

2018년 2학기 컴퓨터공학실험Ⅱ  
CSE3016-05반 3주차 예비 보고서

20171665 이 선 호

2018. 09. 14

# 목 차

## I Gate 구조를 Transistor Level로 그리기

1. AND Gate	.....	3
2. OR Gate	.....	3
3. NOT Gate (Inverter)	.....	

## II AND / OR / NOT Logic

1. AND / OR / NOT Logic의 특성	.....	4
-----------------------------	-------	---

## III Fan-Out

1. Fan-Out	.....	4
------------	-------	---

## IV 전파 지연

1. 전파 지연	.....	4
----------	-------	---

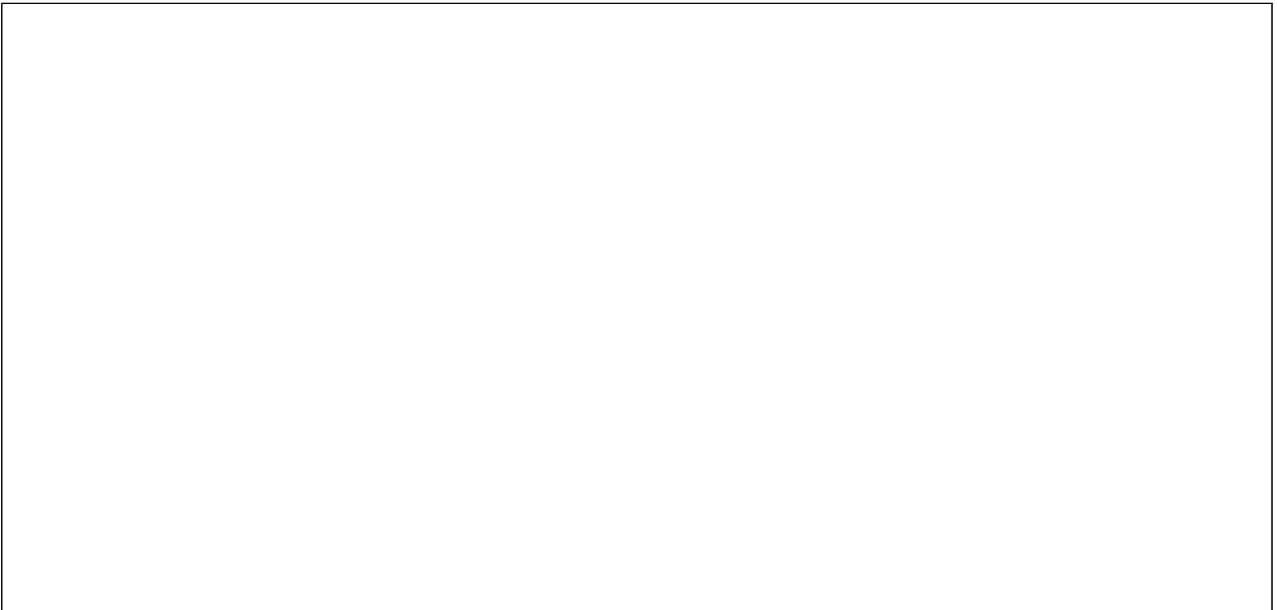
## V Task와 Function

1. Verilog의 Task와 Function	.....	4
----------------------------	-------	---

## I Gate 구조를 Transistor Level로 그리기

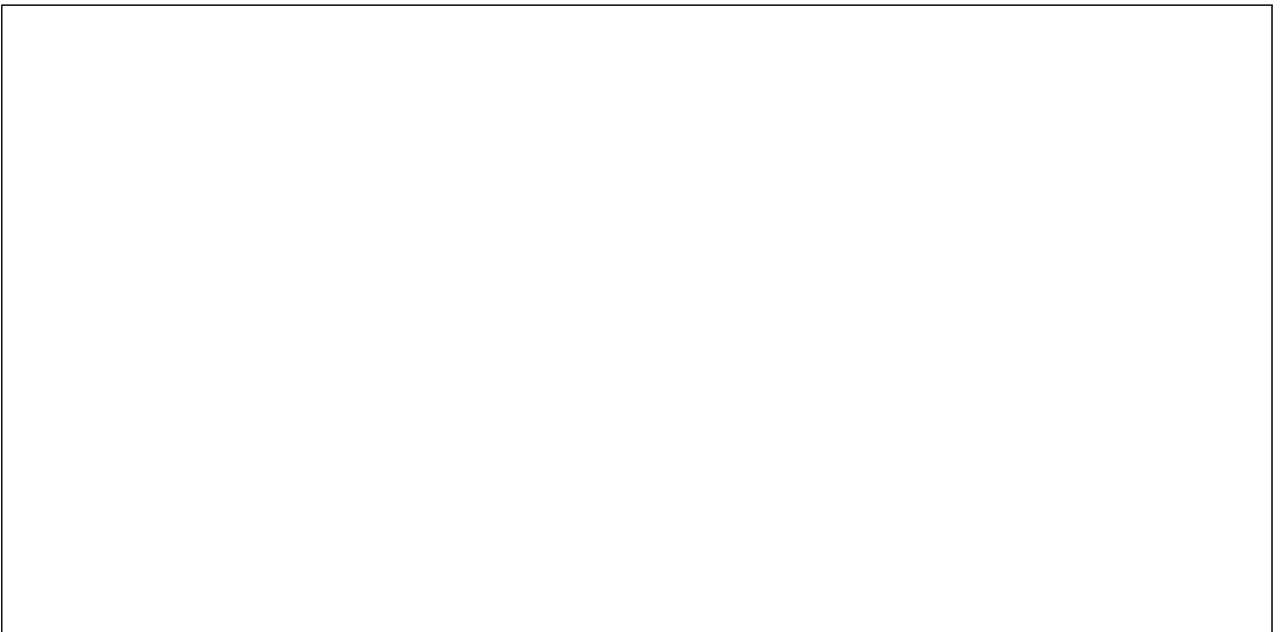
AND, OR, NOT Gate의 구조를 항목별로 나눠서 Transistor-Level로 그렸다.

### 1. AND Gate



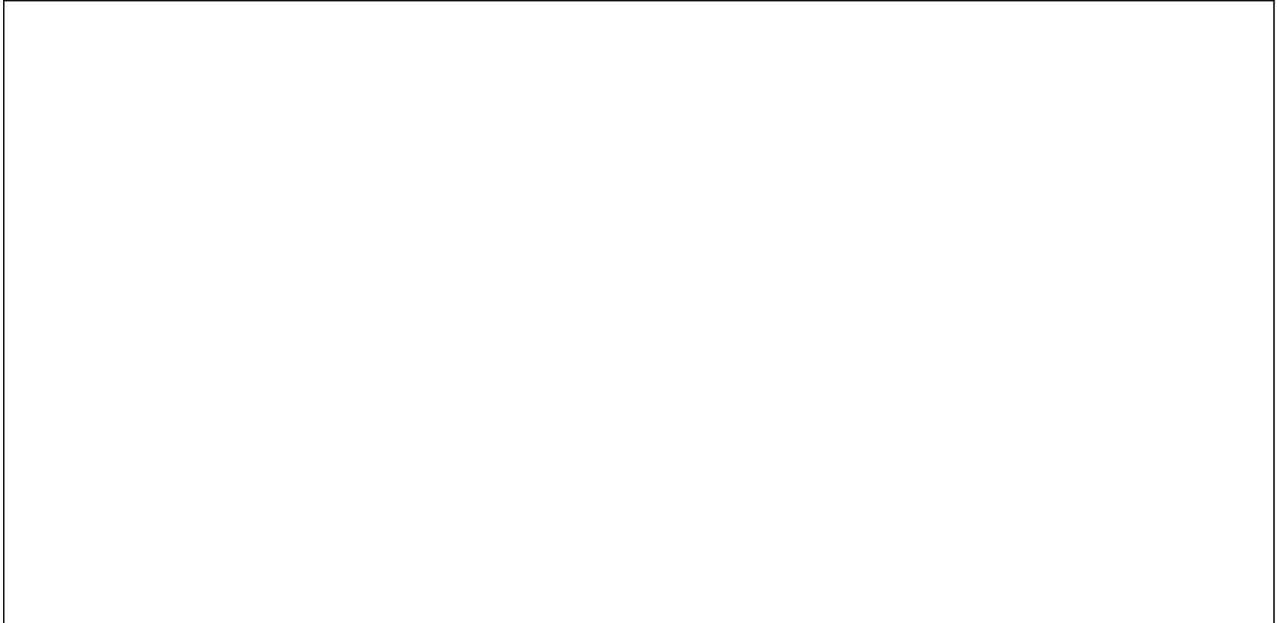
[그림 I -1]

### 2. OR Gate



[그림 I -2]

### 3. NOT Gate (Inverter)



[그림 I-3]

## II AND / OR / NOT Logic

### 1. AND / OR / NOT Logic의 특성

AND Logic은 input 값들이 모두 1일 때 output으로 1을 출력하며, 그렇지 않을 때에는 0을 출력하는 논리이다. 논리식으로 표현하면  $A \cdot B$ 이고 논리 회로 기호로 표현하면 [그림 II-1]과 같으며, Functional Expression으로 나타내면  $F = xy$ 이다. 또한 Transistor-Level로 표현하면 [그림 I-1]이고, 이 때 사용되는 트랜지스터 개수는 6개이며 전파 지연(Gate Delay) 시간은 약 2.4 나노초다. 진리표로 나타내면 [표 II-1]과 같다.

OR Logic은 input 값들 중에 하나라도 1이 있을 때 output으로 1을 출력하며, 그렇지 않고 input 값들이 모두 0일 경우에는 0을 출력하는 논리이다. 논리식으로 표현하면  $A+B$ 이고 논리 회로 기호로 표현하면 [그림 II-2]와 같으며, Functional Expression으로 나타내면  $F = x+y$ 이다. 또한 Transistor-Level로 표현하면 [그림 I-2]이고, 이 때 사용되는 트랜지스터 개수는 6개이며 전파 지연 시간은 약 2.4 나노초다. 진리표로 나타내면 [표 II-2]와 같다.

NOT Logic은 input 값을 부정하여 출력하는 논리이다. 예를 들어, input이 1이면 output으로 0을, 반대로 input이 0이면 output으로 1을 출력한다. 논리식으로 표현하면  $\sim A$  또는  $\overline{A}$ 이고 논리 회로 기호로 표현하면 [그림 II-3]과 같으며, Functional Expression으로 나타내면  $F = x'$ 이다. 또한 Transistor-Level로 표현하면 [그림 I-3]이고, 이 때 사용되는 트랜지스터 개수는 2개이며 전파 지연 시간은 약 1 나노초다. 진리표로 나타내면 [표 II-3]과 같다.

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

[표 II-1]

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

[표 II-2]

x	x'
0	0
0	1
1	0
1	1

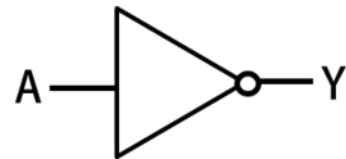
[표 II-3]



[그림 II-1]



[그림 II-2]



[그림 II-3]

### III Fan-Out

#### 1. Fan-Out

대개 디지털 회로에서 널리 쓰이는 TTL(Transistor-Transistor Logic, 트랜지스터와 트랜지스터를 조합한 것이며, 입력에서 신호가 들어오면 어떤 조건에 따라 출력에서 신호가 나오도록 만들어진 논리회로)과 CMOS(Complementary Metal Oxide Semiconductor, 금속 산화막 반도체로서 전계효과 트랜지스터를 사용하는 반도체)와 같은 표준 논리소자들은 1개의 출력 신호에 접속할 수 있는 입력신호의 수에 제한이 있는데 이를 팬 아웃(Fan-Out)이라고 한다. 팬 아웃을 만든 이유는 각 소자의 출력단

에 최대로 흐를 수 있는 전류에 제한이 따르기 때문이다.

TTL은 형식에 따라 다양한 종류가 있지만 같은 종류의 형식은 동일한 팬 아웃 값을 갖는다. 다만 다수의 입력 신호를 분배할 목적으로 제작되어서 같은 형식임에도 불구하고 높은 팬 아웃 값을 가지는 회로 소자도 존재한다. CMOS형 논리 회로는 입력 단자에서 소비되는 전류는 적지만, 많은 입력 단자를 연결하면 각 단자와 배선에서 전기 용량이 커져서 출력 단자에서 용량성 저항이 커지게 된다. 이를 가리켜 팬 아웃이라고 부른다.

만일 팬 아웃을 초과하여 부하를 접속하게 되면 출력단의 회로가 손상될 수 있거나 출력단의 전압강하 때문에 출력 전압이 규정치로 유지되지 못하여 그 다음의 입력단에 입력되는 신호의 논리 상태를 보장할 수 없게 된다. 그래서 팬 아웃을 초과하여 부하를 접속하는 경우 중간에 버퍼를 사용하기도 한다.

## IV 전파 지연

### 1. 전파 지연

디지털회로에서 전파 지연(gate delay)은 논리 회로에 안정되고 유효한 신호가 입력되는 순간부터 논리 회로가 안정되고 유효한 신호를 출력할 때까지 걸리는 시간이다. 경우에 따라서 전달 지연은 입력 신호가 변경되었을 때 출력 신호가 최종 출력 수준의 50%에 도달하는 데 걸리는 시간을 나타내기도 한다. 전파 지연을 감소시키면 데이터를 빠르게 처리하고 전반적인 시스템 성능을 향상시킬 수 있다. 그래서 더 빠른 gate를 사용함으로써 전파 지연을 줄이려는 노력이 요구된다. 전파 지연 시간을 구하는 방법을 예로 들면 아래와 같다.



## V Task와 Function

### 1. Verilog의 Task와 Function

Verilog는 태스크(Task)와 함수(Function)를 이용해 커다란 행위 수준 설계를 작은 단위로 나눈다. 태스크와 함수는 마치 C언어의 함수처럼 서로 비슷한 것으로 보이지만 이 둘 사이에는 확연한 차이가 존재한다. 이를 [표 V-1]에 정리하였다.

태스크(Task)	함수(Function)
다른 태스크나 함수를 사용할 수 있다.	다른 함수를 사용할 수 있지만 다른 태스크는 사용 할 수 없다.
0이 아닌 시뮬레이션 시간에 수행 가능하다.	항상 시뮬레이션 시간 0에 수행된다.
지연, 사건, 또는 타이밍 제어 문장을 포함할 수 있다.	어떤 지연, 사건, 또는 타이밍 제어 문장을 포함할 수 없다.
input, output, 또는 inout을 하나도 가지지 않거나 다수를 가질 수 있다.	적어도 하나 이상의 input 인수를 가져야만 한다.
값을 반환할 수 없지만, output과 input을 통해서 여러 개의 값을 전달할 수 있다.	항상 하나의 값만 반환하며, output 또는 input 인수를 가질 수 없다.

[표 V-1]

그러나 꼭 차이점만 존재하는 것은 아니다. 태스크와 함수는 와이어를 가질 수 없으며, 행위 수준 문장만 포함한다는 공통점이 존재한다.

태스크는 Delay, timing, event 제어 구조가 있거나, 혹은 output이 없거나 둘 이상의 output이 있는 경우, 또는 input이 없는 경우에 사용해야 한다. 태스크를 선언하는 키워드는 task와 endtask이다. 반면에 함수는 event 제어 구조가 없거나, 혹은 단 한 개의 값을 반환하는 경우, 또는 적어도 한 개의 input이 있는 경우에 사용해야 한다. 함수를 선언하는 키워드는 function과 endfunction이다.