

**K L Deemed to be University**  
**Department of Electronics & Communication Engineering**  
**B. Tech ECE Third Year Semester-I**  
**A.Y.2020-21, ODD Semester**

**Course Title: Technical Proficiency & Training -1**

ID Number	Name
180040256	N.Eswar
180040257	K Yogananda Sagar
180040263	P.Sri Vidya

**SINGLE PORT RAM USING VERILOG**

**CONTENTS**

Abstract	1
Introduction	1
Methodology	2-4
Implementation	5-7
Results	8
Applications	9
References	9
THANK YOU	10

## **ABSTRACT**

This Project describes a 64 bit x 8 bit single-port RAM design with common read and write addresses in Verilog.

In computing, memory refers to a device that is used to store information for immediate use in a computer or related computer hardware device. It typically refers to semiconductor memory, specifically metal–oxide–semiconductor memory, where data is stored within MOS memory cells on a silicon integrated circuit chip.

## **INTRODUCTION**

RAM (Random Access Memory) is the internal memory of the CPU for storing data, program, and program result. It is a read/write memory which stores data until the machine is working. As soon as the machine is switched off, data is erased.

Access time in RAM is independent of the address, that is, each storage location inside the memory is as easy to reach as other locations and takes the same amount of time. Data in the RAM can be accessed randomly but it is very expensive

RAM is volatile, i.e. data stored in it is lost when we switch off the computer or if there is a power failure. Hence, a backup Uninterruptible Power System (UPS) is often used with computers. RAM is small, both in terms of its physical size and in the amount of data it can hold.

RAM is of two types –

Static RAM (SRAM)

DynamicRAM(DRAM)

## METHODLOGY

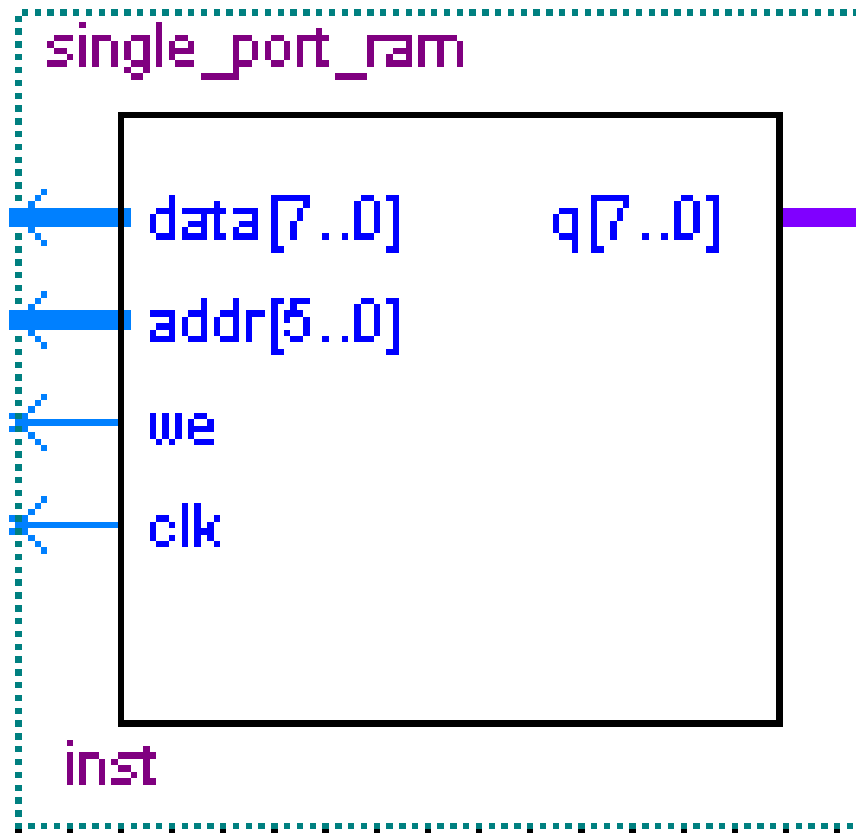


Table 1. Single-Port RAM Port Listing

Port Name	Type	Description
<code>data[7:0]</code>	Input	8-bit data input
<code>addr[5:0]</code>	Input	6-bit address input
<code>we</code>	Input	Write enable input
<code>clk</code>	Input	Clock input
<code>q[7:0]</code>	Output	8-bit data output

## Explanation

### **DYNAMIC RAM:-**

DRAM, unlike SRAM, must be continually refreshed in order to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory as it is cheap and small. All DRAMs are made up of memory cells, which are composed of one capacitor and one transistor.

#### Characteristics of Dynamic RAM

Short data lifetime

Needs to be refreshed continuously

Slower as compared to SRAM

Used as RAM

Smaller in size

Less expensive

Less power consumption



## STATIC RAM:-

- The word static indicates that the memory retains its contents as long as power is being supplied. However, data is lost when the power gets down due to volatile nature. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not be refreshed on a regular basis.
- There is extra space in the matrix, hence SRAM uses more chips than DRAM for the same amount of storage space, making the manufacturing costs higher. SRAM is thus used as cache memory and has very fast access.
- Characteristic of Static RAM
  - Long life
  - No need to refresh
  - Faster
  - Large size
  - Expensive
  - High power consumption



## IMPLEMENTATION/Code

### Single port RAM:

```
module single_port_ram
(
    input [7:0] data,
    input [5:0] addr,
    input we, clk,
    output [7:0] q
);

    reg [7:0] ram[63:0];
    reg [5:0] addr_reg;
    always @ (posedge clk)
    begin
        if (we)
            ram[addr] <= data;
        else
            addr_reg <= addr;
        end
        assign q = ram[addr_reg];
    endmodule
```

### **Single port RAM Tb:**

```
module single_port_ram_tb();  
    reg [7:0] data;  
    reg [5:0] addr;  
    reg we;  
    reg clk;  
    wire [7:0] q;  
    single_port_ram single_port_ram(  
        .data(data),  
        .addr(addr),  
        .we(we),  
        .clk(clk),  
        .q(q)  
    );  
    initial  
    begin  
        clk = 1'b1;  
        forever #50 clk = ~clk;  
    end  
    initial  
    begin  
        data = 8'h01;  
        addr = 5'd0;  
        #100;
```

```
data = 8'h02;
```

```
addr = 5'd01;
```

```
we = 1'b1;
```

```
#100;
```

```
data = 8'h03;
```

```
addr = 5'd02;
```

```
#100;
```

```
addr = 5'd0;
```

```
we = 1'b0;
```

```
#100;
```

```
addr = 5'd1;
```

```
#100;
```

```
addr = 5'd2;
```

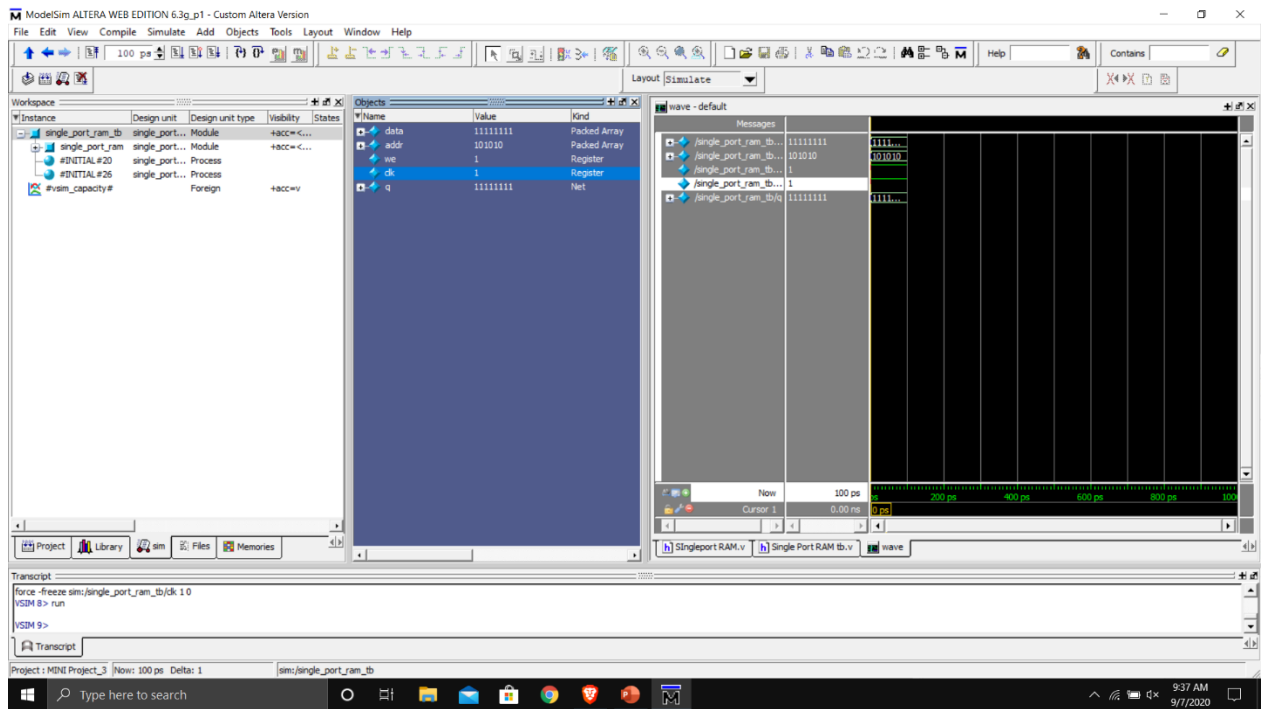
```
#100;
```

```
end
```

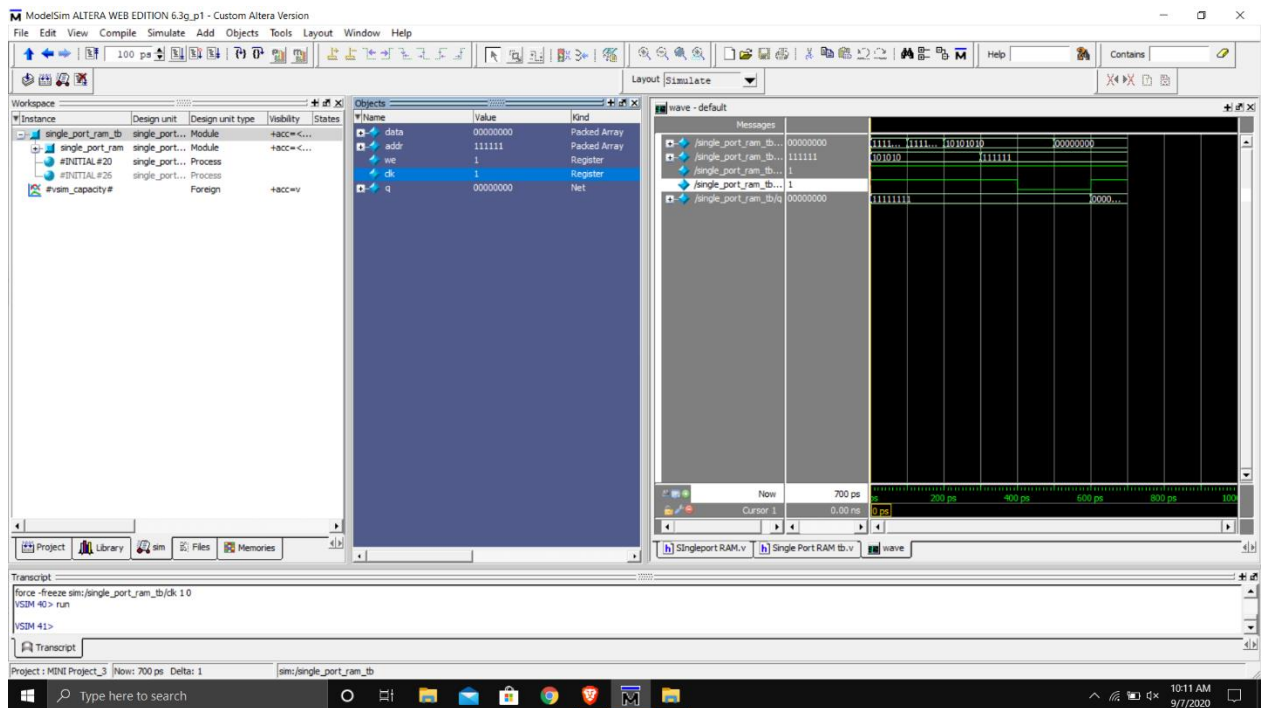


## RESULTS:

### SINGLE PORT RAM-



### TEST BENCH Results:-



## **Applications:-**

**RAM** allows your computer to perform many of its everyday tasks.

Such as loading **applications**.

Browsing the internet.

Editing a spreadsheet.

Experiencing the latest game.

Memory also allows you to switch quickly among these tasks,.

Remembering where you are in one task when you switch to another task.

## **REFERENCE:-**

<https://www.intel.com/content/www/us/en/programmable/support/support-resources/design-examples/design-software/verilog/ver-single-port-ram.html>

<https://www.chipverify.com/verilog/verilog-single-port-ram>

Thank you