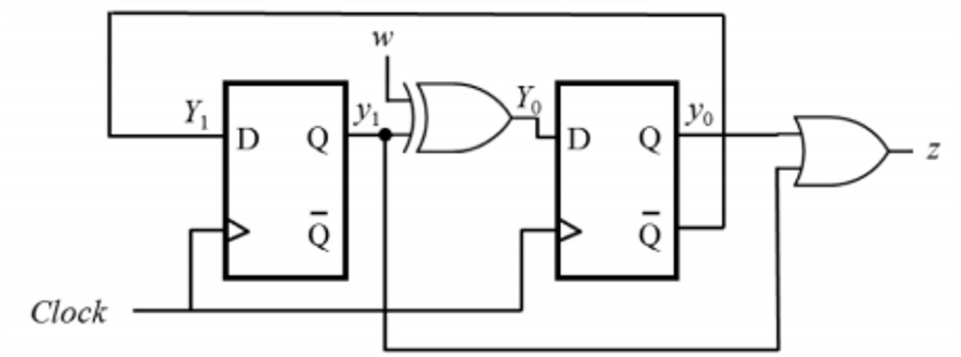
Nome e Matricula: _____

1. Para o circuito abaixo que implementa uma máquina de estados.

(a) Suponha que os 2 flipflops tenham o valor inicial 0. Qual será o valor dos flipflop após três ciclos de clock ? Qual o valor da saída z ?
(b) Desenhe a tabela com descrição da máquina de estados implementada pelo circuito
(c) Desenhe o diagrama de estados da máquina.
(d) Descreva em Verilog com os três formatos (case, equações e memória) a máquina abaixo.



2. A com 2 bits. Se A=0 sequencia 0,1,2,1,3 → 0,1,2,1,3, Se A=1 sequencia 1,3,2,0,3 → 1,3,2,0,3, Se A=2 sequencia 3 → 3 → ….. Se A=3 sequencia 1,2 → 1,2 ….
Fazer a tabela, diagrama de estados e preencher a memória. Não é preciso fazer código em Verilog.

3. Complete o desenho para o código Verilog da Cache.

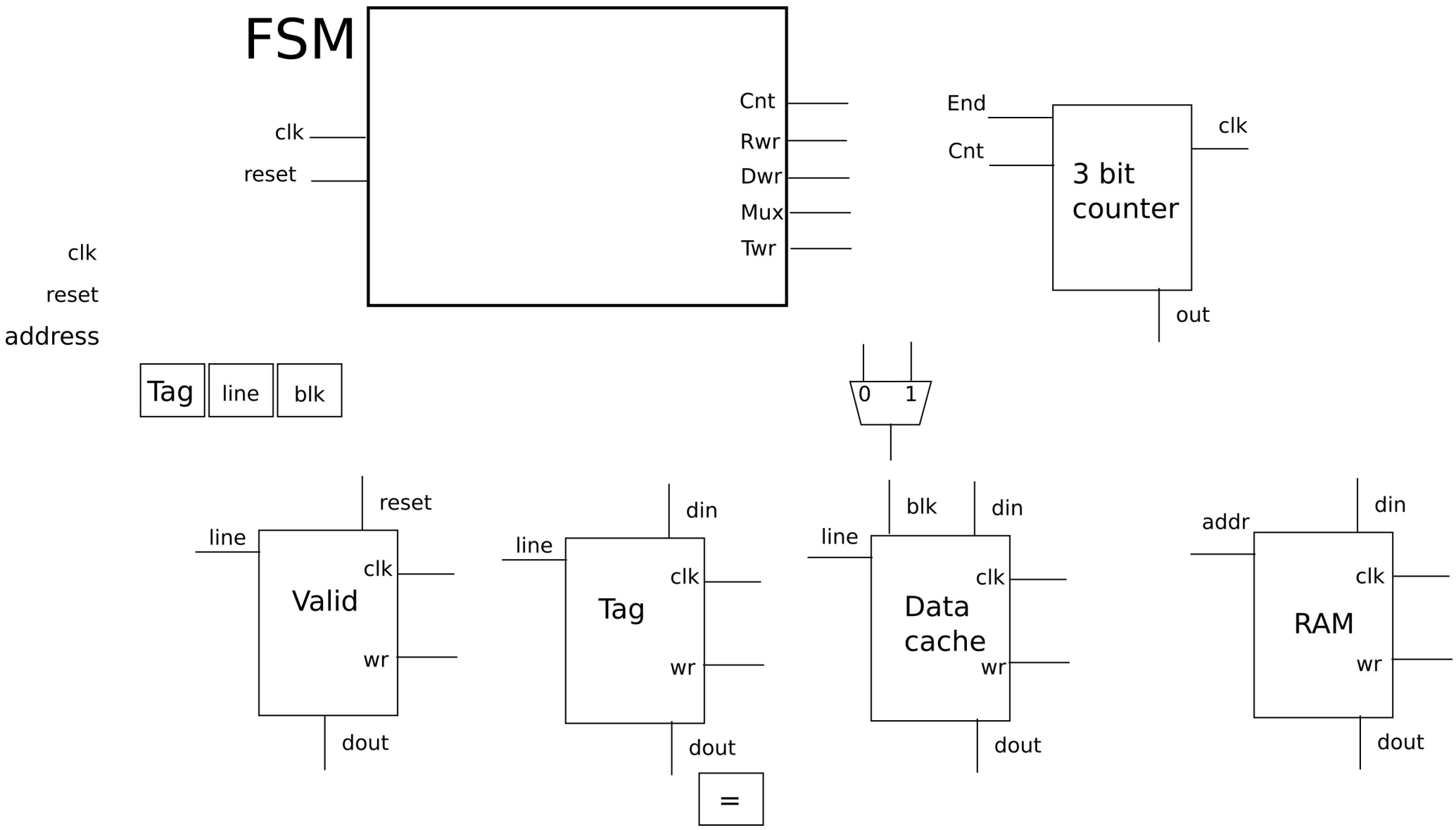
```
module cache_read_only(clk, reset, address, dout, din);
input clk;
input reset;
input [10:0] address;
output [7:0] dout;
input [7:0] din;
output [5:0] debug;
reg [5:0] debug;
reg [7:0] dout;
wire [1:0] line;
wire [2:0] blk;
wire [5:0] tag;
assign tag = address[10:5];
assign line = address[4:3];
assign blk = address[2:0];
wire Twr,Dwr,Rwr;
wire END, cnt, c, v, Dmux;
wire [7:0] Ram2Cache;
wire [2:0] Mux1;
wire [2:0] Muxout;
wire MuxSel;
wire [5:0] Tout;
wire [7:0] Cache2out;
always @(*) begin
 dout = Cache2out;
end
```

```
fsm  FSM( .clk (clk),  .reset (reset),  .c (c) ,  .v (v),  .END (END),  .Twr (Twr),  .Dwr (Dwr),  .Rwr (Rwr),  .Cnt (Cnt),  .Mux (MuxSel) );
valid V( .clk (clk), .line (line), .reset (reset),    .wr (Twr), .dout(v) );
Mtag T( .clk (clk), .line (line),  .din(tag),    .wr(Twr), .dout(Tout), .reset(reset) );
comparator comp(.out(c), .tag(tag), .tag_in(Tout));
datacache dcache( .clk (clk), .line
(line), .blk(Muxout),  .din(Ram2Cache), .wr(Dwr), .dout(Cache2out) );
ram R ( .clk(clk), .addr({tag,line,Mux1}), .din(din), .wr(Rwr), .dout(Ram2Cache), .reset(reset) );
mux DataMux( .din_0 (blk), .din_1(Mux1), .sel(MuxSel) , .mux_out (Muxout));
counter count(.out(Mux1), .clk(clk), .reset(Cnt), .End(END));
endmodule
```

----------Exemplos de Verilog para Questão 1 ----------

```
module ff ( input data, input c, input r, output q);
reg q;
always @(posedge c or negedge r)
begin
 if(r==1'b0)  q <= 1'b0;
 else q <= data;
end
endmodule //End
module statem(input clk, input reset, input d, output t);
reg [1:0] state;
parameter zero=2'd0, one=2'd1, two=2'd2, three=2'd3      ;
assign t = (state == three)? 1:0;
always @(posedge clk or negedge reset)
    begin
        if (reset==0)
            state = zero;
        else
            case (state)
                zero:   if ( d == 1 ) state = one;
                one:  if ( d == 1 ) state = two;
                                else state = zero;
                two:  if ( d == 1 ) state = three;
                                else state = zero;
                three:     state = zero;
            endcase
    end
endmodule
module statePorta(input clk, input res, input d, output [1:0] t);
wire [1:0] e; wire [1:0] p;
assign t = {e[0] & e[1], ~e[0]};
assign p[0] = ~e[0] & d;
assign p[1] = d & (e[0] ^ e[1]);
ff  e0(p[0],clk,res,e[0]);
ff  e1(p[1],clk,res,e[1]);
endmodule
module stateMem(input clk,input res, input d, output [1:0] t);
reg [2:0] StateMachine [0:15];
initial
begin
StateMachine[0] = 4'd0;  StateMachine[1] = 4'd2; StateMachine[2] = 4'd0;  StateMachine[3] = 4'd4;
end
wire [2:0] address; wire [2:0] dout;
```

```
assign address[0] = d;
assign dout = StateMachine[address];
assign t = dout[1:0];
ff st0(dout[2],clk,res,address[1]);
ff st1(dout[3],clk,res,address[2]);
endmodule
```


FSM
clk
reset
Cnt
Rwr
Dwr
Mux
Twr
End
Cnt
clk
3 bit
counter
out
clk
reset
address
Tag
line
blk
0
1
reset
line
clk
Valid
wr
dout
din
line
clk
Tag
wr
dout
=
blk
din
line
clk
Data
cache
wr
dout
din
addr
clk
RAM
wr
dout