**Dual Port RAM:**

It is a Random Access Memory which has 2 ports and can write the information on same time and read the information simultaneously.

**Verilog Code for Dual Port Ram:**

**`timescale 1ns / 1ps**

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

// Create Date: 04.10.2023 00:11:20

// Design Name:

// Module Name: dualportram

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

// Dependencies:

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//////////////////////////////////////////////////////////////////////////////////

module Dualportram(input [7:0]a,

input [7:0]b,

input clk,

input [2:0] addrs\_a,

input [2:0] addrs\_b,

input wra,rda,wrb,rdb,

output [7:0] out\_a,

output [7:0] out\_b);

dualportram dualportram(.a(a),.wra(wra),.rda(rda),.out\_a(out\_a),.addrs\_a(addrs\_a),.b(b),.wrb(wrb),.rdb(rdb),.addrs\_b(addrs\_b),.clk(clk),.out\_b(out\_b));

endmodule

module dualportram(input [7:0]a,

input [7:0]b,

input clk,

input [2:0] addrs\_a,

input [2:0] addrs\_b,

input wra,rda,wrb,rdb,

output reg [7:0] out\_a,

output reg [7:0] out\_b);

reg [7:0] ram1[7:0];

reg [7:0] ram2[7:0];

always @(posedge clk)

begin

if(rda)

ram1[addrs\_a]<= a;

if(wra)

out\_a=ram1[addrs\_a];

if(rdb)

ram2[addrs\_b]<=b;

if(wrb)

out\_b<=ram2[addrs\_b];

end

endmodule

**Test Bench Code for Dual Port Ram:**

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

// Create Date: 03.10.2023 22:30:09

// Design Name:

// Module Name: singleportram\_tb

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

// Dependencies:

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module dualportram\_tb;

reg clk;

reg [7:0] a;

reg [7:0] b;

reg [2:0]addrs\_a;

reg [2:0]addrs\_b;

reg wra,rda,wrb,rdb;

wire [7:0] out\_a;

wire [7:0] out\_b;

//Dut

dualportram dualportram(.a(a),.wra(wra),.rda(rda),.out\_a(out\_a),.addrs\_a(addrs\_a),.b(b),.wrb(wrb),.rdb(rdb),.addrs\_b(addrs\_b),.clk(clk),.out\_b(out\_b));

initial

begin

clk=1'b0;

forever #5 clk=~clk;

end

initial

begin

a=8'b00011000;

b=8'b00010100;

//read operation for a

#10 rda=1;

rdb=1;#10

//read operation for b

rda=1;

rdb=1;

//write operation for Port A

#10 wra=1'b1;addrs\_a=3'b000;

#10 wra=1'b1;addrs\_a=3'b001;

#10 wra=1'b1;addrs\_a=3'b010;

#10 wra=1'b1;addrs\_a=3'b011;

#10 wra=1'b1;addrs\_a=3'b100;

#10 wra=1'b1;addrs\_a=3'b101;

#10 wra=1'b1;addrs\_a=3'b110;

#10 wra=1'b1;addrs\_a=3'b111;

//write operation for Port B

#10 wrb=1'b1; addrs\_b=3'b000;

#10 wrb=1'b1; addrs\_b=3'b001;

#10 wrb=1'b1; addrs\_b=3'b010;

#10 wrb=1'b1; addrs\_b=3'b011;

#10 wrb=1'b1; addrs\_b=3'b100;

#10 wrb=1'b1; addrs\_b=3'b101;

#10 wrb=1'b1; addrs\_b=3'b101;

#10 wrb=1'b1; addrs\_b=3'b110;

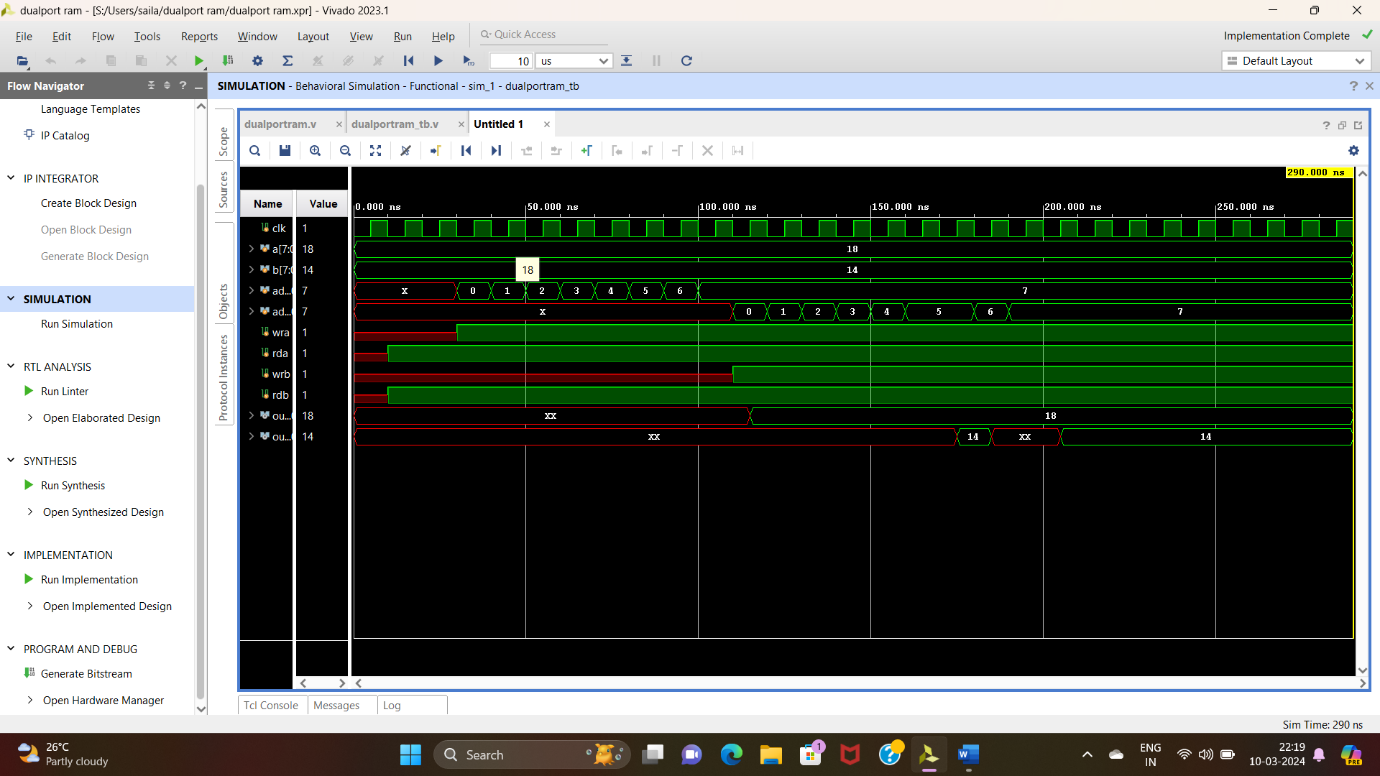
#10 wrb=1'b1; addrs\_b=3'b111;

#100 $finish;

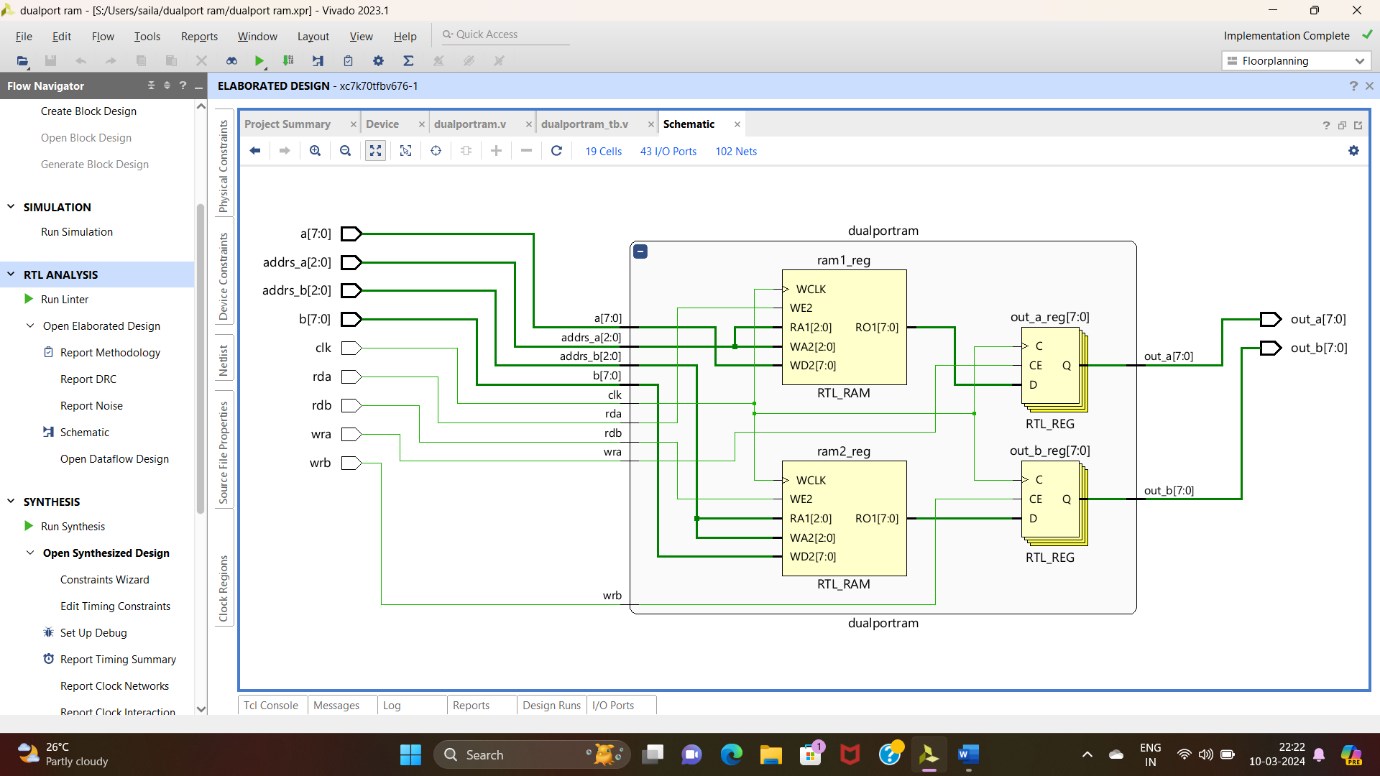
end

endmodule

**Simulation Waveform:**

****

**Schematic Diagram:**

****