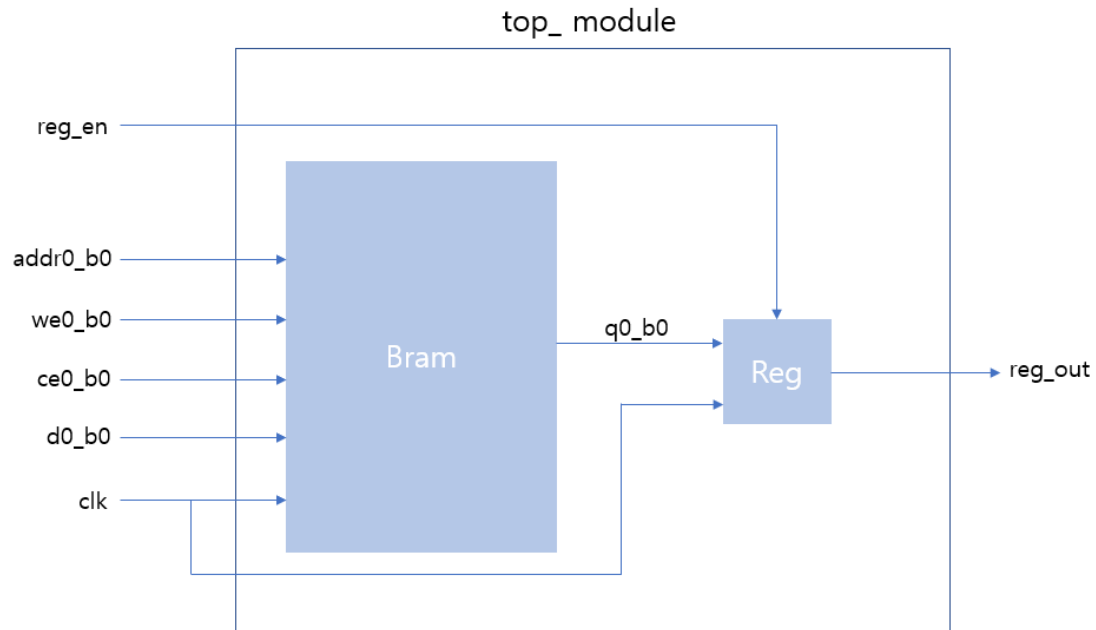


BRAM PRACTICE

- 실습목표 : BRAM의 동작을 이해하고, 다른 module과 data를 주고 받는 동작을 검증해본다.
- 제출기한 : 2022-08-04 23:59
- 제출방법 : 스크린샷 파일과 testbench 파일을 압축해서 카톡방에 올려주세요.

MODULE SPEC



- 이번 실습에서 simulation 할 대상은 왼쪽 그림의 `top_module`이다.
- `Bram`은 실습시간에 다뤘던 내용과 동일하며, 추가적인 부분은 `Reg` 이다.
- `Reg`는 하나의 register로써, clock의 Rising edge에서 `reg_en` 신호가 1'b1로 인가되어 있다면, input data(그림에서는 `q0_b0`)을 저장한다.
- (해당 `top_module`은 이미 만들어져 있으며 `top_module.v`라는 파일로 제공된다.)

TEST PLAN

- 검증할 동작은 다음 순서와 같다.

1) top_module 내부의 BRAM의 1번 주소지에 16'b0000_0000_0000_0001을 write한다.

2) top_module 내부의 BRAM의 1번 주소지에 저장되어 있는 data를 read하여 그 read data를 Reg에 저장한다.

- read data에 해당하는 port인 q0_b0는 top_module.v 내부안에 이미 선언되어 Reg의 input data port로 연결되어 있으므로, testbench를 작성할 시 고려해야 하는 port는 오직 clk, BRAM PORT(4개), reg_en이다.
- Reg의 output data인 reg_out port를 살펴보고 bram에 write된 data가 적절히 read되어 Reg에 저장되는지 검증한다.

TESTBENCH

- Testbench 파일또한 tb_top_module.v 파일로 제공되며, 내부에 주석으로 처리되어 있는 공간을 마저 채워 완성시키면 된다.
- Vivado를 통해 top_module을 검증하고, simulation 결과를 스크린샷하여 tb_top_module.v 파일과 함께 압축하여 제출한다.

```
initial begin
    clk      = 0;
    addr0_b0 = {(ADDR_WIDTH){1'b0}};
    ce0_b0   = 0;
    we0_b0   = 0;
    d0_b0    = {(DATA_WIDTH){1'b0}};
    reg_en   = 0;
end
Park Sang Hyun, 11시간 전 • update
initial begin
    /* Fill the code here
    //
    //
    //
    */
end
```