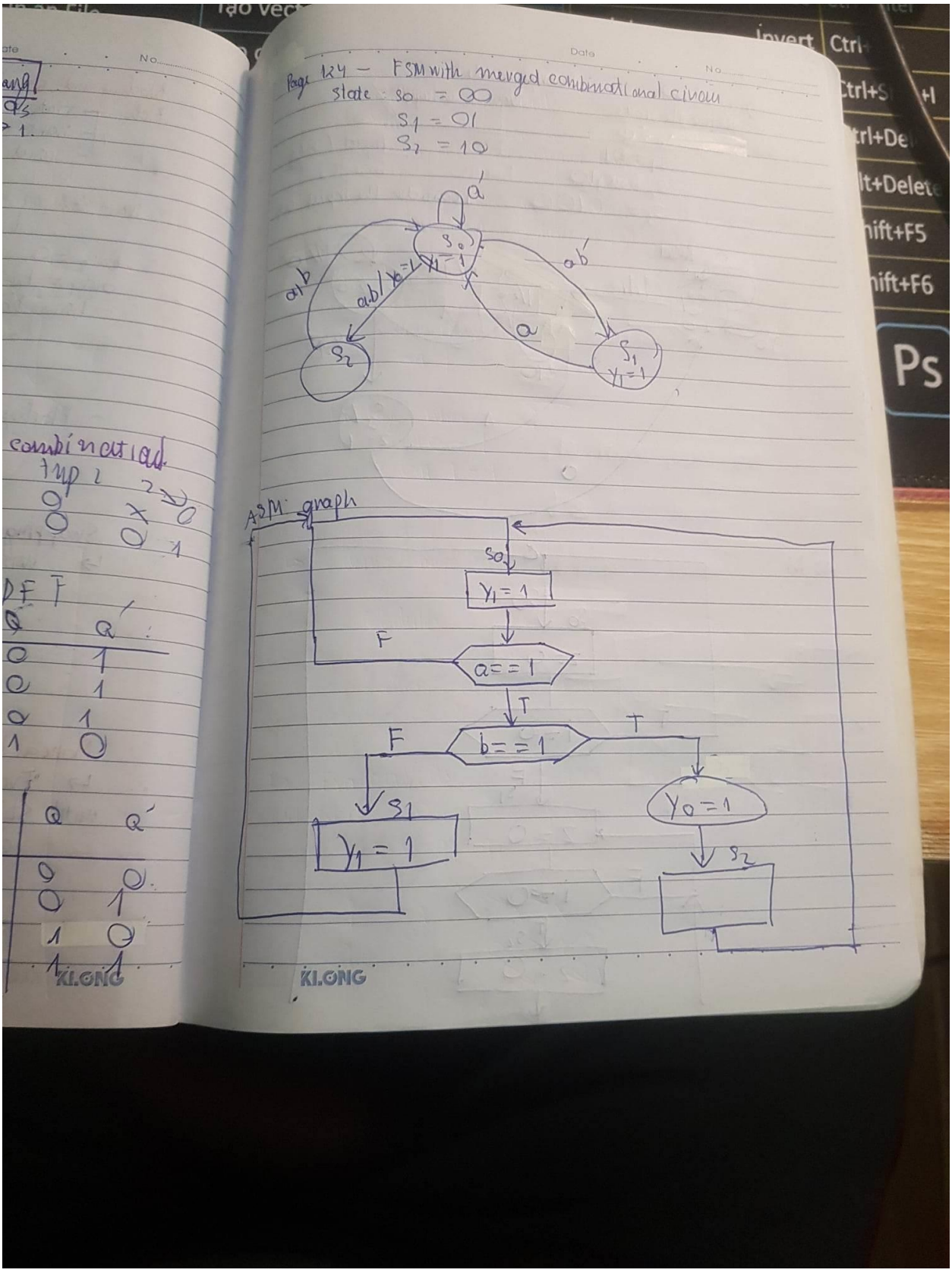
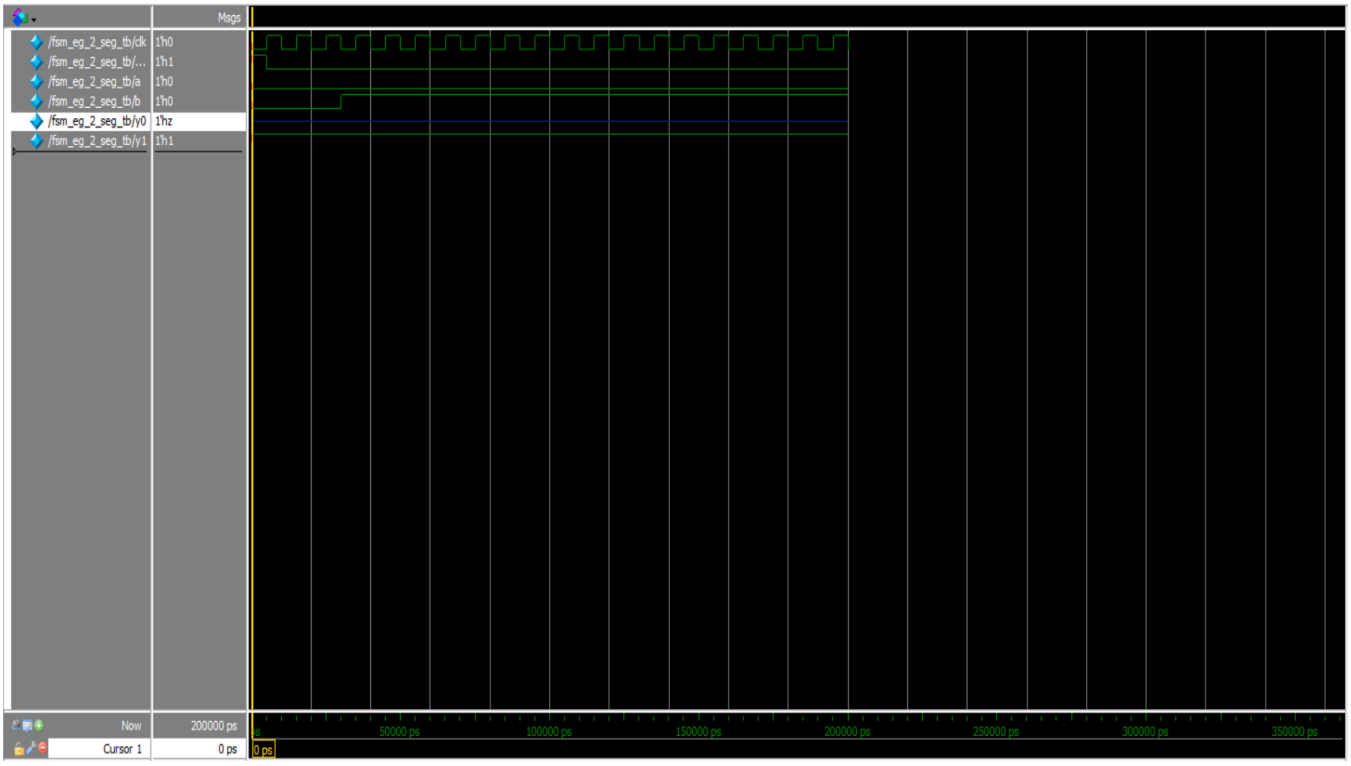
1 a.


Page 124 – FSM with merged combinational circuit
state: s0 = 00
S1 = 01
S2 = 10
a'
S0
Y1 = 1
a'b
ab/Y0=1
ab'
a
S1
Y1=1
S2
ASM graph
S0
Y1 = 1
F
a==1
T
F
b==1
T
S1
Y1 = 1
Y0=1
S2
KLONG
KLONG

Testbench

```
`timescale 1ns / 1ps

module fsm_eg_2_seg_tb;

reg clk, reset;
reg a, b;
wire y0, y1;

fsm_eg_2_seg dut (
    .clk(clk),
    .reset(reset),
    .a(a),
    .b(b),
    .y0(y0),
    .y1(y1)
);

initial begin
    clk = 0;
    reset = 1;
    a = 0;
    b = 0;
    #5 reset = 0;
end

always #5 clk = ~clk;

initial
begin
a = 0;
b = 0;
reset = 1;
#30;
a = $random();
b = $random();
end

endmodule

```

Wave


Msgs
/fsm_eg_2_seg_tb/clk
/fsm_eg_2_seg_tb/...
/fsm_eg_2_seg_tb/a
/fsm_eg_2_seg_tb/b
/fsm_eg_2_seg_tb/y0
/fsm_eg_2_seg_tb/y1
1'h0
1'h1
1'h0
1'h0
1'hz
1'h1
Now
200000 ps
Cursor 1
0 ps
50000 ps
100000 ps
150000 ps
200000 ps
250000 ps
300000 ps
350000 ps

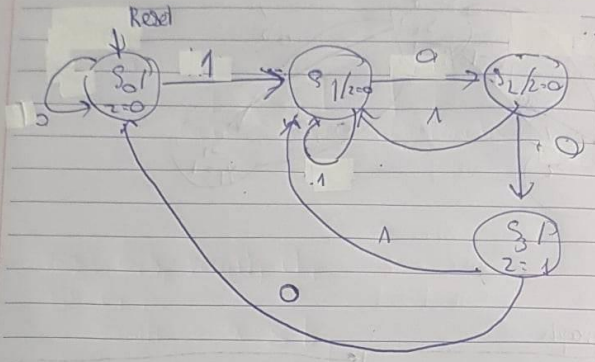
b.

b) 0100 Moore model:
Define the state Non overlapping

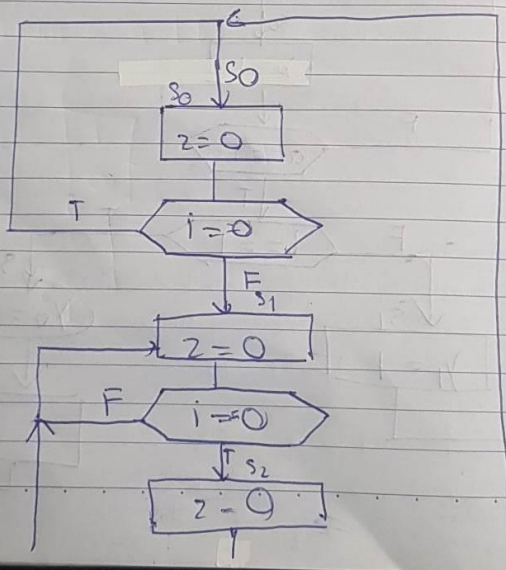
$S_0 = 0$
$S_1 = 01$
$S_2 = 010$
$S_3 = 0100$

Input cases

I consider the input as i.



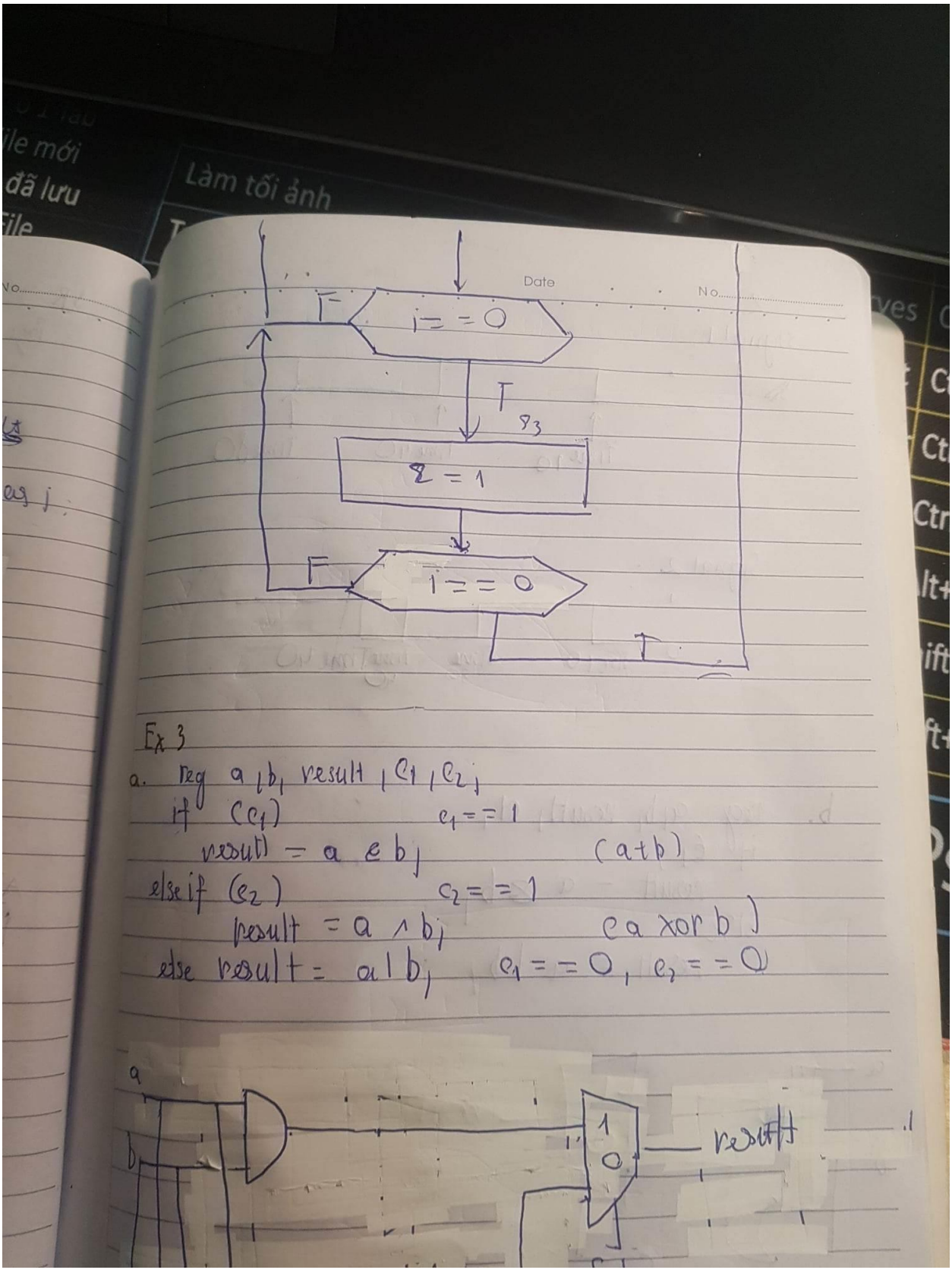
ASM



KI.ONG

Ex 3

a.


Date
No.
i == 0
T
F
s3
s = 1
i == 0
F
T
Ex 3
a. reg a, b, result, c1, c2;
if (c1) c1 == 1
result = a & b; (a+b)
else if (c2) c2 == 1
result = a ^ b; (a xor b)
else result = a | b; c1 == 0, c2 == 0
a
b
1
0
result

RTL file

```
module fsm2(
input wire clk, reset,
input wire a,
output y0
);

localparam [1:0] s0 = 2'b00,
                 s1 = 2'b01,
                 s2 = 2'b10,
           s3 = 2'b11;
// signal declaration
reg [1 : 0] state_reg, state_next;

// state register
always @(posedge clk, posedge reset)
if (reset)
state_reg <= s0;
else state_reg <= state_next;

// next-state logic and output logic
always @*
case(state_reg)
s0:
  if (a==1)
        state_next = s1;
        else
        state_next = s0;
s1:
if(a == 1)
        state_next =s2;
```

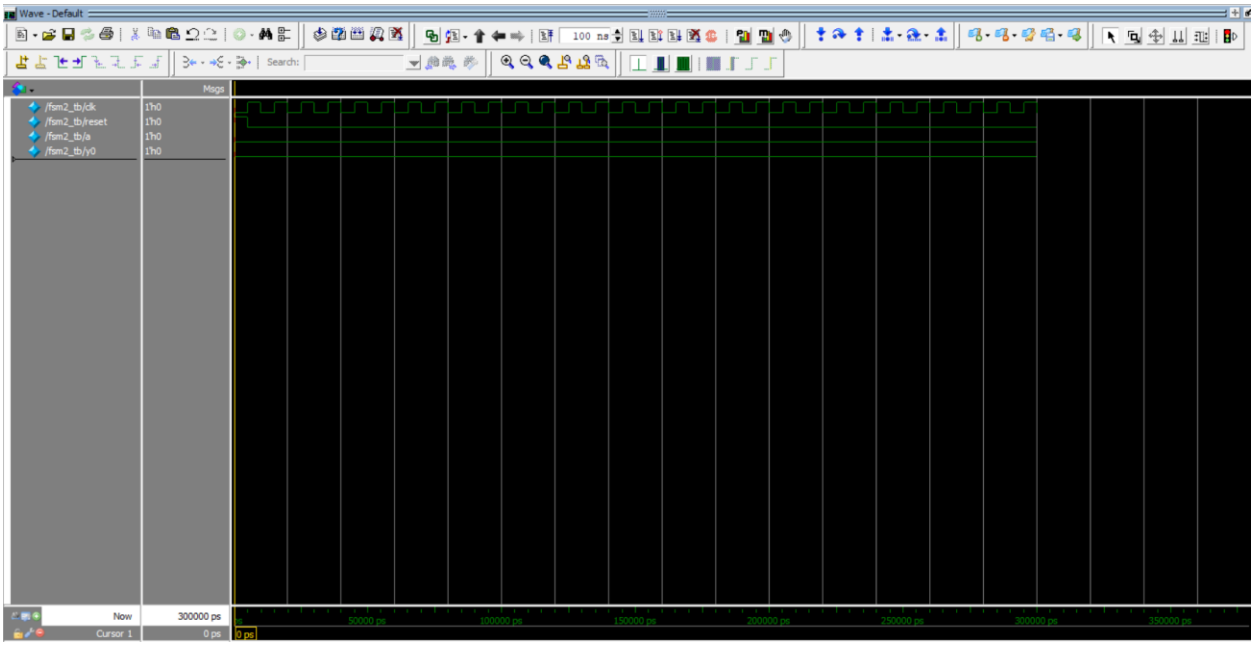
```
else
state_next = s1;

s2:
if(a == 0)
state_next = s3;
else
state_next = s1;

s3:
if ( a == 1)
state_next = s1;
else
state_next = s0;
endcase


assign y0 = (state_reg ==s3);


endmodule
```

Testbench file

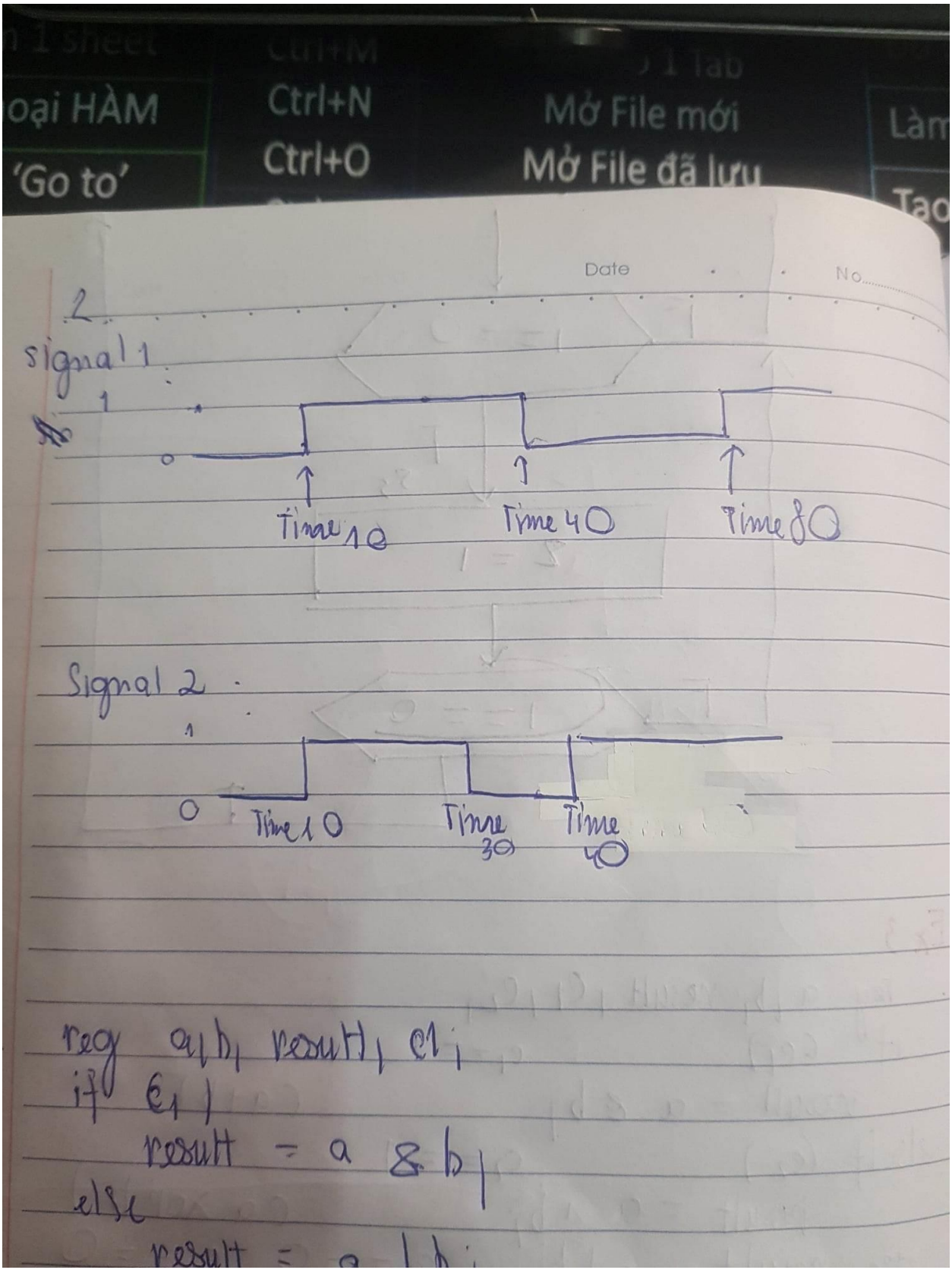
```
`timescale 1ns / 1ps

module fsm2_tb;

reg clk, reset;
reg a;
wire y0;
```

```
fsm2 dut (
    .clk(clk),
    .reset(reset),
    .a(a),
    .y0(y0)
);

initial begin
    clk = 0;
    reset = 1;
    a = 0;
    #5 reset = 0;
end

always #5 clk = ~clk;

initial
begin
a = 0;
reset = 1;
#30;
a = $random();
end

endmodule
```

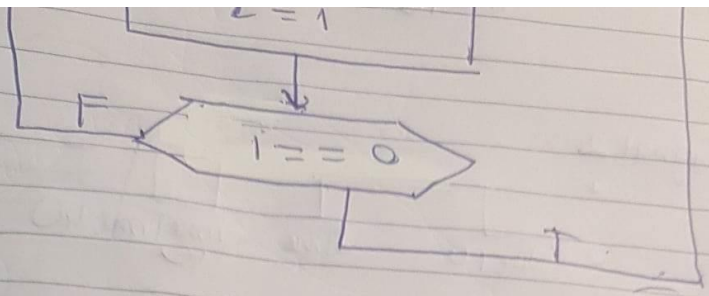
Wave


Wave - Default
Search:
Msgs
/fsm2_tb/clk
/fsm2_tb/reset
/fsm2_tb/a
/fsm2_tb/y0
1'h0
1'h0
1'h0
1'h0
Now
300000 ps
Cursor 1
0 ps
50000 ps
100000 ps
150000 ps
200000 ps
250000 ps
300000 ps
350000 ps

Ex2


oại HÀM
'Go to'
Ctrl+N
Mở File mới
Ctrl+O
Mở File đã lưu
Date
No.
2
signal 1
1
0
Time 10
Time 40
Time 80
Signal 2
1
0
Time 10
Time 30
Time 40
reg a, b, result, c1;
if (c1)
result = a & b;
else
result = a | b;

Ex3



Ex 3

a. reg a, b, result, c_1, c_2;

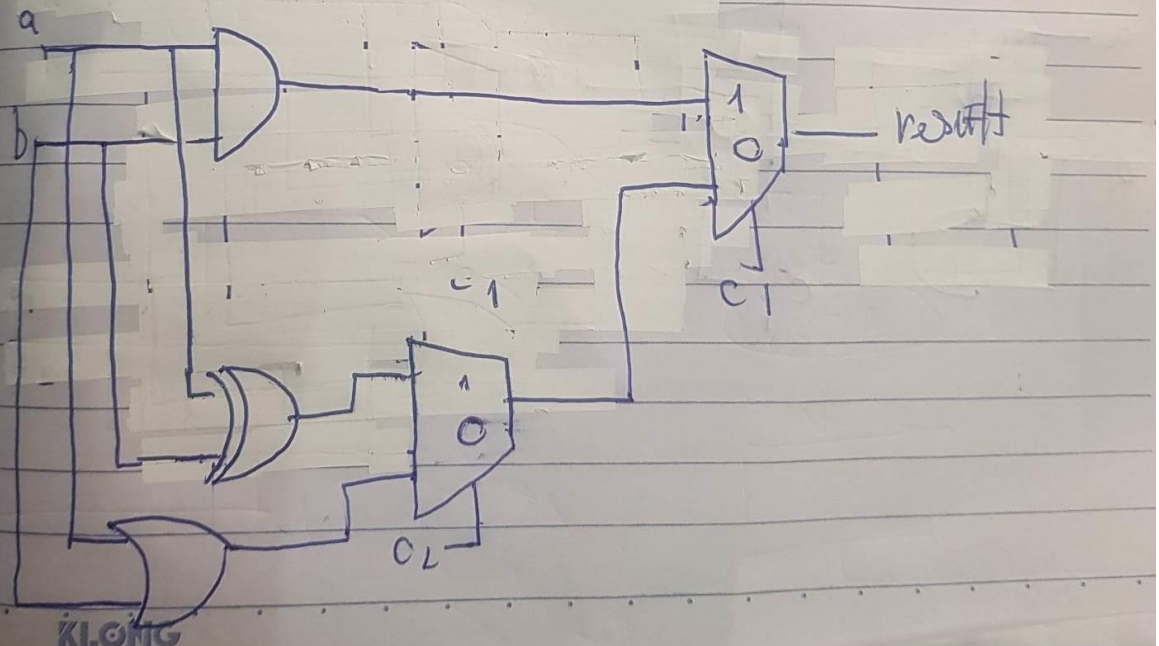
if (c_1) $c_1 == 1$

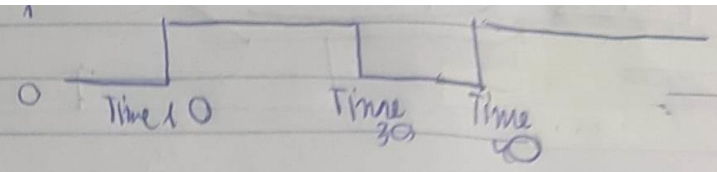
result = a & b; (a+b)

else if (c_2) $c_2 == 1$

result = a ^ b; (a xor b)

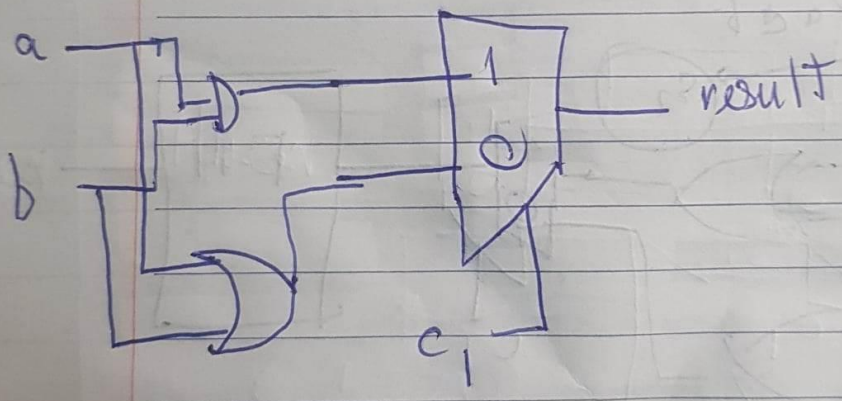
else result = a | b; $c_1 == 0$, $c_2 == 0$





b.

```
reg a, b, result, c1;
if (c1)
    result = a & b;
else
    result = a | b;
```



Date No.

```
c. always @ (a, b, c, sel).
begin
 case(sel)
  2'b00 : out = a;
  2'b01 : out = b;
  2'b10 : out = c;
  default : out = 0;
 endcase
end
```