DAY 12
MEMORY ELEMENTS - II

October 25, 2023

SRAM

Static Random Access Memory (SRAM) is a type of memory that is used in various digital systems to store data temporarily. SRAM is known for its high speed and low power consumption, making it a popular choice for cache memories and other high-speed data storage applications.

In the provided Verilog module, sram, the implementation of a simple SRAM memory block is demonstrated. The SRAM is parameterized with **DATA_WIDTH, ADDR_WIDTH, and MEM_DEPTH,** allowing for customization of the memory size and data width.

SRAM Working Principle

Data Storage: The SRAM stores data in an array of memory cells, each of which holds one bit of data. The size of the memory array is determined by **MEM_DEPTH**, and each data word has a width of **DATA_WIDTH** bits. The array is declared as **reg [DATA_WIDTH-1:0] ram [0:MEM_DEPTH-1];**.

Addressing: Data in the SRAM is accessed using addresses. The input addr specifies which memory location should be accessed. The width of the address bus is A**DDR_WIDTH**, allowing for addressing of up to **2^ADDR_WIDTH** unique memory locations.

Data Access: The SRAM supports both read and write operations, which are controlled by the input signals **WE_b, OE_b,** and

Write Operation: When **CS_b** is low, **WE_b** is low, and **OE_b** is high, a write operation is performed. The data on the data bus is written to the memory location specified by addr. This operation is synchronous to the rising edge of the clock signal clk.

```
if((CS_b == 0) && (WE_b == 0) && (OE_b == 1)) begin
  ram[addr] <= data;
end
```

Read Operation: When **CS_b** is low, **WE_b** is high, and **OE_b** is low, a read operation is performed. The data from the memory location specified by addr is placed onto the data bus. This operation is also synchronous to the rising edge of the clock signal clk. To avoid contention on the data bus, an internal register temp is used to hold the data read from the memory array. The data bus is then driven with the contents of temp when the SRAM is not in write mode.

```
    if((CS_b == 0) && (WE_b == 1) && (OE_b == 0)) begin
```

```
        temp <= ram[addr];
    end
    assign data = ((CS_b == 0) && (WE_b == 1) && (OE_b == 0)) ? temp :
{DATA_WIDTH{1'bz}};
```

Reset Operation: The SRAM module includes a synchronous reset input rstn. When rstn is asserted low, the internal register temp is reset to 0. This ensures that the SRAM is in a known state upon initialization.

Tri-State Bus: The data bus is bidirectional and is put into a high-impedance state when the SRAM is not actively driving it. This allows for other devices on the same bus to drive the bus without causing bus contention.

Specification of today's design

- A 2K x 16 architecture
- Word addressable
- A bidirectional 16 bit Data Bus
- An 11 bit Address Bus
- Low true Output Enable – OE
- Read / Write control. Low true Write Enable → WE, High true Read → WE
- Clock

SRAM Module

```
  module sram#(parameter DATA_WIDTH=16,ADDR_WIDTH=11,MEM_DEPTH=2048)
// ADDR_WIDTH = 11 = log2(2^(1))
(inout [DATA_WIDTH-1:0] data,input WE_b,OE_b,CS_b,clk,rstn,
input [ADDR_WIDTH-1:0] addr);

// Define the memory register
reg [DATA_WIDTH-1] ram [0:MEM_DEPTH-1];
reg [DATA_WIDTH-1:0] temp; // Temporary internal register


always@(posedge clk or negedge rstn) begin
  if(!rstn) begin
    temp <= 16'b0;
  end
  else if((CS_b == 0) && (WE_b == 0) && (OE_b == 1)) // Write operation
      ram[addr] <= data;
  else if((CS_b == 0) && (WE_b == 1) && (OE_b == 0)) // Read operation
      temp <= ram[addr];
end

assign data = ((CS_b == 0) && (WE_b == 1) && (OE_b == 0)) ? temp : 1'bz;

endmodule\textbf{}
```