LAB 2

Digital System Design - 2020 spring

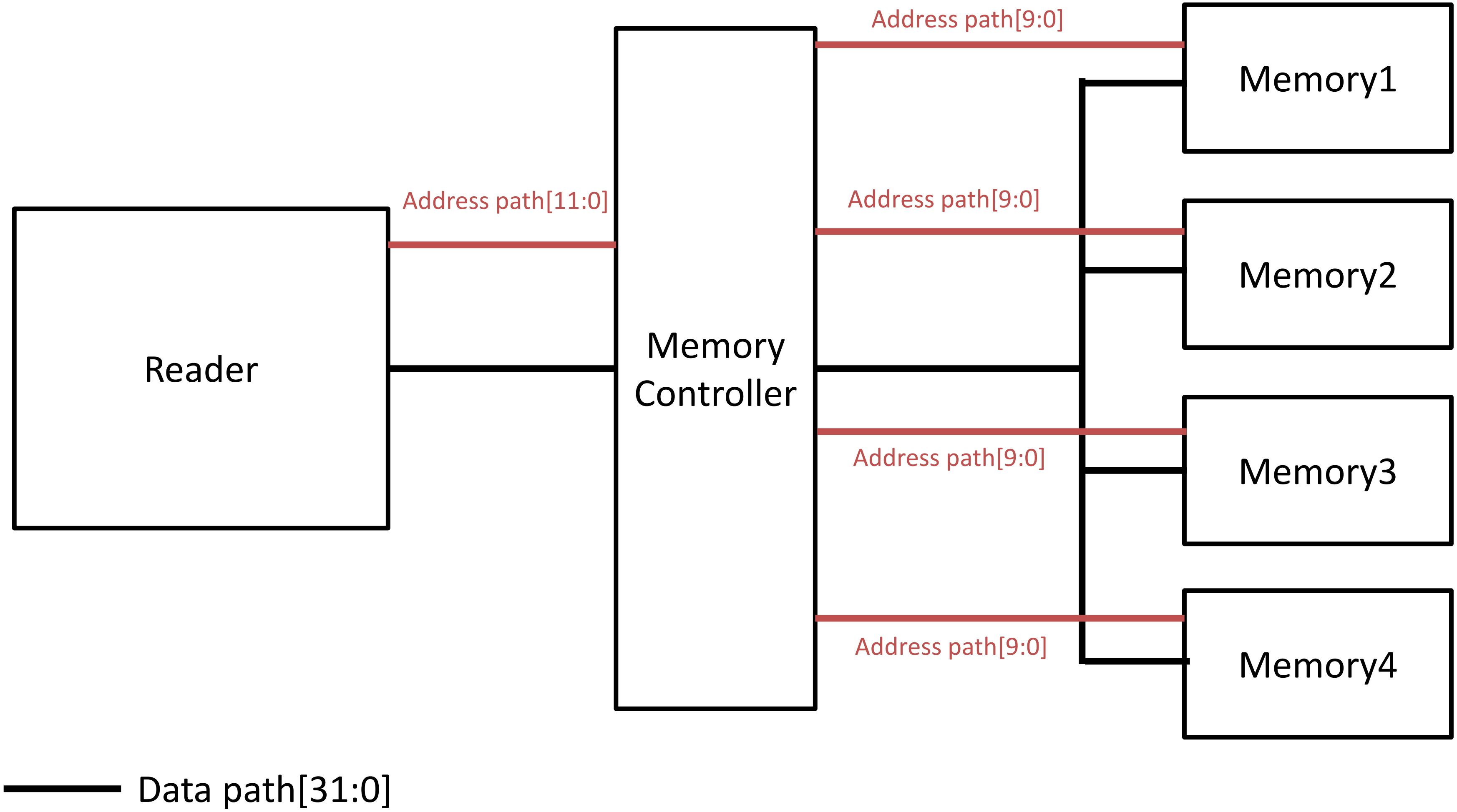
Sungkyunkwan University

Overview

- **Objective**
  - Design ‘memory controller’ for various memories
  - Make ‘read operation’ of LAB2

- **Collaboration Policy**
  - All project must be done by personally
  - If you copy other students’ project, you will fail this course.

Block Diagram



Specification

- **Memory reader:** 2ns Clock period

- 2ns Clock period
- When allow_address == 1’b1, the reader gives the address value, which the reader wants to read, to the memory controller every next posedge clk.
- When read_complete == 1’b1 at posedge, it assume that the data has arrived.
- The data must arrive in the order the addresses were sent.
- Ex)

Allowed	Not allowed
Send address1-> Send address2-> Receive data1-> Receive data2	Send address1-> Send address2 -> Receive data2 -> Receive data1

Specification

➢ **Memory controller:** 2ns clock period

- What you have to make
- Don’t use # syntax in your code.
- You can’t add buffer besides given buffers.
- When rstn ==1, your code performs ‘reset operation’.
- When ‘read signal’ is applied to each memory, each memory has to read data in its buffers using the given address.
- Each ‘path_on’ bit means the memory number which memory controller will use.
- Ex)

path_on[3:0]	Result
0000	No use data path
0001	Memory1 use
0010	Memory2 use
0100	Memory3 use
1000	Memory4 use

read[3:0]	Result
0001	Memory1 read
0011	Memory1 &2 read
1010	Memory2 & 4 read
1111	All memory read
1000	Memory4 read

Specification

- **Memory common option**

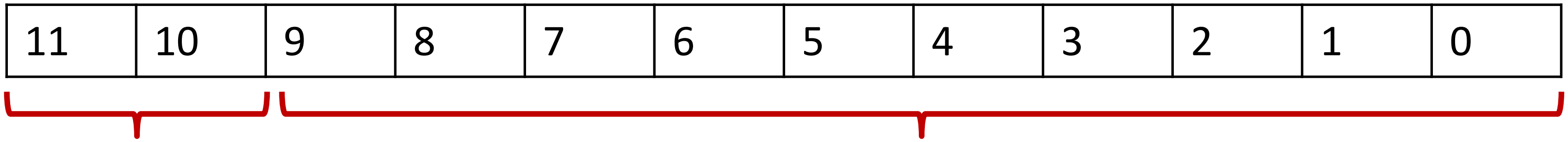
Only one memory uses data path connected to memory controller at once.

Each memory has 1024 32-bit data.

- **Memory1:** 4ns clock period, posedge active
- **Memory2:** 4ns clock period, negedge active
- **Memory3:** 8ns clock period, posedge active
- **Memory4:** 4ns clock period, negedge active, it gives back data 4ns later.

Specification

- **Memory address [11:0]**



11	10	9	8	7	6	5	4	3	2	1	0

- **Memory Number[11:10]**

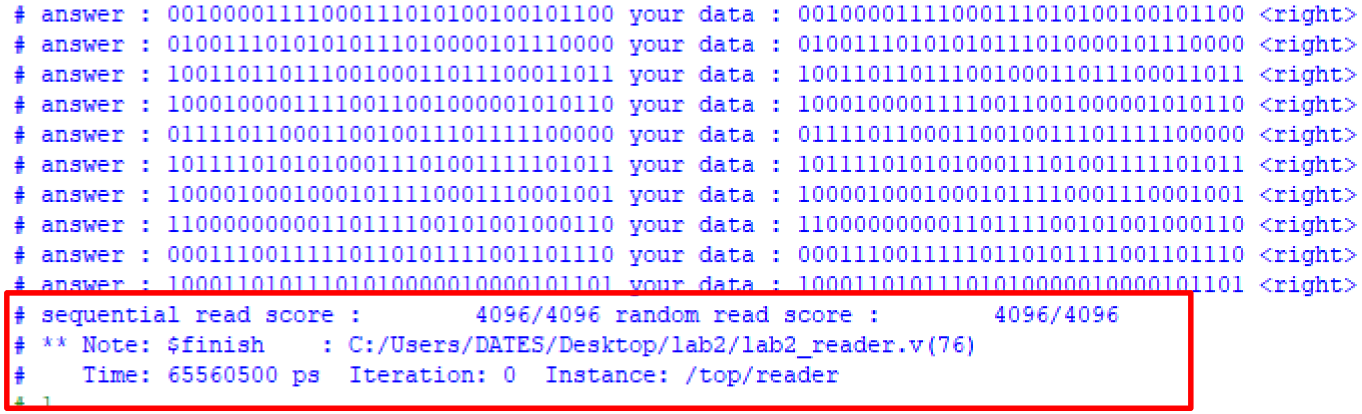
Address in memory

- 00:Memory1
- 01:Memory2
- 10:Memory3
- 11:Memory4

- **Other spec : check the Lab2 skeleton code**

Test_bench

- **Simulate with module “top”**
- **Result will show your score with sequential read & random read and time.**



```
# answer : 00100001111000111010100100101100 your data : 00100001111000111010100100101100 <right>
# answer : 01001110101010111010000101110000 your data : 01001110101010111010000101110000 <right>
# answer : 10011011011100100011011100011011 your data : 10011011011100100011011100011011 <right>
# answer : 10001000011110011001000001010110 your data : 10001000011110011001000001010110 <right>
# answer : 01111011000110010011101111100000 your data : 01111011000110010011101111100000 <right>
# answer : 10111101010100011101001111101011 your data : 10111101010100011101001111101011 <right>
# answer : 10000100010001011110001110001001 your data : 10000100010001011110001110001001 <right>
# answer : 11000000000110111100101001000110 your data : 11000000000110111100101001000110 <right>
# answer : 00011100111110110101111001101110 your data : 00011100111110110101111001101110 <right>
# answer : 10001101011101010000010000101101 your data : 10001101011101010000010000101101 <right>
# sequential read score :        4096/4096 random read score :        4096/4096
# ** Note: $finish    : C:/Users/DATES/Desktop/lab2/lab2_reader.v(76)
#    Time: 65560500 ps  Iteration: 0  Instance: /top/reader
```

Restriction & Precaution

- Never use # Syntax & initial syntax in your code
- You can't add address and data buffer besides given buffers.
- Place the '.bin' files(memory1.bin , random_ans.bin etc) in the folder where your project(.mpf) is for test.
- Your code will be graded with other samples.

Submission

- Submit your **lab2_memory_controller.v file** and **report** on iCampus
- Report must include explanation of your code (ex. state diagram, how to make fast, what you learned, and so on)
- Format : name_ID.v & name_ID.pdf
- Ex) 홍길동_2020000000.v,홍길동_2020000000.pdf

Grade

- **Function test(50)**

Sequential read(25) + Random read(25)

- **The Speed of your memory controller(30)**

Your code will be graded by the rank of memory controller speed from 0 to 30 points

Only code that is perfectly implemented for the function will be graded in terms of speed.

- **The report(20)**

Q&A

- **Only the questions on the I-Campus will be accepted not e-mail.**