In Lab:

1. Structural description을 이용해 half 74x139를 디자인하고 실행해보았다.

코드:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 19:55:07 04/03/2018

// Design Name:

// Module Name: v74x139h\_s

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module v74x139h\_s(

input G\_L,

input A,

input B,

output [3:0] Y\_L

);

wire N\_A, N\_B, G;

not T1(G, G\_L);

not T2(N\_A, A);

not T3(N\_B, B);

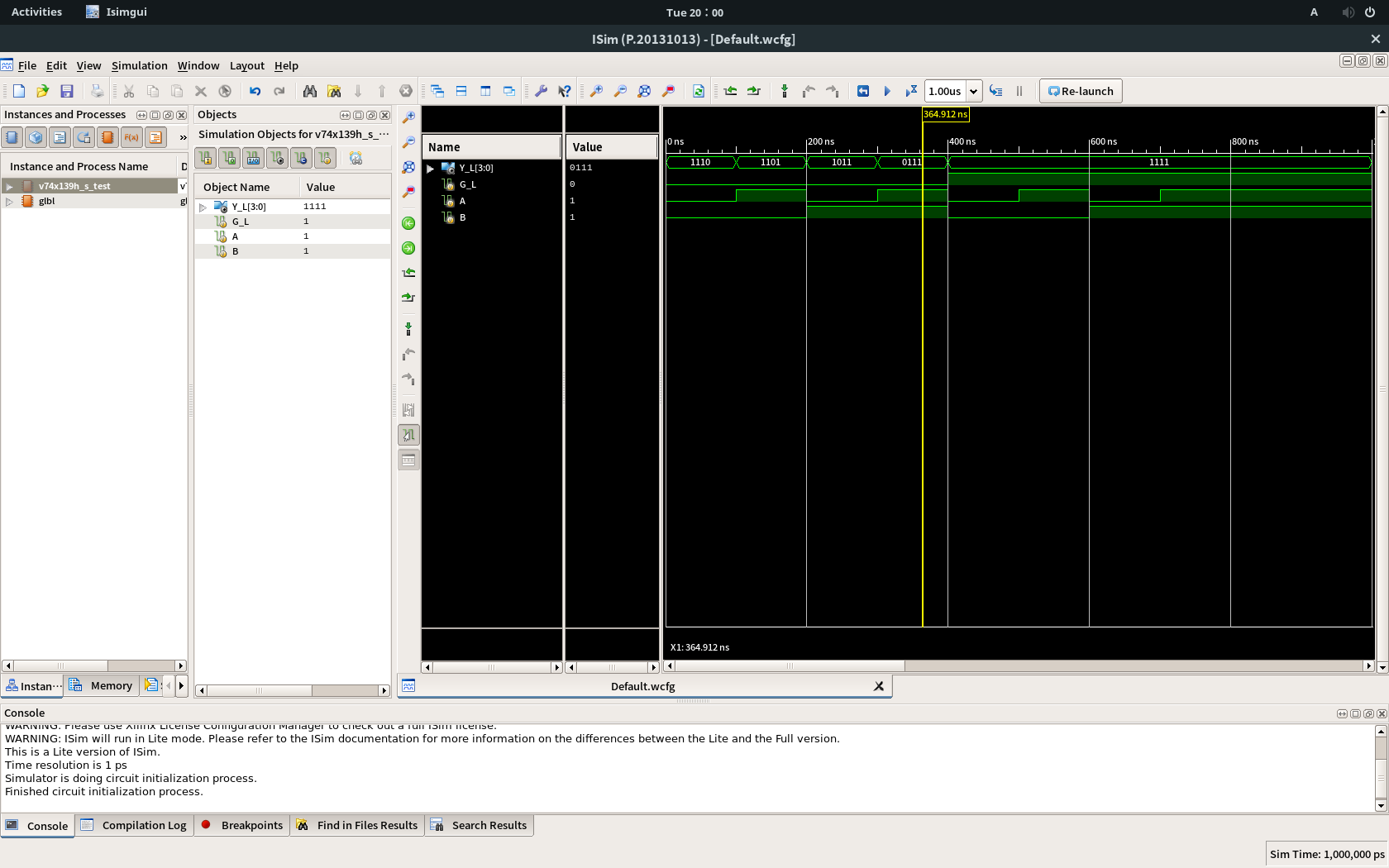
nand T4(Y\_L[0], G, N\_A, N\_B);

nand T5(Y\_L[1], G, A, N\_B);

nand T6(Y\_L[2], G, N\_A, B);

nand T7(Y\_L[3], G, A, B);

endmodule

사진:

2. Data flow description을 이용해 half 74x139를 디자인하고 실행해보았다.

코드:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 20:02:45 04/03/2018

// Design Name:

// Module Name: v74x139h\_d

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module v74x139h\_d(

input G\_L,

input A,

input B,

output [3:0] Y\_L

);

wire [1:0] sel;

wire [3:0] out;

assign sel = {B, A};

assign Y\_L = ~out;

assign out = (sel == 2'b00 && G\_L == 1'b0) ? 4'b0001 :

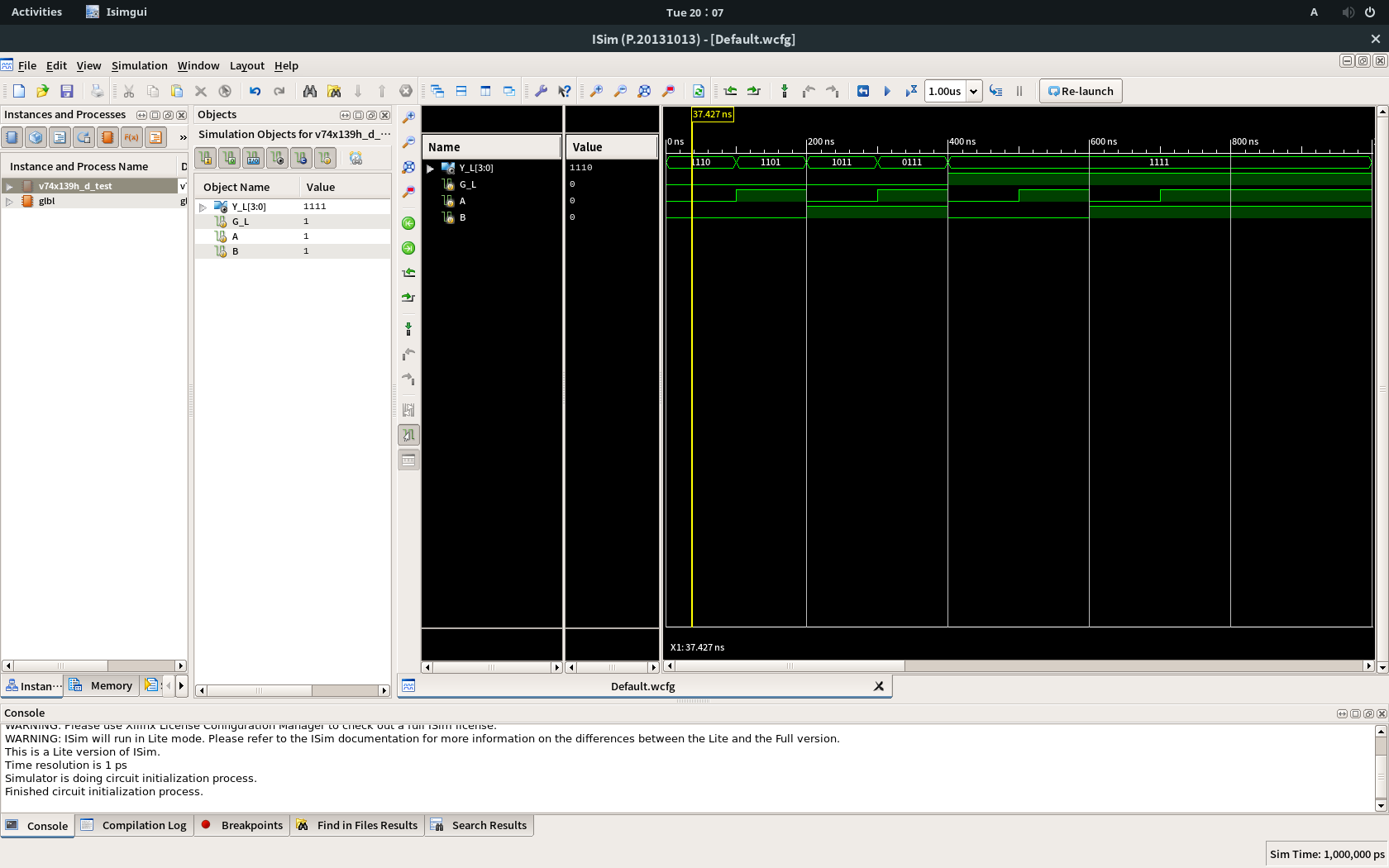
(sel == 2'b01 && G\_L == 1'b0) ? 4'b0010 :

(sel == 2'b10 && G\_L == 1'b0) ? 4'b0100 :

(sel == 2'b11 && G\_L == 1'b0) ? 4'b1000 :

4'b0000;

endmodule

사진:

3. Behavioral description을 이용해 half 74x139를 디자인하고 실행해보았다.

코드:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 20:09:31 04/03/2018

// Design Name:

// Module Name: v74x139h\_b

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module v74x139h\_b(

input G\_L,

input A,

input B,

output [3:0] Y\_L

);

wire [1:0] sel;

reg [3:0] out;

assign sel = {B, A};

assign Y\_L = ~out;

always@(G\_L or sel)

begin

if (G\_L == 1'b0)

begin

case(sel)

2'b00 : out = 4'b0001;

2'b01 : out = 4'b0010;

2'b10 : out = 4'b0100;

2'b11 : out = 4'b1000;

endcase

end

else

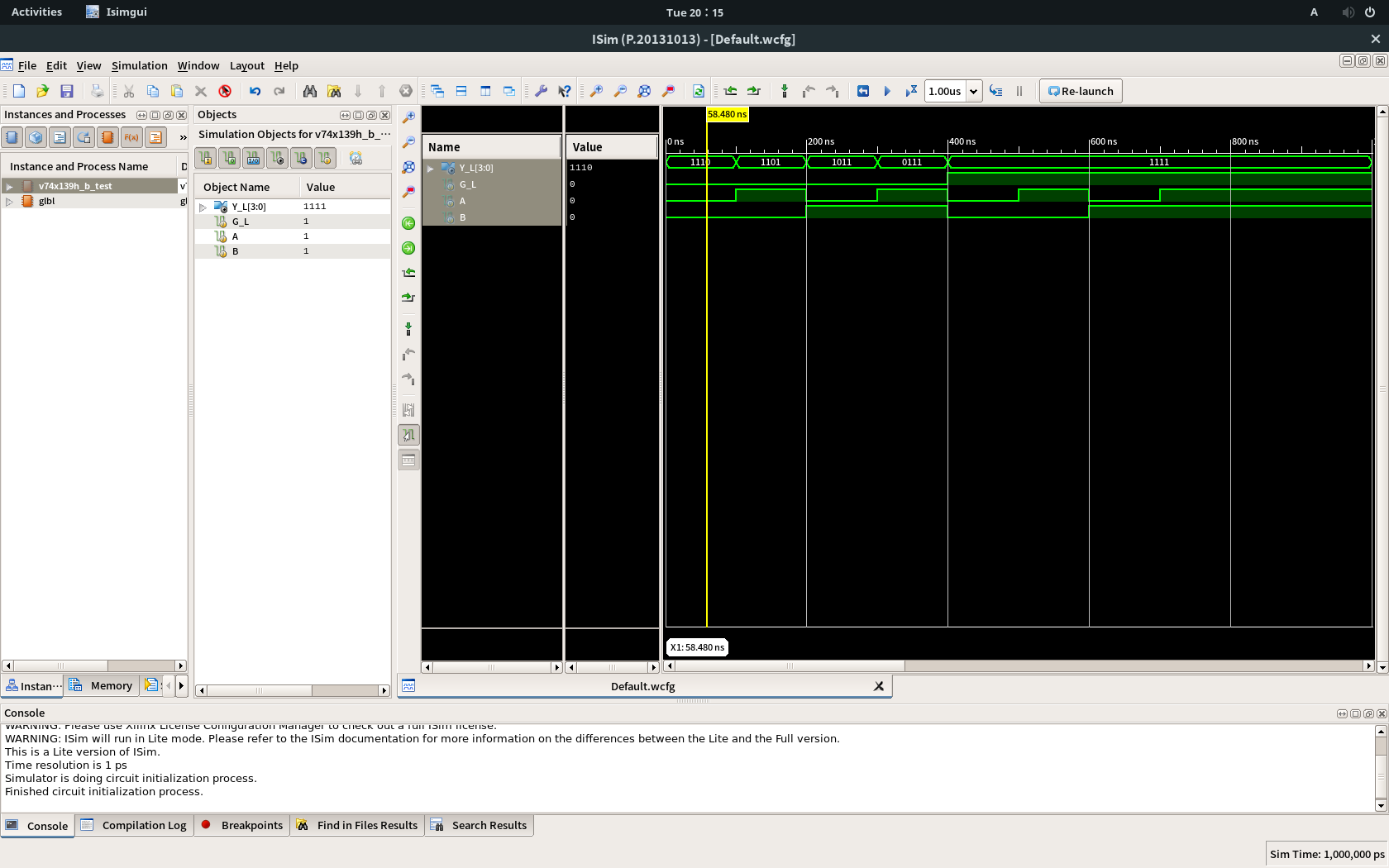
begin

out = 4'b0000;

end

end

endmodule

사진:

부록 1. half 74x139의 테스트코드: 세 종류 모두 동일한 테스트코드 사용

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 19:56:26 04/03/2018

// Design Name: v74x139h\_s

// Module Name: /csehome/pistolstar1797/v74x139/v74x139h\_s\_test.v

// Project Name: v74x139

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: v74x139h\_s

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module v74x139h\_s\_test;

// Inputs

reg G\_L;

reg A;

reg B;

// Outputs

wire [3:0] Y\_L;

// Instantiate the Unit Under Test (UUT)

v74x139h\_s uut (

.G\_L(G\_L),

.A(A),

.B(B),

.Y\_L(Y\_L)

);

initial begin

// Initialize Inputs

G\_L = 0;

A = 0;

B = 0;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

G\_L = 0;

A = 1;

B = 0;

#100 G\_L = 0; A = 0; B = 1;

#100 G\_L = 0; A = 1; B = 1;

#100 G\_L = 1; A = 0; B = 0;

#100 G\_L = 1; A = 1; B = 0;

#100 G\_L = 1; A = 0; B = 1;

#100 G\_L = 1; A = 1; B = 1;

end

endmodule

4. half 74x139 두개를 이용해 74x139를 디자인하고 실행해보았다.

코드:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 20:18:47 04/03/2018

// Design Name:

// Module Name: v74x139

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module v74x139(

input G1\_L,

input G2\_L,

input A1,

input A2,

input B1,

input B2,

output [3:0] Y1\_L,

output [3:0] Y2\_L

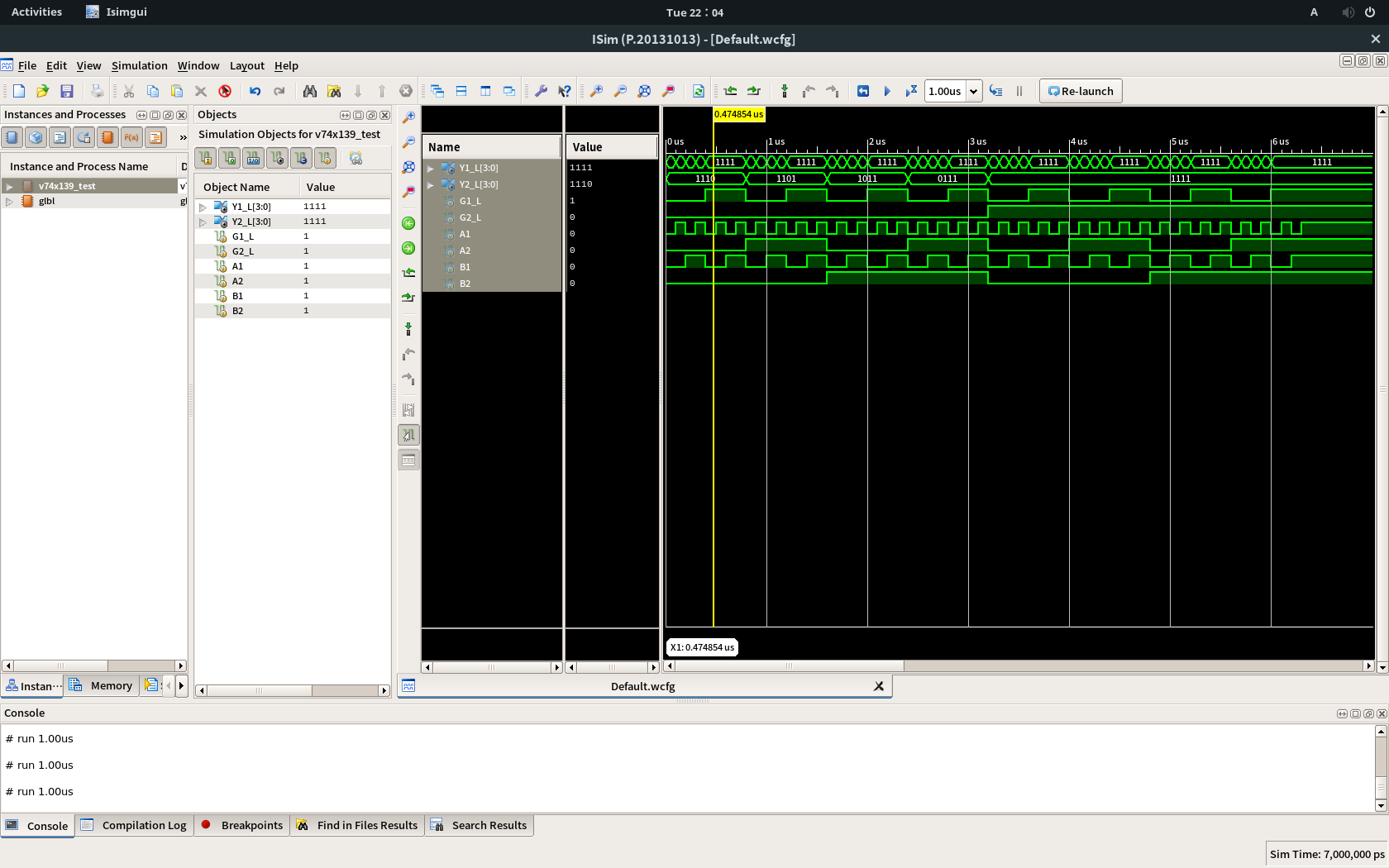
);

v74x139h\_b T1(.G\_L(G1\_L), .A(A1), .B(B1), .Y\_L(Y1\_L));

v74x139h\_b T2(.G\_L(G2\_L), .A(A2), .B(B2), .Y\_L(Y2\_L));

endmodule

사진:



부록 2. 74x139의 테스트코드: for문이 작동하는 줄 모르고 2^6줄을 일일이 손으로 작성함

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 20:24:45 04/03/2018

// Design Name: v74x139

// Module Name: /csehome/pistolstar1797/v74x139/v74x139\_test.v

// Project Name: v74x139

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: v74x139

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module v74x139\_test;

// Inputs

reg G1\_L;

reg G2\_L;

reg A1;

reg A2;

reg B1;

reg B2;

// Outputs

wire [3:0] Y1\_L;

wire [3:0] Y2\_L;

// Instantiate the Unit Under Test (UUT)

v74x139 uut (

.G1\_L(G1\_L),

.G2\_L(G2\_L),

.A1(A1),

.A2(A2),

.B1(B1),

.B2(B2),

.Y1\_L(Y1\_L),

.Y2\_L(Y2\_L)

);

initial begin

// Initialize Inputs

G1\_L = 0;

G2\_L = 0;

A1 = 0;

A2 = 0;

B1 = 0;

B2 = 0;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

G1\_L = 0;

G2\_L = 0;

A1 = 1;

A2 = 0;

B1 = 0;

B2 = 0;

#100 G1\_L = 0; G2\_L = 0; A1 = 0; A2 = 0; B1 = 1; B2 = 0;

#100 G1\_L = 0; G2\_L = 0; A1 = 1; A2 = 0; B1 = 1; B2 = 0;

#100 G1\_L = 1; G2\_L = 0; A1 = 0; A2 = 0; B1 = 0; B2 = 0;

#100 G1\_L = 1; G2\_L = 0; A1 = 1; A2 = 0; B1 = 0; B2 = 0;

#100 G1\_L = 1; G2\_L = 0; A1 = 0; A2 = 0; B1 = 1; B2 = 0;

#100 G1\_L = 1; G2\_L = 0; A1 = 1; A2 = 0; B1 = 1; B2 = 0;

#100 G1\_L = 0; G2\_L = 0; A1 = 0; A2 = 1; B1 = 0; B2 = 0;

#100 G1\_L = 0; G2\_L = 0; A1 = 1; A2 = 1; B1 = 0; B2 = 0;

#100 G1\_L = 0; G2\_L = 0; A1 = 0; A2 = 1; B1 = 1; B2 = 0;

#100 G1\_L = 0; G2\_L = 0; A1 = 1; A2 = 1; B1 = 1; B2 = 0;

#100 G1\_L = 1; G2\_L = 0; A1 = 0; A2 = 1; B1 = 0; B2 = 0;

#100 G1\_L = 1; G2\_L = 0; A1 = 1; A2 = 1; B1 = 0; B2 = 0;

#100 G1\_L = 1; G2\_L = 0; A1 = 0; A2 = 1; B1 = 1; B2 = 0;

#100 G1\_L = 1; G2\_L = 0; A1 = 1; A2 = 1; B1 = 1; B2 = 0;

#100 G1\_L = 0; G2\_L = 0; A1 = 0; A2 = 0; B1 = 0; B2 = 1;

#100 G1\_L = 0; G2\_L = 0; A1 = 1; A2 = 0; B1 = 0; B2 = 1;

#100 G1\_L = 0; G2\_L = 0; A1 = 0; A2 = 0; B1 = 1; B2 = 1;

#100 G1\_L = 0; G2\_L = 0; A1 = 1; A2 = 0; B1 = 1; B2 = 1;

#100 G1\_L = 1; G2\_L = 0; A1 = 0; A2 = 0; B1 = 0; B2 = 1;

#100 G1\_L = 1; G2\_L = 0; A1 = 1; A2 = 0; B1 = 0; B2 = 1;

#100 G1\_L = 1; G2\_L = 0; A1 = 0; A2 = 0; B1 = 1; B2 = 1;

#100 G1\_L = 1; G2\_L = 0; A1 = 1; A2 = 0; B1 = 1; B2 = 1;

#100 G1\_L = 0; G2\_L = 0; A1 = 0; A2 = 1; B1 = 0; B2 = 1;

#100 G1\_L = 0; G2\_L = 0; A1 = 1; A2 = 1; B1 = 0; B2 = 1;

#100 G1\_L = 0; G2\_L = 0; A1 = 0; A2 = 1; B1 = 1; B2 = 1;

#100 G1\_L = 0; G2\_L = 0; A1 = 1; A2 = 1; B1 = 1; B2 = 1;

#100 G1\_L = 1; G2\_L = 0; A1 = 0; A2 = 1; B1 = 0; B2 = 1;

#100 G1\_L = 1; G2\_L = 0; A1 = 1; A2 = 1; B1 = 0; B2 = 1;

#100 G1\_L = 1; G2\_L = 0; A1 = 0; A2 = 1; B1 = 1; B2 = 1;

#100 G1\_L = 1; G2\_L = 0; A1 = 1; A2 = 1; B1 = 1; B2 = 1;

#100 G1\_L = 0; G2\_L = 1; A1 = 0; A2 = 0; B1 = 0; B2 = 0;

#100 G1\_L = 0; G2\_L = 1; A1 = 1; A2 = 0; B1 = 0; B2 = 0;

#100 G1\_L = 0; G2\_L = 1; A1 = 0; A2 = 0; B1 = 1; B2 = 0;

#100 G1\_L = 0; G2\_L = 1; A1 = 1; A2 = 0; B1 = 1; B2 = 0;

#100 G1\_L = 1; G2\_L = 1; A1 = 0; A2 = 0; B1 = 0; B2 = 0;

#100 G1\_L = 1; G2\_L = 1; A1 = 1; A2 = 0; B1 = 0; B2 = 0;

#100 G1\_L = 1; G2\_L = 1; A1 = 0; A2 = 0; B1 = 1; B2 = 0;

#100 G1\_L = 1; G2\_L = 1; A1 = 1; A2 = 0; B1 = 1; B2 = 0;

#100 G1\_L = 0; G2\_L = 1; A1 = 0; A2 = 1; B1 = 0; B2 = 0;

#100 G1\_L = 0; G2\_L = 1; A1 = 1; A2 = 1; B1 = 0; B2 = 0;

#100 G1\_L = 0; G2\_L = 1; A1 = 0; A2 = 1; B1 = 1; B2 = 0;

#100 G1\_L = 0; G2\_L = 1; A1 = 1; A2 = 1; B1 = 1; B2 = 0;

#100 G1\_L = 1; G2\_L = 1; A1 = 0; A2 = 1; B1 = 0; B2 = 0;

#100 G1\_L = 1; G2\_L = 1; A1 = 1; A2 = 1; B1 = 0; B2 = 0;

#100 G1\_L = 1; G2\_L = 1; A1 = 0; A2 = 1; B1 = 1; B2 = 0;

#100 G1\_L = 1; G2\_L = 1; A1 = 1; A2 = 1; B1 = 1; B2 = 0;

#100 G1\_L = 0; G2\_L = 1; A1 = 0; A2 = 0; B1 = 0; B2 = 1;

#100 G1\_L = 0; G2\_L = 1; A1 = 1; A2 = 0; B1 = 0; B2 = 1;

#100 G1\_L = 0; G2\_L = 1; A1 = 0; A2 = 0; B1 = 1; B2 = 1;

#100 G1\_L = 0; G2\_L = 1; A1 = 1; A2 = 0; B1 = 1; B2 = 1;

#100 G1\_L = 1; G2\_L = 1; A1 = 0; A2 = 0; B1 = 0; B2 = 1;

#100 G1\_L = 1; G2\_L = 1; A1 = 1; A2 = 0; B1 = 0; B2 = 1;

#100 G1\_L = 1; G2\_L = 1; A1 = 0; A2 = 0; B1 = 1; B2 = 1;

#100 G1\_L = 1; G2\_L = 1; A1 = 1; A2 = 0; B1 = 1; B2 = 1;

#100 G1\_L = 0; G2\_L = 1; A1 = 0; A2 = 1; B1 = 0; B2 = 1;

#100 G1\_L = 0; G2\_L = 1; A1 = 1; A2 = 1; B1 = 0; B2 = 1;

#100 G1\_L = 0; G2\_L = 1; A1 = 0; A2 = 1; B1 = 1; B2 = 1;

#100 G1\_L = 0; G2\_L = 1; A1 = 1; A2 = 1; B1 = 1; B2 = 1;

#100 G1\_L = 1; G2\_L = 1; A1 = 0; A2 = 1; B1 = 0; B2 = 1;

#100 G1\_L = 1; G2\_L = 1; A1 = 1; A2 = 1; B1 = 0; B2 = 1;

#100 G1\_L = 1; G2\_L = 1; A1 = 0; A2 = 1; B1 = 1; B2 = 1;

#100 G1\_L = 1; G2\_L = 1; A1 = 1; A2 = 1; B1 = 1; B2 = 1;

end

endmodule

5. half 74x139 (2-to-4 decoder)를 응용하여 3-to-8 decoder를 디자인하고 실행해보았다.

코드:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 20:55:17 04/03/2018

// Design Name:

// Module Name: v3to8decoder

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module v3to8decoder(

input G\_L,

input A,

input B,

input C,

output [7:0] Y\_L

);

wire [3:0] out1;

wire [3:0] out2;

assign Y\_L[3:0] = out1;

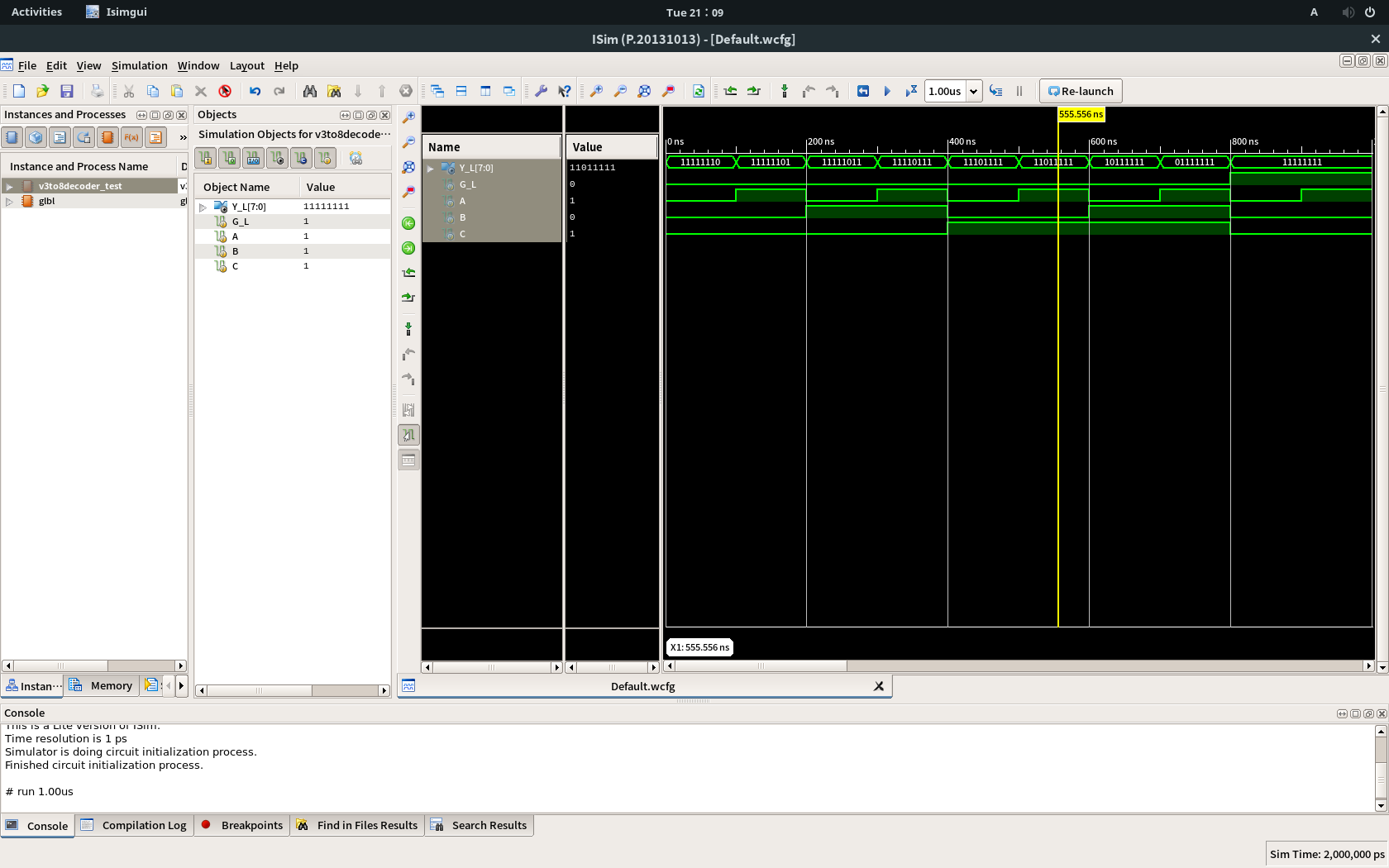
assign Y\_L[7:4] = out2;

v74x139h\_b T1(.G\_L(G\_L | C), .A(A), .B(B), .Y\_L(out1));

v74x139h\_b T2(.G\_L(G\_L | ~C), .A(A), .B(B), .Y\_L(out2));

endmodule

사진:



부록 3. 3-to-8 decoder의 테스트코드:

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 20:56:07 04/03/2018

// Design Name: v3to8decoder

// Module Name: /csehome/pistolstar1797/v74x139/v3to8decoder\_test.v

// Project Name: v74x139

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: v3to8decoder

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module v3to8decoder\_test;

// Inputs

reg G\_L;

reg A;

reg B;

reg C;

// Outputs

wire [7:0] Y\_L;

// Instantiate the Unit Under Test (UUT)

v3to8decoder uut (

.G\_L(G\_L),

.A(A),

.B(B),

.C(C),

.Y\_L(Y\_L)

);

initial begin

// Initialize Inputs

G\_L = 0;

A = 0;

B = 0;

C = 0;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

G\_L = 0;

A = 1;

B = 0;

C = 0;

#100 G\_L = 0; A = 0; B = 1; C = 0;

#100 G\_L = 0; A = 1; B = 1; C = 0;

#100 G\_L = 0; A = 0; B = 0; C = 1;

#100 G\_L = 0; A = 1; B = 0; C = 1;

#100 G\_L = 0; A = 0; B = 1; C = 1;

#100 G\_L = 0; A = 1; B = 1; C = 1;

#100 G\_L = 1; A = 0; B = 0; C = 0;

#100 G\_L = 1; A = 1; B = 0; C = 0;

#100 G\_L = 1; A = 0; B = 1; C = 0;

#100 G\_L = 1; A = 1; B = 1; C = 0;

#100 G\_L = 1; A = 0; B = 0; C = 1;

#100 G\_L = 1; A = 1; B = 0; C = 1;

#100 G\_L = 1; A = 0; B = 1; C = 1;

#100 G\_L = 1; A = 1; B = 1; C = 1;

end

endmodule

Homework:

1. Structural description을 이용해 4-to-1 MUX를 디자인하고 실행해보았다.

코드:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 21:13:14 04/03/2018

// Design Name:

// Module Name: v4to1MUX\_s

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module v4to1MUX\_s(

input [3:0] X,

input [1:0] C,

output Y

);

wire [3:0] T;

wire [1:0] N\_C;

not (N\_C[0], C[0]);

not (N\_C[1], C[1]);

and (T[0], X[0], N\_C[1], N\_C[0]);

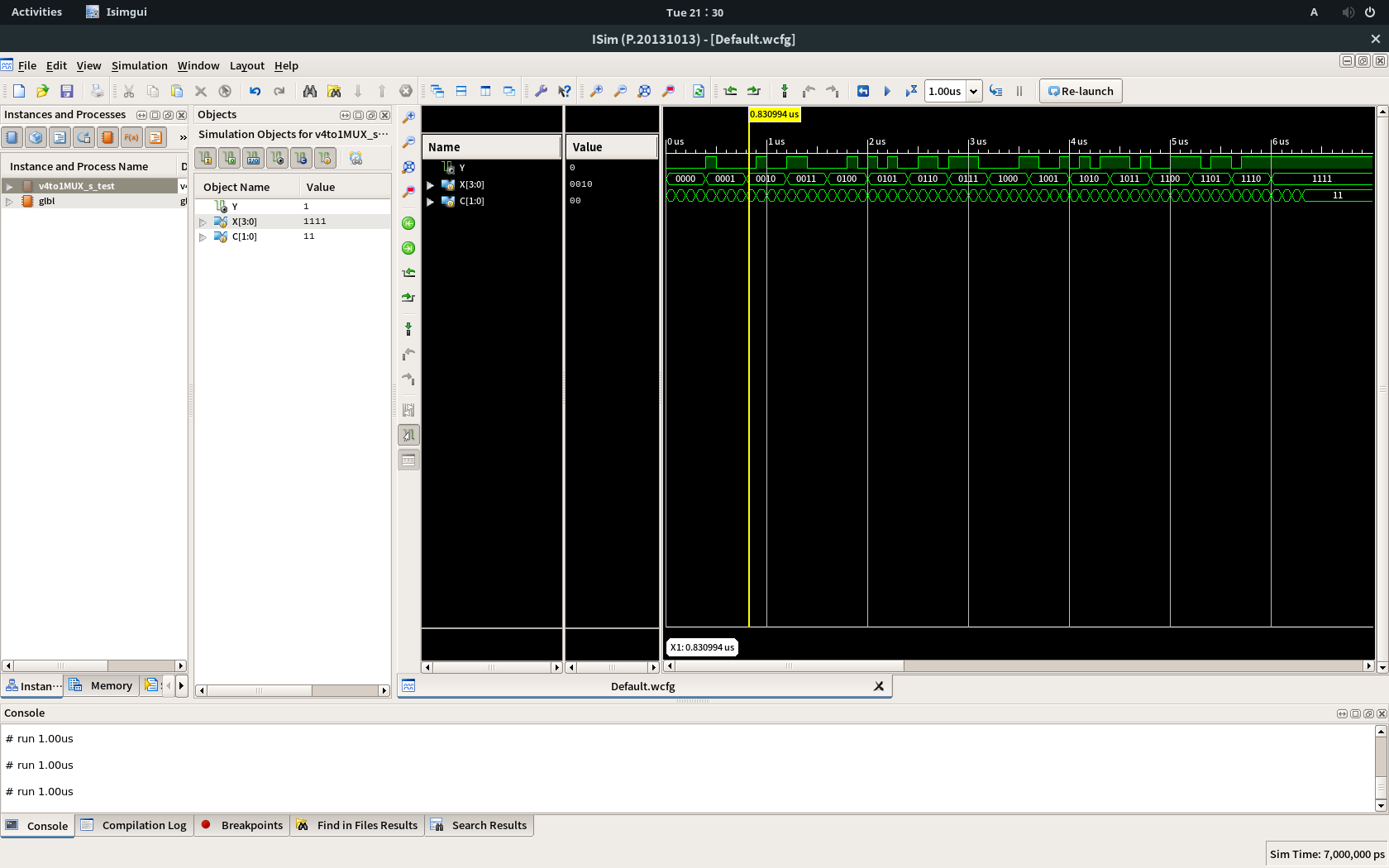
and (T[1], X[1], N\_C[1], C[0]);

and (T[2], X[2], C[1], N\_C[0]);

and (T[3], X[3], C[1], C[0]);

or (Y, T[0], T[1], T[2], T[3]);

endmodule

사진:

2. Data flow description을 이용해 4-to-1 MUX를 디자인하고 실행해보았다.

코드:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 21:33:23 04/03/2018

// Design Name:

// Module Name: v4to1MUX\_d

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module v4to1MUX\_d(

input [3:0] X,

input [1:0] C,

output Y

);

assign Y = (X[0] == 1 && C == 2'b00) ? 1'b1 :

(X[1] == 1 && C == 2'b01) ? 1'b1 :

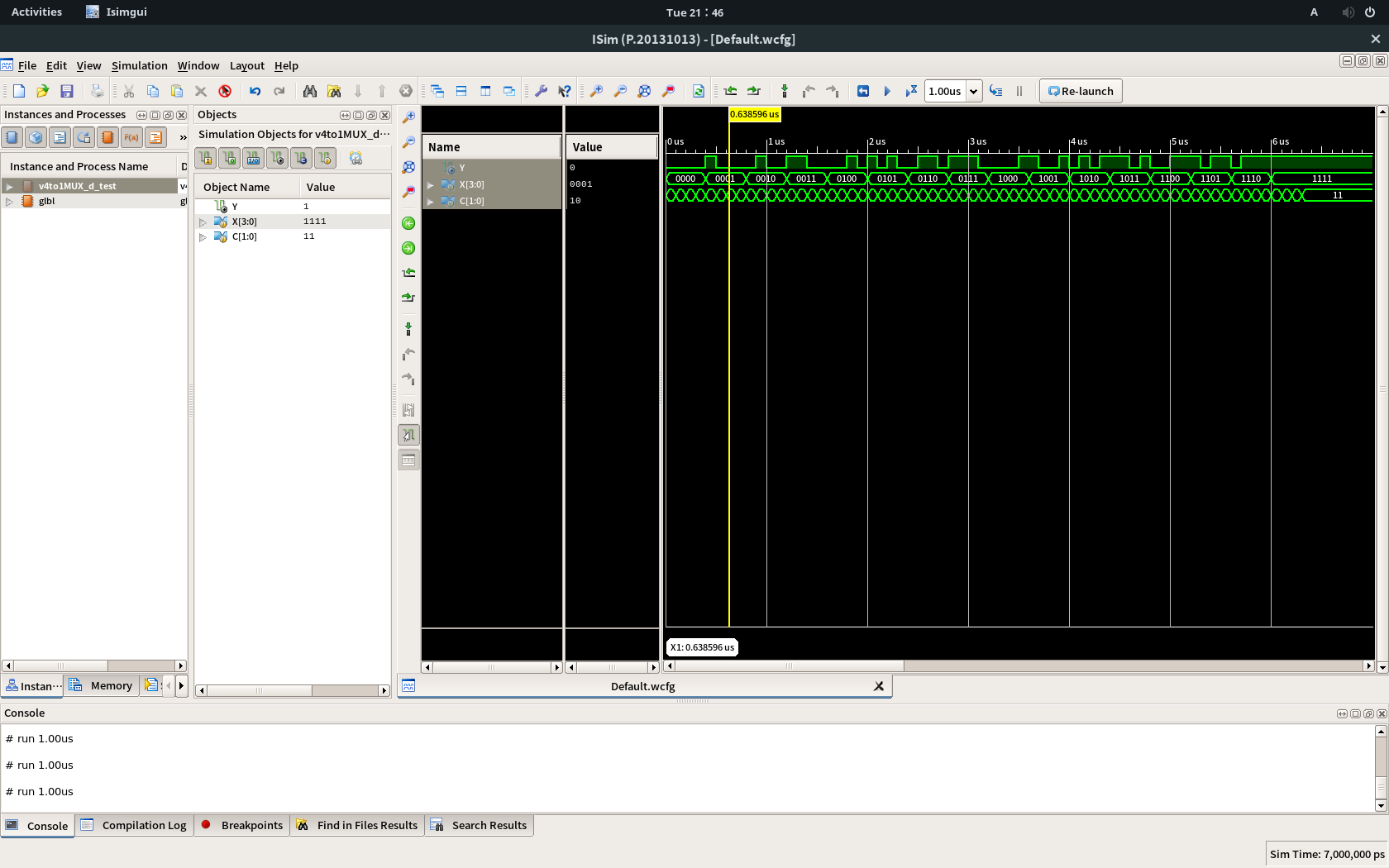
(X[2] == 1 && C == 2'b10) ? 1'b1 :

(X[3] == 1 && C == 2'b11) ? 1'b1 :

1'b0;

endmodule

사진:



3. Behavioral description을 이용해 4-to-1 MUX를 디자인하고 실행해보았다.

코드:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 21:47:40 04/03/2018

// Design Name:

// Module Name: v4to1MUX\_b

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module v4to1MUX\_b(

input [3:0] X,

input [1:0] C,

output Y

);

reg T;

assign Y = T;

always@(X or C)

begin

if(X[C]==1)

begin

T = 1'b1;

end

else

begin

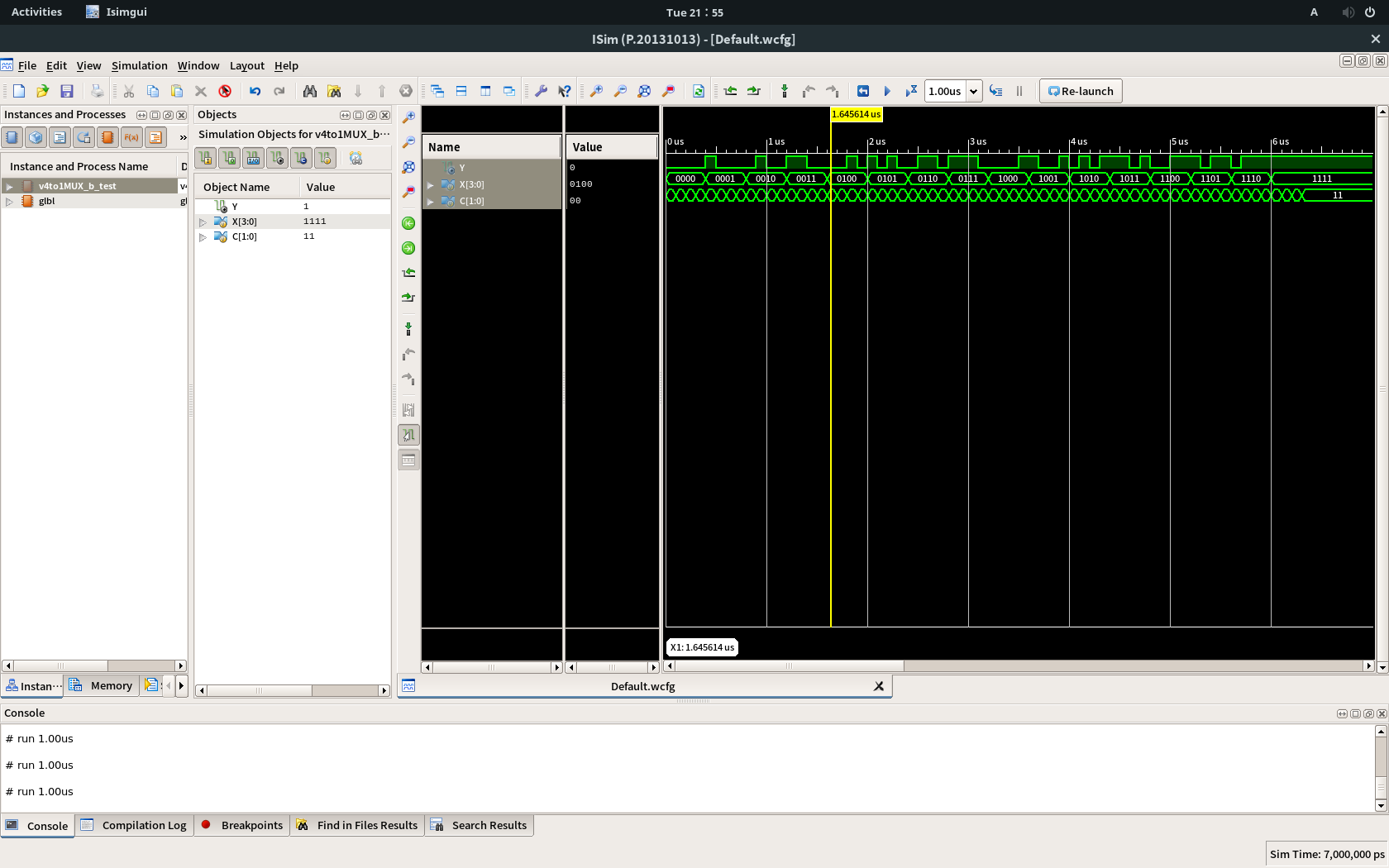
T = 1'b0;

end

end

endmodule

사진:



4. Structural description의 특징: Schematic diagram을 그대로 코드로 작성하는 방법

장점: schematic diagram을 안다면 틀릴 일이 없다.

단점: 게이트의 개수가 많아지면 코드가 길고 복잡해질 수 있다.

5. Data flow description의 특징: Truth table을 그대로 코드로 작성하는 방법

장점: truth table을 안다면 틀릴 일이 없다.

단점: 입/출력 값이 많아지면 코드가 기하급수적으로 길어진다.

6. Behavioral description의 특징: module이 무슨 기능을 하는지를 논리적으로 작성하는 방법

장점: 알고리즘을 잘 파악하고 있다면 짧고 효율적인 코드를 작성할 수 있다.

단점: 알고리즘을 제대로 파악하지 못하면 틀린 결과값을 출력하거나 코드가 산으로 갈 수 있다.