



Lab#7
Parametrized ROM

Digital design principles.

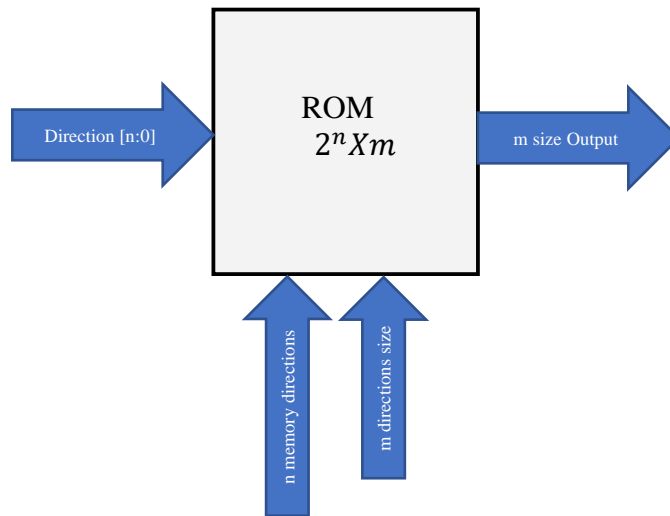
Teacher:
Gloria Castro Muñoz

Work made by:
Christian Aaron Ortega Blanco

Hermosillo, Son. 6 de mayo de 2022.

Instructions

Now that it's possible to read an external file acting like a ROM memory. It would be useful to make it a module that reads a memory of $n \times m$ size, where n is the number of directions on the memory and m is the length of the values on each direction.



Module definition

```
// -----  
// INAOE > ROM parametrized  
// -----  
// Author : christian Aaron Ortega Blanco  
// File  : design.sv  
// Create : 2022-05-06 15:18:53  
// Revise : 2022-05-06 15:18:53  
// Editor : sublime text4, tab size (2)  
// -----  
  
/* ----module----- datalong-> bits columns memlong-> number  
of rows*/  
module Chris      #(parameter datalong, memlong)(input  
[$clog2(memlong-1):0] Selection, output reg [datalong-1:0] char);  
  
    reg [datalong-1:0] Mem [0:memlong-1]; /// size of the memory  
  
    initial $readmemb("mem_c.txt",Mem);    /// direction of the  
memory  
    initial begin  
        assign char=Mem[Selection];        /// assign value of  
Mem[Selection] to the output  
    end  
  
endmodule
```

ROM content

```
10011100000000  
00001110000010  
00000001000010  
00000000010010  
00010001000001  
00011110000000  
00000000010010  
00011010000010  
00001010000010
```

Bench test

```
// -----  
// INAOE > ROM parametrized  
// -----  
// Author : christian Aaron Ortega Blanco  
// File : testbench.sv  
// Create : 2022-05-06 15:18:22  
// Revise : 2022-05-06 15:18:22  
// Editor : sublime text4, tab size (2)  
// -----  
  
`timescale 1ns/100ps  
  
module Chris_TB;  
  
    /* ----- dimension variables ----- */  
    parameter memlong =9; ///// mem localities size  
    parameter datalong =14; ///// word size  
  
    /* ----- vectorizing variables ----- */  
    reg [$clog2(memlong-1):0] Selection;  
    wire [datalong-1:0] char;  
    /* ----- assign module variables Chris #( data dimentions, mem  
    dimentions ) UUT(dir,DataOut); ----- */  
    Chris #( datalong,memlong ) UUT(Selection,char); /// christian  
    has 9 characters 14 bits per character  
  
    initial  
    begin  
        $dumpfile("Chris.vcd");  
        $dumpvars(1,Chris_TB);  
        $display("dir : valor");  
        #1;  
        for(int i=0 ; i<memlong; i++)begin ///// scaning all memory  
locations  
            Selection=i;  
            #1;  
            $display("%b %b",Selection,char);  
  
        end  
  
        $finish;  
    end  
endmodule
```

Results

To test this module we can make a testbench that display every location on the memory, we only have to provide the length of the word and the total of locations.

```
dir : valor
0000 10011100000000
0001 00001110000010
0010 00000001000010
0011 00000000010010
0100 00010001000001
0101 00011110000000
0110 00000000010010
0111 00011010000010
1000 00001010000010
```