**RAM MODULE AND SIMULATION REPORT**

1. **Title**

**“Ram Module and Simulation Report”**

1. **Objective:**

To develop a simple synchronous RAM Module with read and write operations.

This report details the design, Verilog implementation, testbench creation, and simulation results of a RAM Module with read and write Operations.

1. **Module:**

Data Width: 8 bit

Address Width: 4 bits

Memory Depth: 16 locations

1. **Verilog Code:**

module ram #(

parameter DATA\_WIDTH = 8,

parameter ADDR\_WIDTH = 4,

parameter RAM\_DEPTH = 1 << ADDR\_WIDTH // Calculate RAM depth based on address width

)(

input clk,

input we, // Write enable

input [ADDR\_WIDTH-1:0] addr,

input [DATA\_WIDTH-1:0] data\_in,

output reg [DATA\_WIDTH-1:0] data\_out

);

reg [DATA\_WIDTH-1:0] mem [0:RAM\_DEPTH-1];

always @(posedge clk) begin

if (we) begin

mem[addr] <= data\_in; // Write operation

end

data\_out <= mem[addr]; // Read operation (always happens on clock edge)

end

endmodule

1. **Testbench code**

module ram\_tb;

reg clk;

reg we;

reg [3:0] addr;

reg [7:0] data\_in;

wire [7:0] data\_out:

ram #(

.DATA\_WIDTH(8),

.ADDR\_WIDTH(4)

) ram\_inst (

.clk(clk),

.we(we),

.addr(addr),

.data\_in(data\_in),

.data\_out(data\_out)

);

initial begin

clk = 0;

forever #5 clk = ~clk; // Clock generation

end

initial begin

// Test sequence

we = 0; addr = 0; data\_in = 0; #10; // Initialization

// Write operations

we = 1; addr = 0; data\_in = 8'hAA; #10;

we = 1; addr = 1; data\_in = 8'hBB; #10;

we = 1; addr = 2; data\_in = 8'hCC; #10;

we = 1; addr = 0; data\_in = 8'hDD; #10; //overwrite address;

// Read operations

we = 0; addr = 0; #10;

we = 0; addr = 1; #10;

we = 0; addr = 2; #10;

we = 0; addr = 0; #10;

$finish;

end

initial begin

$monitor("Time=%0t clk=%b we=%b addr=%h data\_in=%h data\_out=%h", $time, clk, we, addr, data\_in, data\_out);

$dumpfile("ram.vcd");

$dumpvars(0, ram\_tb);

end

endmodule

1. **Testbench Analysis:**

Write Operations:

Wrote 0xAA to address 0.

Wrote 0xBB to address 1.

Wrote 0xCC to address 2.

Overwrote address 0 with 0xDD.

Read Operations:

Read 0xDD from address 0.

Read 0xBB from address 1.

Read 0xCC from address 2.

Read 0xDD from address 0 (after overwrite).

1. **Compilation Report**

Flow Status Successful - Tue Feb 04 22:06:16 2025

Quartus Prime Version 20.1.1 Build 720 11/11/2020 SJ Lite Edition

Revision Name ram

Top-level Entity Name ram

Family Cyclone V

Device 5CGXFC7C7F23C8

Timing Models Final

Logic utilization (in ALMs) 1 / 56,480 (< 1 %)

Total registers 0

Total pins 22 / 268 (8 %)

Total virtual pins 0

Total block memory bits 128 / 7,024,640 (< 1 %)

Total DSP Blocks 0 / 156 (0 %)

Total HSSI RX PCSs 0 / 6 (0 %)

Total HSSI PMA RX Deserializers 0 / 6 (0 %)

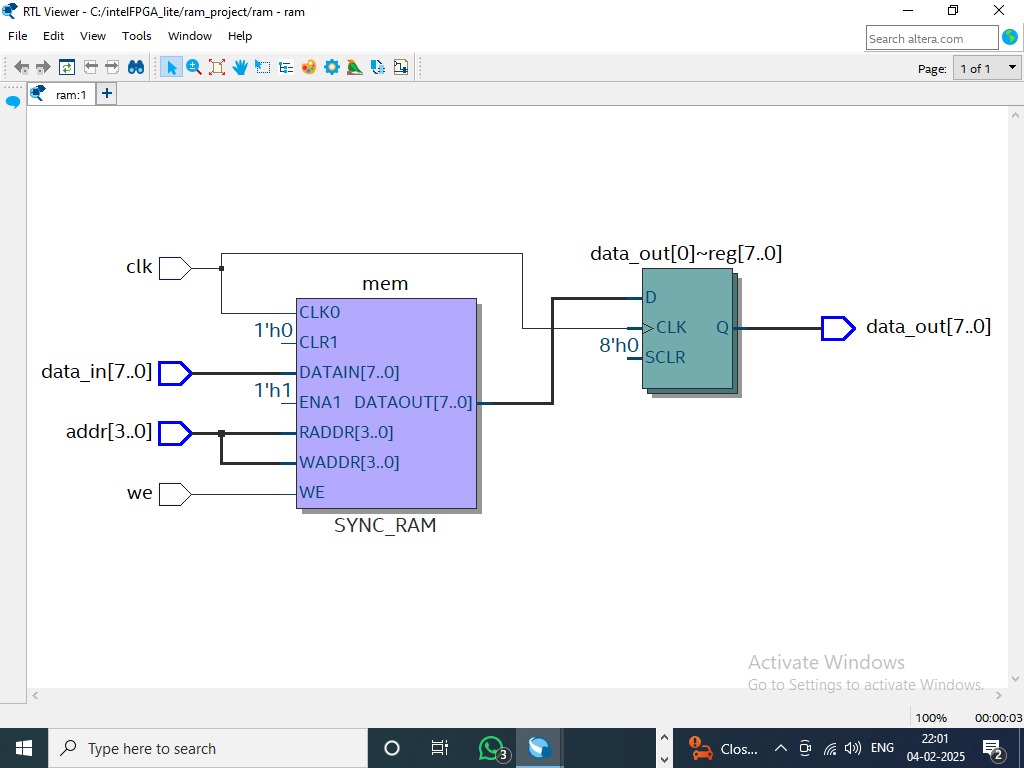
Total HSSI TX PCSs 0 / 6 (0 %)

Total HSSI PMA TX Serializers 0 / 6 (0 %)

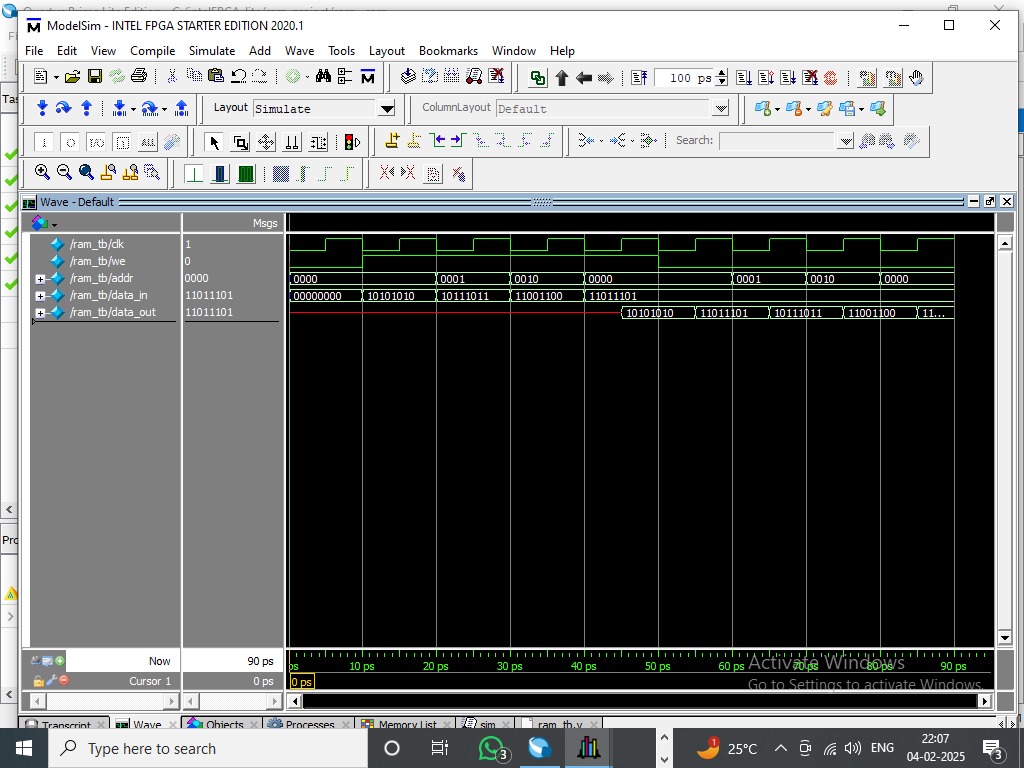
Total PLLs 0 / 13 (0 %)

Total DLLs 0 / 4 (0 %)

1. **RTL Viewer**



1. **Waveform**



1. **Observation:**

All write and read operations occurred correctly on the rising edge of the clock.

Overwriting of data at a memory location functioned as expected.

No unexpected behaviour or errors were observed during the simulation.

1. **Conclusion:**

The simulation successfully demonstrated the correct functionality of the synchronous RAM module, including write, read, and overwrite operations. The module operates as expected according to the design specifications.