things lab
Biomedical Engineeringe at Dankook University

02. Advanced Logic Gate

Content

1. Full Adder Gate
2. Half Subtractor Gate

Full Adder Gate

Full Adder Gate

1. What is Full Adder Gate?

- Full Adder는 다음으로 이루어져 있다.

  Input

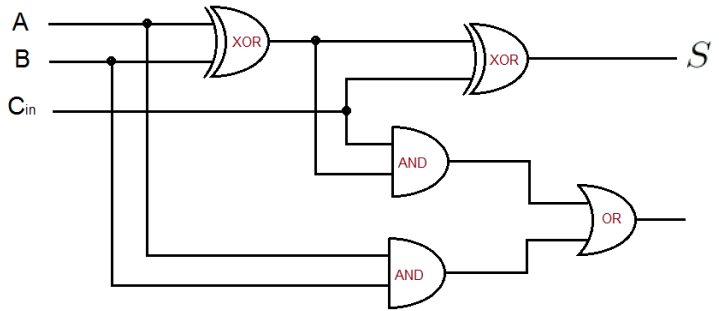
  - 1 bit carry-in x 1
  - 1 bit binary x 2

  Output

  - 1 bit sum x 1
  - 1 bit carry-out x 1

- 예를 들어, A와 B가 1bit binary 값이고, Cin이 선행 bit로부터 반입된 경우, Sum 과 Cout은 다음과 같다.

$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = (A \wedge B) \vee (B \wedge C_{in}) \vee (C_{in} \wedge A)$$



Design of Full Adder Gate

Full Adder Gate

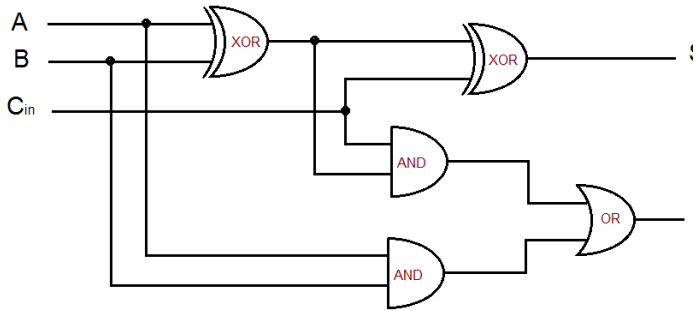
2. How to code in Verilog

- full_adder.v

```
1   module full_adder (input In1, In2, Cin, output Sum, Cout);
2
3   assign Sum  = (In1 ^ In2) ^ Cin;
4
5   assign Cout = (In1 & In2) | (In2 & Cin) | (Cin & In1);
6
7   endmodule
```

Line 1 : Module 이름, Input, Output 값 설정.
Line 3 : 출력될 Sum 정의
Line 5 : 출력될 Cout 정의
Line 7 : module 종료

+) 지난 시간에 배운 내용
- XOR => ^
- AND => &
- OR => |



Design of Full Adder Gate

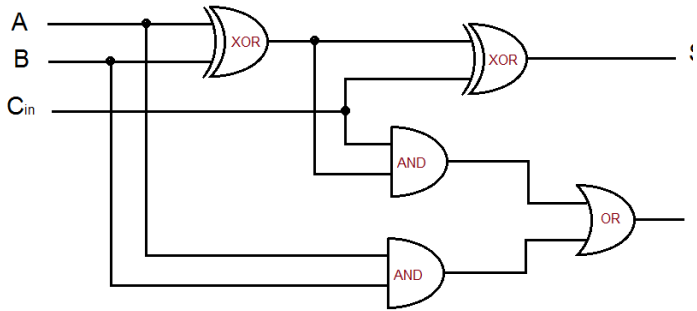
Full Adder Gate

2. How to code in Verilog

- full_adder_tb.v

```
1   module full_adder_tb;
2       reg Cin, In1, In2;
3       wire Count, sum;
4       full_adder uut ( .In1(In1), .In2(In2), .Cin(Cin), .Sum(Sum), .Cout(Cout));
5
6       initial begin
7           $dumpfile("full_adder.vcd");
8           $dumpvars(0, full_adder_tb);
9
10          Cin = 0; In1 = 0; In2 = 0;
11          #10 In1 = 1;
12          #10 In2 = 1;
13          #10 Cin = 1;
14          #10 $finish;
15      end
16
17  endmodule
```

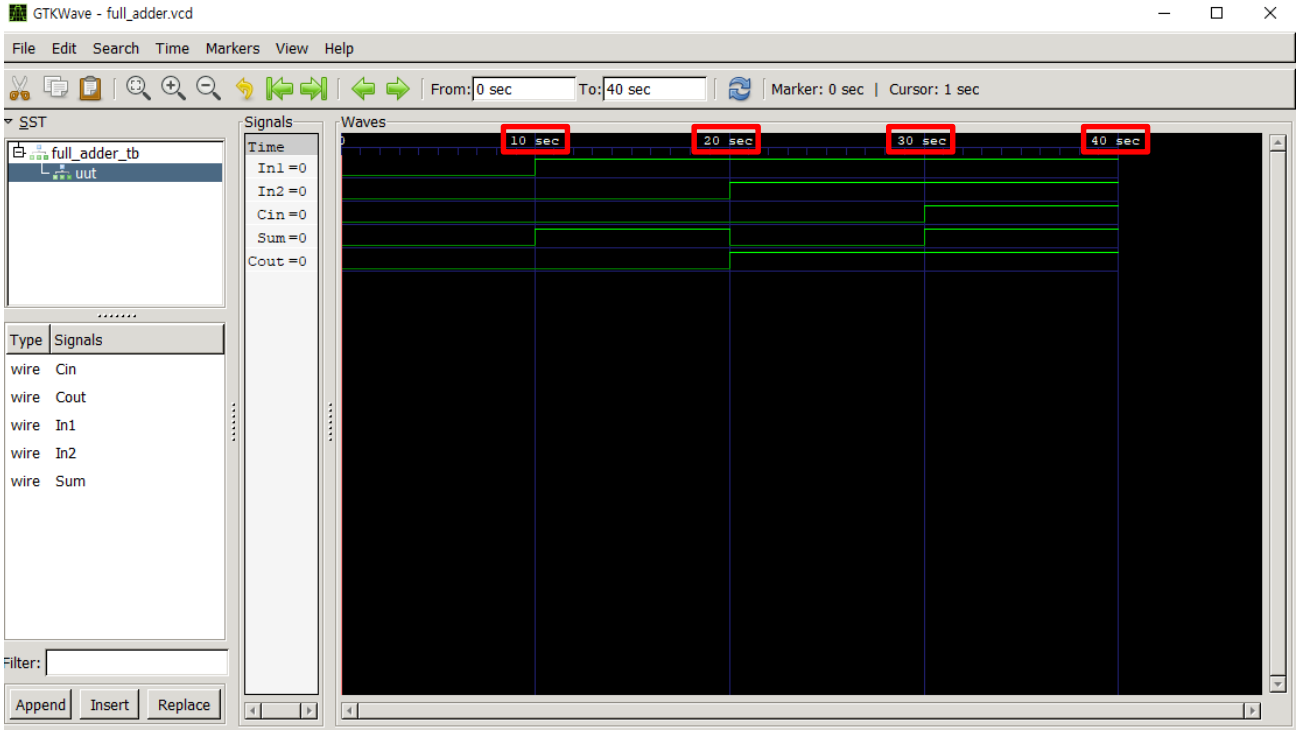
Line 1 : test_bench할 module 이름 설정
Line 2 : Input값은 reg로 설정
Line 3 : Output값은 wire로 설정
Line 4 : full_adder에 대한 Instantiate the Unit Under Test
Line 6 : time을 0으로 설정한다.
Line 7 : GTKWave를 위한 dumpfile name 설정
Line 8 : dumpvars
 - First parameter : '0' 은 2번째에 나올 모듈을 모두 dump 한다는 뜻
 - Second parameter : dump할 모듈 이름.
Line 10 : 초기값 설정
Line 11-13 : 10초 뒤 값을 1로 변경 (병렬로 실행됨을 방지 하기 위함)
Line 14 : 10초 뒤 종료



Design of Full Adder Gate

Full Adder Gate

3. Check the GTKWave



Half Subtractor

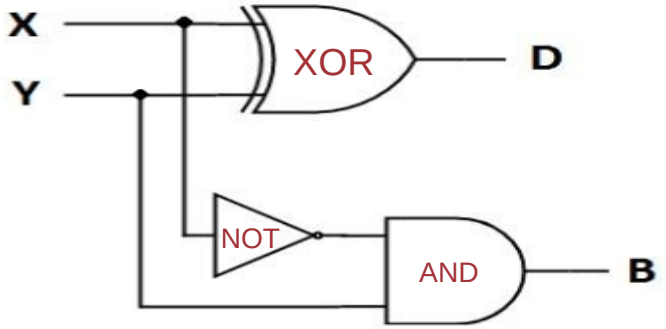
Half Subtractor

1. What is Half Subtractor Gate?

- Half Subtractor는 다음으로 이루어져 있다.

  Input
  - 1 bit binary x 2

  Output
  - 1 bit difference x 1
  - 1 bit borrow x 1



Design of Half Subtractor Gate

- 예를들어, X와 Y는1bit binary 값이고, difference를 D, borrow를 B라고 했을 때, Half Subtractor는 다음과 같다.

$$D = X \oplus Y$$

$$B = \neg X \wedge Y$$

Half Subtractor

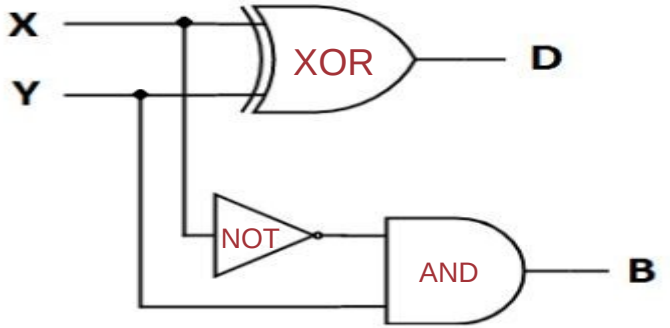
2. How to code in Verilog

- half_subtractor.v

```
1   module half_subtractor (input X, Y, output D, B);
2
3   assign D  = X ^ Y;
4
5   assign B = ~X & Y;
6
7   endmodule
```

Line 1 : Module 이름, Input, Output 값 설정.
Line 3 : 출력될 Difference 정의
Line 5 : 출력될 Borrow정의
Line 7 : module 종료

+) 지난 시간에 배운 내용
- XOR => ^
- AND => &
- NOT => ~



Design of Half Subtractor Gate

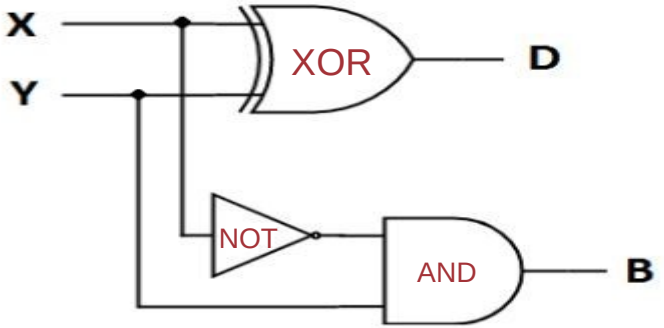
Half Subtractor

2. How to code in Verilog

- half_subtractor_tb.v

```
1  module half_subtractor_tb;
2      reg X, Y;
3      wire D, B;
4      half_subtractor uut ( .X(X), .Y(Y), .D(D), .B(B));
5
6      initial begin
7          $dumpfile("half_subtractor.vcd");
8          $dumpvars(0, half_subtractor_tb);
9
10         X = 0; Y = 0;
11         #10 X = 1;
12         #10 Y = 1;
13         #10 $finish;
14     end
15
16 endmodule
```

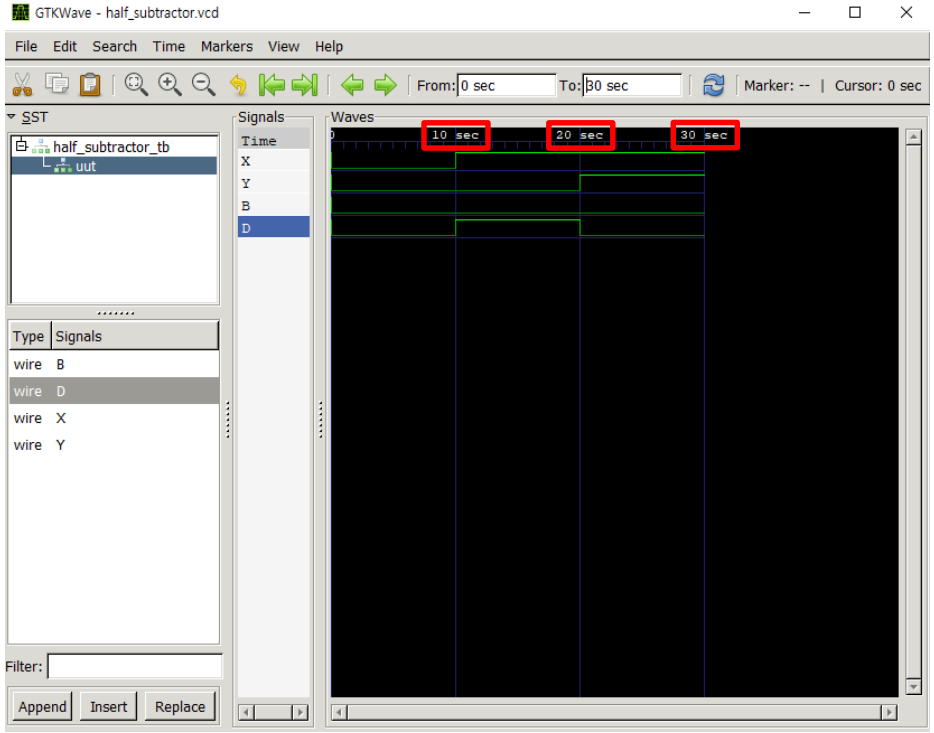
Line 1 : test_bench할 module 이름 설정
Line 2 : Input값은 reg로 설정
Line 3 : Output값은 wire로 설정
Line 4 : half_subtractor 에 대한 Instantiate the Unit Under Test
Line 6 : time을 0으로 설정한다.
Line 7 : GTKWave를 위한 dumpfile name 설정
Line 8 : dumpvars
 - First parameter : '0' 은 2번째에 나올 모듈을 모두 dump 한다는 뜻
 - Second parameter : dump할 모듈 이름.
Line 10 : 초기값 설정
Line 11-12 : 10초 뒤 값을 1로 변경 (병렬로 실행됨을 방지 하기 위함)
Line 13 : 10초 뒤 종료



Design of Half Subtractor Gate

Half Subtractor

3. Check the GTKWave



Assignment 02

- Half Adder Gate
- Full Subtractor Gate

위 두 Gate를 찾아보고 주석을 포함하여 아래와 같은 형식으로 3개 파일을 제출하라.

제출파일
1. [Gate이름].v
2. [Gate이름]_tb.v
3. GTKWave 결과 스크린샷