하드웨어 시스템 및 설계

LAB-03

이다운

2016-13919

my_add

```
module my_add #(
    parameter BITWIDTH = 32
)
(
    input [BITWIDTH-1:0] ain,
    input [BITWIDTH-1:0] bin,
    output[BITWIDTH-1:0] dout,
    output overflow
 );


assign {overflow, dout} = ain + bin;

endmodule
```

my_add 모듈은 add 연산을 하는 모듈로 input으로 ain, bin을 받고 output으로 연산 결과값인 dout과 overflow 여부를 나타내는 overflow가 있다.

assign {overflow, dout} = ain+ bin 으로 구현하였는데 이렇게 하면 dout에는 +연산 결과 값이 나오고 만약 이 결과값이 BITWIDTH 범위를 초과한다면 overflow에 1의 값이 아니면 0의 값을 가지게 한다.

my_mul

```
23  module my_mul #(
24      parameter BITWIDTH =32
25  )
26  (
27      input [BITWIDTH-1:0] ain,
28      input [BITWIDTH-1:0] bin,
29      output [2*BITWIDTH-1:0] dout
30  );
31  assign dout = ain * bin;
32
33  endmodule
34
```

my_mul 모듈은 곱셈 연산을 하는 모듈로 input으로 ain, bin을 가지고 output으로 dout을 가진다.

*연산자를 사용하여 ain과 bin 곱셈 값을 dout이 가지도록 하였다.

my_fusemul

```
module my_fusemul #(
    parameter BITWIDTH =32
)
(
    input [BITWIDTH-1:0] ain,
    input [BITWIDTH-1:0] bin,
    input en,
    input clk,
    output reg [2*BITWIDTH-1:0] dout

);
always@(posedge clk) begin
  if(en ==0) begin
    dout =0;

  end
  else begin
  dout = dout + ain*bin;
  end
end


endmodule
```

my_fuse 모듈은 input으로 ain,bin,en,clk output으로 dout을 가지는데 en이 0일 일 때 dout의 값을 0으로 en이 1일때는 dout 값은 ain과bin 곱셈값과 그 전의 dout 값의 합이 된다. (dout = dout+ ain x bin)

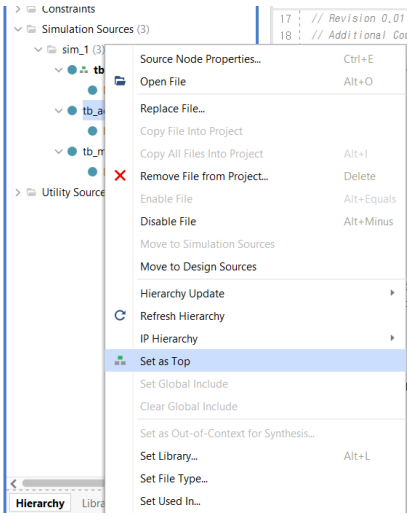
en 신호와 곱셈 연산은 모두 클럭 clk와 동기화 시켜주었고 clk이 양의 값으로 변할 때 즉 posedge할 때 연산을 실시한다.

posedge clk 을 탐지하는 always 블록 안에 만약 en이 0이면 dout을 0으로 초기화 하고 아니면 dout = dout +ain x bin 연산을 하도록 구현하였다.

Discussion & Others

주어진 testbench에서는 input 값을 범위를 2**31이하로 하여 my_add 연산 수행 시 overflow가 발생하지 않는다. 만약 overflow을 발생시키게 하려면 2**32 -1 로 바꿔서 하면 된다.

원하는 모듈을 시뮬레이션을 돌리고 싶으면 그 전에 그 모듈을 top module로 설정해줘야 한다. (아래 사진 참고)



result의 waveform에 있는 모든 숫자는 unsigned int를 10진수로 나타낸 수이다.