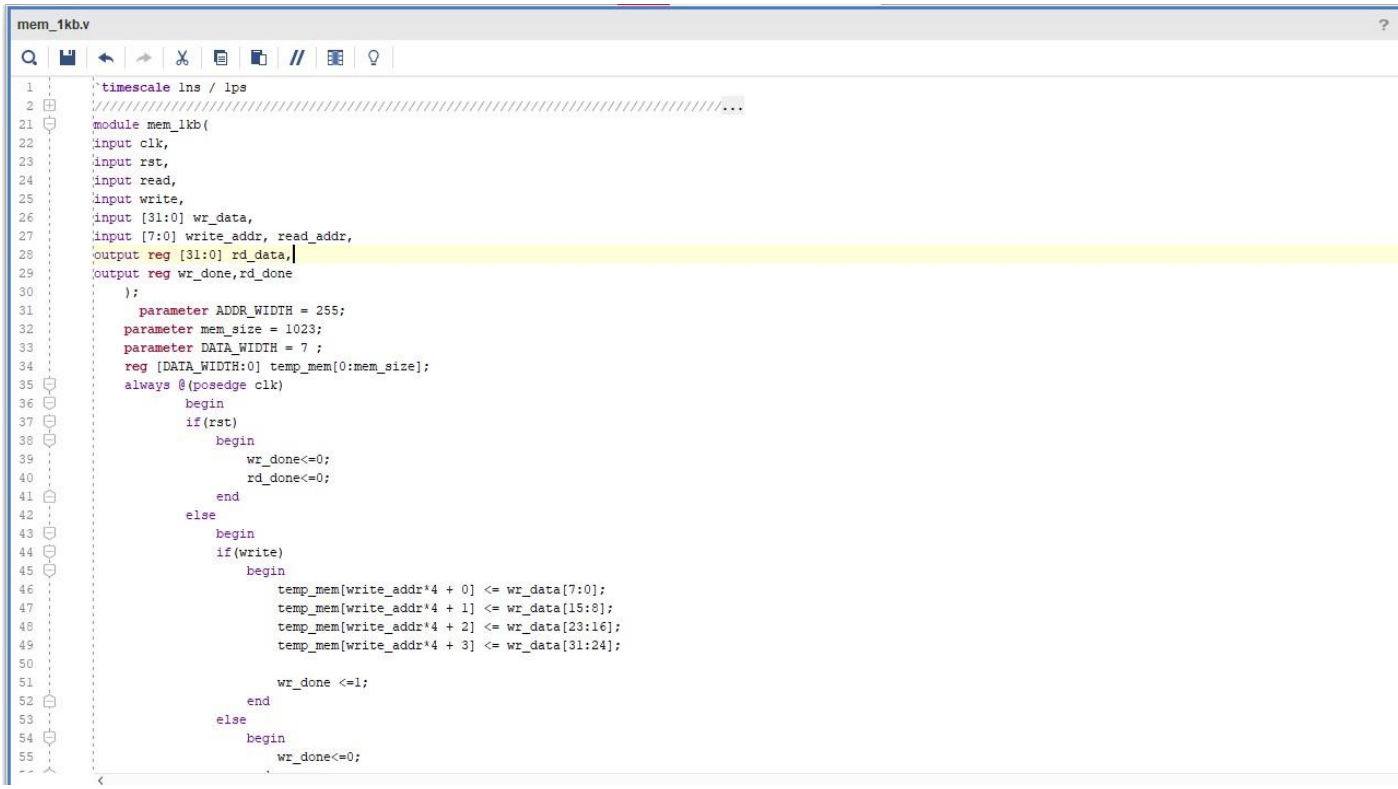
CS201P MEMORY

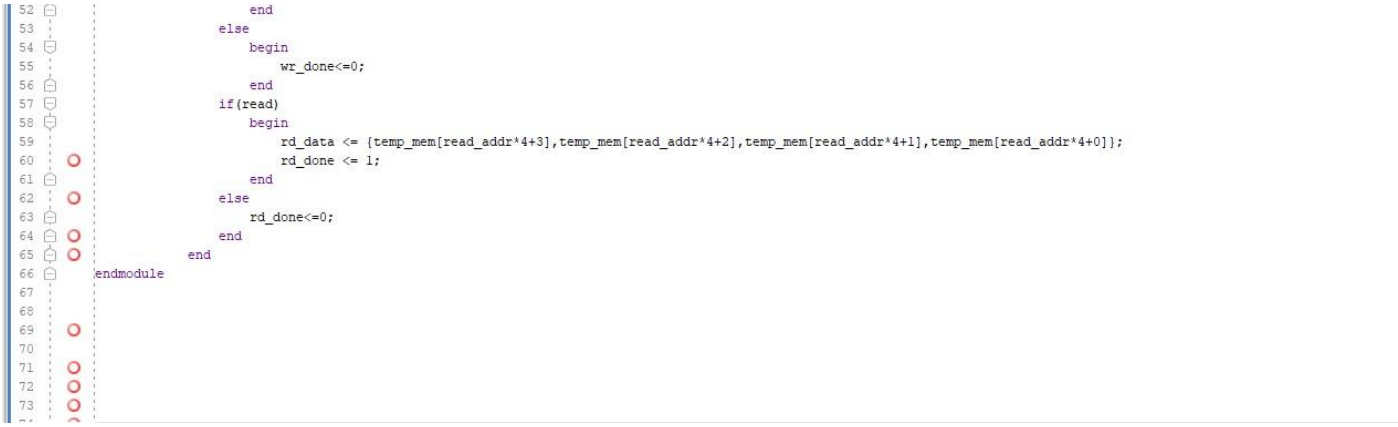
B20218 Narmit Kumar

1kb Memory code:



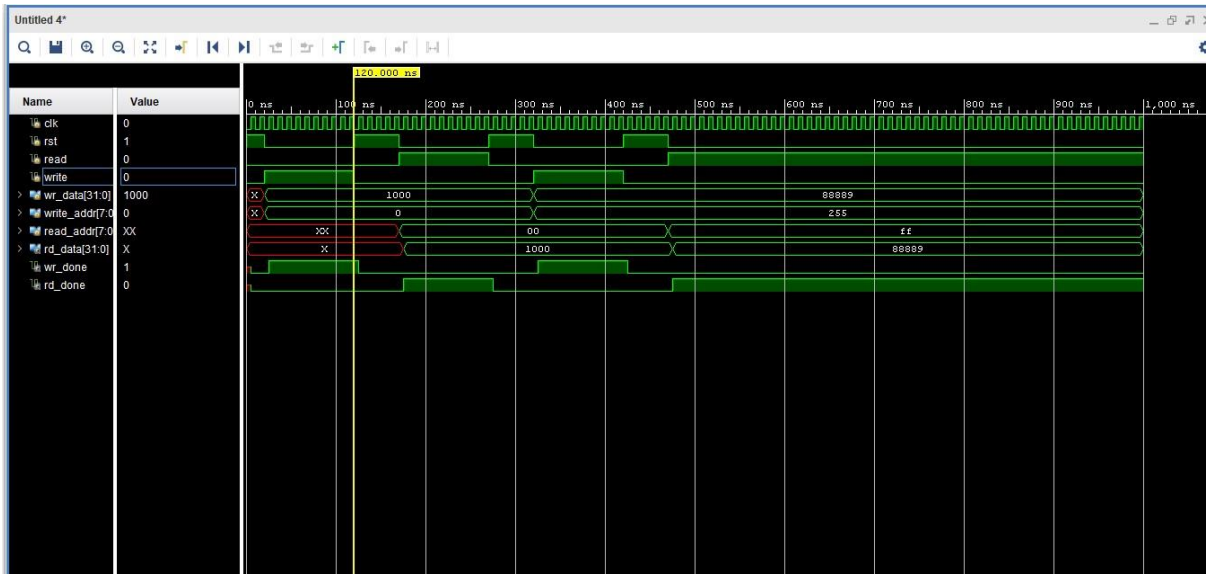
mem_1kb.v

```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////...
21  module mem_1kb(
22  input clk,
23  input rst,
24  input read,
25  input write,
26  input [31:0] wr_data,
27  input [7:0] write_addr, read_addr,
28  output reg [31:0] rd_data,
29  output reg wr_done,rd_done
30      );
31        parameter ADDR_WIDTH = 255;
32      parameter mem_size = 1023;
33      parameter DATA_WIDTH = 7 ;
34      reg [DATA_WIDTH:0] temp_mem[0:mem_size];
35      always @(posedge clk)
36              begin
37              if(rst)
38                  begin
39                      wr_done<=0;
40                      rd_done<=0;
41                  end
42              else
43                  begin
44                  if(write)
45                      begin
46                          temp_mem[write_addr*4 + 0] <= wr_data[7:0];
47                          temp_mem[write_addr*4 + 1] <= wr_data[15:8];
48                          temp_mem[write_addr*4 + 2] <= wr_data[23:16];
49                          temp_mem[write_addr*4 + 3] <= wr_data[31:24];
50
51                          wr_done <=1;
52                      end
53                  else
54                      begin
55                          wr_done<=0;
```



```
52                      end
53                  else
54                      begin
55                          wr_done<=0;
56                      end
57                  if(read)
58                      begin
59                          rd_data <= {temp_mem[read_addr*4+3],temp_mem[read_addr*4+2],temp_mem[read_addr*4+1],temp_mem[read_addr*4+0]};
60                          rd_done <= 1;
61                      end
62                  else
63                      rd_done<=0;
64                  end
65              end
66  endmodule
67
68
69
70
71
72
73
```

Output:



3rd

Question

Testbench

:

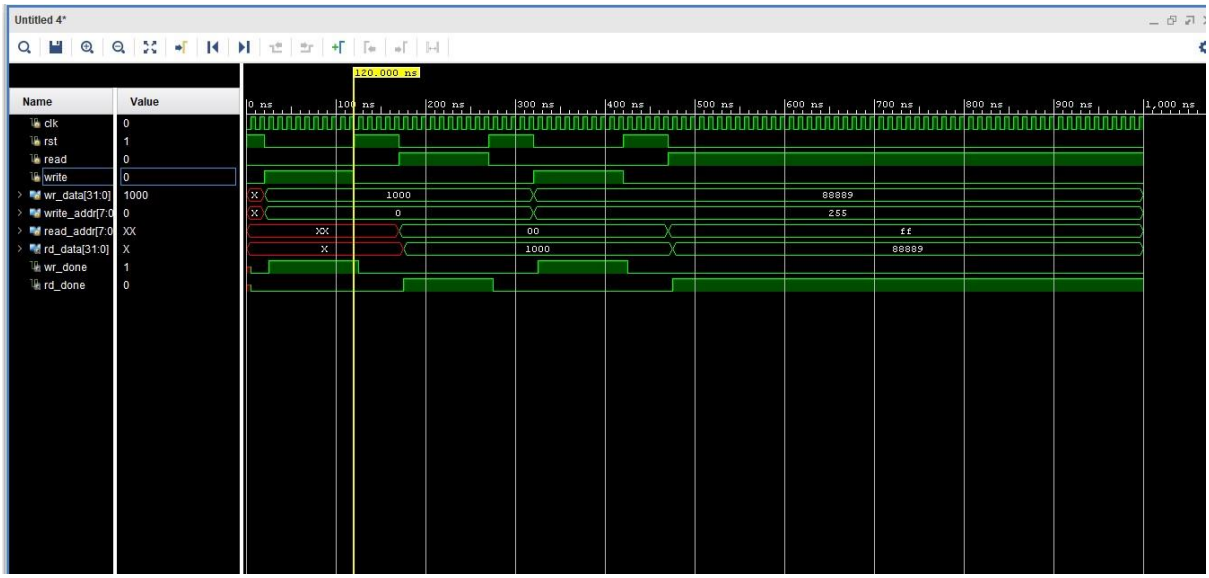
test_mem1kb.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////...
module test_mem1kb();

 reg clk, rst, read, write;
 reg [31:0] wr_data;
 reg [7:0] write_addr, read_addr;


 wire [31:0] rd_data;
 wire wr_done, rd_done;

 mem_1kb 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 = 1'b1;
read = 1'b0;
write = 1'b0;

#20;
           // write_addr and read_addr can take value from 0 to 255 8bit data
           // write data should be 32 bit
```

test_mem1kb.v

```
 rst = 1'b0;
 write = 1'b1;
 write_addr = 8'd0;
 wr_data = 32'd1000;

 #100;

 rst = 1'b1;
 write = 1'b0;
 #50;

 rst = 1'b0;
 read = 1'b1;
 read_addr = 8'd0;

 #100;

  rst = 1'b1;
  read = 1'b0;
  #50;

  rst = 1'b0;
  write = 1'b1;
  write_addr = 8'd255;
  wr_data = 32'd88889;
  #100;

  rst = 1'b1;
  write = 1'b0;
  #50;

   rst = 1'b0;
   read = 1'b1;
   read_addr = 8'd255;

  end
endmodule
```

Output:


Untitled 4*
120.000 ns
Name
Value
clk
0
rst
1
read
0
write
0
wr_data[31:0]
1000
write_addr[7:0]
0
read_addr[7:0]
XX
rd_data[31:0]
X
wr_done
1
rd_done
0
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1,000 ns
1000
88889
255
00
ff