



컴퓨터공학 기초 실험2

Lab #10

Memory & Bus

32×32 Memory

PRACTICE I

Project Properties

➤ New Project Wizard

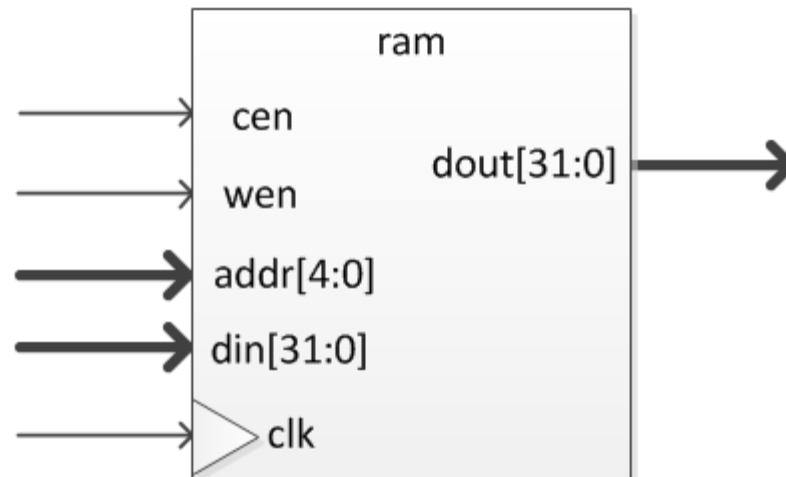
- ✓ Project name : ram
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

➤ Verilog file

- ✓ Add files:
- ✓ New files: ram.v

Functional Description

- Address에 기반하여 데이터를 저장하는 hardware
- Features
 - ✓ Address 의 bandwidth는 5bit이다.
 - ✓ Data의 bandwidth는 32bit이다.
 - ✓ RAM의 내부에 32개의 데이터를 address에 기반하여 저장한다.



Pin Description

Direction	Port name	Description
Input	clk	Clock
	cen	Chip enable
	wen	Write enable
	addr[4:0]	Address
	din[31:0]	Data in
Output	dout[31:0]	Data out

- cen과 wen이 모두 1이면 address가 가리키는 memory에 din을 write한다. 이 때 dout은 0을 출력한다.
- cen이 1이고, wen이 0이면, address가 가리키는 memory의 값을 dout에 write한다.
- cen이 0이면, dout은 0이 된다.

Memory Declaration & Coding

➤ VerilogHDL에서의 메모리 선언

✓ `reg[wordsize-1:0] variable_name[0:storage-1];`

➤ 메모리 초기화 필수

✓ `initial` 구문 내에서 작성

✓ `for`문을 이용하여 초기화

```
module ram(clk, cen, wen, addr, din, dout);
    input          clk;
    input          cen, wen;
    input [4:0]    addr;
    input [31:0]   din;
    output reg [31:0] dout;

    reg [31:0] mem [0:31];
    integer i;

    initial begin
        //memory initialization
        //Fill the initial syntax
        end

    always @(posedge clk)
    begin
        // read/write performance
        end
endmodule
```

32-bit Bus with 8-bit address

PRACTICE II

Project Properties

➤ New Project Wizard

- ✓ Project name : bus
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

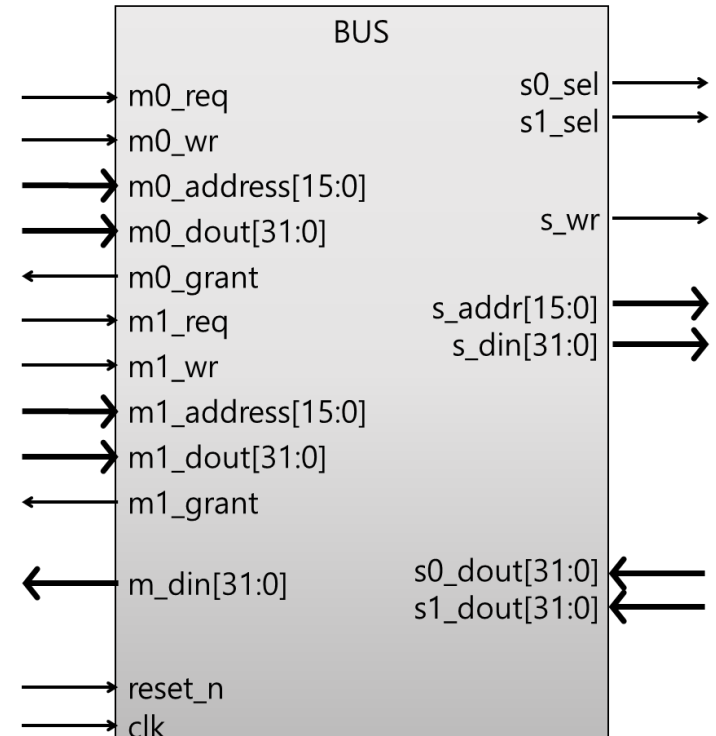
➤ Verilog file (예시)

- ✓ Add files: mux2_32bit.v, mux3_32bit, mux2_8bit.v, mux2.v
- ✓ New files: bus.v, bus_arbit.v, bus_addr.v

Functional Description

- Bus는 여러 component들 간에 data를 전송(transfer)할 수 있도록 연결해주는 component이다.
 - ✓ Bus는 새로운 component들을 추가하기가 쉬우며, 가격이 저렴한 특징을 가지고 있다.
- Feature
 - ✓ 2개의 master와 2개의 slave를 가지고 있다.
 - ✓ Address의 bandwidth는 8 bits이다.
 - ✓ Data의 bandwidth는 32bits이다.
 - ✓ 각 slave가 가지는 주소 범위는 다음과 같다.

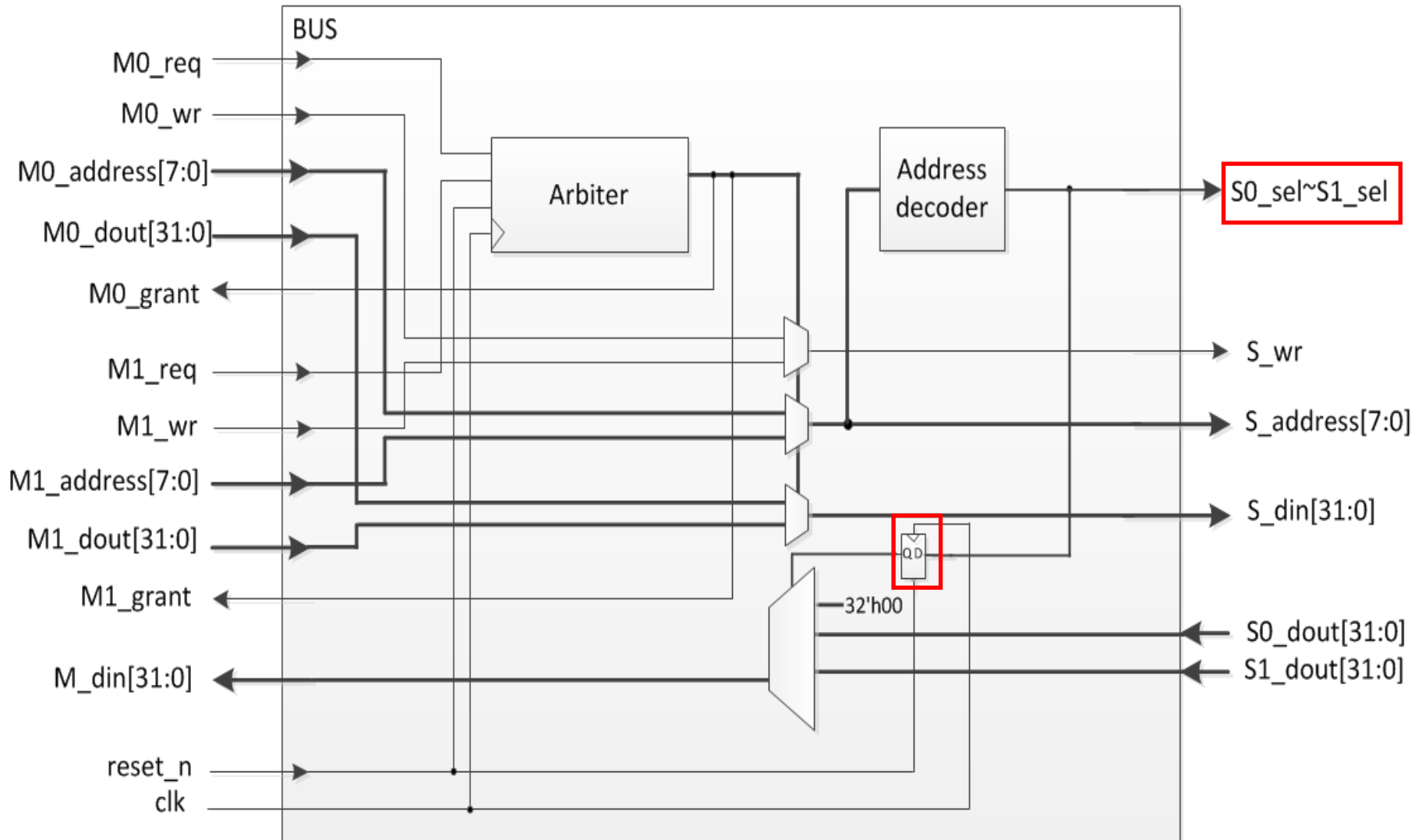
Memory map	
Slave 0	0x00 ~ 0x1F
Slave 1	0x20 ~ 0x3F



Pin Description

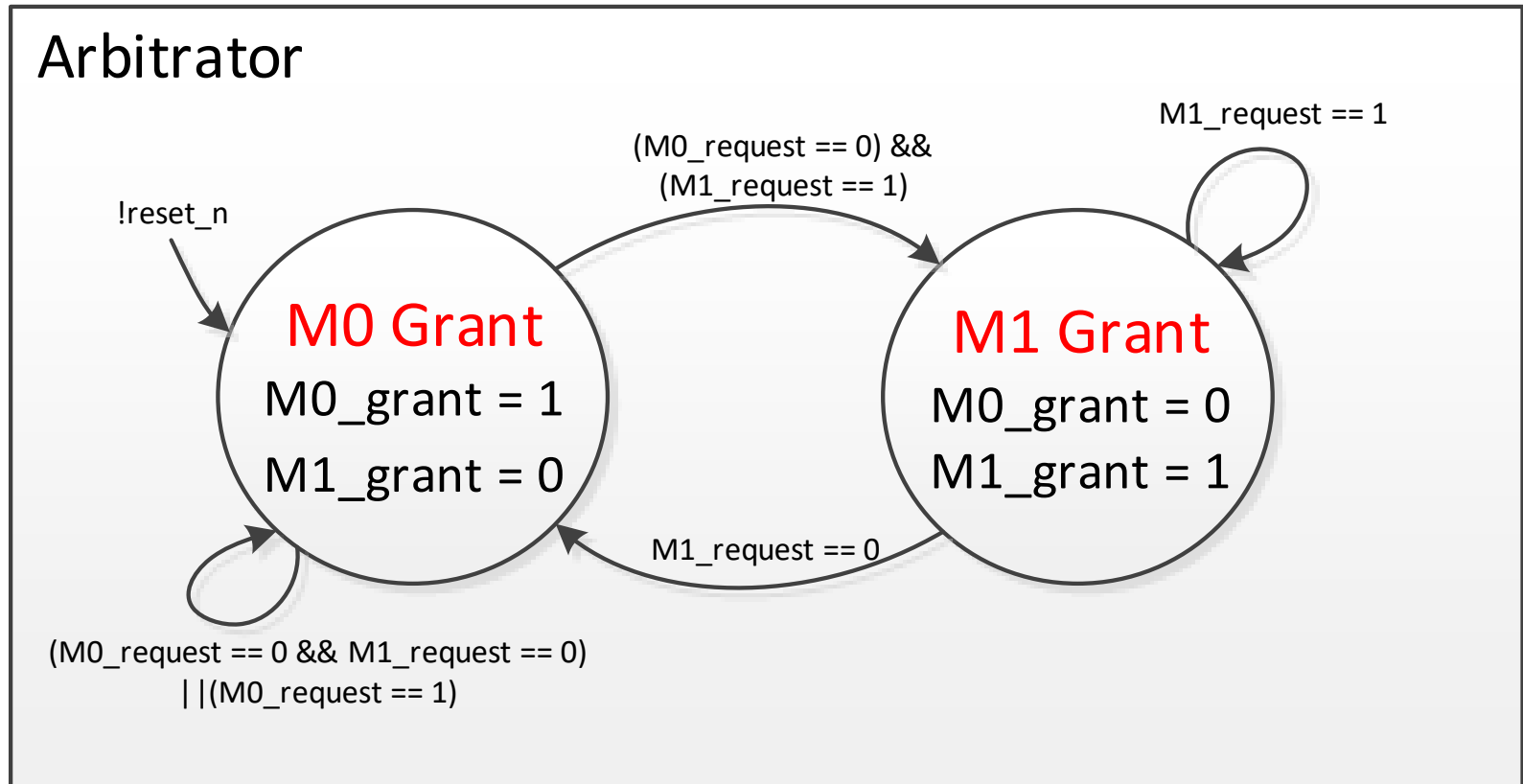
Direction	Port name	Description
Input	clk	Clock
	reset_n	Active low reset
	m0_req	Master 0 request
	m0_wr	Master 0 write/read
	m0_address[7:0]	Master 0 address
	m0_dout[31:0]	Master 0 data output
	m1_req	Master 1 request
	m1_wr	Master 1 write/read
	m1_address[7:0]	Master 1 address
	m1_dout[31:0]	Master 1 data out
	s0_dout[31:0]	Slave 0 data out
	s1_dout[31:0]	Slave 1 data out
	m0_grant	Master 0 grant
	m1_grant	Master 1 grant
Output	m_din[31:0]	Master data input
	s0_sel	Slave 0 select
	s1_sel	Slave 1 select
	s_address[7:0]	Slave address
	s_wr	Slave write/read
	s_din[31:0]	Slave data input

Design



Arbiter

➤ Finite State Diagram



Address Decoder

Address decoder

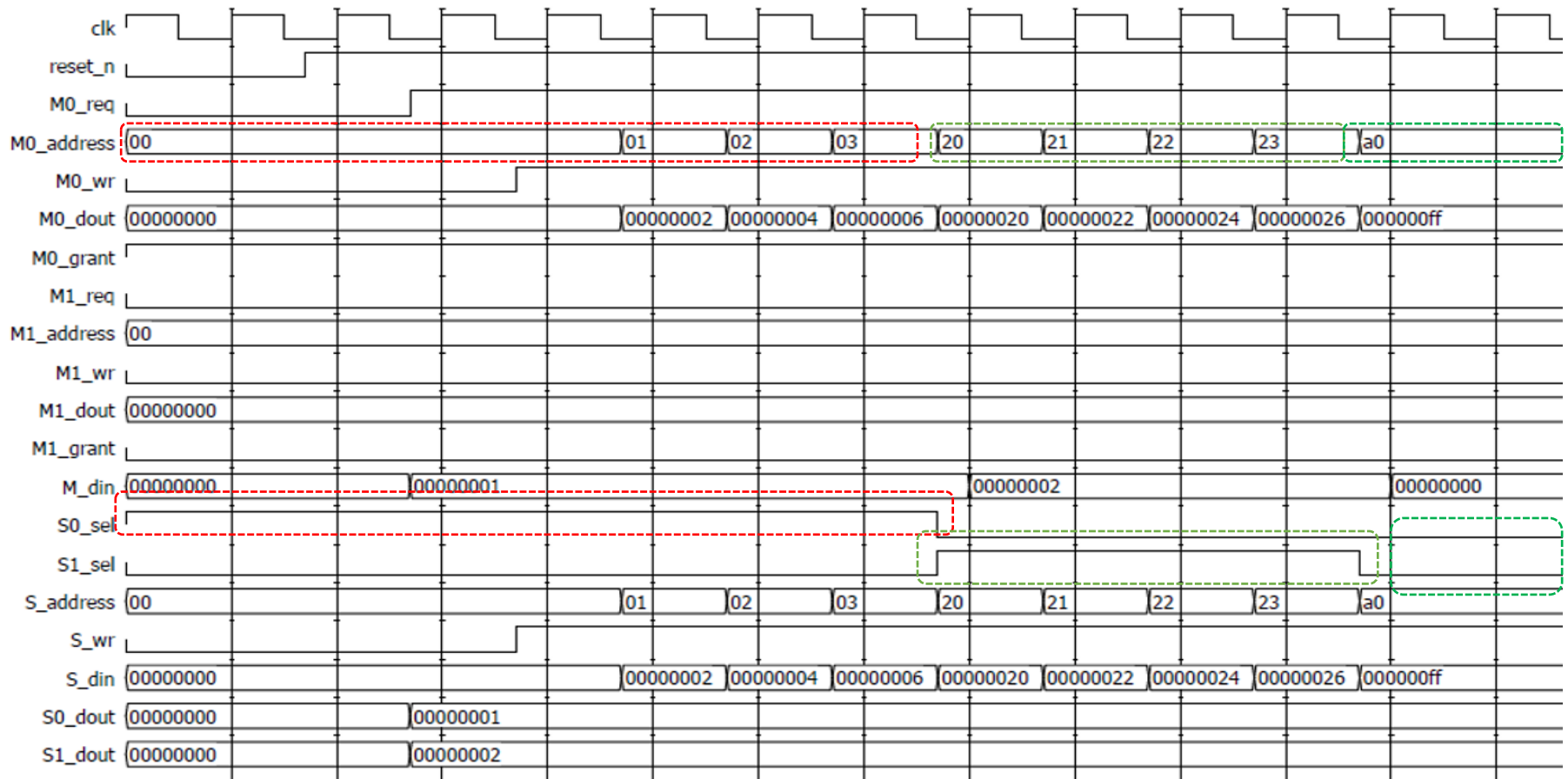
```
if((address >= 0x00) && (address < 0x20))  
    {s0_sel, s1_sel} = 2'b10;  
else if((address >= 0x20) && (address < 0x40))  
    {s0_sel, s1_sel} = 2'b01;  
...  
else  
    {s0_sel, s1_sel} = 2'b00;
```

Address Decoder

Address decoder

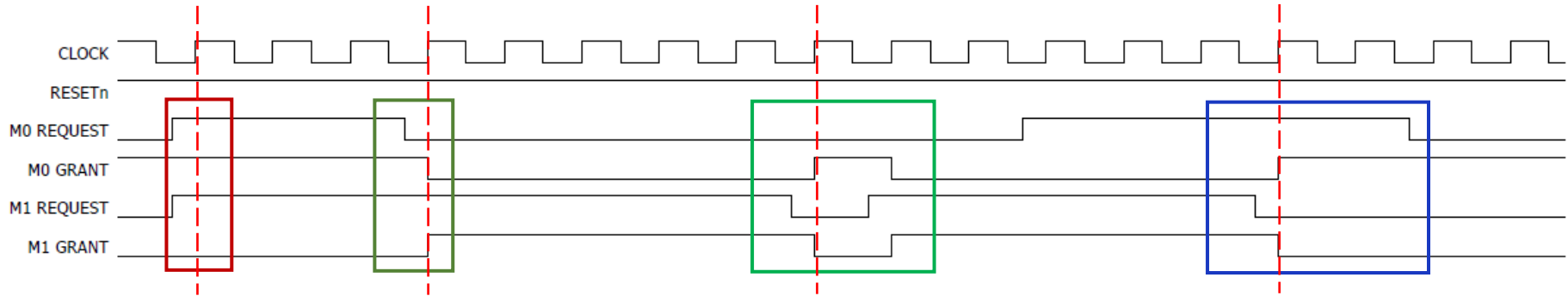
```
upper_3bit = address[7:5];  
if(upper_3bit == 3'b000)  
    {s0_sel, s1_sel} = 2'b10;  
else if(upper_3bit == 3'b001)  
    {s0_sel, s1_sel} = 2'b01;  
...  
else  
    {s0_sel, s1_sel} = 2'b00;
```

Testbench I



- 입력을 통해 들어오는 address에 따라 slave0 또는 slave1이 선택된다
 - ✓ Slave 0과 slave 1에 해당하지 않은 주소가 들어왔을 경우에는 어느 slave도 선택되지 말아야 한다.

Testbench II



- Master는 bus를 통해 data를 transfer하고자 할 때, 자신에게 해당하는 request signal을 1로 한 이후에, 그에 대한 확인으로 grant signal을 받은 후 data transfer를 올바르게 할 수 있다.
- Master가 grant signal을 받은 후 request signal이 1인 동안에는 bus의 소유권을 빼앗기지 않고 data transfer를 계속 할 수 있다.
- 만약 두 개의 master 모두 request를 하고 있지 않다면 grant는 master 0이 받는다.

Assignment 10

➤ Report

- ✓ 자세한 사항은 homework & practice document 참고

➤ Submission

- ✓ Soft copy
 - 수업 후 1 주일까지 제출 (딜레이 2일 (20% 감점))

Q&A

THANK YOU