TASK 2 : Simple Synchronous RAM Module With Read & Write Operations

BASIC INFORMATION :

Synchronous Random Access Memory (SRAM) is a type of memory that relies on a clock signal to control the timing of read and write operations. Unlike asynchronous RAM, which operates independently of the clock, synchronous RAM ensures that all operations happen in sync with a clock signal, improving timing accuracy and predictability.

MATLAB CODE:

```
RAM_SIZE = 16;

ADDR_WIDTH = 4;

DATA_WIDTH = 8;

ram_memory = zeros(RAM_SIZE, DATA_WIDTH);

clk = 0;

clk_period = 10;

reset = 0;

we = 0

addr = 0;

data_in = 0;

data_out = 0;

num_cycles = 20;

memory_history = zeros(RAM_SIZE, DATA_WIDTH, num_cycles);  % To store RAM state at each cycle

for t = 1:num_cycles

   clk = ~clk;

   if t <= 3

      reset = 0;

   else

      reset = 1;

   end

   if reset == 0

      ram_memory = zeros(RAM_SIZE, DATA_WIDTH);

      data_out = 0;
```

```
    else
        if we == 1
            ram_memory(addr + 1, :) = data_in;
        end
        if we == 0
            data_out = ram_memory(addr + 1, :);
        end
    end
    if t == 5
        addr = 4;
        data_in = 170;
        we = 1;
    end
    if t == 10
        addr = 5;
        data_in = 204;
        we = 1;
    end
    if t == 15
        addr = 4;
        we = 0;
    end
    if t == 18
        addr = 5;
        we = 0;
    end
    memory_history(:, :, t) = ram_memory;  % Store RAM state at this cycl
    pause(clk_period / 1000);
end
% Visualize the RAM content over time (using a subplot for better visualization)
figure;
```

```
for t = 1:num_cycles
    subplot(num_cycles, 1, t);
    stem(ram_memory(:, 1), 'filled');  % Plot for the first data column (8 bits)
    title(['Cycle ' num2str(t)]);
    xlabel('Address');
    ylabel('Data Value');
    pause(0.1);  % Pause between cycles for better visualization
end
```

SIMULATION OUTPUT: