Objective of the Lab/Program. (2 points)

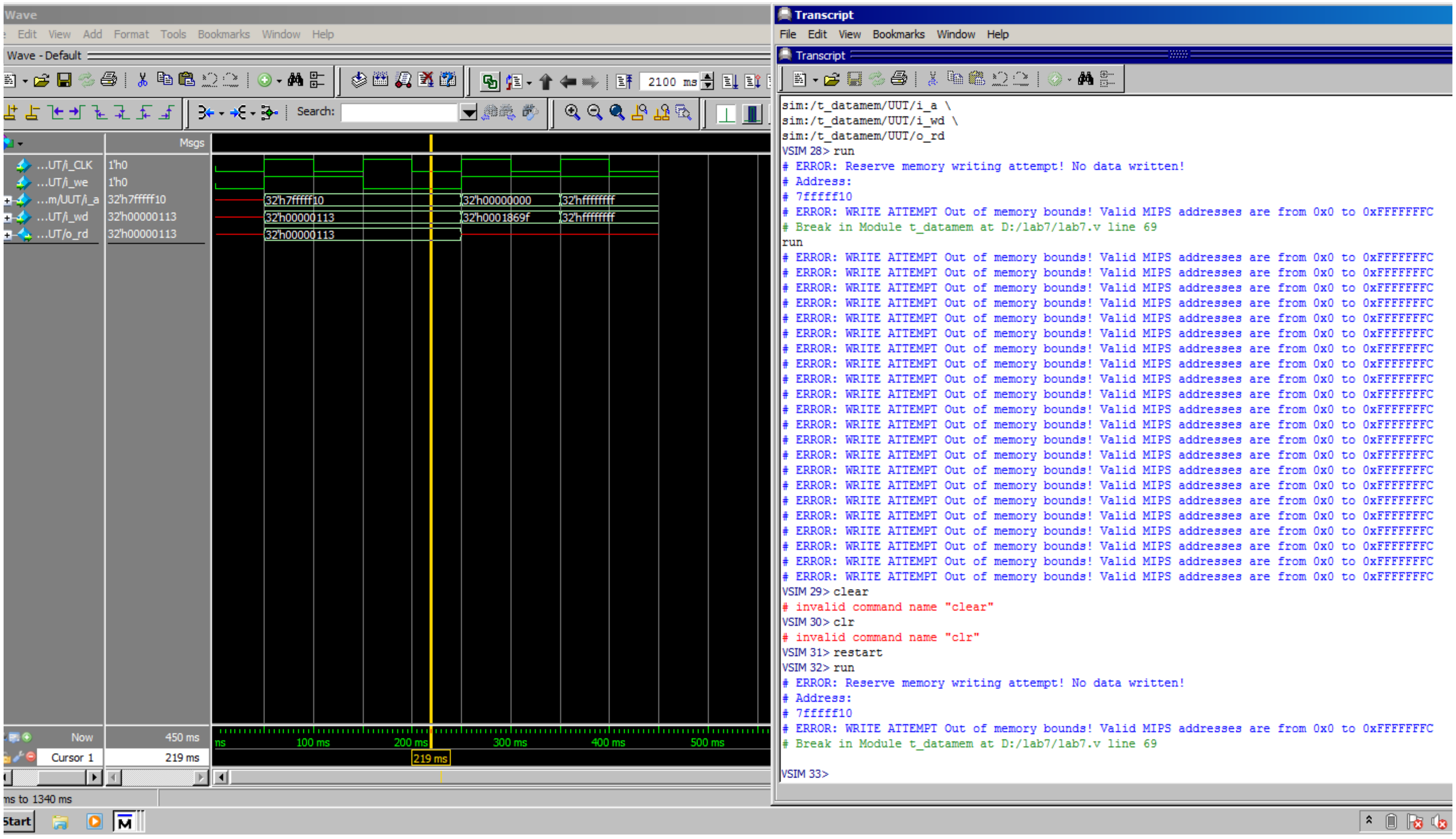
The objective of this lab is to demonstrate an HDL implementation of a characteristic element of the Von-Neumann architecture: the unified memory. This implementation does not feature any caching or virtual memory techniques, thus it is simple main memory. The implementation features memory privilege and out-of-bounds checks.

VHDL/Verilog Source Codes (8 points)

```
//Lab 7:
//Implement a data memory unit.
//Write a value into the memory location and read a value from the
memory location. Write always happens on the rising edge of the clock
and when WE (write enable) is set high. The data and address lines are
32 bits wide.
//Due 07/31/18: Demo: 08:00; Report: 23:59
module datamem
    (
        i_a, i_wd, i_we, i_CLK,
        o_rd
    );
    input i_we, i_CLK;
    input[31:0] i_a, i_wd;
    output[31:0] o_rd;

    reg [31:0] memory [2147483392:2147483644];
    //Read code
    assign o_rd = memory[i_a];

    //Write code
    always begin
        @(posedge i_CLK)
            if(i_a < 32'hfffffffc && i_a >= 32'h0)
            begin
                if(i_a > 32'h003ffffc && i_a < 32'h80000000)//checks for
reserved memory
                    if(i_we)
                        memory[i_a] <= i_wd;
                else
                begin
                        $display("ERROR: Reserve memory writing attempt!
No data written!\nAddress:");
                        $display("%h", i_a);
                end
            end
            else
                $display("ERROR: WRITE ATTEMPT Out of memory bounds!
Valid MIPS addresses are from 0x0 to 0xFFFFFFFC");
    end
endmodule
`timescale 1ms/1ms
```

```
module t_datamem;
    reg[31:0] t_i_a, t_i_wd;
    reg t_i_CLK = 1'b0;
    reg t_i_we = 1'b0;
    wire[31:0] t_o_rd;

    datamem UUT
    (
        .i_a(t_i_a),
        .i_wd(t_i_wd),
        .i_we(t_i_we),
        .i_CLK(t_i_CLK),
        .o_rd(t_o_rd)
    );
    always #50 t_i_CLK <= !t_i_CLK;

    initial begin
        #50
        t_i_we <= 1'b1;
        t_i_a <= 32'h7fffff10;
        t_i_wd <= 32'd275;
        #100
        t_i_we <= 1'b0;
        t_i_a <= 32'h7fffff10;
        #100
        t_i_we <= 1'b1;
        t_i_a <= 32'h0;
        t_i_wd <= 32'd99999;
        #100
        t_i_a <= 32'hffffffff;
        t_i_wd <= 32'hffffffff;
        #100
        $stop;
    end
endmodule
```

Screen shot of the simulation (waveforms). (8 points)



```
sim:/t_datamem/UUT/i_a \
sim:/t_datamem/UUT/i_wd \
sim:/t_datamem/UUT/o_rd
VSIM 28> run
# ERROR: Reserve memory writing attempt! No data written!
# Address:
# 7fffff10
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# Break in Module t_datamem at D:/lab7/lab7.v line 69
run
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
VSIM 29> clear
# invalid command name "clear"
VSIM 30> clr
# invalid command name "clr"
VSIM 31> restart
VSIM 32> run
# ERROR: Reserve memory writing attempt! No data written!
# Address:
# 7fffff10
# ERROR: WRITE ATTEMPT Out of memory bounds! Valid MIPS addresses are from 0x0 to 0xFFFFFFFC
# Break in Module t_datamem at D:/lab7/lab7.v line 69

VSIM 33>
```

Note: cropped to fit page. Please ignore lines above/before VSIM 32, as they are test messages.

Conclusion and References. (2 points)

Similar to the register file implementation from laboratory assignment 6, two values are written, then read sequentially (due to having only one data input and output). A protected and an out-of-bounds memory write is attempted. The memory behaves as described in the assignment file (not included). Therefore, it is possible to implement a unified memory with safety checks in a standardized HDL.

Dranishnikov, Peter A. "EEL4768CENGR01 Lab Assignment #06"
Patterson, David A. & Hennessy, John L. *Computer Organization and Design* 3rd edition
Harris, David Money & Harris, Sarah L. *Digital Design and Computer Architecture* 2nd edition