한 연산에 사용됨
문법	task (...) endtask	function (...) endfunction

cf) **Blocking 방식**: 호출된 즉시 연산을 수행하고, 연산이 끝나면 호출한 위치로 값을 반환하는 방식. function은 task와 달리 동기적 연산을 수행하므로 blocking 방식으로 작업을 처리한다. 반대로 task는 시간 지연이 포함된 작업, 비동기적 작업을 수행할 수 있다.