14th Day

Synchronous FIFO

Design

```
module sync_fifo(input clk,rst,wr,rd,[7:0]din,output reg [7:0]dout,output full,empty );


reg [7:0]mem[0:8];


reg [2:0]wr_ptr;

reg [2:0]rd_ptr;


always @(posedge clk)

begin

  if(rst)

  begin

  dout<=0;

  wr_ptr<=0;

  rd_ptr<=0;

  for(integer i=0;i<8;i=i+1)

    mem[i]<=0;

  end


  else

  begin

    if((wr==1'b1) && (full==1'b0))

    begin

    mem[wr_ptr]<=din;

    wr_ptr<=wr_ptr+1;

    end
```

```
        else if((rd==1'b1)&&(empty==1'b0))
        begin
        dout<=mem[rd_ptr];
        rd_ptr<=rd_ptr+1;
        end
     end

 end

        assign empty=((wr_ptr-rd_ptr)==0)?1'b1:1'b0;
        assign full=((wr_ptr-rd_ptr)==7)?1'b1:1'b0;

endmodule
```

Testbench

```
module sync_fifo_tb();

reg clk,rst,wr,rd;
reg [7:0]din;
wire full,empty;
wire [7:0]dout;

sync_fifo dut(clk,rst,wr,rd,din,dout,full,empty);

always #5 clk=~clk;

task clk_rst;
    begin
    clk=1'b0;
    rst=1'b1;

    #15;

    rst=1'b0;

    end

endtask


task wr_stimulus;

    begin

    wr=1'b1;

    repeat(7)

    begin

    din=$random;

    #10;

    end

    wr=1'b0;

    end

endtask


task rd_stimulus;

     begin

     rd=1'b1;

     #70;

     rd=1'b0;

     end

endtask


initial

begin

clk_rst;

wr_stimulus;
```

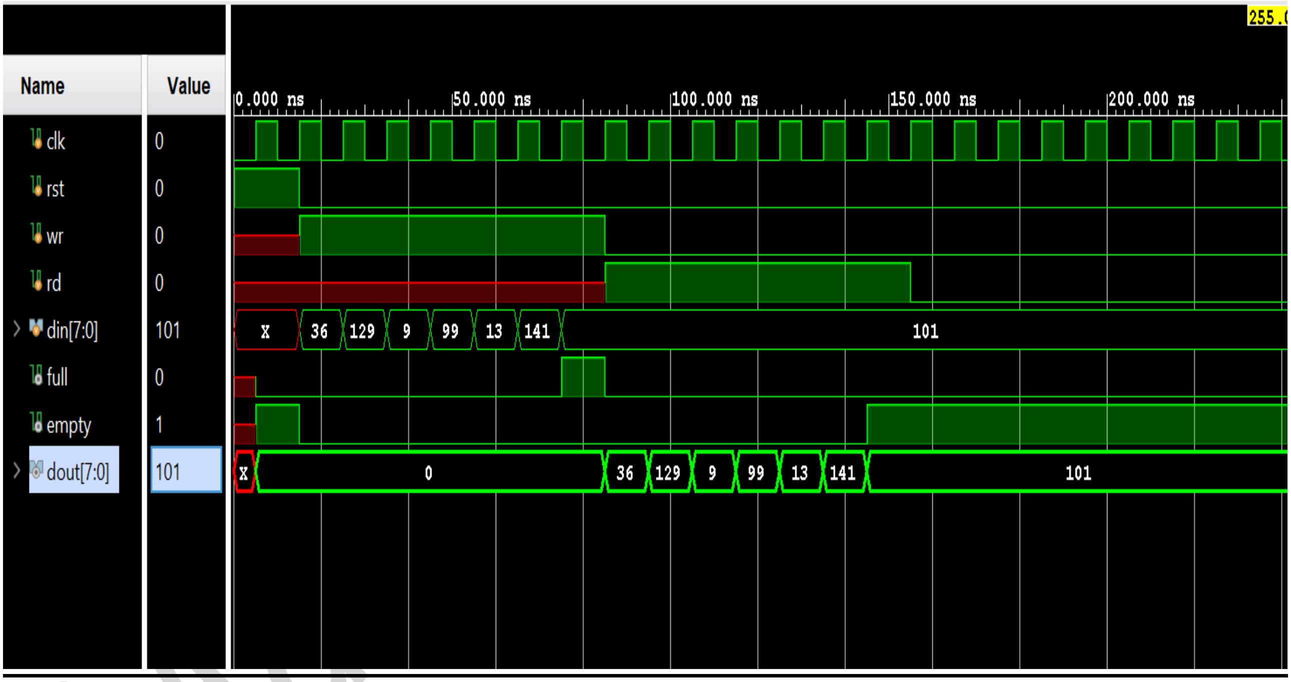
```
rd_stimulus;

#100;$finish;

end


endmodule
```

Simulation


255.(
Name
Value
0.000 ns
50.000 ns
100.000 ns
150.000 ns
200.000 ns
clk 0
rst 0
wr 0
rd 0
din[7:0] 101
X 36 129 9 99 13 141 101
full 0
empty 1
dout[7:0] 101
X 0 36 129 9 99 13 141 101