DAY-50

#100DAYSOFRTL

Aim:- Implementation of 32-bit RAM using Verilog.

RTL CODE:-

```
////Day-50
////DATE:-19/02/2024
////32-BIT RAM
module Day_50(input clk,wr, rd,rst,
input [31:0] data_in,
input [4:0] addr,
output  [31:0]  data_out);
reg [31:0] ram [0:31];
always @(posedge clk) begin
if(rst) begin
ram[addr]<=32'd0;
end
else begin
if(wr)
ram[addr]<=data_in;
end
end
assign data_out=rd?ram[addr]:32'dz;
endmodule

```

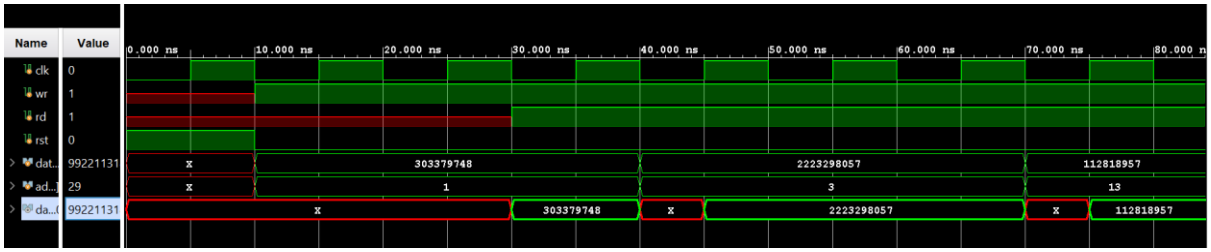
OUTPUT:-

```
wr=1,data_in= 303379748,addr= 1
rd=1,addr= 1,data_out= 303379748
wr=1,data_in=2223298057,addr= 3
rd=1,addr= 3,data_out=2223298057
wr=1,data_in= 112818957,addr=13
rd=1,addr=13,data_out= 112818957
wr=1,data_in=2999092325,addr=18
rd=1,addr=18,data_out=2999092325
wr=1,data_in=  15983361,addr=13
rd=1,addr=13,data_out=  15983361
wr=1,data_in= 992211318,addr=29
rd=1,addr=29,data_out= 992211318
wr=1,data_in=1993627629,addr=12
rd=1,addr=12,data_out=1993627629
wr=1,data_in=2097015289,addr= 6
rd=1,addr= 6,data_out=2097015289
wr=1,data_in=3807872197,addr=10
rd=1,addr=10,data_out=3807872197
```

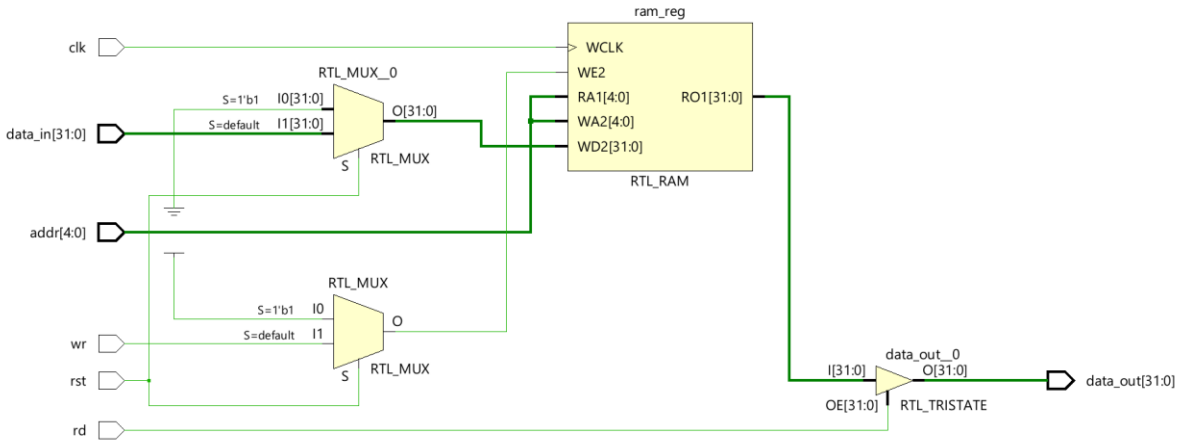
TESTBENCH:-

```
1  module Day_50_tb();
2  reg clk,wr,rd,rst;
3  reg [31:0] data_in;
4  reg [4:0] addr;
5  wire [31:0] data_out;
6  Day_50 uut(clk,wr,rd,rst,data_in,addr,data_out);
7  always #5 clk=~clk;
8  initial clk=0;
9  initial begin
10 rst=1; #10;
11 rst=0;
12 for(int i=0; i<15; i=i+1) begin
13 wr=1'b1;
14 data_in=$random();
15 addr=$random();
16 #10;
17 $display("wr=%b,data_in=%d,addr=%d",wr,data_in,addr);
18 #10;
19 rd=1'b1;
20 #10;
21 $display("rd=%b,addr=%d,data_out=%d",wr,addr,data_out);
22 end
23 end
24 initial begin
25 #400;
26 $finish();
27 end
28 endmodule
29
```

WAVEFORMS:-



SCHEMATIC:-


ram_reg
clk
WCLK
WE2
RA1[4:0]
RO1[31:0]
WA2[4:0]
WD2[31:0]
RTL_RAM
RTL_MUX__0
S=1'b1
I0[31:0]
S=default
I1[31:0]
O[31:0]
S
RTL_MUX
data_in[31:0]
addr[4:0]
RTL_MUX
S=1'b1
I0
S=default
I1
O
S
RTL_MUX
wr
rst
rd
data_out__0
I[31:0]
O[31:0]
OE[31:0]
RTL_TRISTATE
data_out[31:0]