CMPE 260 – Digital Systems Design II

Lab Exercise 6: SRAM with Built in Memory Controller

Hrishikesh Moholkar

Performed: 5/3/2017

Submitted: 5/10/2017

Lab Section: 01L1

Professor: Richard Cliver

Lab instructor: Mdshahriar Shamim

TA: Iosif Grigoryev

Table of Contents:

Abstract

This lab deals with the development of a memory unit that performs asynchronous read and write to the SRAM with the help of synchronous memory controller. The memory controller communicates with the instructions of the outer synchronous components and synchronously performs actions on the asynchronous SRAM. The memory controller is a Moore state machine while SRAM is an array of standard logic vector which stores the data of the standard logic vector type. The IO- bus encloses the memory unit and the output is displayed by the seven segment display. VHDL code is written to simulate the working of the memory unit. Behavioral and Post-route simulation are run in order to test the functioning of the memory unit. The code is then loaded and tested on the Nexy3-Board. The results obtained are satisfactory. The data can be written to desired address and can be read from the required address.

Design Methodology

The memory unit consists of two main components:

1) SRAM
2) Memory Controller

Memory controller:
The inputs to the memory controller are as follows:

1) Clk
2) Reset (synchronous, active low)
3) Bus_id (for selection of the SRAM)
4) rw(instruction for reading or writing the data)
5) ready(starting point of the reading or writing process)
6) burst(reading the data from the consecutive addresses)
7) addr(3 bits)

Outputs of the memory controller include WE, OE and offset.
The memory controller is a **Moore state-machine**.
The following is the state diagram for the memory controller:

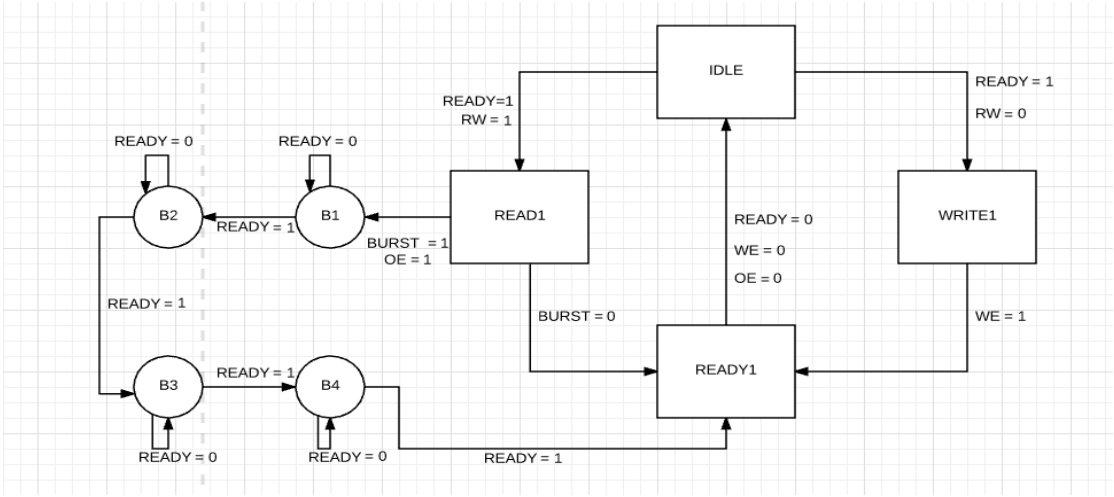


Figure 1.0: Memory Controller

The memory controller has five states. The states are as follows:
IDLE, READ1, WRITE1, READY1, and B1, B2, B3, B4.
When the user wants to write the data to SRAM, READY variable should be set to 1 and RW variable should be set to 1. At that moment, WE is set to 1 and the data is being written to the SRAM to its required address until READY is 1. Once READY is 0 then it stops writing and it goes into IDLE state. When the user wants to read the data from the SRAM, then RW is set to 1 and READY is set to 1. If BURST is set to 0 then it performs normal read operation from the given address otherwise it reads data from the next four consecutive addresses. Whenever the memory controller goes to B1 state, the offset is 01, when in B2 state, the offset is 10, when in B3 state, the offset 11 and in B4 state, the offset is 00.

SRAM:
The inputs to the SRAM are as follows:

1) OE
2) WE
3) ADDR(3 bits long)
4) DATA_IN(4 bits long)

The output of the SRAM is DATA_OUT.
Whenever OE is set to 1 and WE is set to 0 then the output data should be data at the address. When OE is set to 0 and WE is set to 1 then the output data should be the input data as the write operation is being performed. Otherwise the output data should remain unchanged.

IO Bus:

This component wraps the memory controller and SRAM together. The output from the SRAM is sent to seven segment decoder module which then sends the separated signals into seven segment display to output the data.

Wrapper:

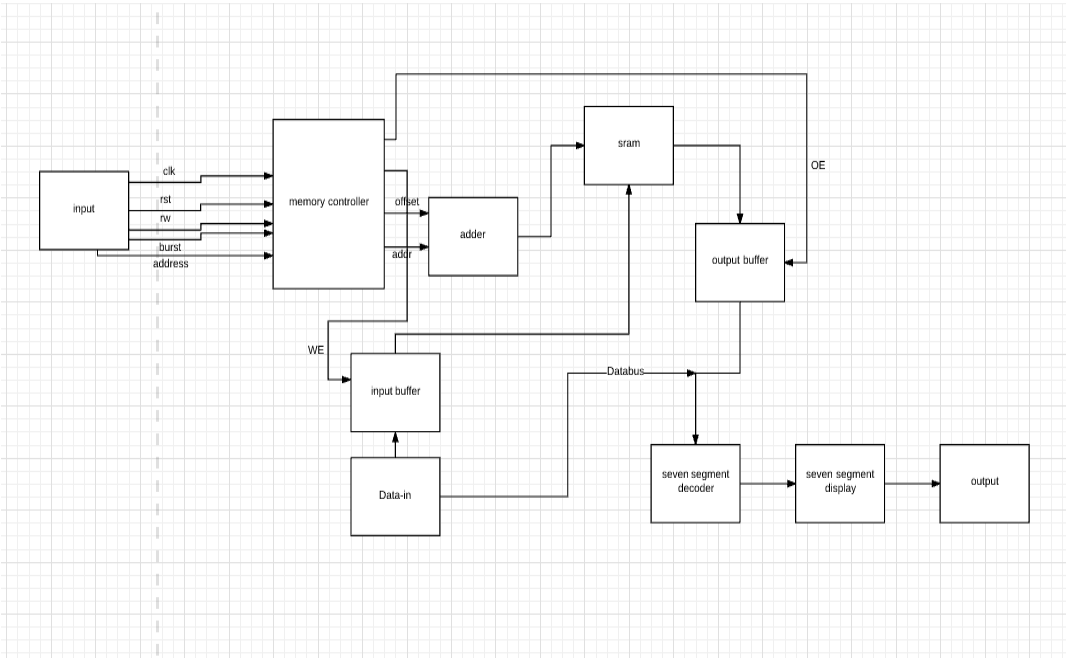


Figure 1.1: Wrapper

Results

Behavioral Simulation:

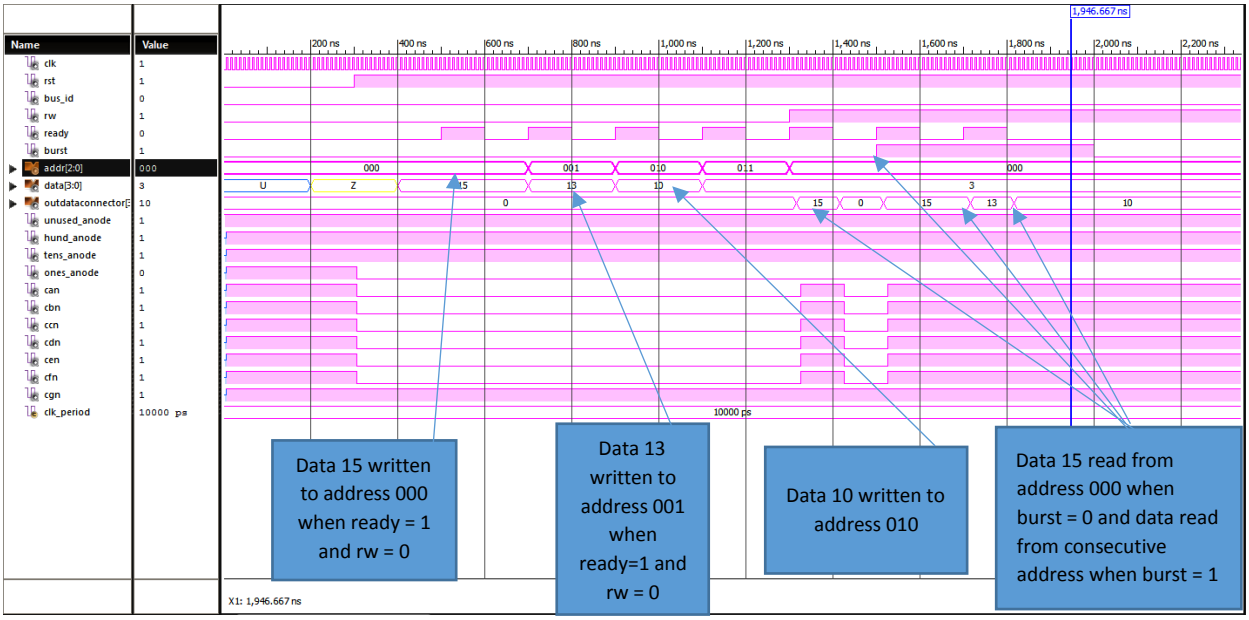


Figure 1.2: Memory unit simulation

As per the simulation, when the rw is 0 and ready is 1 then the data 15 is written to the address 000. Data 13 is written to the address 001 and data 13 is written 010. When rw is 1 then it reads the data from the given address where burst is 0. If burst is set to 1 then it reads all data from the addresses following the address 000.

Post-Route Simulation:

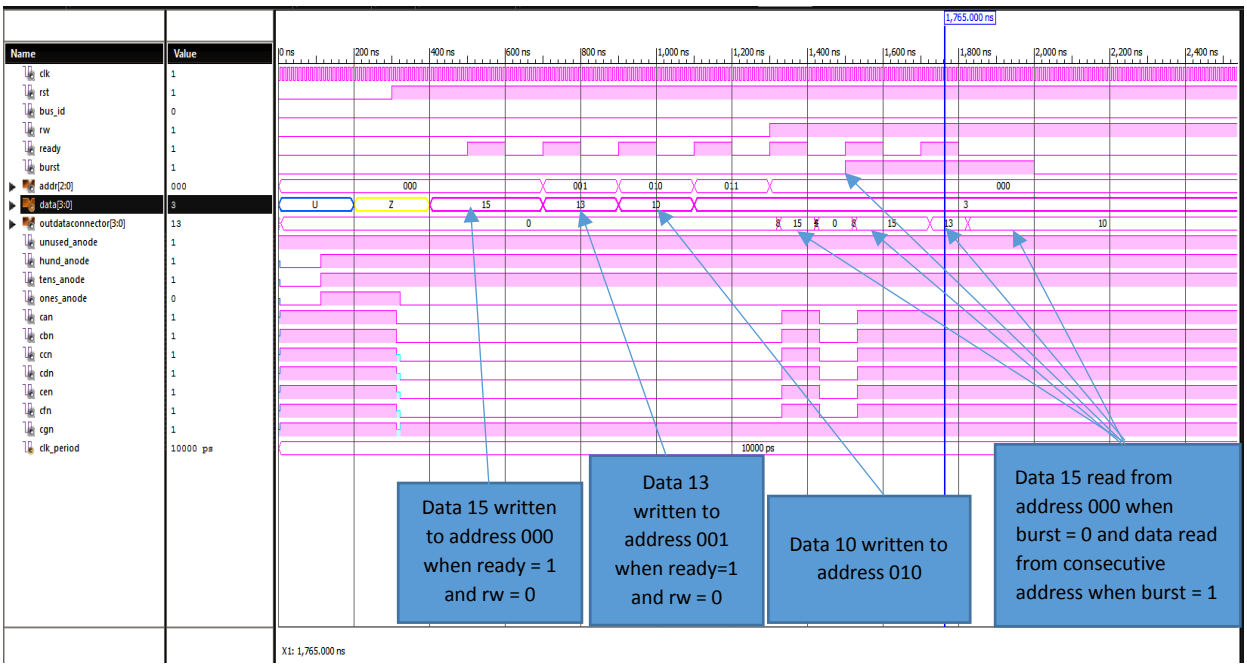


Figure 1.3: Memory unit simulation

As per the simulation, when the rw is 0 and ready is 1 then the data 15 is written to the address 000. Data 13 is written to the address 001 and data 13 is written 010. When rw is 1 then it reads the data from the given address where burst is 0. If burst is set to 1 then it reads all data from the addresses following the address 000. The output of post-route simulation and behavioral simulation are nearly same. The out data connector which is the in-out port writes "zzzz" when it is reading.

Area used:

```
Device Utilization Summary:

Slice Logic Utilization:
  Number of Slice Registers:                    72 out of  18,224    1%
    Number used as Flip Flops:                  40
    Number used as Latches:                     32
    Number used as Latch-thrus:                  0
    Number used as AND/OR logics:                0
  Number of Slice LUTs:                         67 out of   9,112    1%
    Number used as logic:                       66 out of   9,112    1%
      Number using O6 output only:              37
      Number using O5 output only:              17
      Number using O5 and O6:                   12
      Number used as ROM:                        0
    Number used as Memory:                       0 out of   2,176    0%
    Number used exclusively as route-thrus:      1
      Number with same-slice register load:      0
      Number with same-slice carry load:         1
      Number with other load:                    0

Slice Logic Distribution:
  Number of occupied Slices:                    29 out of   2,278    1%
  Number of MUXCYs used:                        20 out of   4,556    1%
  Number of LUT Flip Flop pairs used:           90
    Number with an unused Flip Flop:            19 out of      90   21%
    Number with an unused LUT:                  23 out of      90   25%
    Number of fully used LUT-FF pairs:          48 out of      90   53%
    Number of slice register sites lost
      to control set restrictions:               0 out of  18,224    0%

  A LUT Flip Flop pair for this architecture represents one LUT paired with
  one Flip Flop within a slice.  A control set is a unique combination of
  clock, reset, set, and enable signals for a registered element.
  The Slice Logic Distribution report is not meaningful if the design is
  over-mapped for a non-slice resource or if Placement fails.

IO Utilization:
  Number of bonded IOBs:                        27 out of     232   11%
    Number of LOCed IOBs:                       25 out of      27   92%
```

Figure 1.4: IObus.par file

Area used by flip-flop: 40
LUT used: 67 out of 9112
Occupied slices: 29 out of 2278

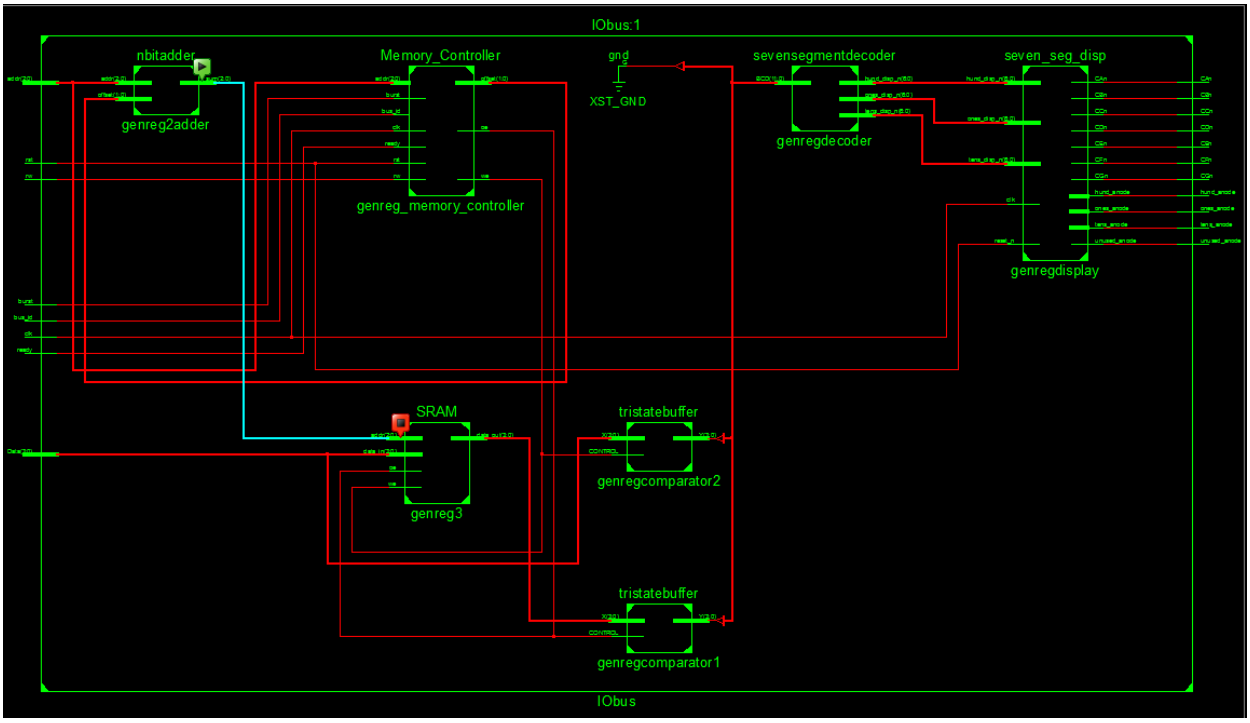
Timing Result:

Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
Autotimespec constraint for clock net clk_BUFGP	SETUP	N/A	5.088ns	N/A	0
	HOLD	0.415ns		0	0

Figure 1.5: IObus.par file timing result

The best case achievable timing is 5.088 ns.

RTL Schematic:

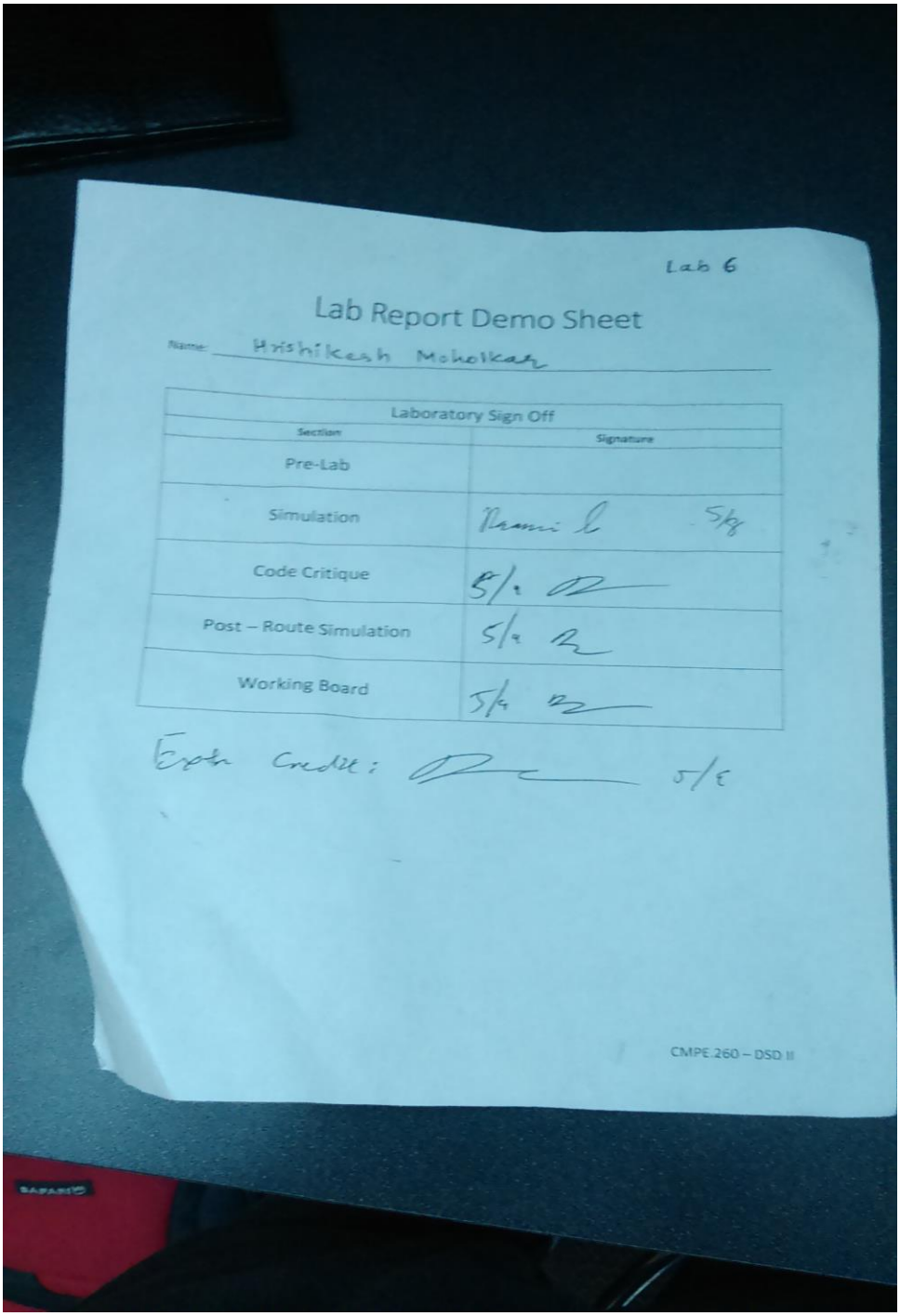


Memory unit

Conclusion

The data can be correctly read from and correctly written to the SRAM with the help of memory controller. The tristate buffers used in the RTL schematic helps to use one bus for input as well as output. The VHDL code tested on NEXYS-3 board shows the correct result. Another SRAM and Memory Controller can be added to the original structure by using a multiplexer and based on the bus number of SRAM, it can be used to read and write data to and from selected SRAM.

Sign-off sheet



Lab 6

Lab Report Demo Sheet

Name: Hrishikesh Moholkar

Laboratory Sign Off	
Section	Signature
Pre-Lab	
Simulation	Hrishikesh 5/8
Code Critique	5/8
Post – Route Simulation	5/8
Working Board	5/8

Extra Credit: 5/8

CMPE 260 – DSD II