Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	**Computer Organization and Architecture**
Course Number:	**COE608**
Semester/Year (e.g.F2016)	**Winter 2023**

Instructor:	**Demetres Kostas**

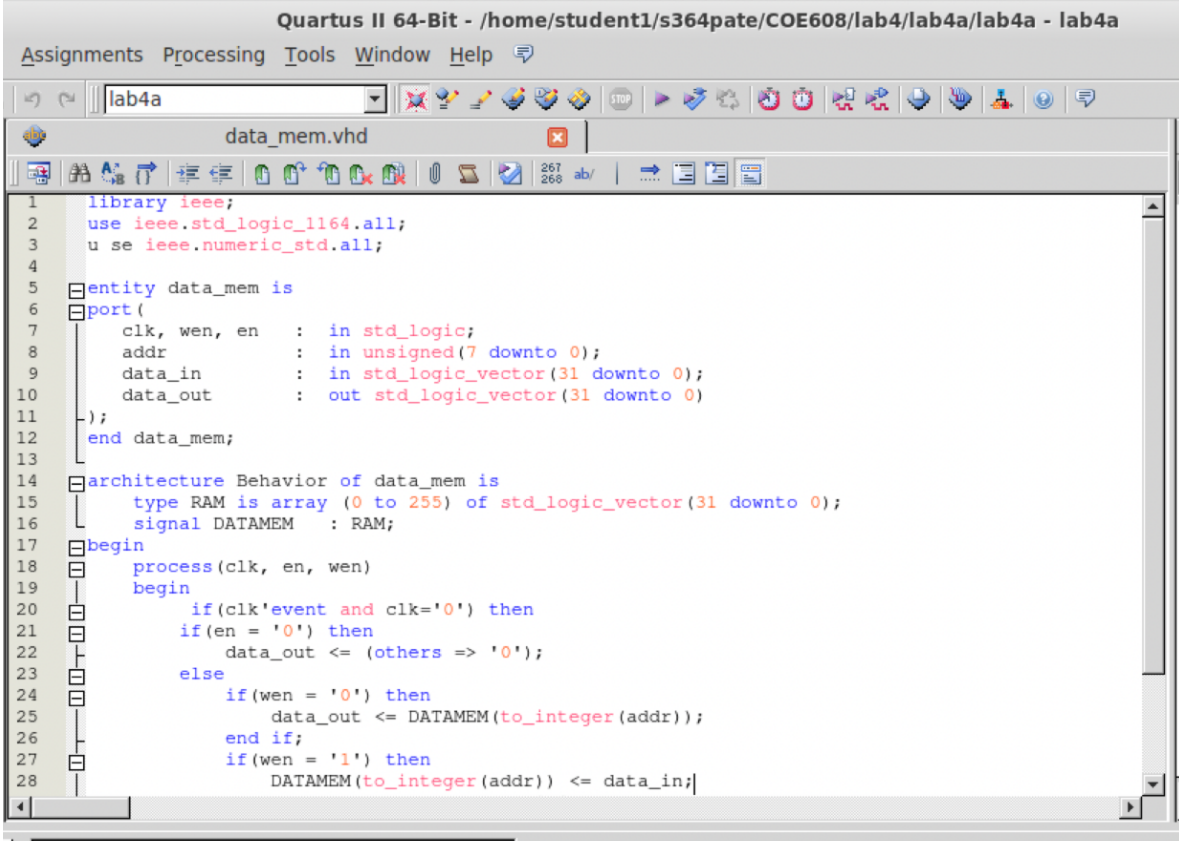
Assignment/Lab Number:	*4a*
Assignment/Lab Title:	*Data Memory Module*

Submission Date:	
Due Date:	*Wednesday March 1 2023 3:00pm*

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Patel	**Stuti**		**04**	**S.P**

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf

Data Memory Part 1

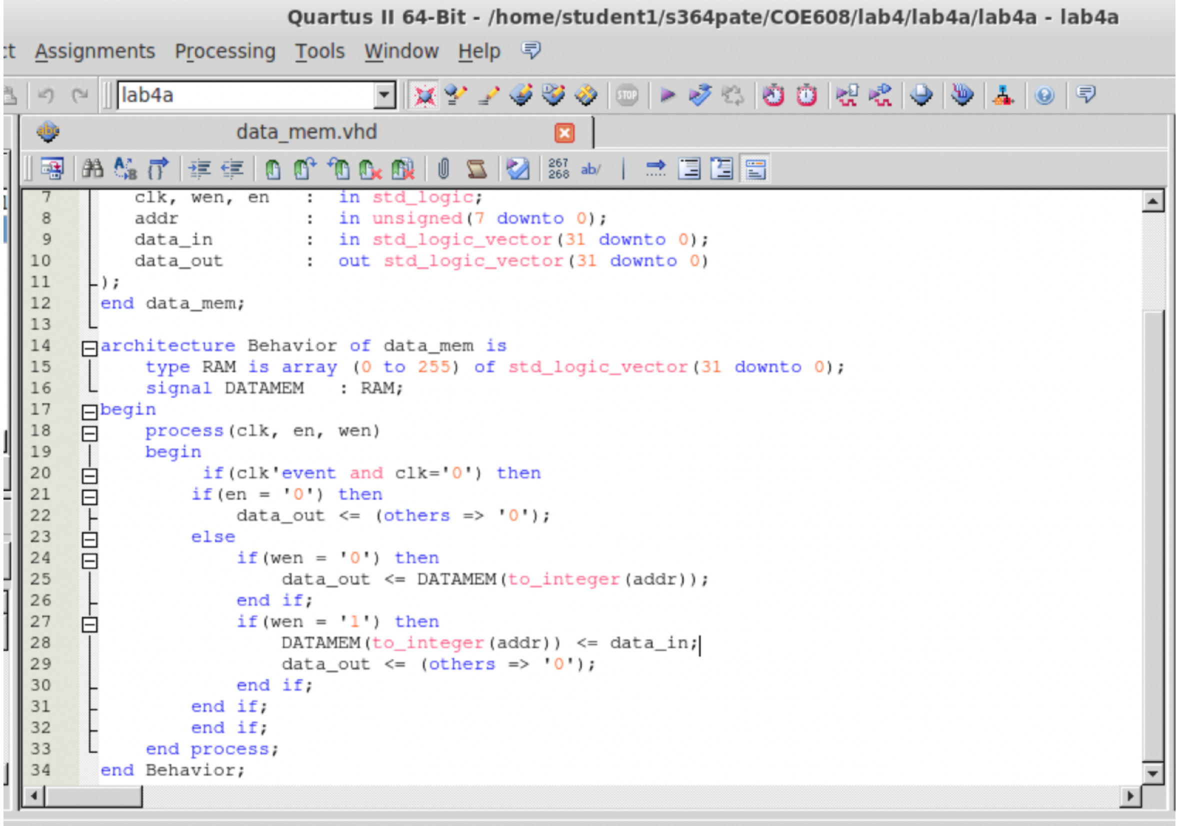

Quartus II 64-Bit - /home/student1/s364pate/COE608/lab4/lab4a/lab4a - lab4a
Assignments
Processing
Tools
Window
Help
lab4a
data_mem.vhd

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

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

architecture Behavior of data_mem is
    type RAM is array (0 to 255) of std_logic_vector(31 downto 0);
    signal DATAMEM    : RAM;
begin
    process(clk, en, wen)
    begin
         if(clk'event and clk='0') then
        if(en = '0') then
            data_out <= (others => '0');
        else
            if(wen = '0') then
                data_out <= DATAMEM(to_integer(addr));
            end if;
            if(wen = '1') then
                DATAMEM(to_integer(addr))  <= data_in;
```

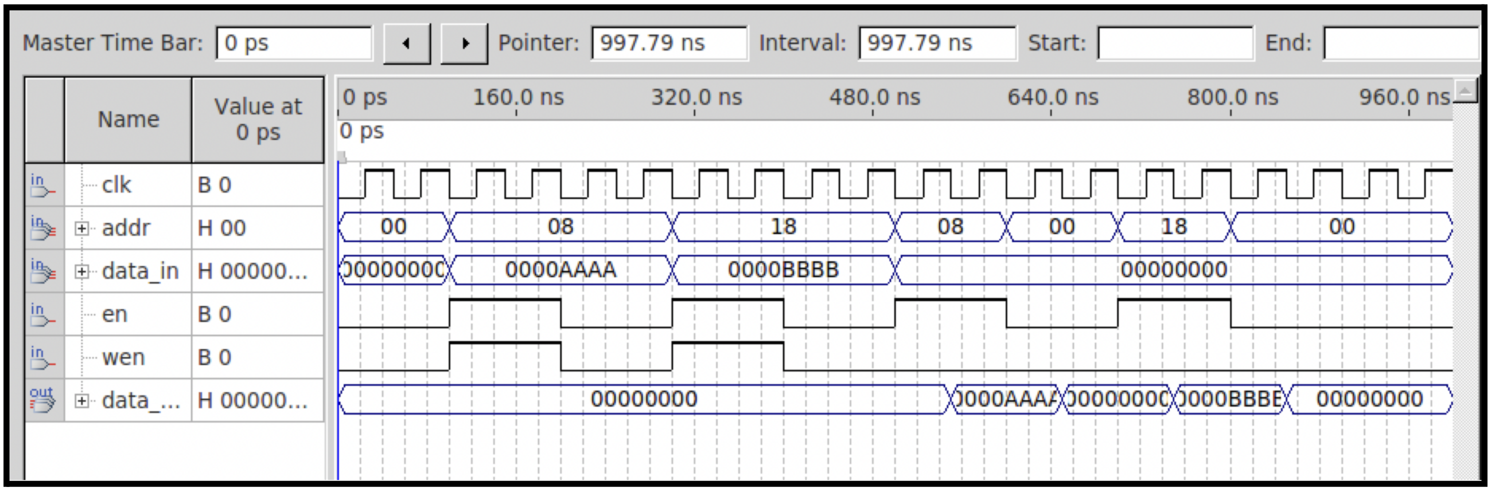
Data Memory Part 2


Quartus II 64-Bit - /home/student1/s364pate/COE608/lab4/lab4a/lab4a - lab4a
Assignments
Processing
Tools
Window
Help
lab4a
data_mem.vhd

```
    clk, wen, en    :  in std_logic;
    addr            :  in unsigned(7 downto 0);
    data_in         :  in std_logic_vector(31 downto 0);
    data_out        :  out std_logic_vector(31 downto 0)
);
end data_mem;

architecture Behavior of data_mem is
    type RAM is array (0 to 255) of std_logic_vector(31 downto 0);
    signal DATAMEM    : RAM;
begin
    process(clk, en, wen)
    begin
         if(clk'event and clk='0') then
        if(en = '0') then
            data_out <= (others => '0');
        else
            if(wen = '0') then
                data_out <= DATAMEM(to_integer(addr));
            end if;
            if(wen = '1') then
                DATAMEM(to_integer(addr))  <= data_in;
                data_out <= (others => '0');
            end if;
        end if;
        end if;
    end process;
end Behavior;
```

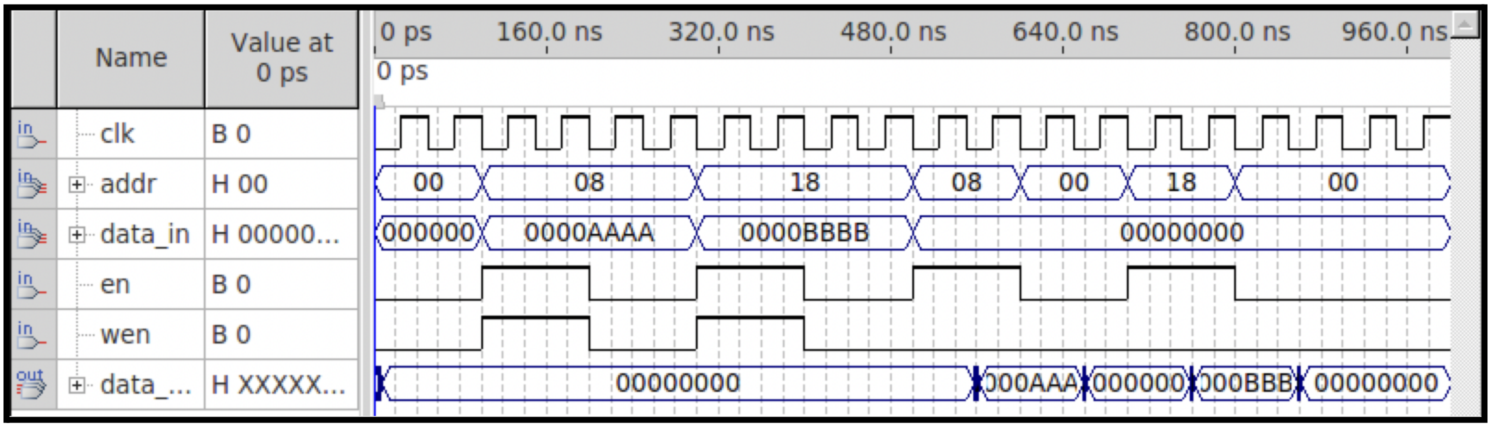
Functional Simulation



Master Time Bar: 0 ps Pointer: 997.79 ns Interval: 997.79 ns Start: End:

Name	Value at 0 ps
clk	B 0
addr	H 00
data_in	H 00000...
en	B 0
wen	B 0
data_...	H 00000...

Time Simulation



Name	Value at 0 ps
clk	B 0
addr	H 00
data_in	H 00000...
en	B 0
wen	B 0
data_...	H XXXXX...

Description of Input and Output Functionalities

The waveforms above indicate a visual representation of how the data memory unit works. To briefly summarize, the data memory unit consists of an 8-bit address input (addr), 32-bit data input and output (data_in and data_out), and other control lines such as clock, write enable (wen), and enable (en):

- The data memory unit consists of a memory array that stores data at specific memory locations, which are addressed by the addr input signal.
- The en signal is used to enable or disable the data memory unit. When en is high, the data memory unit is enabled and can perform read and write operations. When en is low, the data memory unit is disabled and the output is set to all zeros.
- The clk signal is used to synchronize the operations of the data memory unit. The unit only performs a read or write operation on the rising edge of the clk signal.
- The wen signal is used to specify the type of operation to be performed. When wen is high, a write operation is performed. The data on the data_in signal is written to the memory location specified by the addr signal. When wen is low, a read operation is performed. The data stored at the memory location specified by the addr signal is retrieved and sent to the data_out signal.
- The data_in signal is used to provide data to be written to the memory. The data_out signal is used to provide the data retrieved from the memory during a read operation.

When the “en” and “wen” signals are both set to '1', the data_out signal will display '0'. This is because the “wen” signal has been activated to store new data into the data_memory, so there is no output available at this point. On the other hand, when the “en” signal is '1' but the wen signal is '0', the data memory unit retrieves the stored data and outputs it. Therefore, you can observe that the input data values of "0000AAAA" and "0000BBBB" are outputted by the data_out signal when en is set to '1' and wen is set to '0'.

Explanation of the Delays:

You must also submit a short report (half page) describing the timing characteristics of your memory unit, meaning worst-case delays for reads and writes of various inputs. Students need to perform both functional and timing simulations.

The timing characteristics of a memory unit refer to the length of time it takes the unit to complete a read or write operation. Timing characteristics are affected by a variety of parameters, including clock frequency, memory capacity, data bus width, memory technology, and memory unit architecture. In general, memory units have different access times for reads and writes. The access time is the time it takes the memory unit to execute an operation once the input signals have stabilized. The access time comprises the propagation delay via the memory array as well as any extra delays introduced by the control circuitry.

The memory unit has a few timing characteristics to consider, based on the simulation results. The memory unit has a worst case delay of 7.064 ns, which occurs at the beginning of the simulation, the end of writing data, and the end of reading data. The delay at the start of the simulation is due to the time it takes to initialize the simulation. The delay at the end of data write is because the storage unit takes time to store the written data, and the delay at the end of data read is because the output must be stable after the data is retrieved . It is important to note that there is a short delay of 5.298 ns after reading/outputting the first input. This is because the data has already been retrieved and is ready for output. Similarly, the delay after reading/outputting the second input is also 7.064 ns. This is because the storage unit needs time to stabilize before retrieving data. Overall, the memory unit has few delays to consider, but the worst-case delay of 7.064ns occurs at the beginning and end of the simulation, as well as at the end of writing and reading data. These delays can affect the overall performance of the system and should be considered when designing any system that uses the storage device.