# 결 과 보 고 서

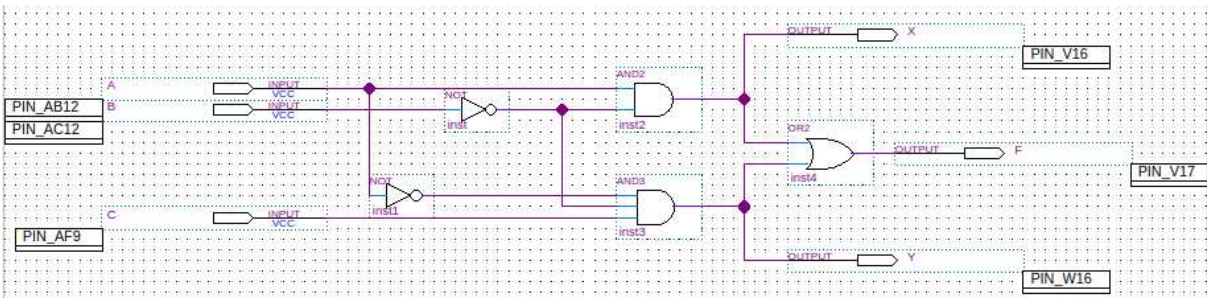
## 실험 3. 부울대수의 간소화(1)

**분반 : 타**
**성명 : 김동현**
**학번 : 20160428**
**실험일: 17/03/31**

# 1. 실험 결과

## 과정 1

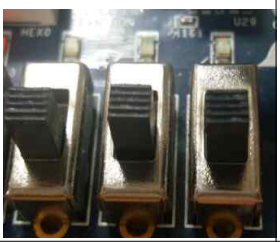
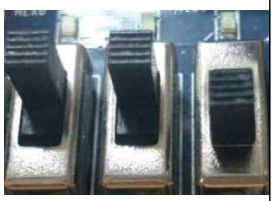
**F = AB' +A'B'C에 대해 Schematic을 만들고 Modelsim을 이용해서 결과 값을 작성하시오**




**[그림 3] block diagram**

| Input | | | Output | | |
|---|---|---|---|---|---|
| A | B | C | X | Y | F |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

**[표 4] 결과 값**

| | | | |
|---|---|---|---|
| A : 0 / B : 0 / C : 0<br>X : 0 / Y : 0 / Z : 0 | A : 0 / B : 0 / C : 1<br>X : 0 / Y : 1 / Z : 1 | A : 0 / B : 1 / C : 0<br>X : 0 / Y : 0 / Z : 0 | A : 0 / B : 1 / C : 1<br>X : 0 / Y : 0 / Z : 0 |
|  |  |  |  |
| A : 1 / B : 0 / C : 0<br>X : 1 / Y : 0 / Z : 1 | A : 1 / B : 0 / C : 1<br>X : 1 / Y : 0 / Z : 1 | A : 1 / B : 1 / C : 0<br>X : 0 / Y : 0 / Z : 0 | A : 1 / B : 1 / C : 1<br>X : 0 / Y : 0 / Z : 0 |
|  |  |  |  |

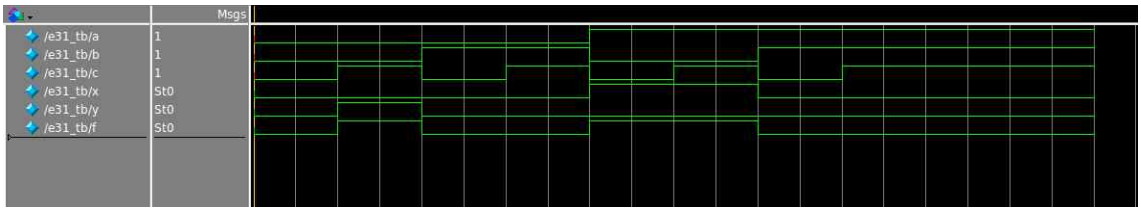
방법 1

```
`timescale 1ns/1ps
module e31_tb;
reg a,b,c;
wire x,y,f;

e31 uut(.A(a),.B(b),.C(c),.X(x),.Y(y),.F(f));

//1
initial begin
a=0; b=0; c=0;

#10   a=0; b=0; c=1;
#10   a=0; b=1; c=0;
#10   a=0; b=1; c=1;
#10   a=1; b=0; c=0;
#10   a=1; b=0; c=1;
#10   a=1; b=1; c=0;
#10   a=1; b=1; c=1;

end
```

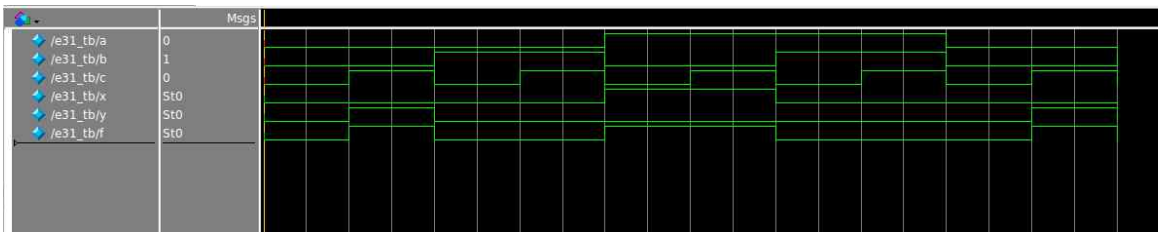
방법 2

```
//2
/*
always begin
#10 c = ~c;
end

always begin
#20 b = ~b;
end

always begin
#40 a = ~a;
end
*/
endmodule
```

**Verilog code**




**방법 1**

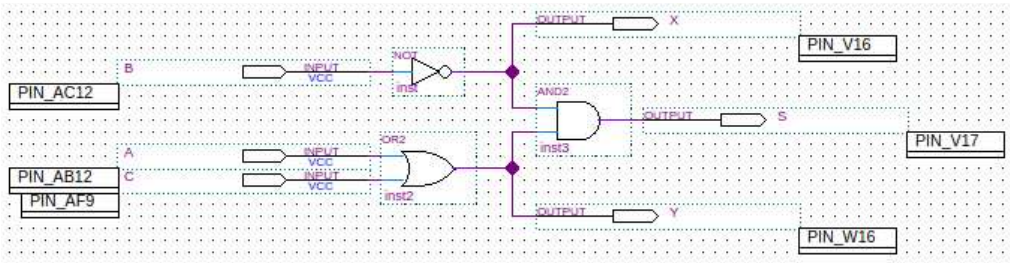



**방법 2**

**방법1은 신호가 입력된 것으로 한 번만 바뀐 후 변하지 않지만, 방법2는 일정한 주기로 신호가 계속해서 바뀐다.**

**과정 2**

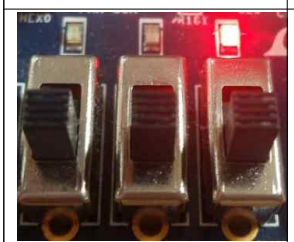
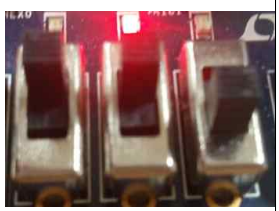
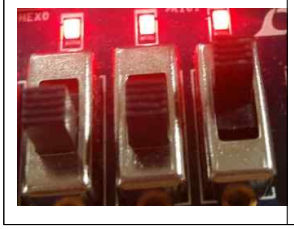
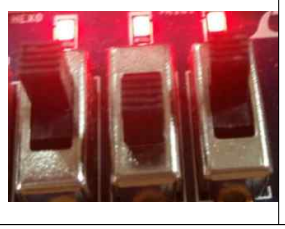
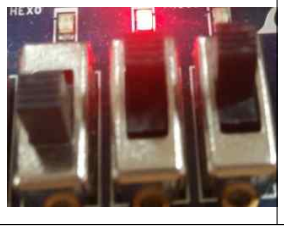
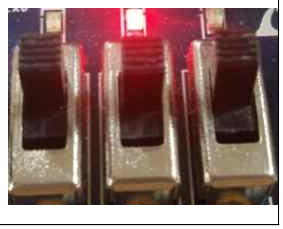
**S = B'( A + C )에 대해 Schematic을 만들고 Modelsim을 이용해서 결과 값을 작성하시오**




**[그림 4] block diagram**

| Input | | | Output | | |
|---|---|---|---|---|---|
| A | B | C | X | Y | S |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |

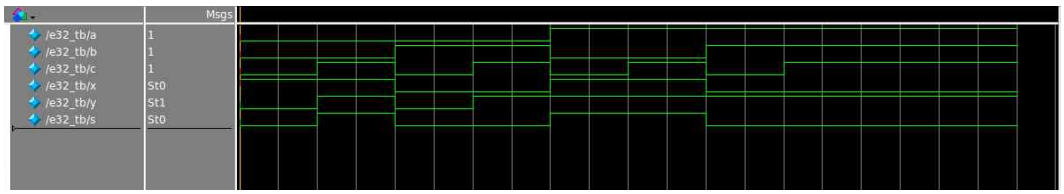
**[표 6] 결과 값**

| A : 0 / B : 0 / C : 0<br>X : 1 / Y : 0 / Z : 0 | A : 0 / B : 0 / C : 1<br>X : 1 / Y : 1 / Z : 1 | A : 0 / B : 1 / C : 0<br>X : 0 / Y : 0 / Z : 0 | A : 0 / B : 1 / C : 1<br>X : 0 / Y : 1 / Z : 0 |
|---|---|---|---|
|  |  |  |  |
| A : 1 / B : 0 / C : 0<br>X : 1 / Y : 1 / Z : 1 | A : 1 / B : 0 / C : 1<br>X : 1 / Y : 1 / Z : 1 | A : 1 / B : 1 / C : 0<br>X : 0 / Y : 1 / Z : 0 | A : 1 / B : 1 / C : 1<br>X : 0 / Y : 1 / Z : 0 |
|  |  |  |  |

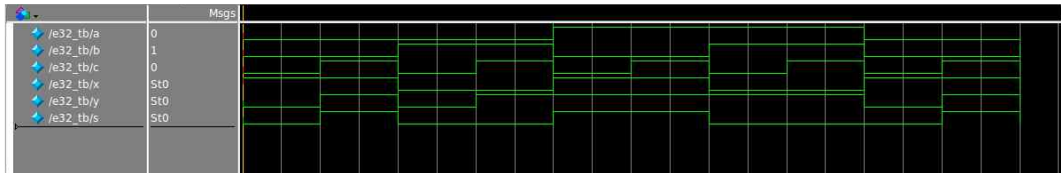
방법 1

방법 2

```
`timescale 1ns/1ps
module e32_tb;
reg a,b,c;
wire x,y,s;

e32 uut(.A(a),.B(b),.C(c),.X(x),.Y(y),.S(s));

//1
initial begin
a=0; b=0; c=0;

#10   a=0; b=0; c=1;
#10   a=0; b=1; c=0;
#10   a=0; b=1; c=1;
#10   a=1; b=0; c=0;
#10   a=1; b=0; c=1;
#10   a=1; b=1; c=0;
#10   a=1; b=1; c=1;

end

//2
/*
always begin
#10 c = ~c;
end

always begin
#20 b = ~b;
end

always begin
#40 a = ~a;
end
*/
endmodule
```




**방법 1**




**방법 2**

## 2. 실험 고찰

1) 입력 a, b와 c신호의 Simulation 입력 값이다. 표와 같이 Simulation을 진행 하고 싶다면 Test bench의 나머지 부분을 작성하시오 ( 각 신호의 Delay는 50ns이다. )

방법 1

```
initial begin
a=0; b=0; c=0;

    #50 a=0; b=0; c=1;
    #50 a=0; b=1; c=0;
    #50 a=0; b=1; c=1;
    #50 a=1; b=0; c=0;
    #50 a=1; b=0; c=1;
    #50 a=1; b=1; c=0;
    #50 a=1; b=1; c=1;
```

방법 2

```
initial begin
a=0; b=0; c=0;
end

always begin
 #50 a=~a;
end

always begin
 #100 b=~b;
end

always begin
 #200 c=~c;
end
```

## 3. 결론

논리게이트를 작성하고 Test bench 코드 작성을 통해 시뮬레이터를 사용할 때 결과 값을 확인할 수 있었다.

Test bench를 작성할 때는 하나의 initial문에 모든 신호를 넣거나 하나의 initial문에 하나의 신호만 넣는 방법을 예비보고서를 작성할 때 확인할 수 있었다. 실제 실험에서는 always문을 사용하여 좀더  간결하게 소스작성을 해보았다.