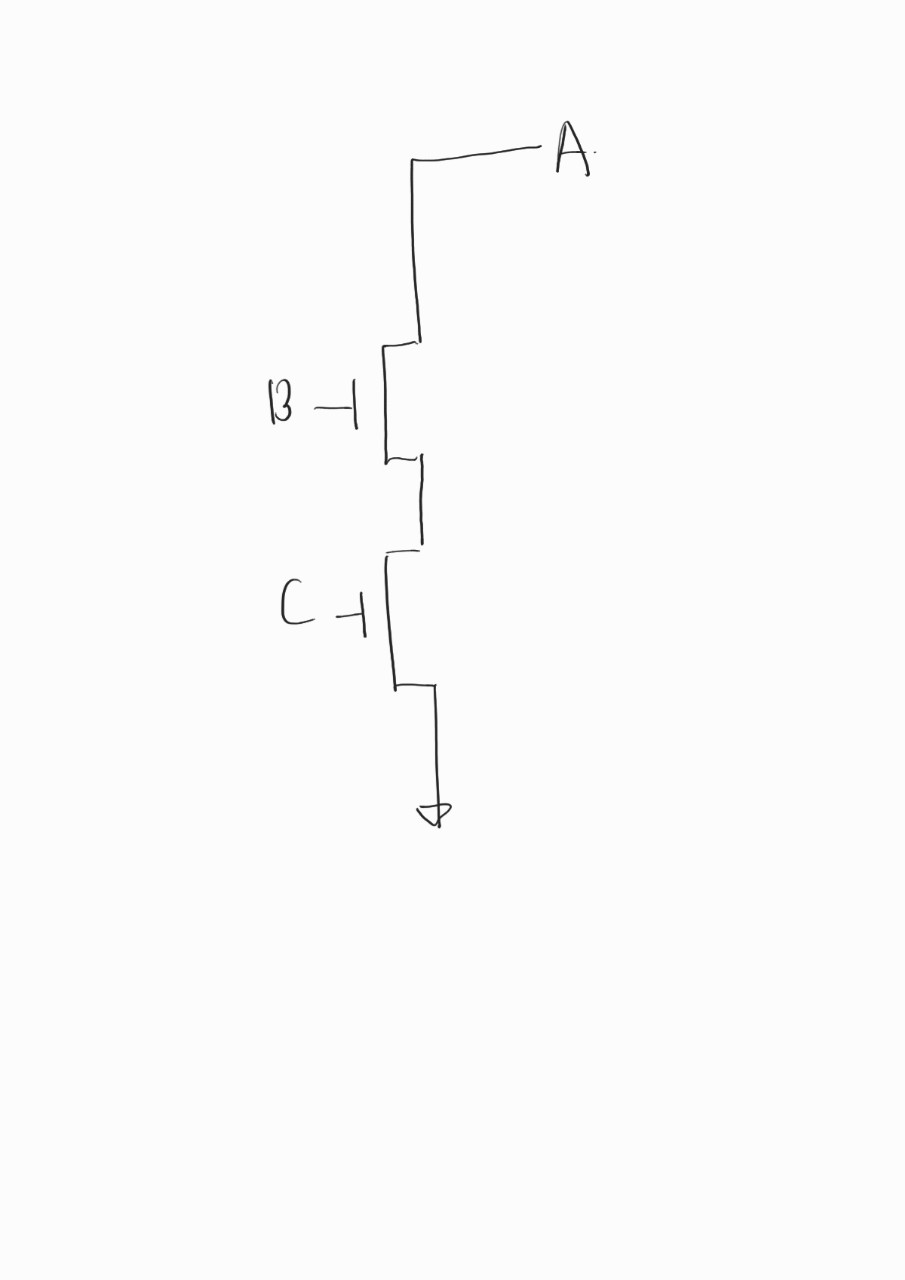
3주차 예비보고서

전공: 수학과 학년: 4학년 학번: 20171273 이름: 심현우

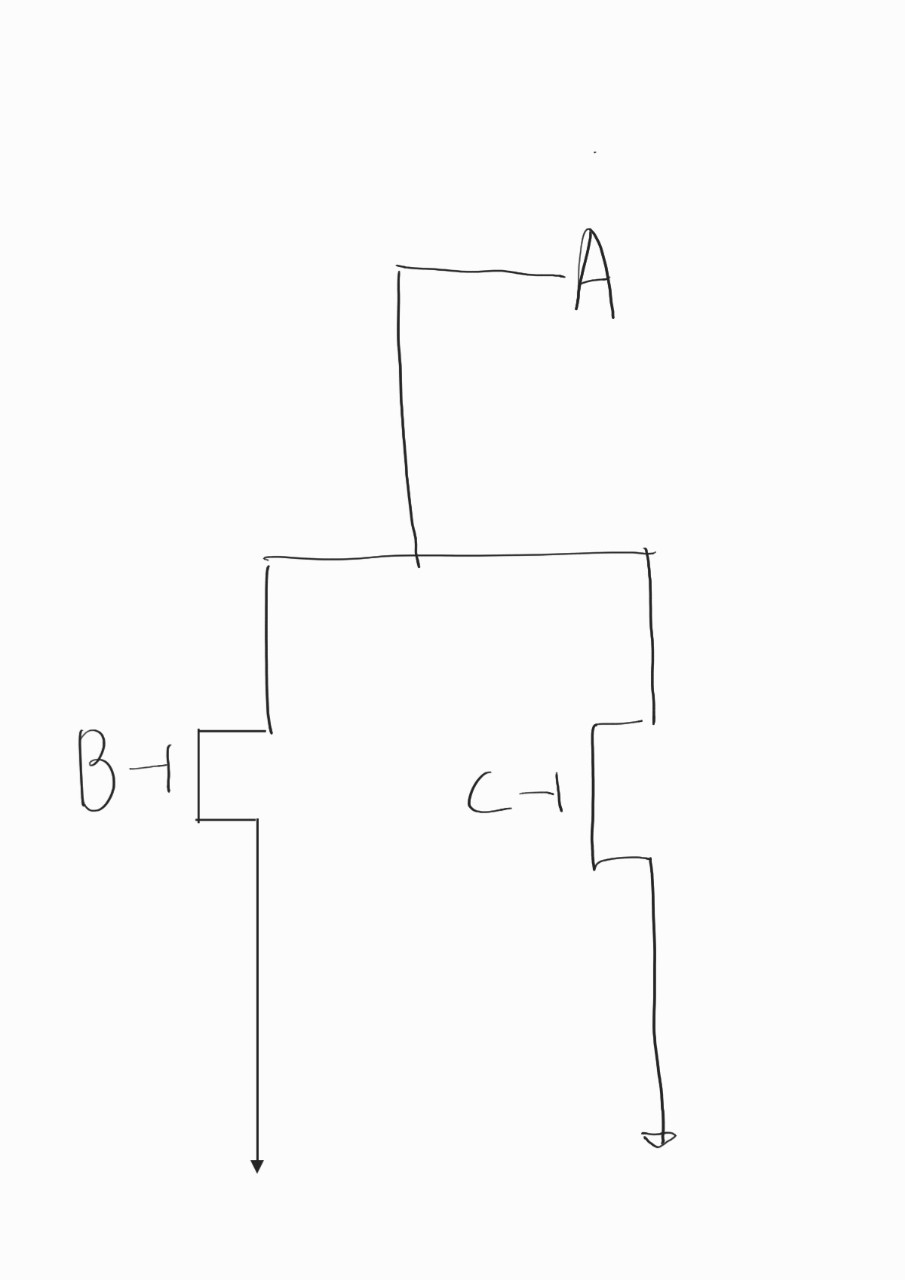
1. **논리게이트 AND/OR/NOT의 구조를 Transistor-Level로 그리시오.**
2. AND

논리게이트인 AND는 논리의 곱을 표현한다. 식으로 구현하면 A = BC 이다. 이를 transistor-level로 그리게 되면 다음과 같다.



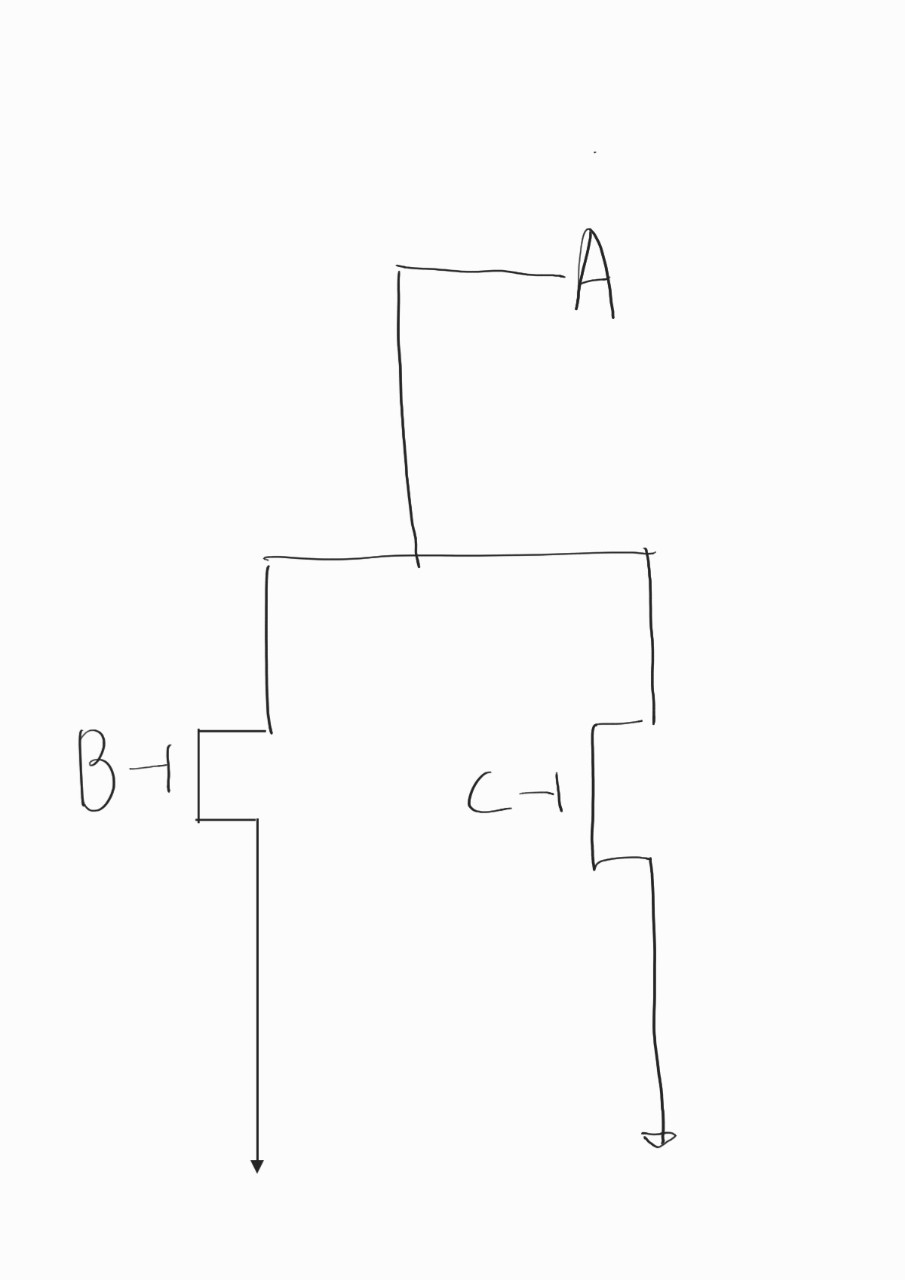
1. OR

논리게이트인 OR는 논리의 합을 의미한다. 식으로 구현하면 A = B + C 이다. 이를 transistor-level로 그리게 되면 다음과 같다.



1. NOT

논리게이트인 NOT은 단순 input과 output에 관한 게이트이다. 예를 들어 1이 input으로 들어오면 0을 output으로 하는 반대 연산을 한다. 식으로 구현하면 A = 이다. 이를 transistor-level로 나타내게 되면 다음과 같다.

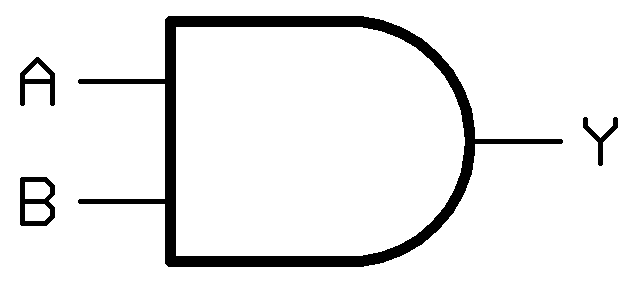


1. **AND/OR/NOT Logic의 특성에 대해서 조사하시오.**
2. AND

AND는 위에서 잠깐 언급했던 것과 같이 논리의 결합 중 논리 곱에 해당한다. 이는 digit와 연관해서 truth table, 즉 진리표를 생각하면 쉽게 이해할 수 있다.

|  |  |  |
| --- | --- | --- |
| A | B | AB |
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

위의 진리표는 A와 B에 대한 input이고 그에 따른 논리 곱, 즉 AND의 결과이다. 이는 실제로 곱하는 연산과 똑같이 결과를 갖게 된다. 이를 logic gate symbol로 나타내면 다음과 같다.

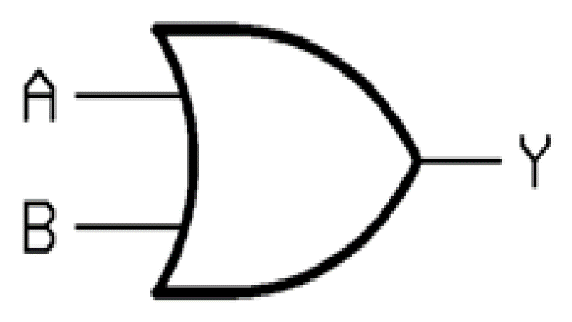


1. OR

AND와 마찬가지로 회로에서 여럿 input을 연산을 통해 하나의 output으로 만드는 logic gate이다. 이는 AND와는 다르게 논리의 합에 해당한다. OR의 말 그대로 해석하여 input 중 하나만 1이라도 결과를 1로 만드는 역할을 한다. 진리표로 나타내면 다음과 같다.

|  |  |  |
| --- | --- | --- |
| A | B | A+B |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

위의 진리표는 A와 B input에 대한 OR gate의 결과값이다. 이를 logic gate symbol로 나타내면 다음과 같다.

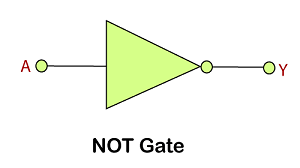


1. NOT

NOT은 여러 input을 받지는 않지만 어떠한 input에 대한 특정 계산을 통해output을 내는 logic gate이다. Logic 자체는 1의 값을 0으로, 0의 값을 1로 바꿔주는 역할을 한다. 진리표로 보면 다음과 같다.

|  |  |
| --- | --- |
| A |  |
| 1 | 0 |
| 0 | 1 |

위와 같은 반대의 값을 결과로 갖는다. 이를 logic gate symbol로 나타내면 다음과 같다.



1. **Fan-out에 대하여 조사하시오.**

Fan-out은 디지털 회로에서 gate에 의해 출력된 결과값이 다른 gate의 input으로 들어갈 때 접속할 수 있는 gate의 제한을 Fan-out이라고 한다. Fan-out을 지정하는 가장 큰 이유는 회로 특성상 전류가 흐를 수 있는 양이 제한적이기 때문이다. 만약 제한을 넘겨 Fan-out을 초과하게 되면 회로가 손상될 수 있다. 이는 안정적인 회로 운영을 위해 Fan-out을 지정한 것이라 볼 수 있다. 또한 회로의 결과에 대한 신뢰를 보장하기 위해 Fan-out을 지정하기도 한다. 이는 Fan-out을 초과하는 input 접속 시 신호 상태를 보장할 수 없기 때문에 결과에 대한 신뢰도가 떨어질 수 있다. 따라서 Fan-out은 반드시 지정해야 하며 Fan-out은 Input과 output current에 비에 따라 정해진다.

1. **전파지연에 대하여 조사하시오.**

전파 지연이란 회로에서 신호 값이 입력부터 출력까지 전달되는데 걸리는 시간으로 두가지, T(PLH)와 T(PHL)로 나뉜다. T(PLH)는 Propagation from low to high의 약자로 결과값이 low, 즉 0에서 1로 바뀌는 데 걸리는 시간을 의미한다. 반대로 T(PLH)는 1에서 0으로 변하는 데 걸리는 시간을 의미한다. 따라서 총 전파 지연 시간을 구하는 공식은 (T(PLH)+T(PHL))\*1/2이다.

1. **Verilog의 Task 및 Function에 대해 조사하시오.**

verilog에서 task란 코드를 실행할 수 있는 일련의 과정을 의미한다. 이는 크게 보면 timing을 제어하고 다른 task와 function을 호출할 수 있다. Function의 기능보다 조금 더 포괄적이고 여러 결과를 계산할 수 있다. Task는 task ; input, input, output, begin (code) end endtask으로 작성하며 위에 지정한 input과 output, 밑의 코드를 통해 작업을 주고받는다. 반면 function의 경우 task보다 좁은 범위에서 사용된다. Function을 사용하기 위해서는 조건이 갖춰져야 하는데, 우선 delay, timing, event control이 없어야 하며, 무조건 하나의 값을 반환한다. 또한 적어도 하나의 입력이 있어야 하며, task와 다르게 output을 인수로 갖지 않아야 한다. Function은 function ; input ; begin (code) end endfunction으로 사용된다.