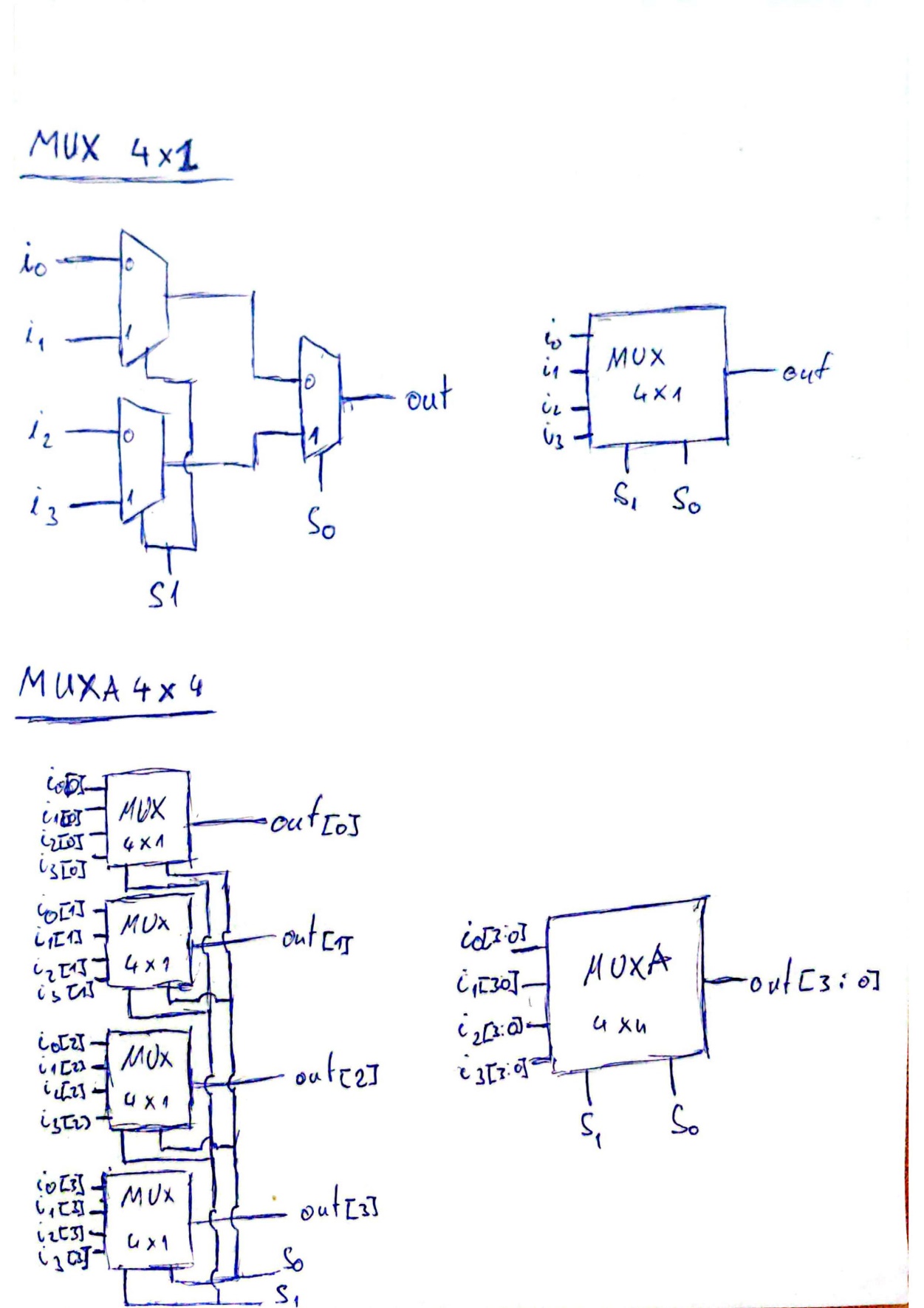
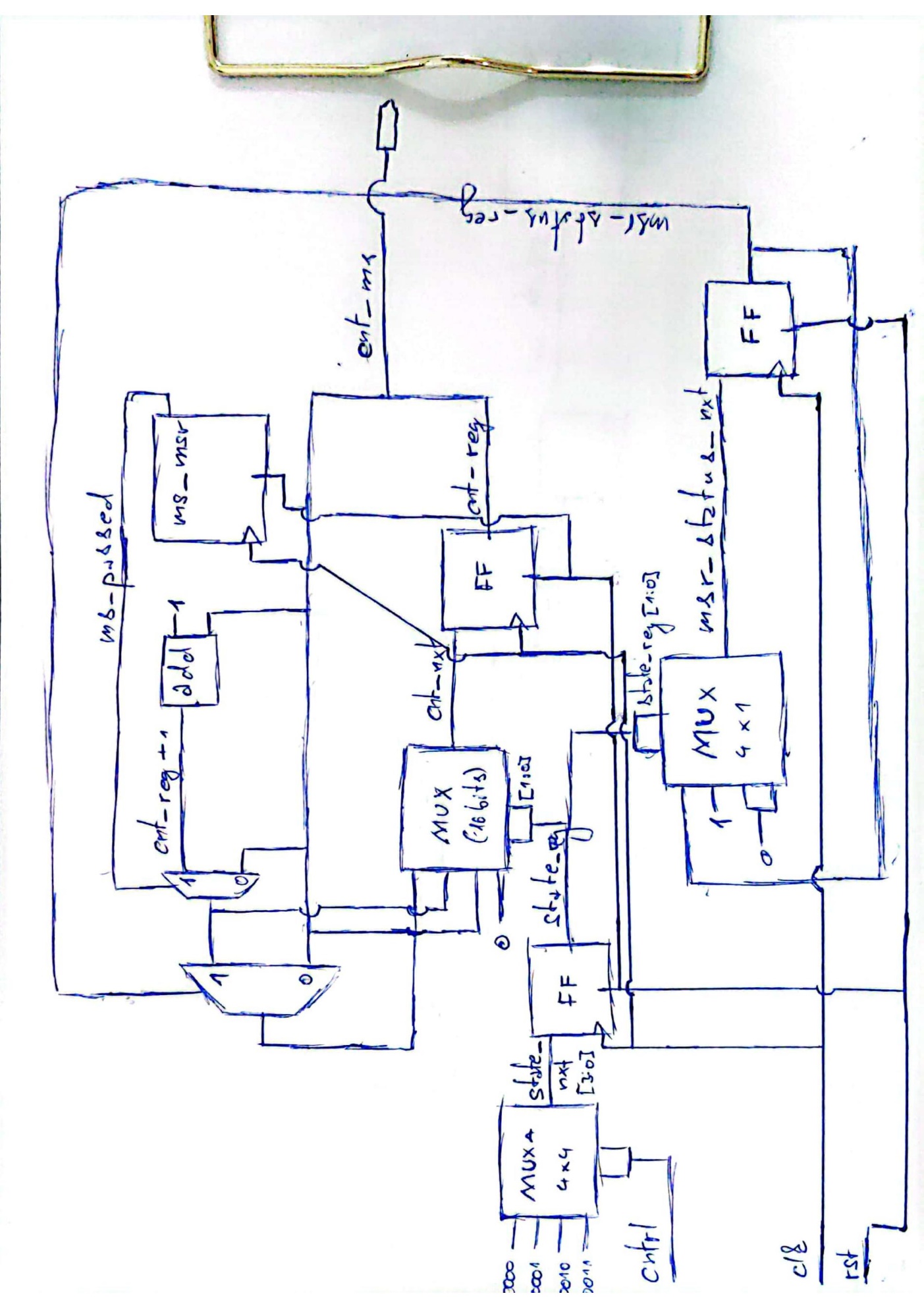
**Adaugarea de opcode-uri**



****

\*MUXA 4x4 ar fi de fapt o implementare posibilă pentru “case”. În acest caz, primul bit este ignorat la ieșire.

**Module:**

module ms\_cnt(

input clk, rst,

input [2:0] cntrl,

output [15:0] cnt\_ms

);

reg [15:0] cnt\_reg, cnt\_nxt;

reg [2:0] state\_reg, state\_nxt;

reg msr\_status\_reg, msr\_status\_nxt;

wire ms\_passed;

wire [15:0] cnt\_clk;

localparam S0 = 3'b000;

localparam S1 = 3'b001;

localparam S2 = 3'b010;

localparam S3 = 3'b011;

ms\_msr ms\_msr\_1(

.clk(clk),

.rst(rst),

.ms\_passed(ms\_passed),

.cnt\_clk(cnt\_clk)

);

always @(\*) begin

case(state\_reg)

S0: begin

if(msr\_status\_reg & ms\_passed)

cnt\_nxt = cnt\_reg + 1;

else

cnt\_nxt = cnt\_reg;

end

S1: begin

msr\_status\_nxt = 1;

if(ms\_passed)

cnt\_nxt = cnt\_reg + 1;

end

S2: begin

msr\_status\_nxt = 0;

cnt\_nxt = cnt\_reg;

end

S3: begin

msr\_status\_nxt = 0;

cnt\_nxt = 0;

end

endcase

case(cntrl)

S0: begin

state\_nxt = S0;

end

S1: begin

state\_nxt = S1;

end

S2: begin

state\_nxt = S2;

end

S3: begin

state\_nxt = S3;

end

endcase

end

always @(negedge clk or posedge rst) begin

if(rst == 1) begin

cnt\_reg <= 0;

msr\_status\_reg <= 0;

state\_reg <= 0;

end

else begin

cnt\_reg <= cnt\_nxt;

msr\_status\_reg <= msr\_status\_nxt;

state\_reg <= state\_nxt;

end

end

assign cnt\_ms = cnt\_reg;

endmodule

**Testbench:**

//correct functionality for a timescale of 1ns/<precision>

module test;

initial begin

$dumpfile("dump.vcd");

$dumpvars(1, test);

end

reg clk, rst;

reg [2:0] cntrl;

wire [15:0] cnt\_ms;

parameter period = 200; //200ns periods correspond to 5 MHz

parameter nr\_p\_s = 5000; //nr of periods in 1s

ms\_cnt ms\_cnt\_1(

.clk(clk),

.rst(rst),

.cnt\_ms(cnt\_ms),

.cntrl(cntrl)

);

initial begin

rst = 1;

clk = 0;

#10 rst = 0;

cntrl = 3'b001;

end

initial begin

repeat(80000)

#(period/2) clk = ~clk;

end

initial begin

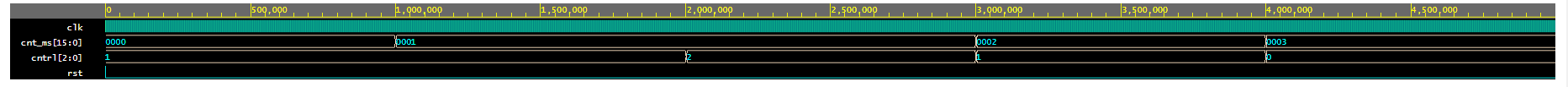
#(2 \* nr\_p\_s \* period) cntrl = 3'b010;

#(1 \* nr\_p\_s \* period) cntrl = 3'b001;

#(1 \* nr\_p\_s \* period) cntrl = 3'b000;

#(1 \* nr\_p\_s \* period) cntrl = 3'b011;

#(1 \* nr\_p\_s \* period) cntrl = 3'b001;

 end

endmodule

**Functionality:**

See waveform