DESIGN AND SIMULATION OF AN 8-BIT RAM

By:-
KOTHAPALLI MAHITH VATHSAV

APRIL 26 , 2025

KOTHAPALLI MAHITH VATHSAV

Objective:-

The main objective is to design a synchronous 8-bit Random Access Memory (RAM) module using Verilog HDL with the ability to reset, write data to multiple memory locations, and read data conditionally, simulating a real-time data storage and retrieval system at the RTL level.

Theory:-

RAM is one of the fundamental components in any digital system or processor. It is used for storing data temporarily during the execution of tasks. Understanding how RAM works and implementing it at the register-transfer level (RTL) is crucial in the VLSI industry, especially in memory design, system-on-chip (SoC) integration, and FPGA/ASIC prototyping.

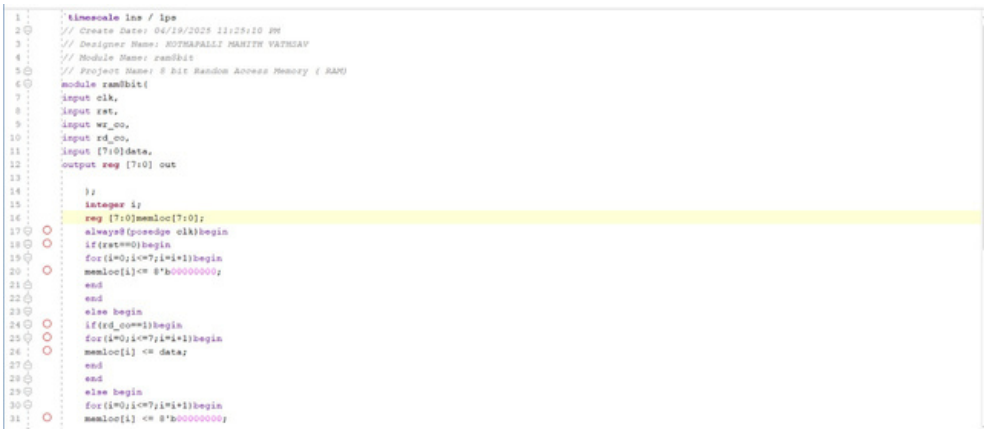
This project was undertaken to build a simple yet scalable memory block using behavioral modeling in Verilog, aiming to reinforce concepts like synchronous operation, control signals, arrayed memory registers, and simulation-driven data flow.

Ports	Direction/width	Description
clk	input(1 bit)	Clock signal it is provide the synchronous operation
rst	input(1 bit)	if rst pin is low initially the memory will be filled with binary 0 and then if it is high the data will be loaded to the memory or else it will write the data from the memory
wr_co	input(1 bit)	control signal to write operation
rd_co	input(1 bit)	control signal to the read operation
data	input(8 bit)	data that need to be loaded
out	output (8 bit)	output data

Internal Architecture:-

- Memory Array:-
- A register array of 8 elements, each 8 bits wide (reg [7:0] memloc[0:7];), acts as the RAM storage.
- Control Logic:
- If rst is high, the memory is cleared (memloc[i] <= 8'b0).
- If rd_co is high (treated as write in this design), all memory locations are updated with the same input data.
- If wr_co is high, data from a specific location (e.g., memloc[7]) is output.

Design part:-



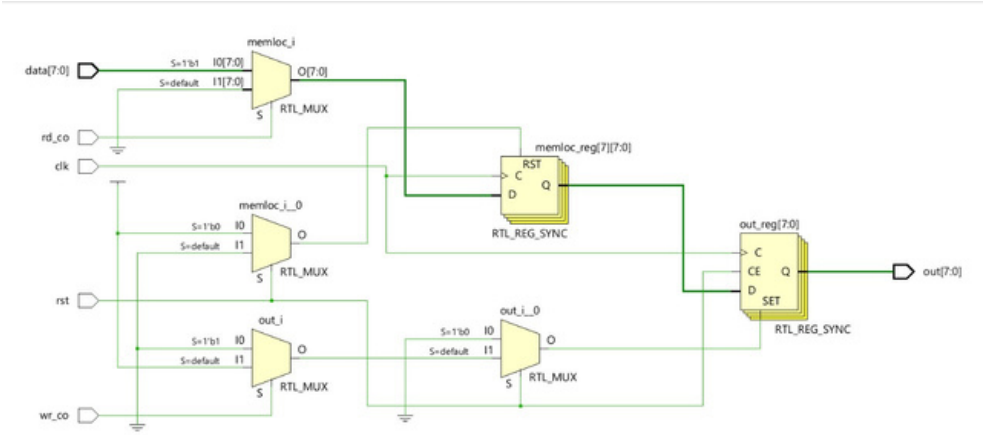
```
timescale 1ns / 1ps
// Create Date: 04/19/2025 11:25:10 PM
// Designer Name: KOTHAPALLI MAHITH VATHSAV
// Module Name: ram8bit
// Project Name: 8 bit Random Access Memory ( RAM)
module ram8bit(
input clk,
input rst,
input wr_co,
input rd_co,
input [7:0]data,
output reg [7:0] out

    );
    integer i;
    reg [7:0]memloc[7:0];
    always@(posedge clk)begin
    if(rst==0)begin
    for(i=0;i<=7;i=i+1)begin
    memloc[i]<= 8'b00000000;
    end
    end
    else begin
    if(rd_co==1)begin
    for(i=0;i<=7;i=i+1)begin
    memloc[i] <= data;
    end
    end
    else begin
    for(i=0;i<=7;i=i+1)begin
    memloc[i] <= 8'b00000000;
```

```
    if(wr_co==1)begin
    for(i=0;i<=7;i=i+1)begin
    out<= memloc[i];
    end
    end
    else begin
    out<= 8'b11111111;
    end
    end
    end
```

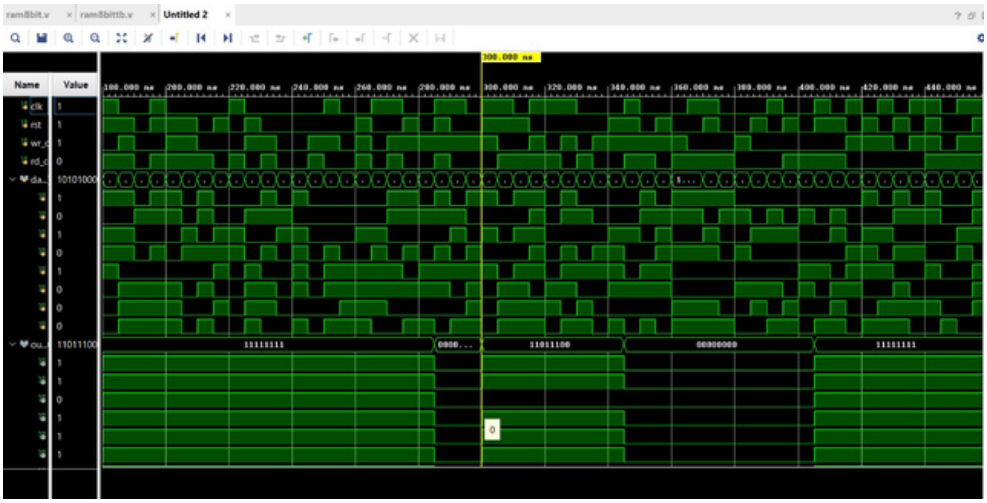
Testbench part:-

```
`timescale 1ns / 1ps
// Create Date: 04/20/2025 12:51:35 AM
// Designer Name: KOTHAPALLI MAHITH VATHSAV
// Module Name: ram8bittb
// Project Name: 8 BIT RAM
module ram8bittb;
reg clk;
reg rst;
reg wr_co;
reg rd_co;
reg [7:0]data;
wire [7:0]out;
ram8bit dut(
.clk(clk),
.rst(rst),
.wr_co(wr_co),
.rd_co(rd_co),
.data(data),
.out(out)
);
initial begin
$monitor("clk=%b,rst=%b,wr_co=%b,rd_co=%b,data=%b,out=%b",clk,rst,wr_co,rd_co,data,out);
end
initial begin
repeat(200)begin
clk=$urandom_range(0,1);
rst=$urandom_range(0,1);
wr_co = $urandom_range(0,1);
rd_co = $urandom_range(0,1);
data = $urandom_range(0,255);
#5;
```

Eloborated design part:-



Output waveform part:-



Thank you!

Want to discuss on this :-

vathsavkothapalli12092004@gmail.com

Follow this page for more VLSI content :-

KOTHAPALLI MAHITH VATHSAV

KOTHAPALLI MAHITH VATHSAV