CS201P MEMORY

B20218 Narmit Kumar

SRAM_16x4 code:

SRAM_16x4.v

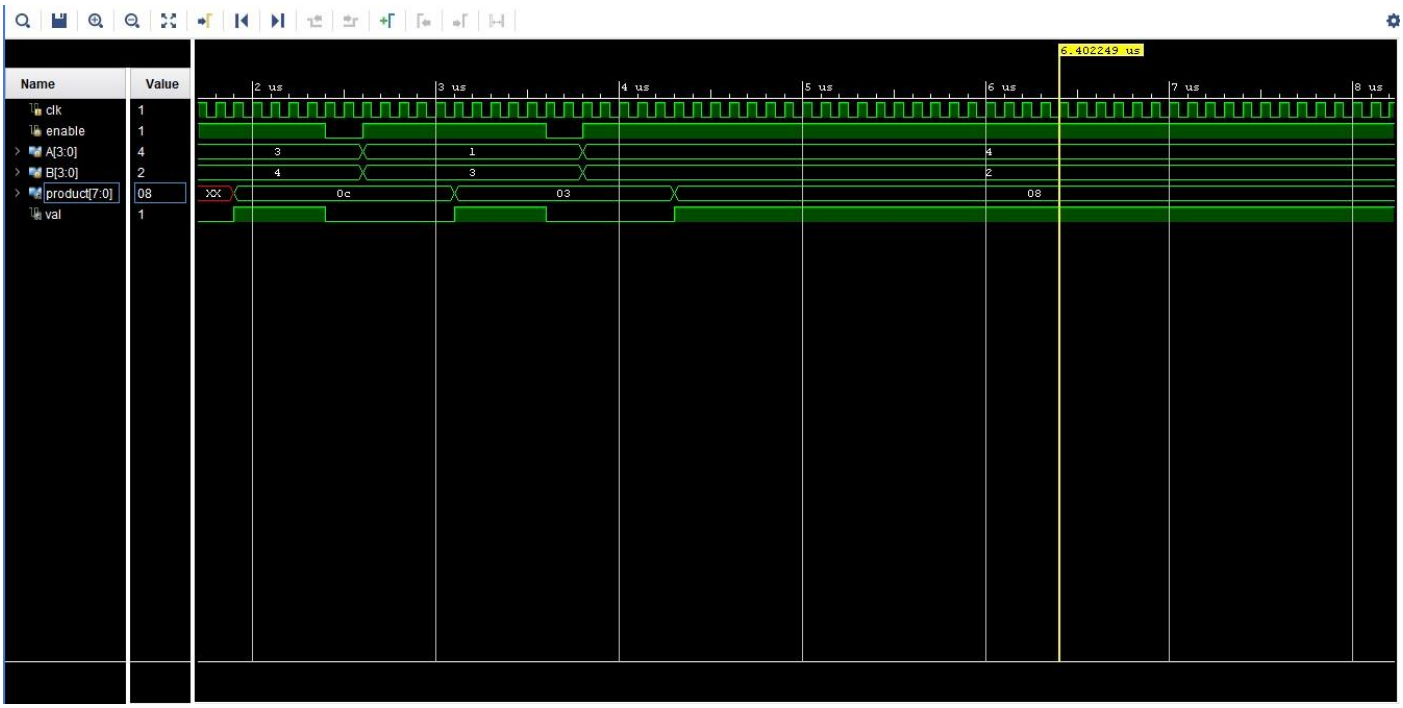
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////...
module SRAM_16x4(
    input clk, input rst, input read, input write, input [3:0] wr_data, input [3:0] write_addr, read_addr,
    output reg [3:0] rd_data,output reg wr_done,rd_done
    );
    parameter ADDR_WIDTH = 4 ; parameter mem_size = 15 ; parameter DATA_WIDTH = 3 ;
    // creates a array of size 16 X 4, means 16 locations each to store a 4-bit data
    reg [DATA_WIDTH:0] temp_mem[0:mem_size];
    always @(posedge clk)
    begin
        if(rst)
            begin
            wr_done<=0;
            rd_done<=0;
            end
        else
        begin
            if(write)
                begin
                temp_mem[write_addr] <= wr_data;
                wr_done <=1;
            end
            else
                begin
                wr_done<=0;
            end
        if(read)
            begin
            rd_data <= temp_mem[read_addr];
            rd_done <= 1;
        end
        else
            rd_done<=0;
        end
    end
endmodule
```

SRAM_16x4 testbench:

test_SRAM_16x4.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////...
module test_SRAM_16x4;

reg clk, rst,read,write;
reg [3:0] wr_data;
reg [3:0] write_addr, read_addr;
wire [3:0] rd_data;
wire wr_done,rd_done;

SRAM_16x4 uut(.clk(clk),
.rst(rst),
.read(read),
.write(write),
.wr_data(wr_data),
.write_addr(write_addr),
.read_addr(read_addr),
.rd_data(rd_data),
.wr_done(wr_done),
.rd_done(rd_done)
);

always #5 clk = !clk;
initial begin
    clk = 1'b0;
    rst=0;
    read= 0;
    write = 1;
    wr_data=4'b0101;
    write_addr=4'd3;
    read_addr=4'd0;
    #100;
    #10;

    rst=0;
    read= 1;
    write = 0;
     write = 0;
     wr_data=0;
     write_addr=0;
     read_addr=4'd3;
     #100;
     #10;

     rst=0;
     read= 0;
     write = 1;
     wr_data=4'b0100;
     write_addr=4'd3;
     read_addr=4'd0;

     #100;
     #10;

      rst=0;
      read= 1;
      write = 0;
      wr_data=0;
      write_addr=0;
      read_addr=4'd3;



end
endmodule
```

Output :


Name
Value
clk
1
enable
1
A[3:0]
4
B[3:0]
2
product[7:0]
08
val
1
6.402249 us
2 us
3 us
4 us
5 us
6 us
7 us
8 us
3
1
4
4
3
2
xx
0c
03
08