**BÀI TẬP THIẾT KẾ IC**

Họ và tên: Hà Minh Khuê

Lớp: 15LVDT

Câu 1:

Code:

module cau1(x1, x2, x3, y);

input x1, x2, x3;

output y;

wire g, h, k;

and(g, x1, x2);

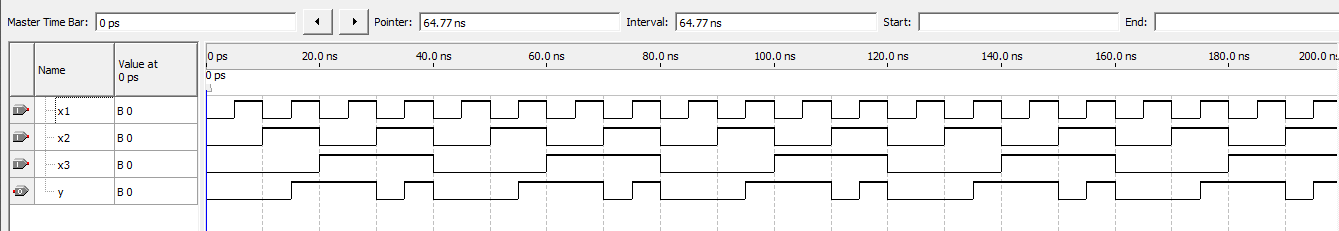
not(k, x2);

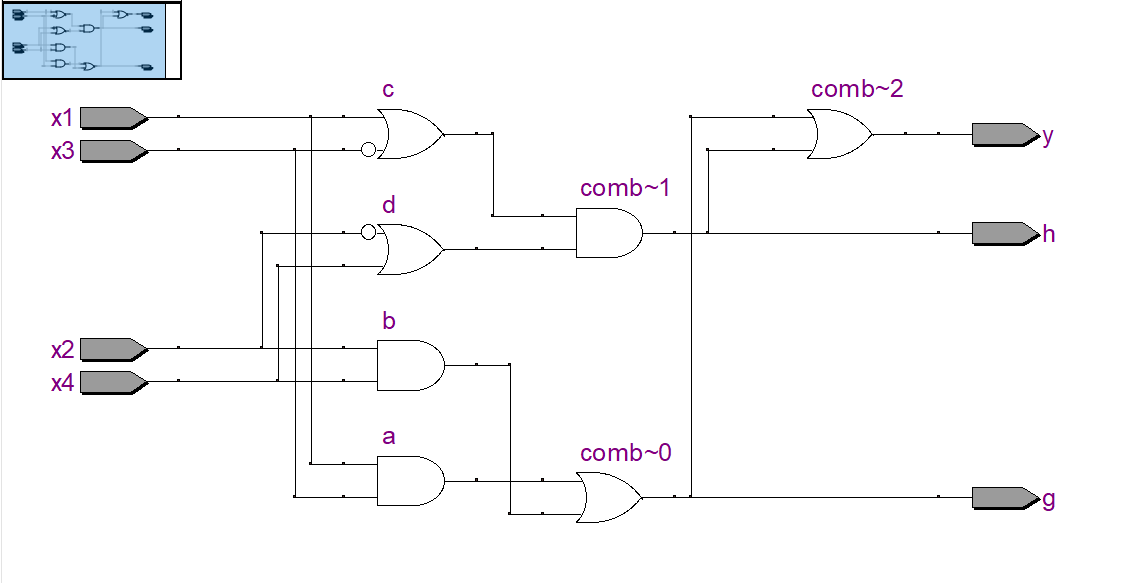
and(h, k, x3);

or(y, g, h);

endmodule

Simulation:



 RTL Viewer:

Câu 2:

Code:

module cau2(x1, x2, x3, x4, y, g, h);

input x1, x2, x3, x4;

output y, g, h;

wire a, b, c, d;

and(a, x1, x3);

and(b, x2, x4);

or(c, x1, ~x3);

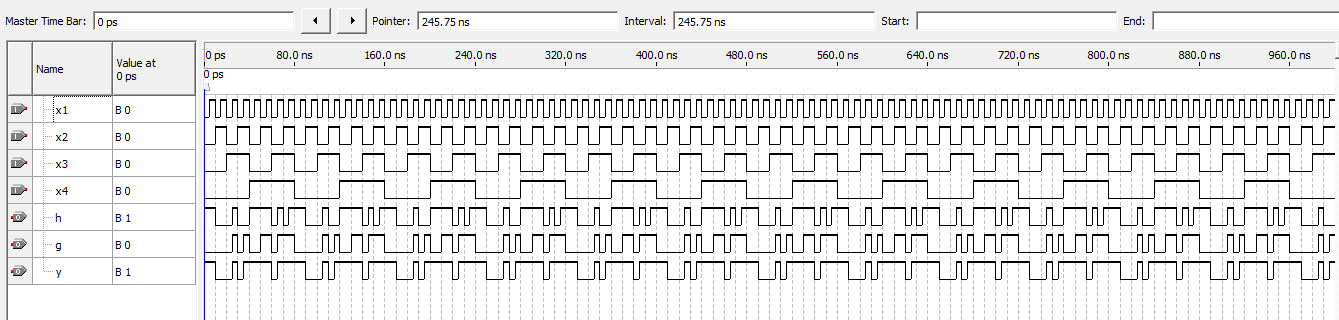
or(d, x4, ~x2);

or(g, a, b);

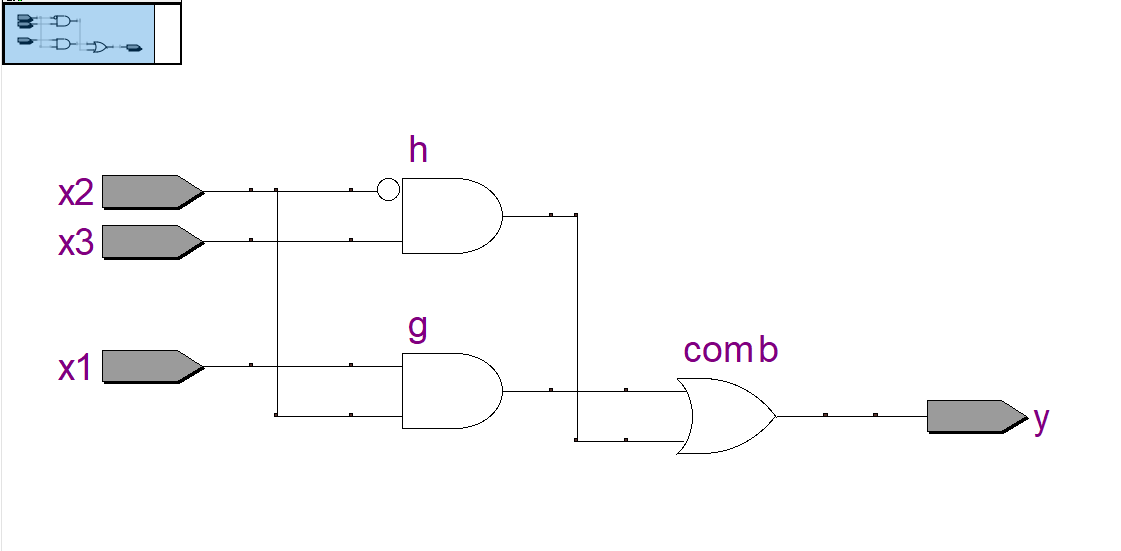
and(h, c, d);

or(y, g, h);

endmodule

Simulation:

RTL Viewer:



Câu 3:

**Blocking**

Code:

module blck (ck,data\_in,data\_out);

input ck;

input [7:0] data\_in;

output [7:0] data\_out;

reg [7:0] data\_out;

reg [7:0] stage\_1;

reg [7:0] stage\_2;

reg [7:0] stage\_3;

always @ (posedge ck) begin

stage\_1 = data\_in;

stage\_2 = stage\_1;

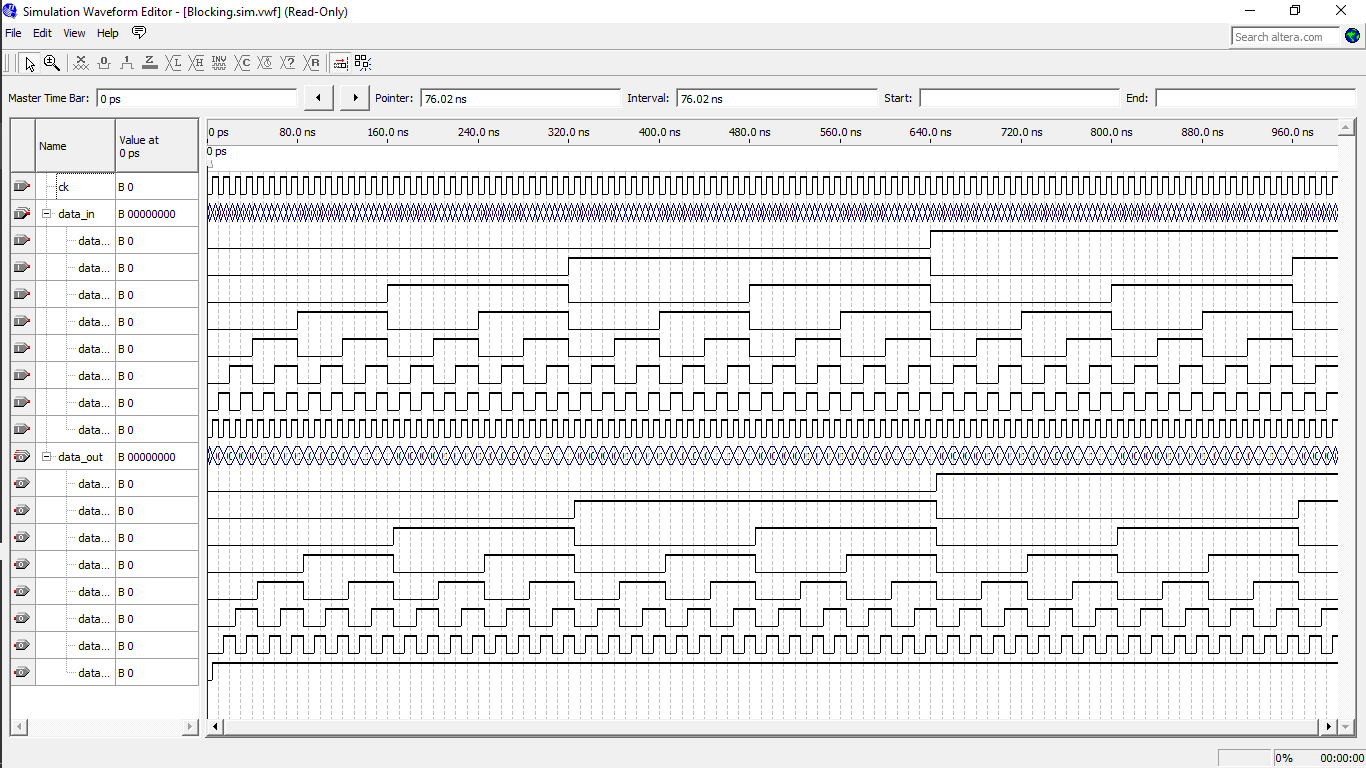
stage\_3 = stage\_2;

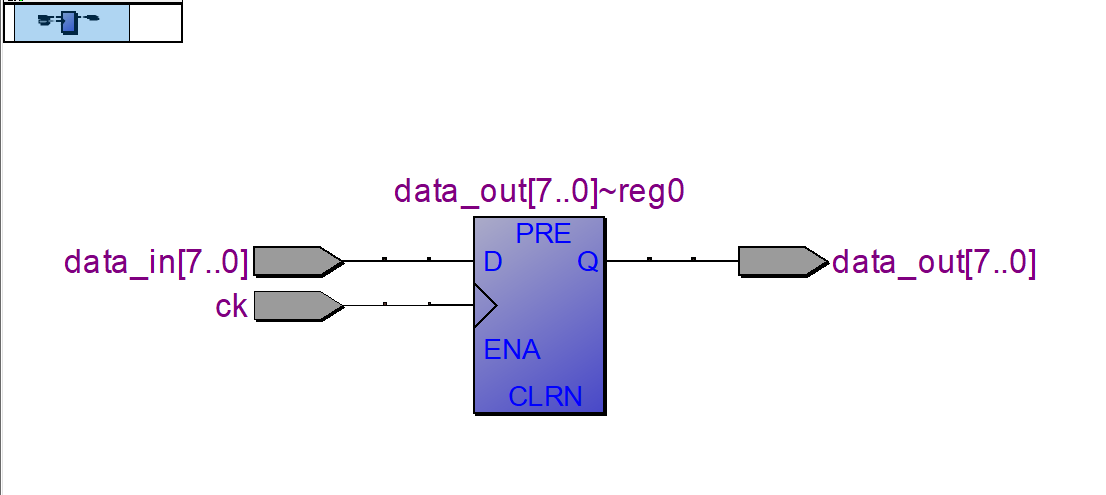
data\_out= stage\_3;

end

endmodule

Simulation:



 RTL Viewer:

**None-Blocking**

Code:

module Non\_Blocking (ck,data\_in,data\_out);

input ck;

input [7:0] data\_in;

output [7:0] data\_out;

reg [7:0] data\_out;

reg [7:0] stage\_1;

reg [7:0] stage\_2;

reg [7:0] stage\_3;

always @ (posedge ck) begin

stage\_1 <= data\_in;

stage\_2 <= stage\_1;

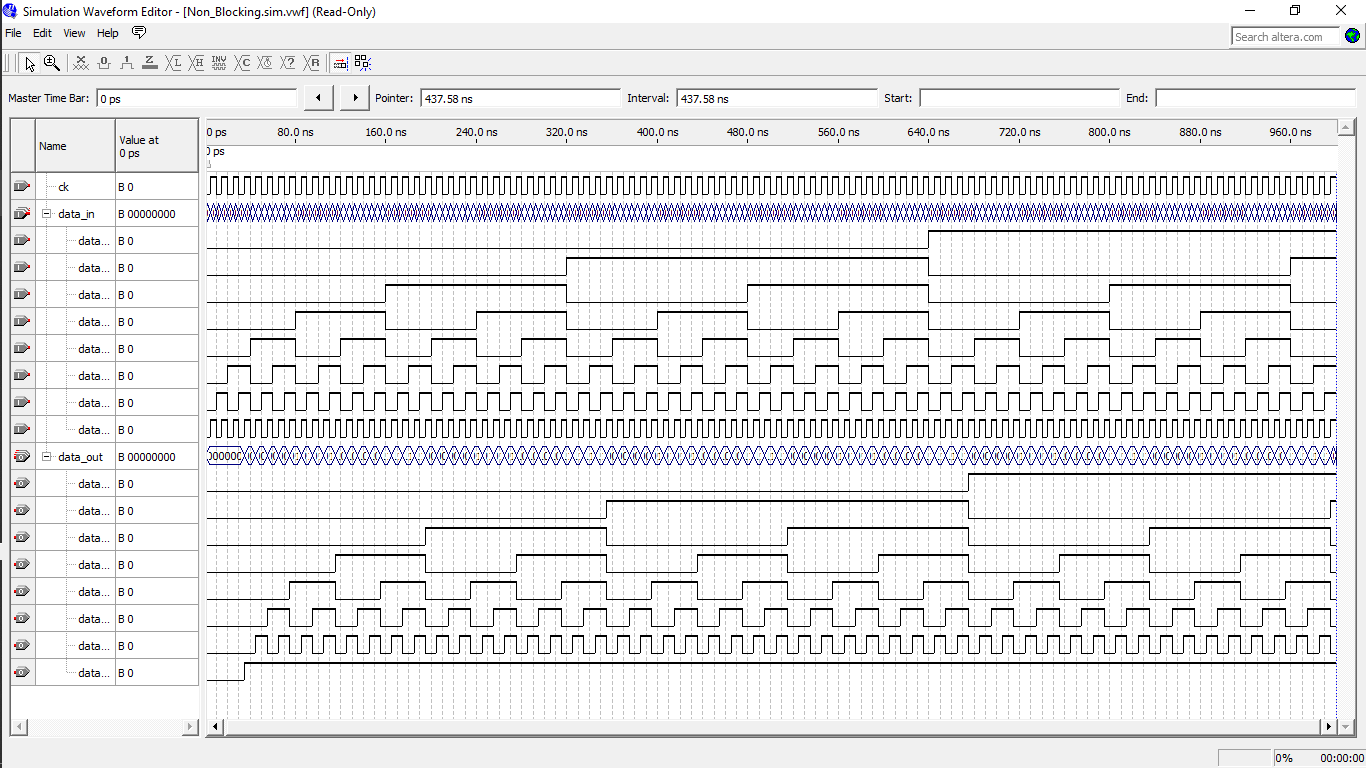
stage\_3 <= stage\_2;

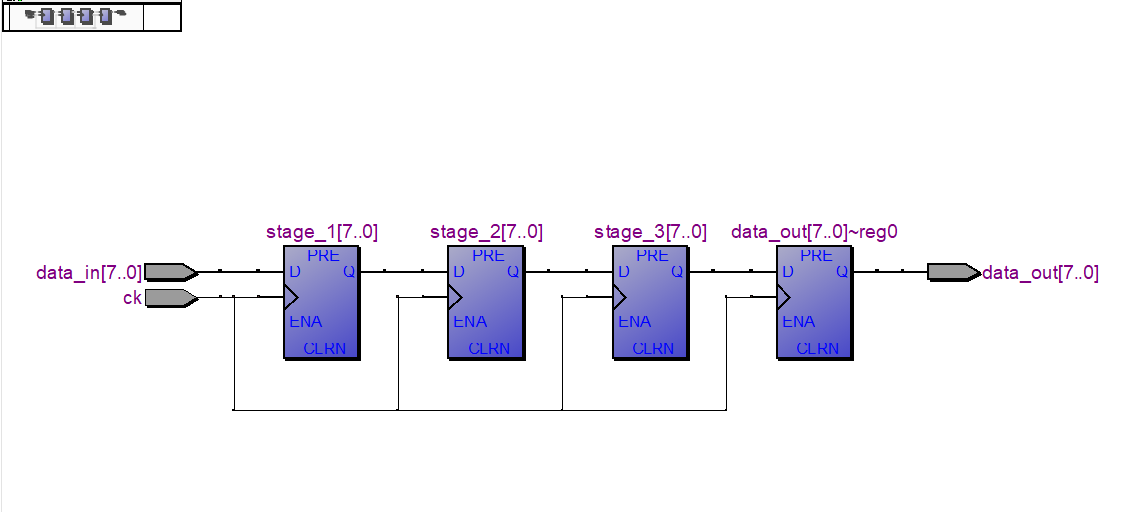
data\_out <= stage\_3;

end

endmodule

Simulation:



 RTL Viewer: