Problem Set 5

Magson Gao (magson@mit.edu)
6.111 Fall 2018

September 27, 2018

1 Problem A

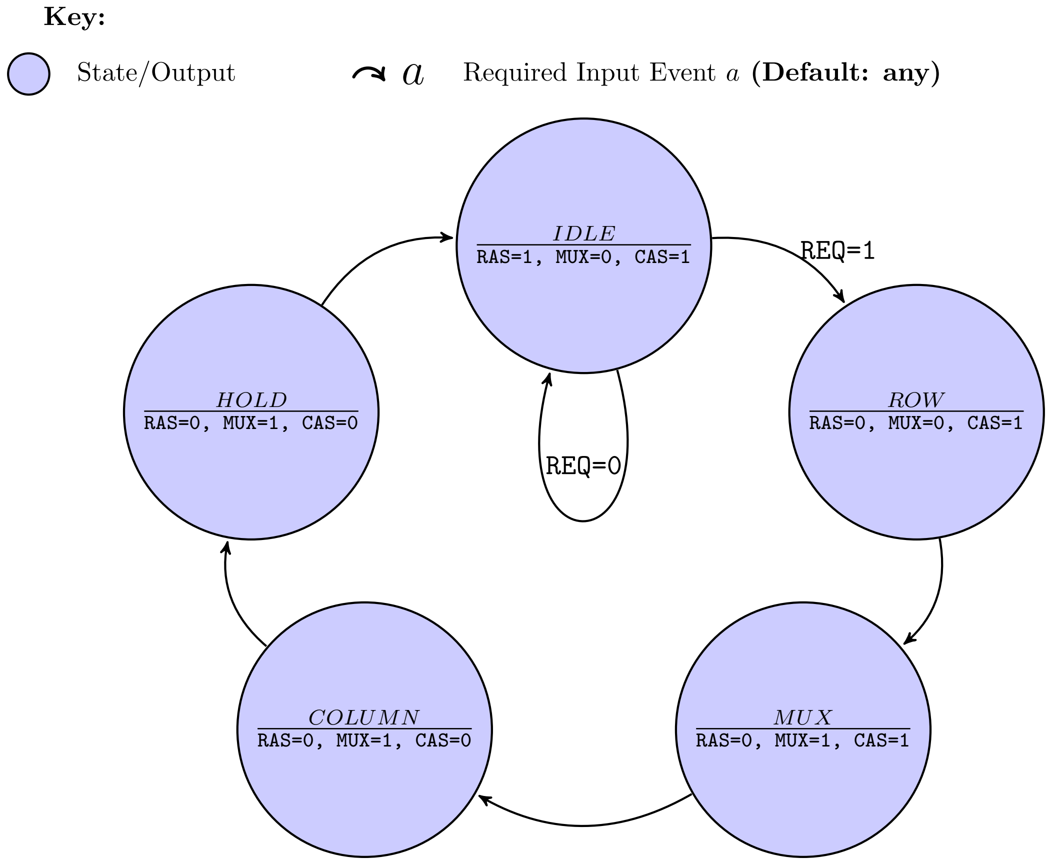


Figure 1.1: The finite state machine diagram for the given system.

2 Problem B

Code:

```
module mem_controller(
            input        clk_in,
            input        req_in,
            output       ras_out,
            output       mux_out,
            output       cas_out
          );

  localparam STATE_IDLE = 4'b1010;
  localparam STATE_ROW  = 4'b0010;
  localparam STATE_MUX  = 4'b0110;
  localparam STATE_COL  = 4'b0100;
  localparam STATE_HOLD = 4'b0101;

  reg [3:0] state = STATE_IDLE;

  assign ras_out = state[3];
  assign mux_out = state[2];
  assign cas_out = state[1];

  // state machine
  always @(posedge clk_in)
  begin
    case (state)
      STATE_IDLE : state <= req_in ? STATE_ROW : STATE_IDLE;
      STATE_ROW  : state <= STATE_MUX;
      STATE_MUX  : state <= STATE_COL;
      STATE_COL  : state <= STATE_HOLD;
      STATE_HOLD : state <= STATE_IDLE;
      default    : state <= STATE_IDLE;
    endcase
  end

endmodule
```

Testbench:

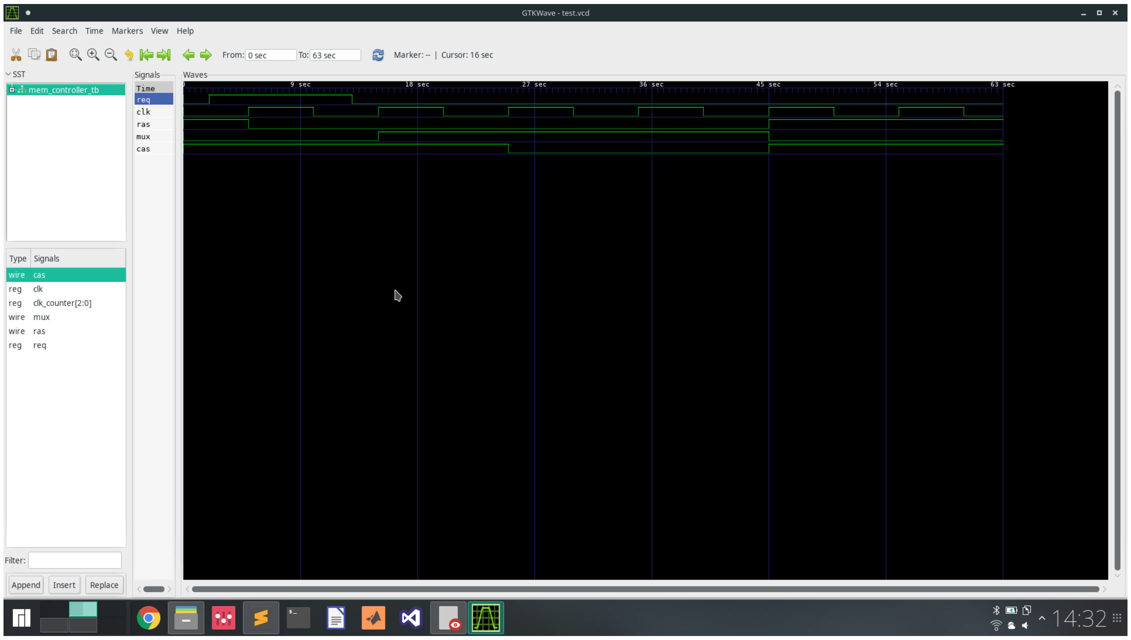
```
`define assert(signal, value) \
        if (signal !== value) \
        begin \
            $display("ASSERTION FAILED in %m at time %0t: signal != value", $time); \
        end

module mem_controller_tb;
  reg         clk;
  reg         req;
  wire        ras;
  wire        mux;
  wire        cas;
  reg [2:0]   clk_counter;

  initial
  begin
      $dumpfile("test.vcd");
    $dumpvars(0,mem_controller_tb);
    clk         = 0;
    clk_counter = 0;
    req         = 0;

    #2;
    req = 1;

    #11;
    req = 0;
    #50;
    $finish;
  end


  mem_controller m1(
            .clk_in    (clk),
            .req_in    (req),
            .ras_out   (ras),
            .mux_out   (mux),
            .cas_out   (cas)
            );

  always @(posedge clk)
  begin
    if ((clk_counter > 0) | ((req == 1) & (clk_counter == 0)))
    begin
      clk_counter = clk_counter + 1;

      if (clk_counter == 1)
      begin
        `assert(ras, 0);
        `assert(mux, 0);
        `assert(cas, 1);
      end
      else if (clk_counter == 2)
      begin
```

```
        ‘assert(ras, 0);
        ‘assert(mux, 1);
        ‘assert(cas, 1);
      end
      else if (clk_counter == 3)
      begin
        ‘assert(ras, 0);
        ‘assert(mux, 1);
        ‘assert(cas, 0);
      end
      else
      begin
        ‘assert(ras, 0);
        ‘assert(mux, 1);
        ‘assert(cas, 0);
        clk_counter = 0;
      end
    end
  end

  always #5 clk = ~clk;

endmodule
```

3 Problem C

Outputs are encoded in the state which is synchronous, thus no glitches occur.

4 Problem D



From: 0 sec To: 63 sec Marker: -- | Cursor: 16 sec

Type	Signals
wire	cas
reg	clk
reg	clk_counter[2:0]
wire	mux
wire	ras
reg	req