

디지털논리회로 2 - Assignment 3 보고서

2021202085 전한아솔

1. 과제 설명

이번 과제는 3개의 master와 8개의 slave를 가지는 bus를 구현하는 과제이다.

<Bus>

Bus는 여러 요소들 간에 데이터를 전송할 수 있도록 연결해주는 장치이다. Master-slave의 구조를 가지는데, Master의 명령에 따라서 Slave가 해당 동작을 수행한다. 마스터는 여러 개 존재할 수 있다. 따라서 마스터의 요청을 받아서 어느 마스터의 요청을 수행할 지 결정하는 중재자(Arbiter)가 필요하다. 또한 각 마스터는 주소를 같이 전송하는데, 해당 주소 값에 따라서 어떤 Slave에게 명령할 지 선택할 수 있다.

<Arbiter>

Arbiter는 위에서 설명했듯, 여러 마스터의 요청 중 어떤 마스터에게 우선권을 부여할지 선택하는 모듈이다. Arbiter에 입력으로 모든 마스터의 요청이 들어오면, Arbiter는 그에 대한 응답으로 각 마스터에 해당하는 grant를 출력한다. 따라서 grant가 1인 마스터의 요청을 수행하도록 한다.

마스터는 $M_0 > M_1 > M_2 \dots$ 의 순서로 우선권을 가지므로 어떠한 마스터도 요청을 하지 않는 상태에서는 M_0 의 grant가 1이된다. 예를 들어, M_0, M_1 가 요청을 했다면 우선권을 M_0 에 있으므로 M_0 의 grant가 1이된다. 따라서 순서가 앞선 마스터가 요청을

하지 않은 상태에서 다른 마스터가 요청을 해야, 해당 마스터의 grant가 1이 될 수 있다.

2. 설계 과정

<Arbiter>

입력으로 받은 마스터의 요청에 따라서 어떤 마스터의 명령을 수행할지 정하는 기능의 모듈이다.

마스터는 총 3개가 존재하고, M0가 요청하면 다른 마스터의 요청은 0, 1과 상관없이 무조건 M0에게 우선권을 부여하도록 if문을 통해서 m0_req가 1이면 m0의 grant가 1이 되도록 한다.

M0_req가 0이고, m1_req가 1이라면 m2_req는 0, 1 상관없이 m1_grant가 1이 되도록 한다.

M0_req와 m1_req가 모두 0일 때, m2_req가 1이라면 m2_grant가 1이 되도록 한다.

아무런 요청이 없다면 m0_grant가 1이 되도록 한다.

grant에 따라 각 마스터의 write 중, 하나를 골라, S_wr의 output으로 출력한다. 따라서 1-bit Mux(3 input)를 사용하여 grant의 값이 1인 마스터의 write를 S_wr에 저장하는 방식으로 구현했다.

S_din과 S_address도 마찬가지로 grant의 값이 1인 마스터의 dout을 S_din으로 출력하고, grant의 값이 1인 마스터의 주소를 S_address로 출력한다. Dout은 32 bits이므로, S_wr에서 사용한 Mux와 같은 기능이지만 32 bits의 입력과 출력을 가지도록 구현했고 주소는 8 bits이므로 8 bits의 입력과 출력을 가지도록 구현했다.

따라서 S_wr, S_address, S_din은 같은 방식으로 동작하지만 입출력의 bit만 다른 Mux 3개를 사용하여 출력했다.

<Address decoder>

각 마스터의 주소는 8 bits로 이루어지는데, 이 주소로 8개의 Slave 중 하나를 고른다.

MSB부터 3개의 bit를 이용하여

000->Slave 0

001->Slave 1

010->Slave 2

011->Slave 3

100->Slave 4

101->Slave 5

110->Slave 6

111->Slave 7

의 구조로 Slave를 선택한다. 따라서 입력으로 주어진 주소의 [7:5] bits를 확인하여 알맞은 Slave 번호에 1, 나머지는 모두 0으로 구성된 8-bits output을 출력한다.

이렇게 Address decoder로 출력된 결과를 이용하여 Slaver의 dout 중 하나를 골라서 M_din으로 출력해야 하는데, MUX로 8개의 input 중 하나를 고르기 위해서는, 3 bits의 selection bits가 필요하다. 따라서 Address decoder의 출력을 3 bits로 바꿔 줄 로직인 encoder를 설계했다.

<Encoder>

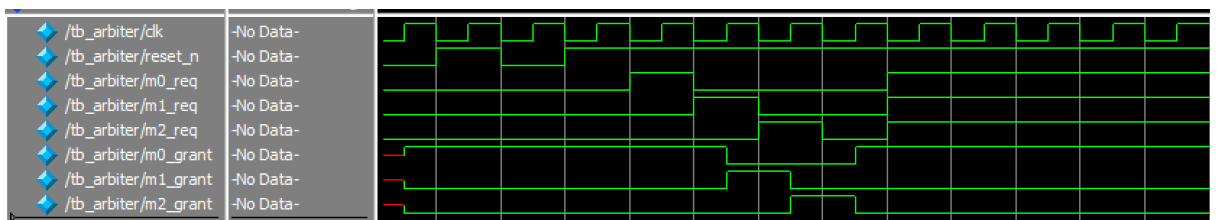
입력으로 들어오는 8 bits 값 중, 1이 존재하는 bit에 따라 3 bits를 output으로 출력한다. 예를 들어, 00001000이라면 LSB(0번

째)부터 3번째에 bit가 1이므로 output은 011이다.

이렇게 인코딩 된 output으로 3-to-8 Mux를 구성하여 Slave의 dout 중 하나를 골라 M_din으로 출력하도록 구현했다.

3. 설계 검증

<Arbiter>



모든 Master의 요청이 없으면 m0_grant가 1인 결과를 볼 수 있다.

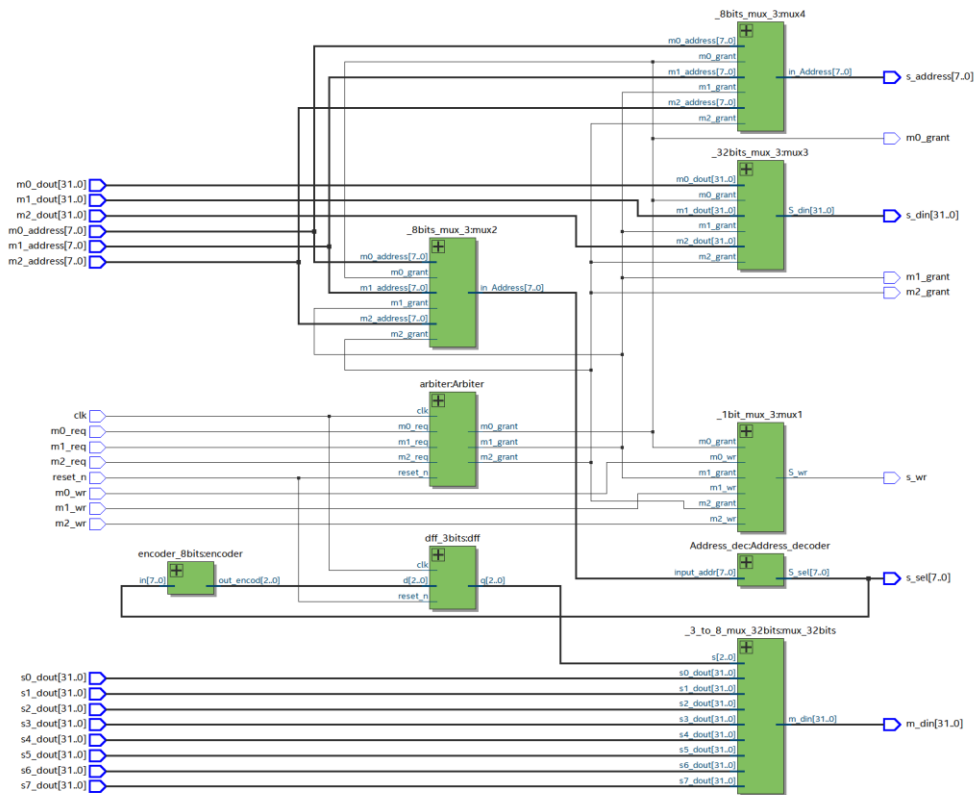
M0_req가 0이고, m1_req가 1인 경우에 m1_grant가 1, m0_req와 m1_req가 0이고, m2_req가 1인 경우에 m2_grant가 1인 결과를 볼 수 있다. 마지막 부분에서 모든 요청이 모두 1이라면 m0에 우선권을 부여하므로 m0_grant가 1인 결과를 볼 수 있다.

<Address decoder>

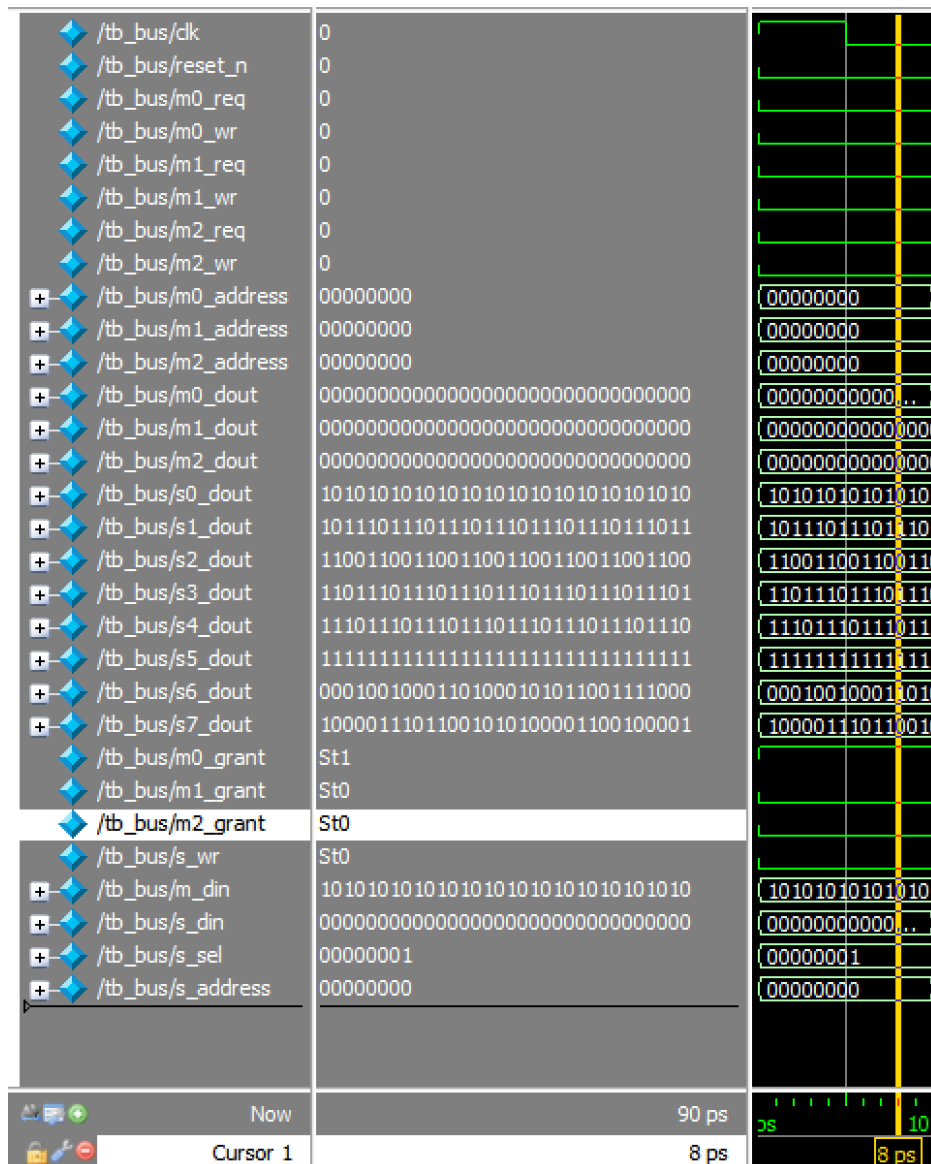
tb_Address_dec/input_addr	00000000	00000000	00100000	01000000	01100000	10000000	10100000	11000000	11100000	11111000	00001000
tb_Address_dec/S_sel	00000001	00000010	00000100	00001000	00010000	00100000	01000000	10000000			00000001

제안에서 나와있는 Slave의 주소범위에 따라서 입력의 MSB부터 3 bits를 확인하여 알맞게 디코딩 된 결과를 출력하는 모습을 볼 수 있다.

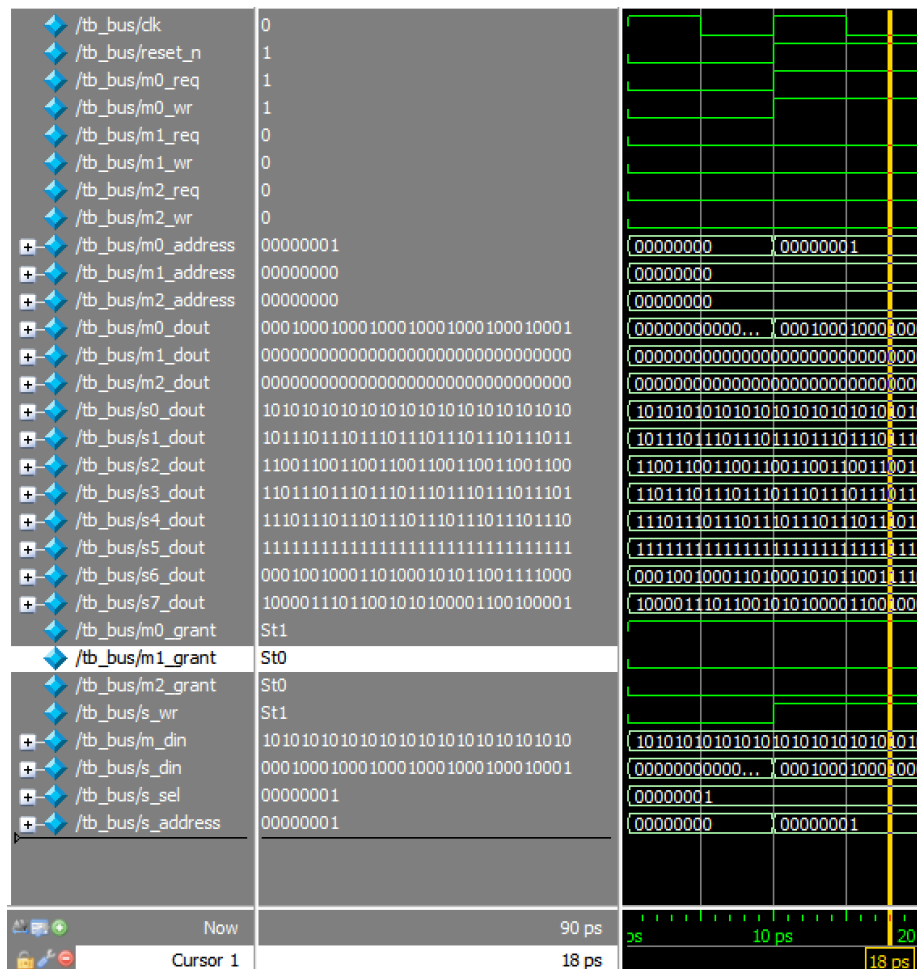
<Bus>



강의 자료의 bus와 비슷한 구조를 가지지만 master 3개, slave 8개인 bus이므로 각 모듈의 구조가 달라졌다. 특히, Arbiter의 output이 3개 이므로 Mux와 연결되는 select가 input이 3개이고, S_sel 또한 강의자료에서는 1bit씩 여러 개가 존재했으나, 이번 과제에서는 8 bits의 bus로 구성된다.

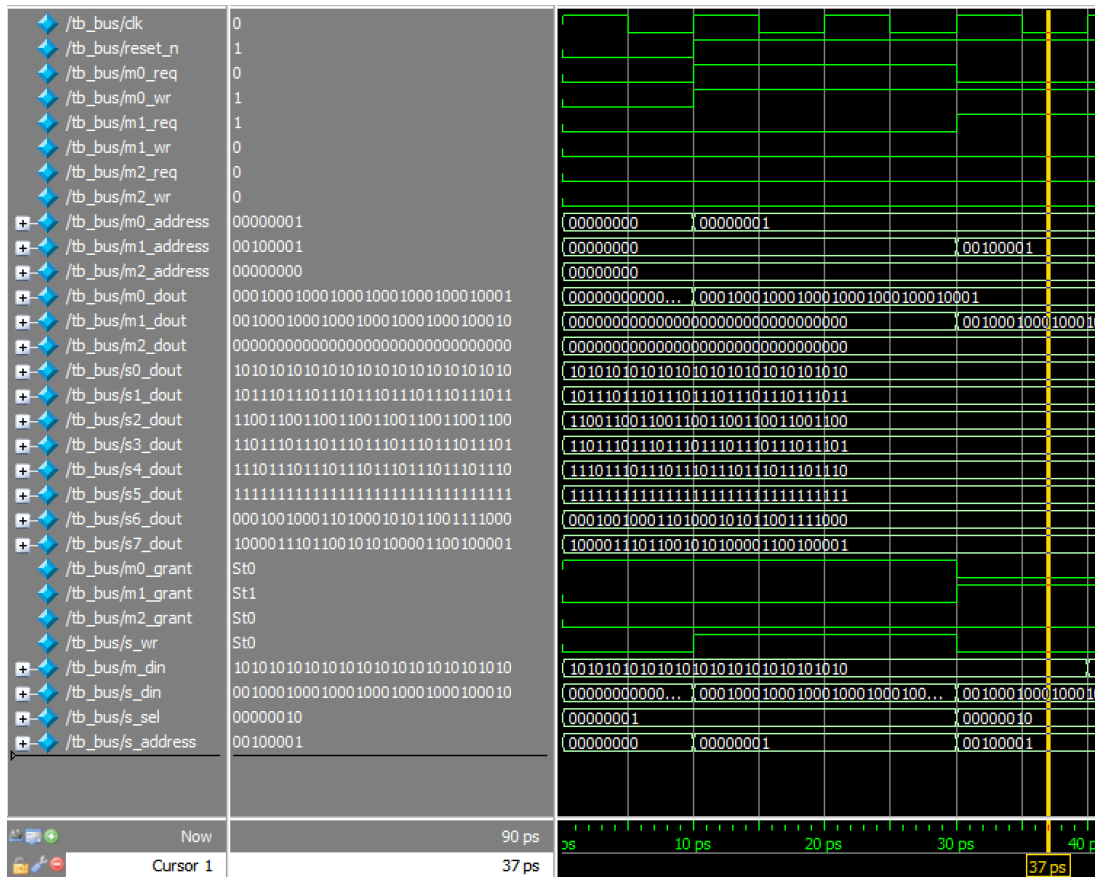


처음 테스트 벤치가 시작되면, reset_n이 0이고, 모든 요청이 없다. 따라서 m0_grant가 1이고 s_din은 0이다. M0_address의 상위 3bits가 000이므로 s_sel의 0번째 bit가 1, m_din은 s0_dout을 출력하는 결과를 볼 수 있다.



Reset_n이 1이 되고 m0_req가 1이므로 m0_grant가 1이다. M0의 address의 상위 3bits는 000이므로 s_sel은 1, m_din은 s0_dout을 출력한다.

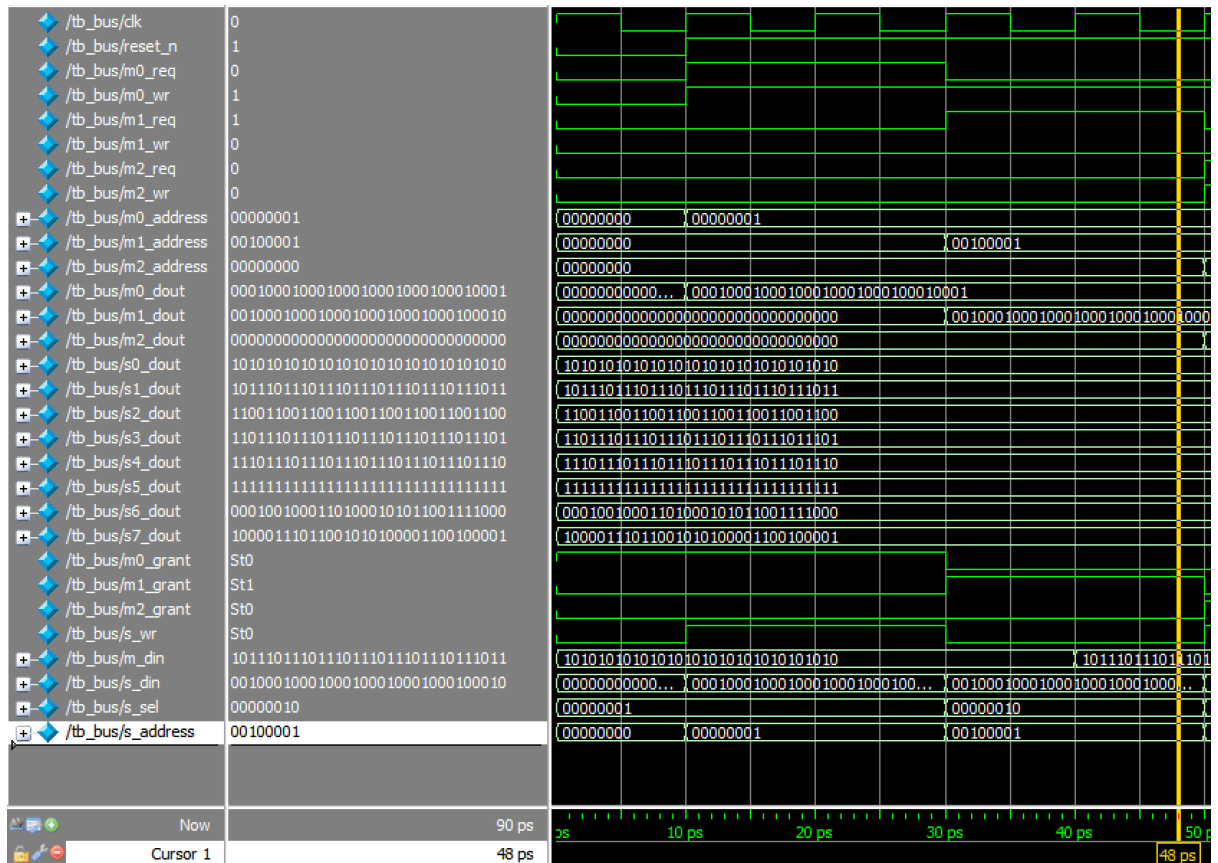
현재 m0의 명령을 수행하고 있으므로 s_din은 m0_dout을 출력하는 모습을 볼 수 있고 M0_wr이 1이므로 s_wr도 1임을 볼 수 있다.



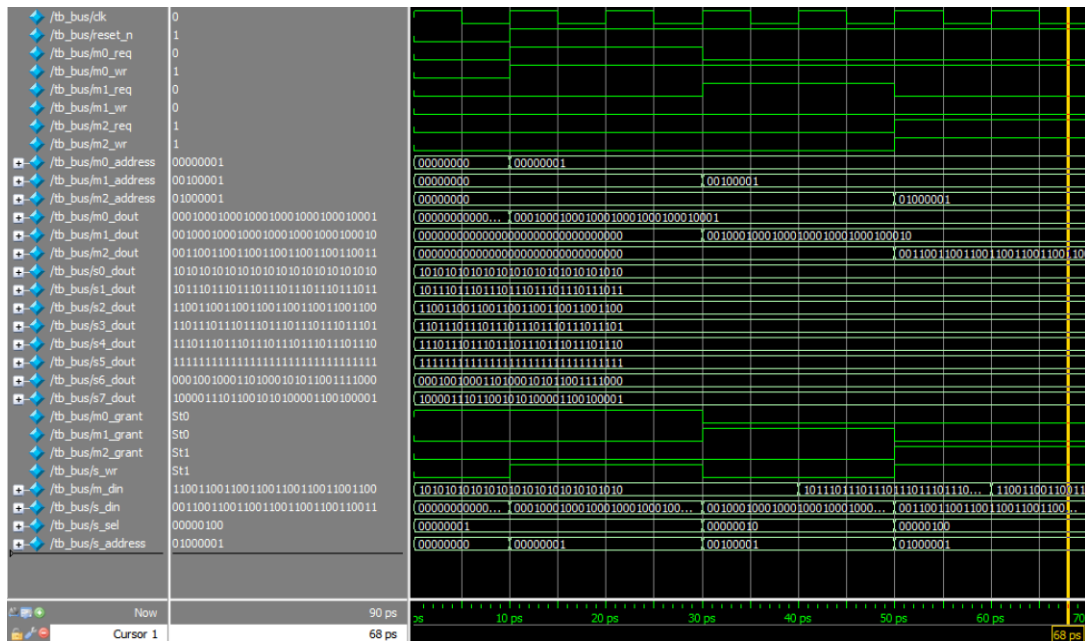
M0의 req가 0이고 m1의 req가 1이므로 m1_grant가 1이 되고
M1_wr은 0이므로 s_wr도 0이 되는 결과를 볼 수 있다.

현재 m1의 명령을 수행하고 m1_address의 상위 3 bits는 001이다. 따라서 s_sel의 1번째 bit가 1임을 볼 수 있다. m_din은 s1_dout을 출력해야 하지만 Address decoder의 output이 D Flip-Flop을 거쳐서 Mux에 입력되어 m_din을 결정하므로 다음 클럭(negative edge)에 동작하여 m_din을 s1_dout으로 바꾼다.

S_din은 알맞게 m1_dout을 출력하는 결과를 볼 수 있다.

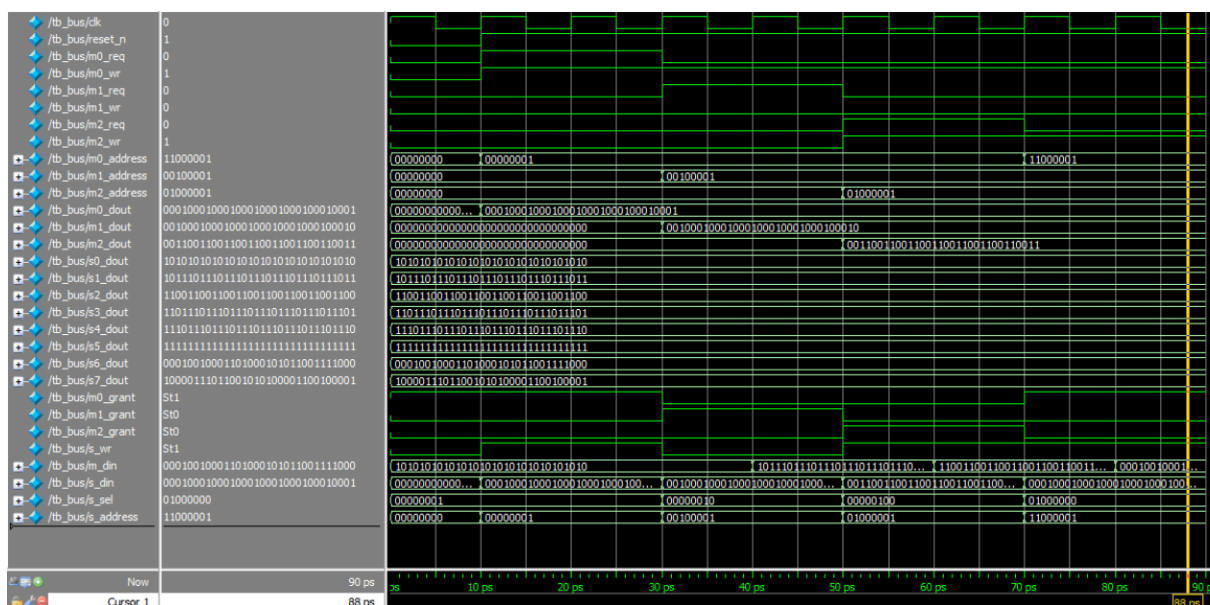


다음 클럭이 지난 후, m_din이 s1_dout과 동일하게 바뀔 수 있다.



M0, m1의 req가 모두 0이고 m2의 req만 1이므로 m2_grant가 1이고 m2_address의 상위 3bits는 010이다. 따라서 s_sel의 2번째 bit가 1이 된 결과를 볼 수 있다.

마찬가지로, m_din은 한 클럭이 지난 후 변경되고, s_din은 m2_dout을 알맞게 출력하는 결과를 볼 수 있다.



어떤 요청도 없으므로 m0_grant는 1, m0_address의 상위 3bits는 110

이므로 s_sel의 6번째 bit가 1이 된 결과를 볼 수 있다.

따라서 s_din은 m0_dout과 같은 값을 가지고 m_din은 s6_dout을 가짐을 볼 수 있다.

4. 고찰

강의자료에 나와있는 bus의 회로에서 input인데 이름이 dout이고 output인데 이름이 din으로 적혀 있어서 테스트 벤치를 보고 과제를 이해하는 과정에서 계속 헷갈렸다. 따라서 베릴로그 코드를 작성하는 과정에서 회로도를 계속 보면서 input, output의 선언이나 코드를 작성하는 과정에서 헷갈리지 않도록 했다.

M_din이 s_din보다 한 클럭 뒤에 값이 바뀌는 현상이 존재해서 생각해본 결과, s_din은 arbiter에서 한 클럭 동안 동작한 결과를 가지고 바로 값을 출력할 수 있지만, m_din은 arbiter를 통과한 결과를 Address decoder에 전달하고, Address decoder의 결과 값으로 D-FF을 한 번 더 거쳐서 출력된다. 따라서 m_din의 결과는 한 클럭 뒤에 바뀌는 것이 맞다고 생각하여 해당 상태로 과제를 제출했다.