**CDAC Feb 2015**

16 x 16 Synchronous Memory

Name: Bhrigu Bhargava

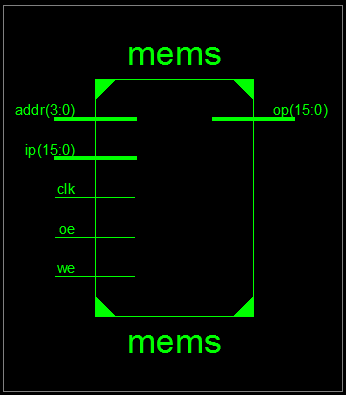
**16 x 16 Synchronous Memory**

# Design Approach:

A Ram memory is non-volatile which is used to store the data using the d flip-flops .The outputs of the each d flip-flop is concatenated and passed as a single input to the multiplexer and depending on the read address the output is read.

* We design 16 x 16 memory by using Behavior Modeling and Gate Level Modeling model .
* Here we design the 16 x 16 Ram memory using the component of d-flipflop and the component of ram memory .
* We need selection line for selecting row of 16 D-flip flop one by one ,by using input addr(4 bit ) and we(1 bit ) hence consider the decoder at input side to provide the enable circuitry for the memory elements.
* And a multiplexer at the output side to read the data. The selection line for the multiplexer will be the read address .

**Block Diagram:**

****

**Fig 1:- Block Diagram using example of 16 x 16 rom**

**Source Code:**

module memb (output reg [15:0] op, input [15:0] ip, input [3:0] addr, input we, oe,clk);

reg [15:0] mem [15:0];

always @ (posedge clk)

begin

if(oe==1'b1)

op=mem[addr];

else if(we==1'b1)

mem[addr]=ip;

end

endmodule

module mems (output [15:0] op, input [15:0] ip,input [3:0] addr, input we,oe,clk);

wire [15:0] en1;

wire [15:0] ip1,ip2,ip3,ip4,ip5,ip6,ip7,ip8,ip9,ip10,ip11,ip12,ip13,ip14,ip15,ip16,top;

deco\_4x16 deco1 (en1,we,addr);

dff2 d1 (ip1, ip,clk,en1[0]);

dff2 d2 (ip2, ip,clk,en1[1]);

dff2 d3 (ip3, ip,clk,en1[2]);

dff2 d4 (ip4, ip,clk,en1[3]);

dff2 d5 (ip5, ip,clk,en1[4]);

dff2 d6 (ip6, ip,clk,en1[5]);

dff2 d7 (ip7, ip,clk,en1[6]);

dff2 d8 (ip8, ip,clk,en1[7]);

dff2 d9 (ip9, ip,clk,en1[8]);

dff2 d10 (ip10, ip,clk,en1[9]);

dff2 d11 (ip11, ip,clk,en1[10]);

dff2 d12 (ip12, ip,clk,en1[11]);

dff2 d13 (ip13, ip,clk,en1[12]);

dff2 d14 (ip14, ip,clk,en1[13]);

dff2 d15 (ip15, ip,clk,en1[14]);

dff2 d16 (ip16, ip,clk,en1[15]);

mux16 m1 (top,ip1,ip2,ip3,ip4,ip5,ip6,ip7,ip8,ip9,ip10,ip11,ip12,ip13,ip14,ip15,ip16,addr);

mux2 m2 (op,top,oe);

endmodule

//---------------------DFF----------------------//

module dff2 (output reg [15:0] op, input [15:0] d, input clk,en);

always @ (posedge clk)

begin

if (en==1'b1)

op=d;

end

endmodule

//-------------------Mux 2:1-------------------------//

module mux2 (output reg [15:0] op,input [15:0] ip, input sel);

always @ (\*)

begin

case (sel)

0: ;

1: op=ip;

endcase

end

endmodule

//-------------------Mux 16:1-------------------------//

module mux16 (output reg [15:0] op,input [15:0] ip1,ip2,ip3,ip4,ip5,ip6,ip7,ip8,ip9,ip10,ip11,ip12,ip13,ip14,ip15,ip16, input [3:0] sel);

always @ (\*)

begin

case (sel)

0: op=ip1;

1: op=ip2;

2: op=ip3;

3: op=ip4;

4: op=ip5;

5: op=ip6;

6: op=ip7;

7: op=ip8;

8: op=ip9;

9: op=ip10;

10: op=ip11;

11: op=ip12;

12: op=ip13;

13: op=ip14;

14: op=ip15;

15: op=ip16;

endcase

end

endmodule

//-------------------Decoder 4:16---------------------//

module deco\_4x16 (d,en,ip);

input [3:0] ip;

output [15:0] d;

input en;

wire [4:0] t2;

wire [15:0] t1;

assign t1=16'd1;

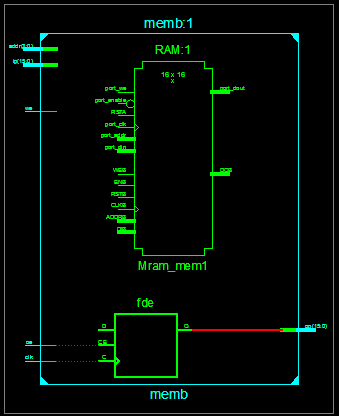
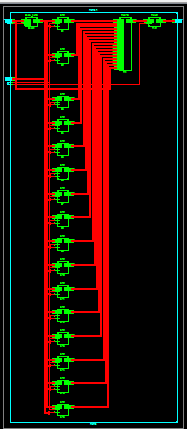
assign t2={en,ip};

assign d=t1<<t2-16;

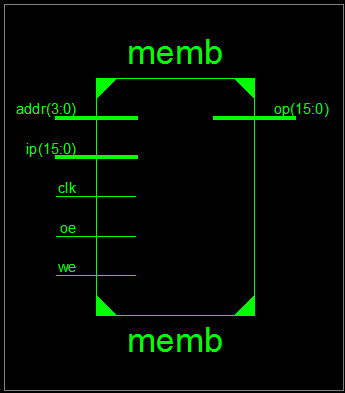
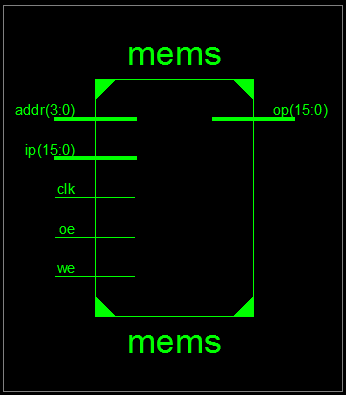
endmodule

**Synthesis:**

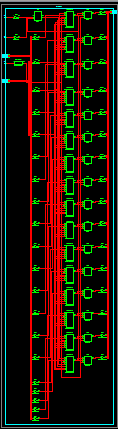
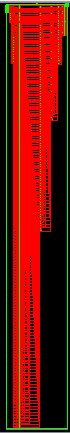
1. RTL Schematic



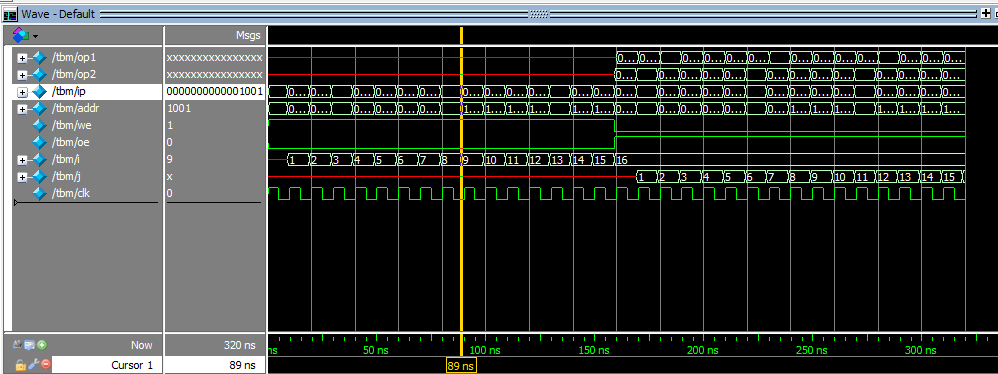
1. Block Diagram

****

1. Tech Schematic



1. Simulation Waveform Result

****

**Error:**

Corrected typo in code

**Verified by:**

Dharamvir Chundawat (150240133007)