Name: Xinyu Hadrian Hu, and Duan Wei Zhang

Student Number: 500194233, and 500824903

TA: Jasminder Singh

Date: June 28, 2020

COE 608: Computer Architecture and Design

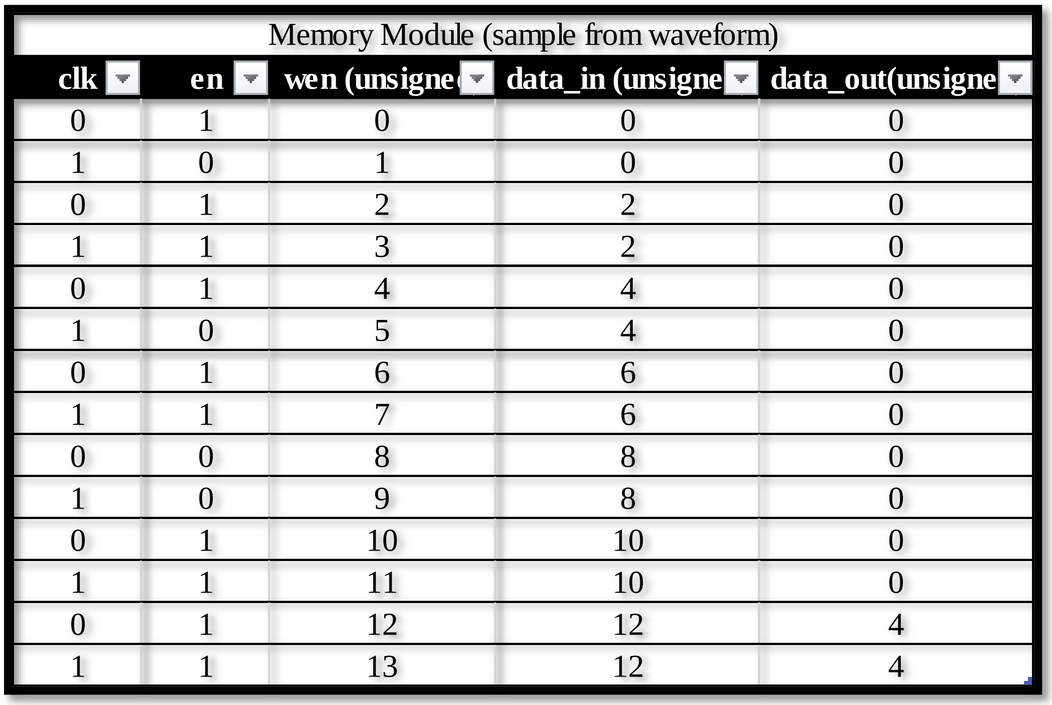
COE 608: Lab 4, Part 1 Report

Objective

The purpose of this lab is to create a memory module for the 32-bit ALU, and later the instruction set architecture (ISA) for the million instructions per second (MIPS) ISA.

Design and Implementation

Provided below is the truth table for the memory module. The address is an 8-bit data, and the data-in and data-outs are 32-bit data. See Figure 1: Memory Module Truth Table.



Memory Module (sample from waveform)

clk	en	wen (unsigne	data_in (unsigne	data_out(unsigne
0	1	0	0	0
1	0	1	0	0
0	1	2	2	0
1	1	3	2	0
0	1	4	4	0
1	0	5	4	0
0	1	6	6	0
1	1	7	6	0
0	0	8	8	0
1	0	9	8	0
0	1	10	10	0
1	1	11	10	0
0	1	12	12	4
1	1	13	12	4

Figure 1: Memory Module Truth Table

Observations and Results

Below are the observations from functional and timing waveforms: See Figure 2: Memory Module Functional Waveform and Figure 3: Memory Module Timing Waveform.

Memory Module: Functional and Timing

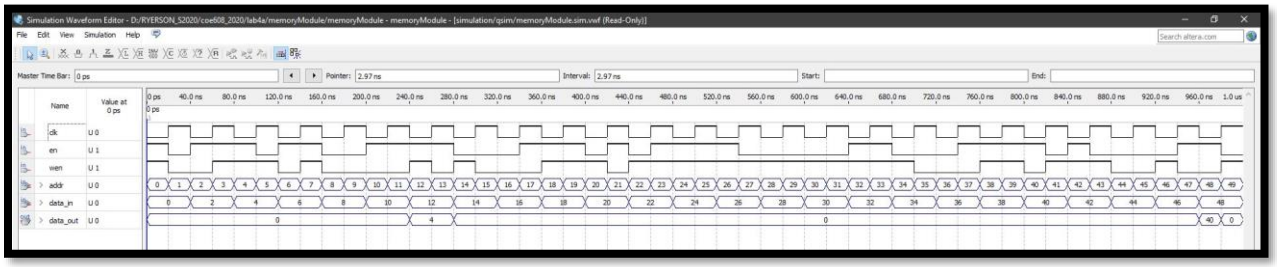

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab4a/memoryModule/memoryModule - memoryModule - [simulation/qsim/memoryModule.sim.vwf (Read-Only)]
File Edit View Simulation Help
Master Time Bar: 0 ps
Pointer: 2.97 ns
Interval: 2.97 ns
Start:
End:
Name
Value at 0 ps
clk U 0
en U 1
wen U 1
addr U 0
data_in U 0
data_out U 0

Figure 2: Memory Module Functional Waveform

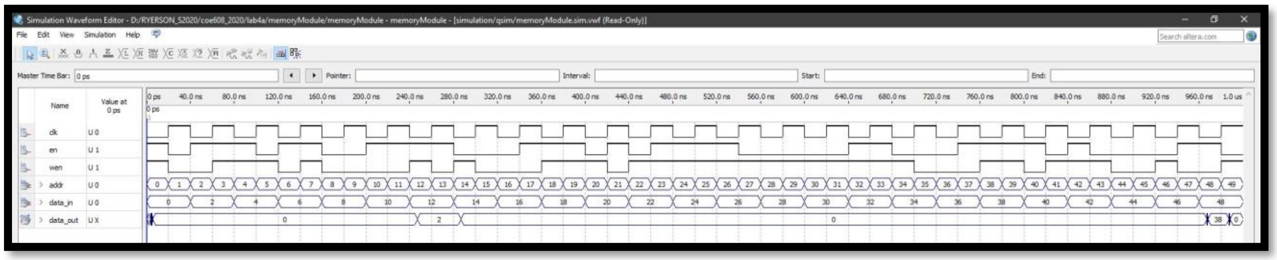

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab4a/memoryModule/memoryModule - memoryModule - [simulation/qsim/memoryModule.sim.vwf (Read-Only)]
File Edit View Simulation Help
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
clk U 0
en U 1
wen U 1
addr U 0
data_in U 0
data_out U X

Figure 3: Memory Module Timing Waveform

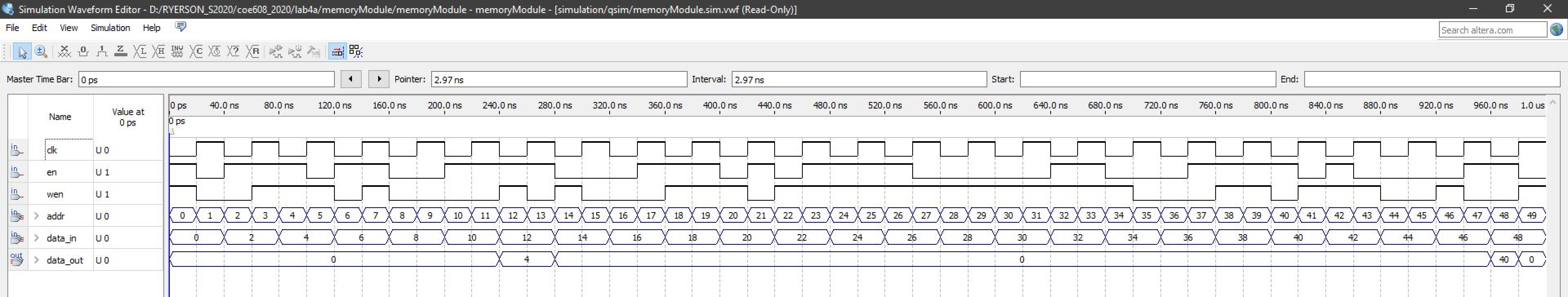
Discussions and Conclusions

The memory module works as intended. The worst case delays for the memory module is 5 ns, as can be observed from the timing simulation waveform.

Appendix: VHDL Codes and Screenshots of Waveforms

I have provided the VHDL and screenshots of the waveforms for easier reading.

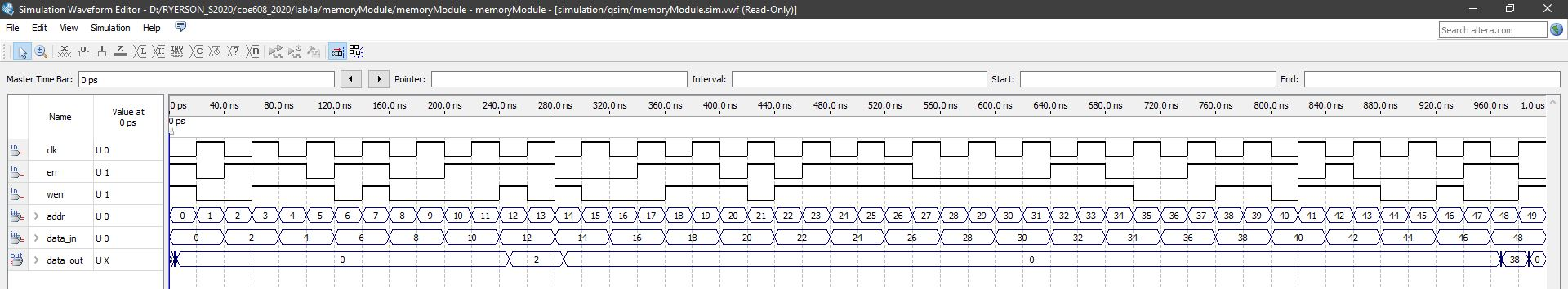
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity memoryModule is
   port (
      clk: in std_logic;
      addr: in unsigned (7 downto 0);
      data_in : in std_logic_vector(31 downto 0);
      wen, en: in std_logic;
      data_out: out std_logic_vector(31 downto 0));
end memoryModule;

architecture description of memoryModule is
   type memory2D is array (7 downto 0) of std_logic_vector(31 downto 0);
   signal memoryModuleArray : memory2D;
      begin
         process(clk,en,wen)
            begin
               if falling_edge(clk) then
                  if (en = '1' and wen = '0') then
                     data_out <= memoryModuleArray(to_integer(unsigned(addr)));
                  elsif (en = '1' and wen = '1') then
                     memoryModuleArray(to_integer(unsigned(addr))) <= data_in;
                     data_out <= (others => '0');
                  end if;
               end if;
               if en = '0' then
                  data_out <= (others => '0');
               end if;
         end process;
end description;


```



Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab4a/memoryModule/memoryModule - memoryModule - [simulation/qsim/memoryModule.sim.vwf (Read-Only)]

Master Time Bar: 0 ps
Pointer: 2.97 ns
Interval: 2.97 ns
Start:
End:

	Name	Value at 0 ps
in	clk	U 0
in	en	U 1
in	wen	U 1
in	addr	U 0
in	data_in	U 0
out	data_out	U 0



Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab4a/memoryModule/memoryModule - memoryModule - [simulation/qsim/memoryModule.sim.vwf (Read-Only)]

File Edit View Simulation Help

Master Time Bar: 0 ps

Pointer:

Interval:

Start:

End:

Name	Value at 0 ps
clk	U 0
en	U 1
wen	U 1
addr	U 0
data_in	U 0
data_out	U X