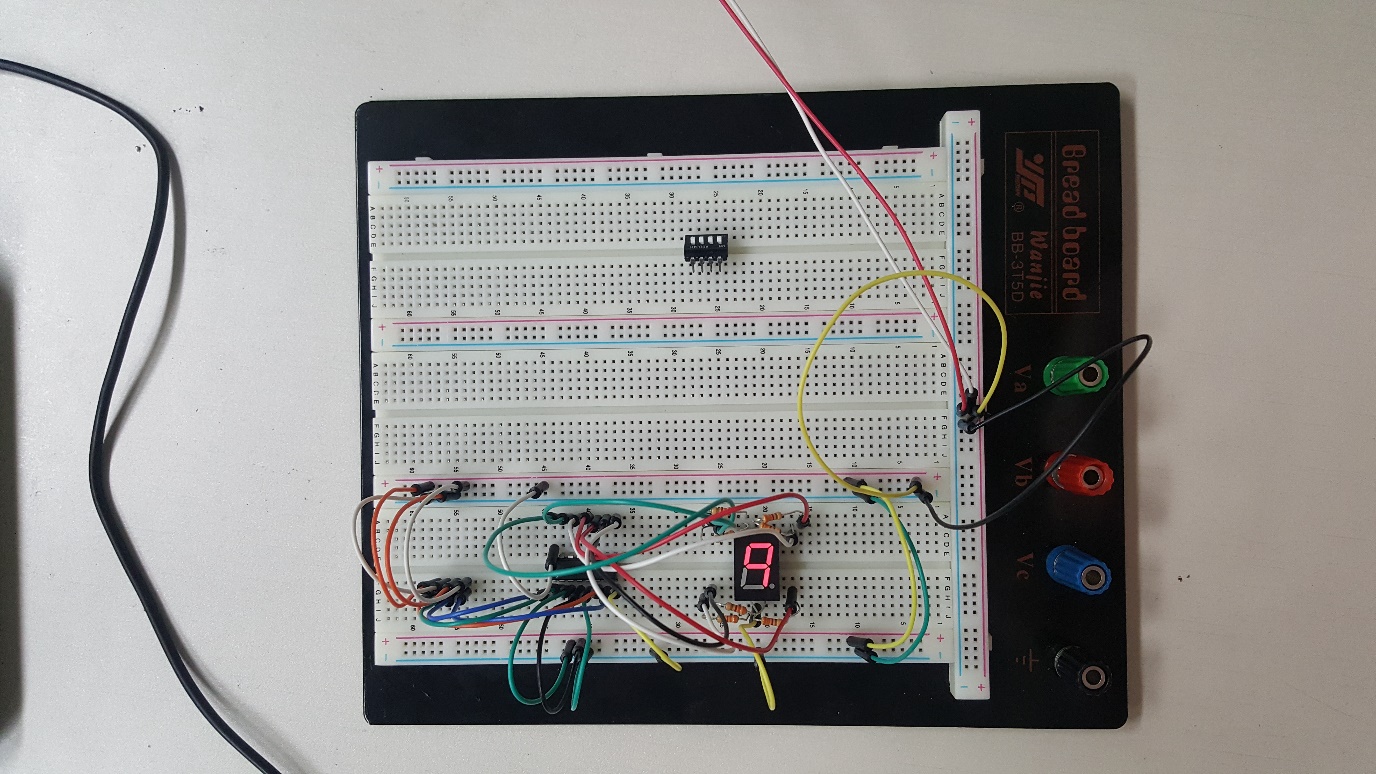
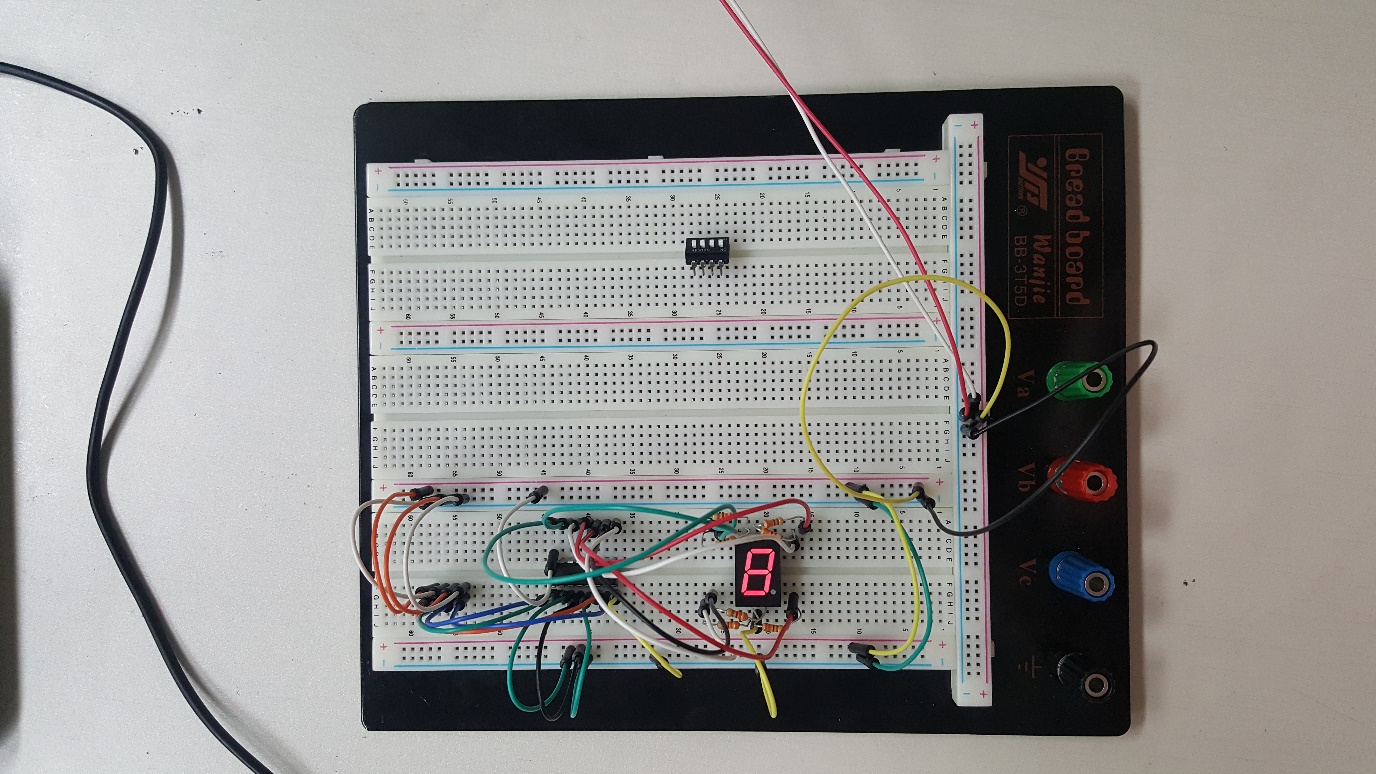
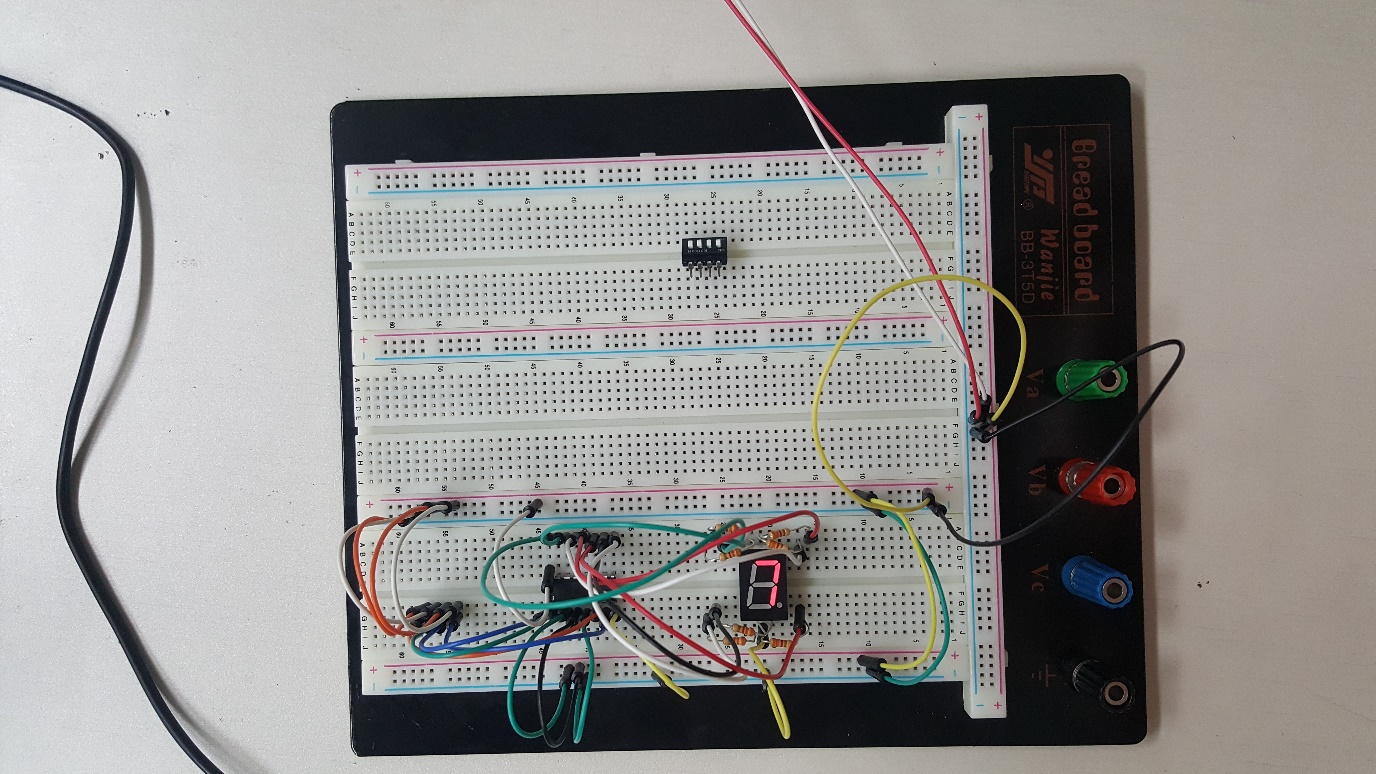
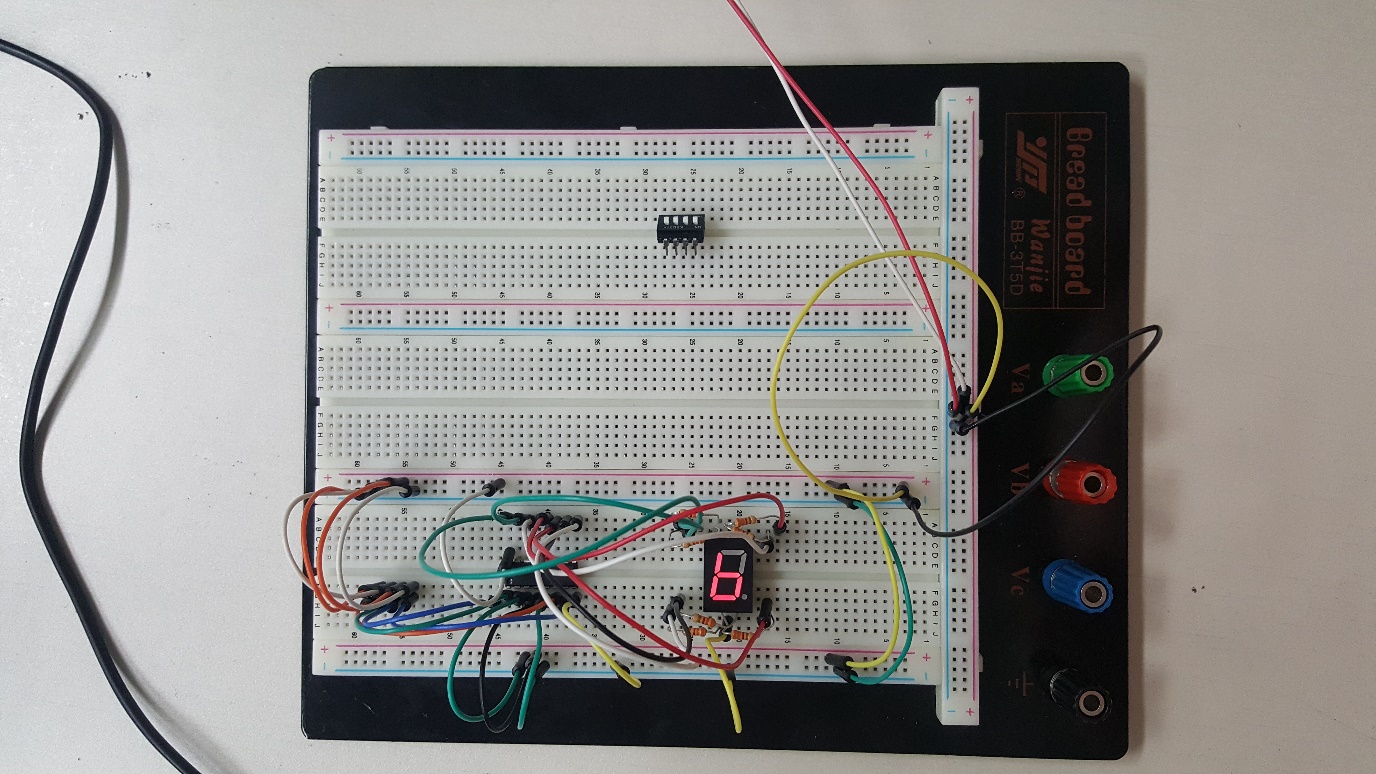
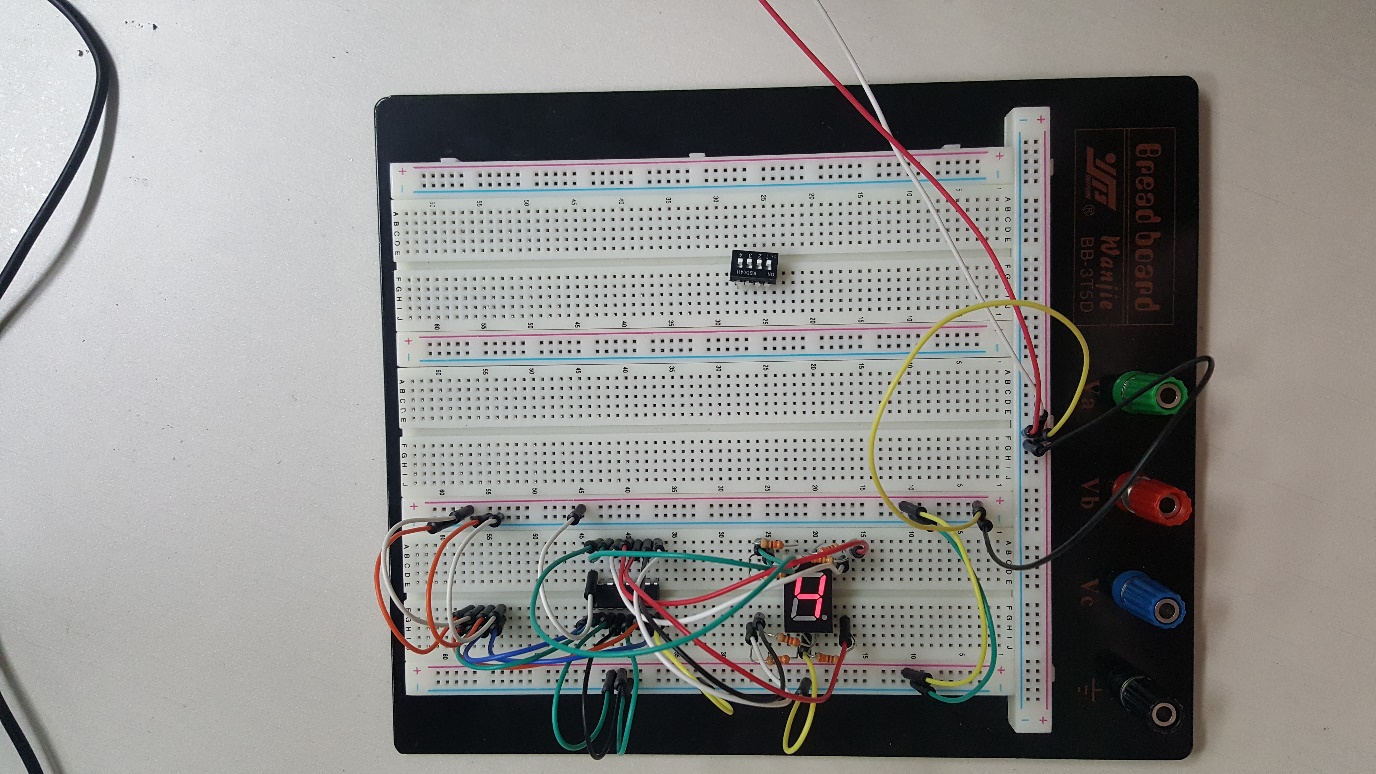
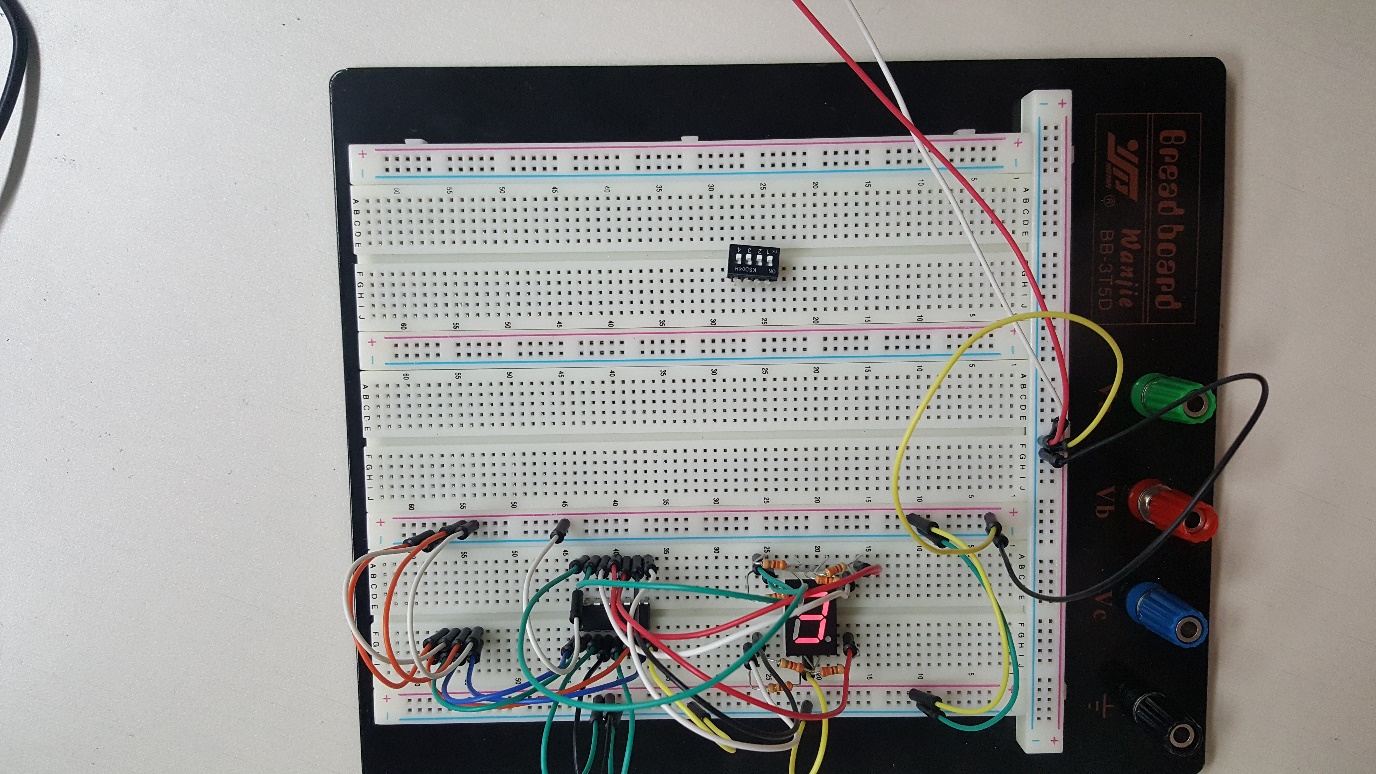
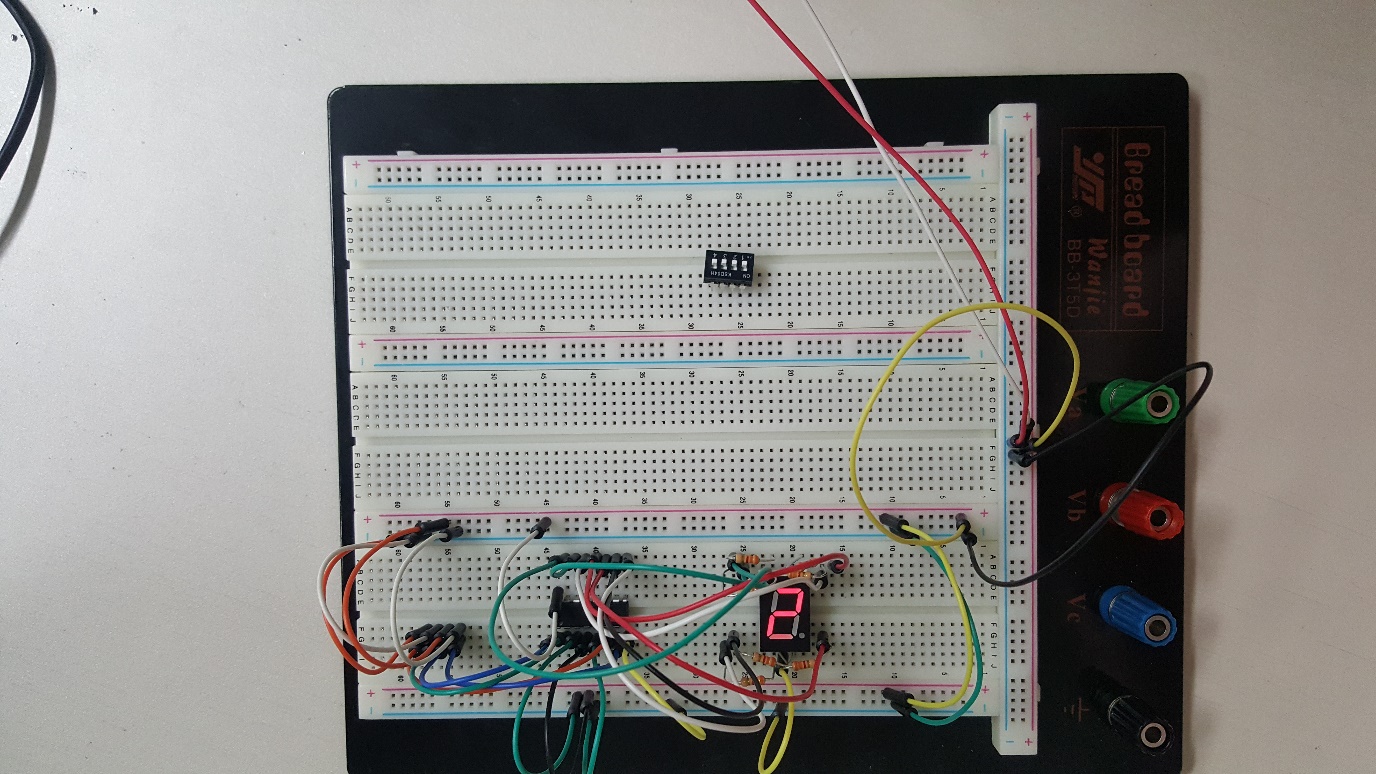
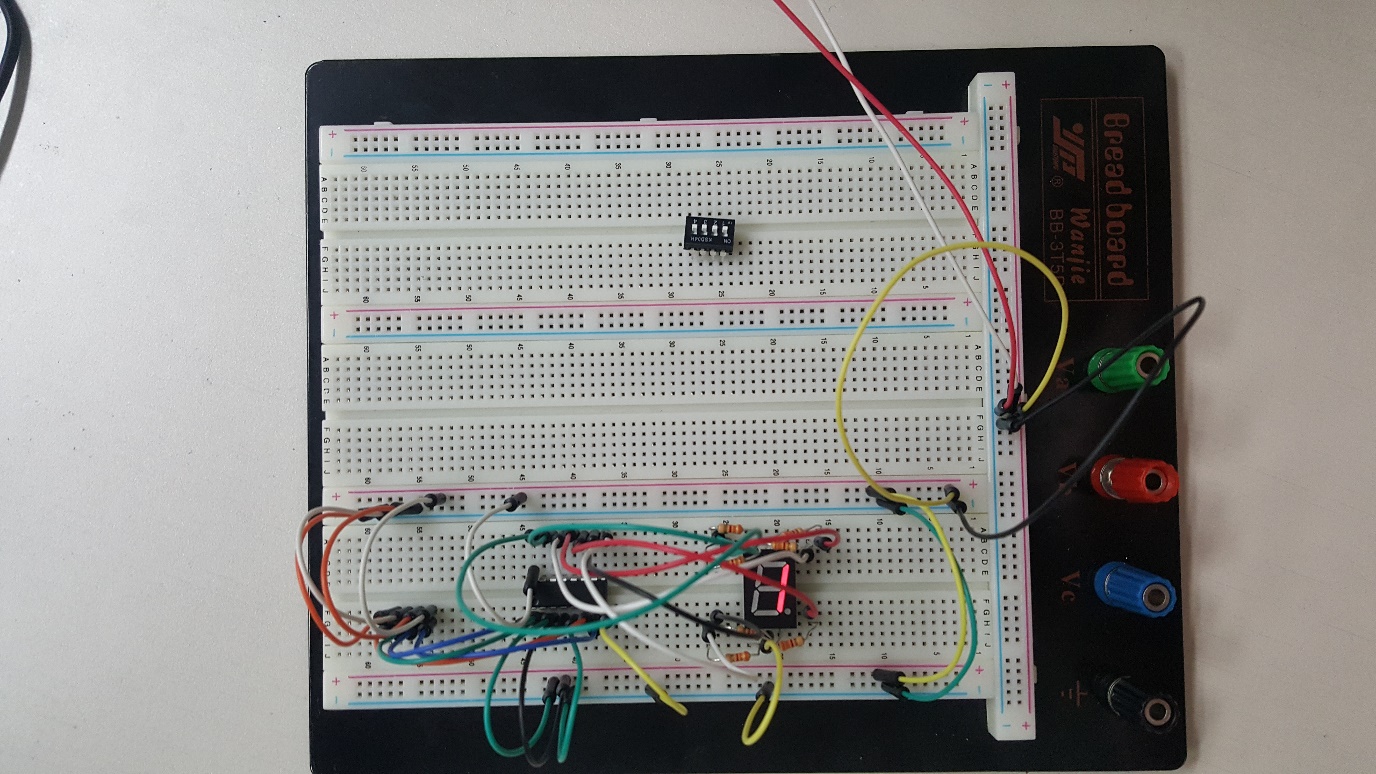
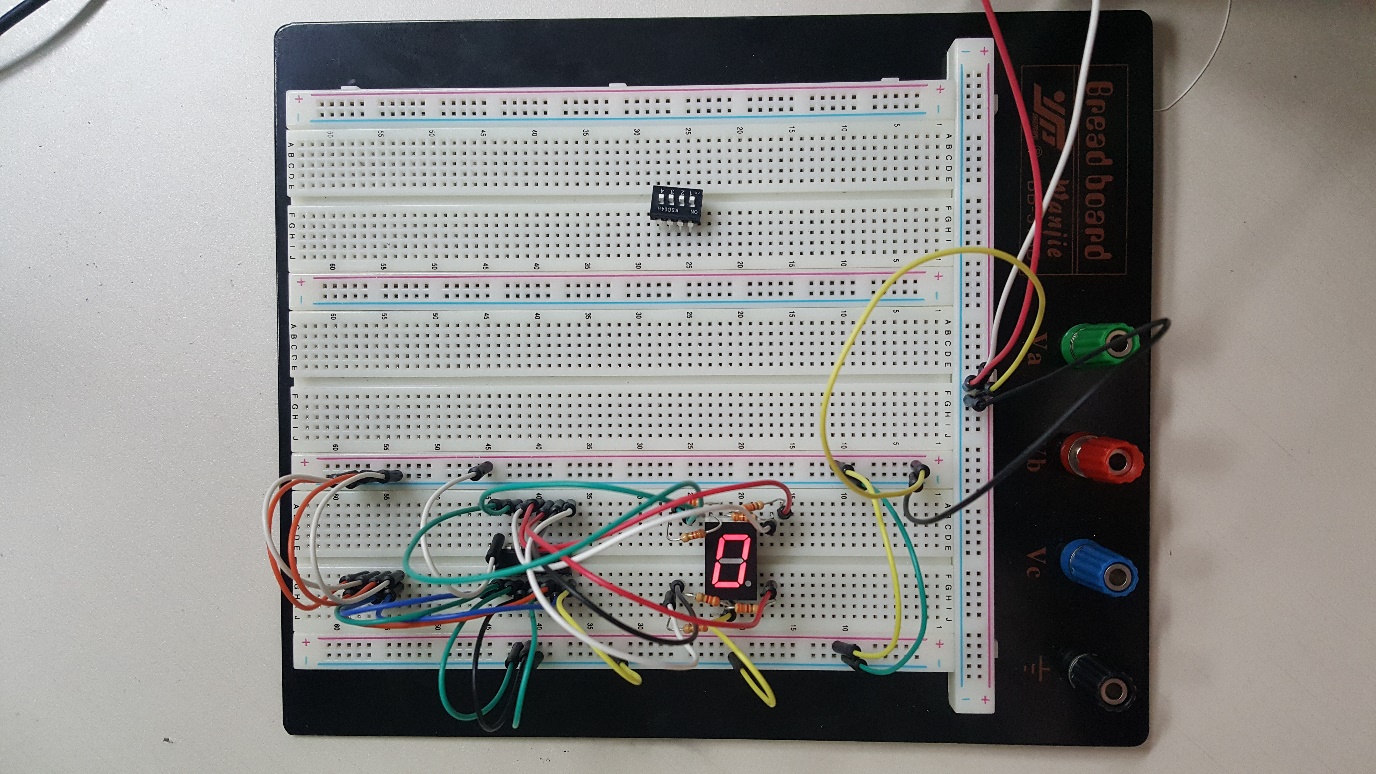
At lab time:

1. 7-segment display를 universal board 위에 7448 (BCD to 7-segment decoder)과 dip switch로 구현하기

-> universal board가 아닌 breadboard 위에 구현하였으며, dip switch가 제대로 부착되지 않아 수동으로 입력 값을 조절하였다.

사진: 

2. BCD to 7-segment decoder를 Xilinx ISE상에 구현하기

2-(1). In structural description

코드:

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date: 19:50:41 04/10/2018

// Design Name:

// Module Name: BCDto7\_s

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module BCDto7\_s(

input A,

input B,

input C,

input D,

output [6:0] out

);

wire NA, NB, NC, ND;

wire BD, NBND, CD, NCND, CND, NBC, BND, BNC, BNCD;

not(NA, A);

not(NB, B);

not(NC, C);

not(ND, D);

and(BD, B, D);

and(CD, C, D);

and(NBND, NB, ND);

and(NCND, NC, ND);

and(CND, C, ND);

and(NBC, NB, C);

and(BND, B, ND);

and(BNC, B, NC);

and(BNCD, B, NC, D);

or(out[6], A, BD, C, NBND);

or(out[5], NCND, CD, NB);

or(out[4], B, NC, D);

or(out[3], NBND, CND, BNCD, NBC);

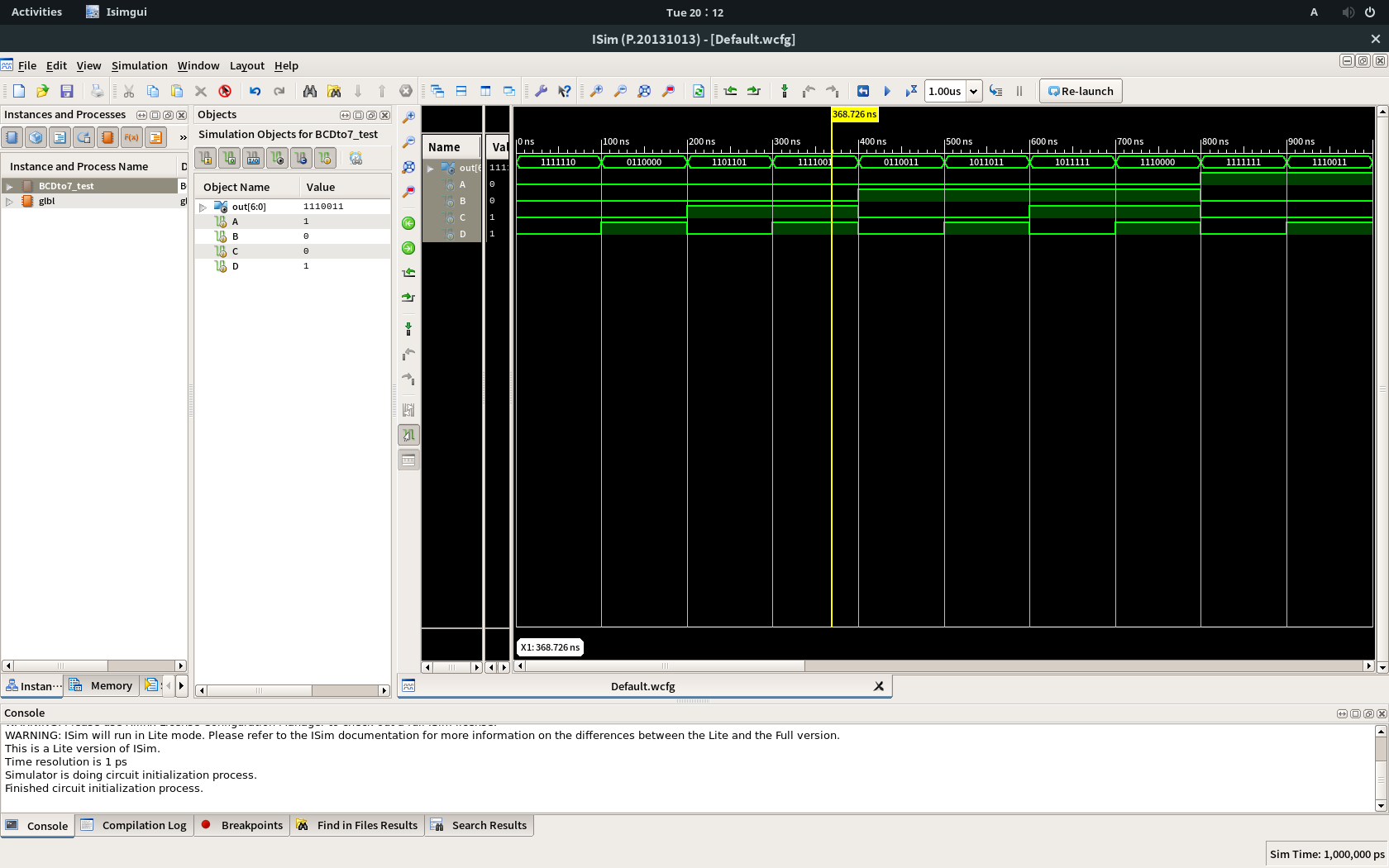
or(out[2], NBND, CND);

or(out[1], A, NCND, BND, BNC);

or(out[0], A, CND, BNC, NBC);

endmodule

사진:



2-(2). In data-flow style description

코드:

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date: 20:16:36 04/10/2018

// Design Name:

// Module Name: BCDto7\_d

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module BCDto7\_d(

input A,

input B,

input C,

input D,

output [6:0] OUT

);

wire [3:0] in;

assign in = {A, B, C, D};

assign OUT = (in == 4'b0000) ? 7'b1111110 :

(in == 4'b0001) ? 7'b0110000 :

(in == 4'b0010) ? 7'b1101101 :

(in == 4'b0011) ? 7'b1111001 :

(in == 4'b0100) ? 7'b0110011 :

(in == 4'b0101) ? 7'b1011011 :

(in == 4'b0110) ? 7'b1011111 :

(in == 4'b0111) ? 7'b1110000 :

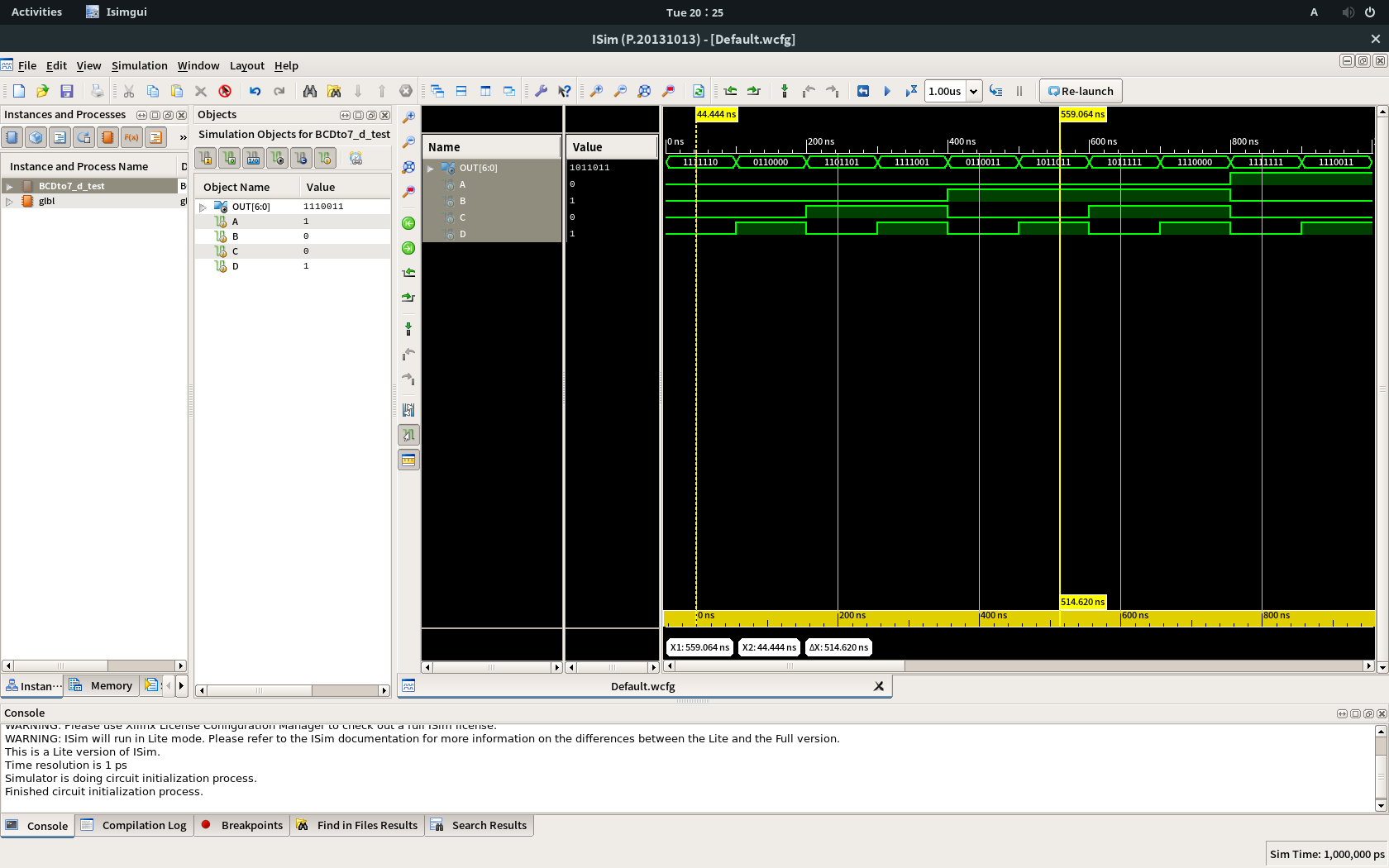
(in == 4'b1000) ? 7'b1111111 :

(in == 4'b1001) ? 7'b1110011 :

7'b0000000;

endmodule

사진:



2-(3). In behavioral description

코드:

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date: 20:27:40 04/10/2018

// Design Name:

// Module Name: BCDto7\_b

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module BCDto7\_b(

input A,

input B,

input C,

input D,

output [6:0] OUT

);

wire [3:0] in;

reg [6:0] t;

assign in = {A, B, C, D};

assign OUT = t;

always@(in)

begin

case(in)

4'b0000 : t = 7'b1111110;

4'b0001 : t = 7'b0110000;

4'b0010 : t = 7'b1101101;

4'b0011 : t = 7'b1111001;

4'b0100 : t = 7'b0110011;

4'b0101 : t = 7'b1011011;

4'b0110 : t = 7'b1011111;

4'b0111 : t = 7'b1110000;

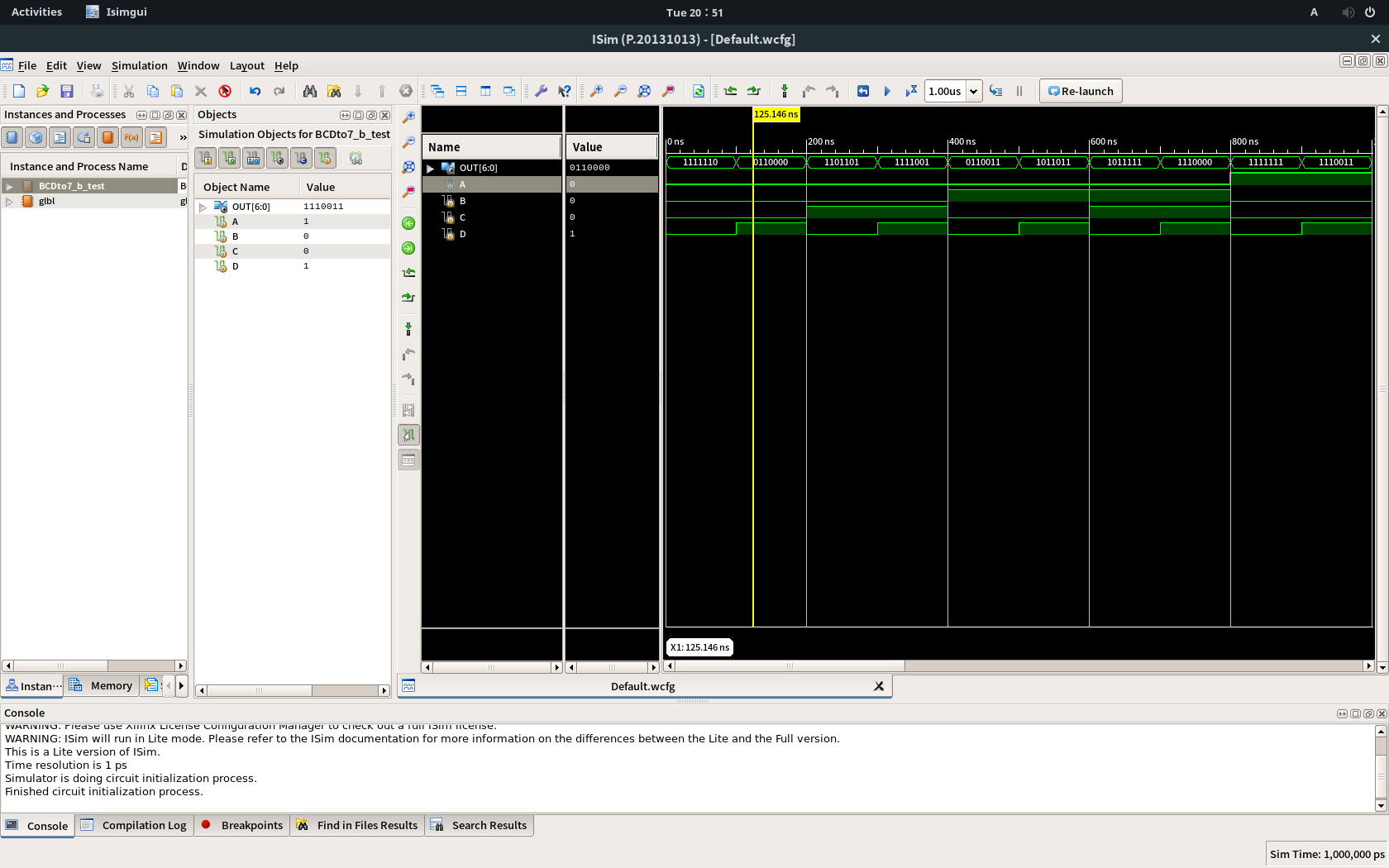
4'b1000 : t = 7'b1111111;

4'b1001 : t = 7'b1110011;

endcase

end

endmodule

사진: 

Homework:

1. Klingon number system decoder for 7-segment를 Xilinx ISE 상에 구현하기

1-(1). In structural description

코드:

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date: 21:31:33 04/10/2018

// Design Name:

// Module Name: Klingon\_s

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module Klingon\_s(

input A,

input B,

input C,

input D,

output [6:0] OUT

);

wire NA, NB, NC, ND;

wire BND, NCND, NBNCND, BD, NANBNCND;

wire BNCD, NBCD, BCND, BCD, AD;

not(NA, A);

not(NB, B);

not(NC, C);

not(ND, D);

and(BND, B, ND);

and(NCND, NC, ND);

and(NBNCND, NB, NC, ND);

and(BD, B, D);

and(NANBNCND, NA, NB, NC, ND);

and(BNCD, B, NC, D);

and(NBCD, NB, C, D);

and(BCND, B, C, ND);

and(BCD, B, C, D);

and(AD, A, D);

and(NANB, NA, NB);

and(OUT[6], NA, NB);

or(OUT[5], A, BND, NCND);

or(OUT[4], A, NBNCND, BD);

or(OUT[3], NANBNCND, BNCD, NBCD);

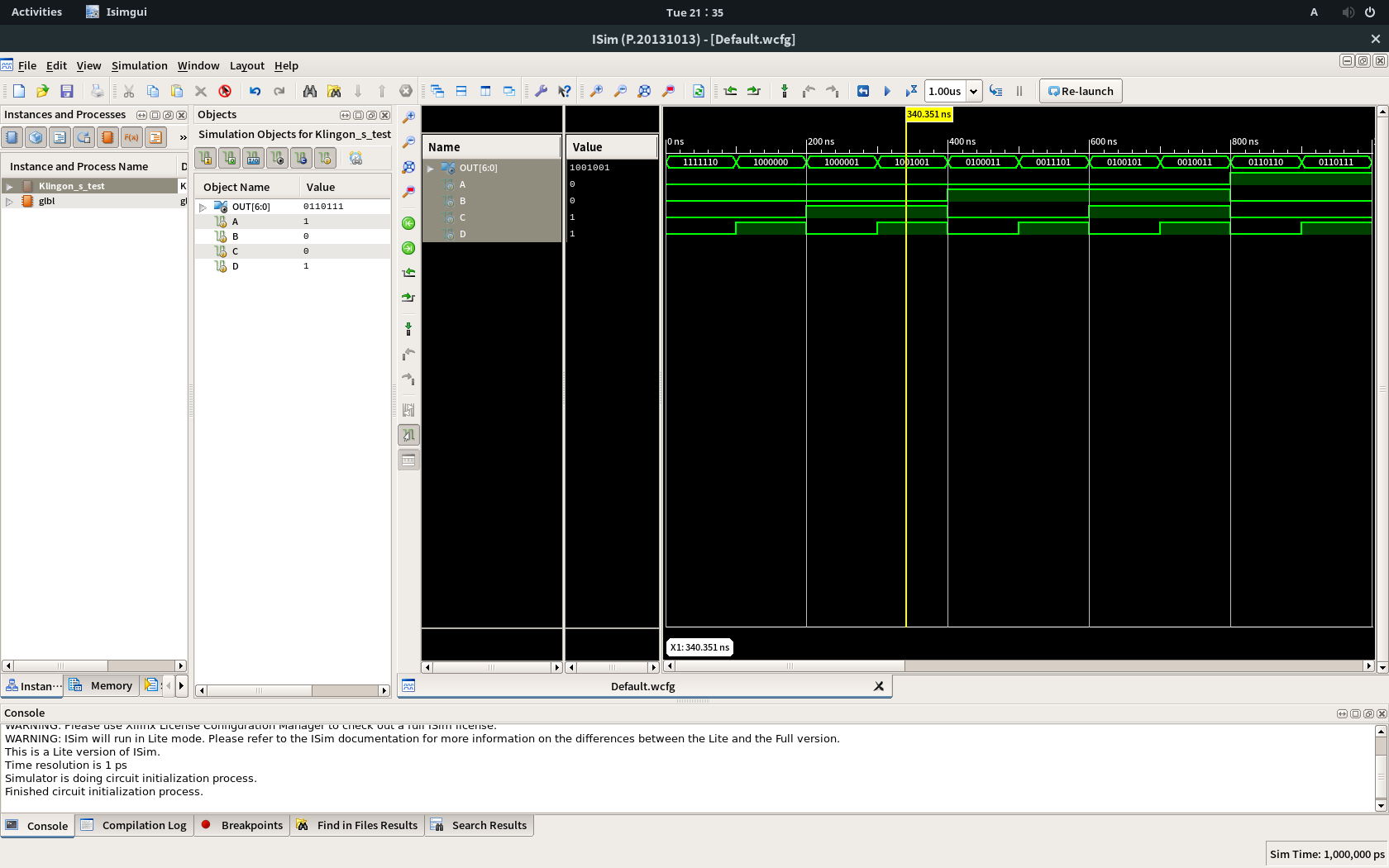
or(OUT[2], A, BNCD, BCND, NBNCND);

or(OUT[1], A, NCND, BCD);

or(OUT[0], AD, B, C);

endmodule

사진:



1-(2). In data-flow style description

코드:

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date: 21:36:53 04/10/2018

// Design Name:

// Module Name: Klingon\_d

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module Klingon\_d(

input A,

input B,

input C,

input D,

output [6:0] OUT

);

wire [3:0] in = {A, B, C, D};

assign OUT = (in == 4'b0000) ? 7'b1111110 :

(in == 4'b0001) ? 7'b1000000 :

(in == 4'b0010) ? 7'b1000001 :

(in == 4'b0011) ? 7'b1001001 :

(in == 4'b0100) ? 7'b0100011 :

(in == 4'b0101) ? 7'b0011101 :

(in == 4'b0110) ? 7'b0100101 :

(in == 4'b0111) ? 7'b0010011 :

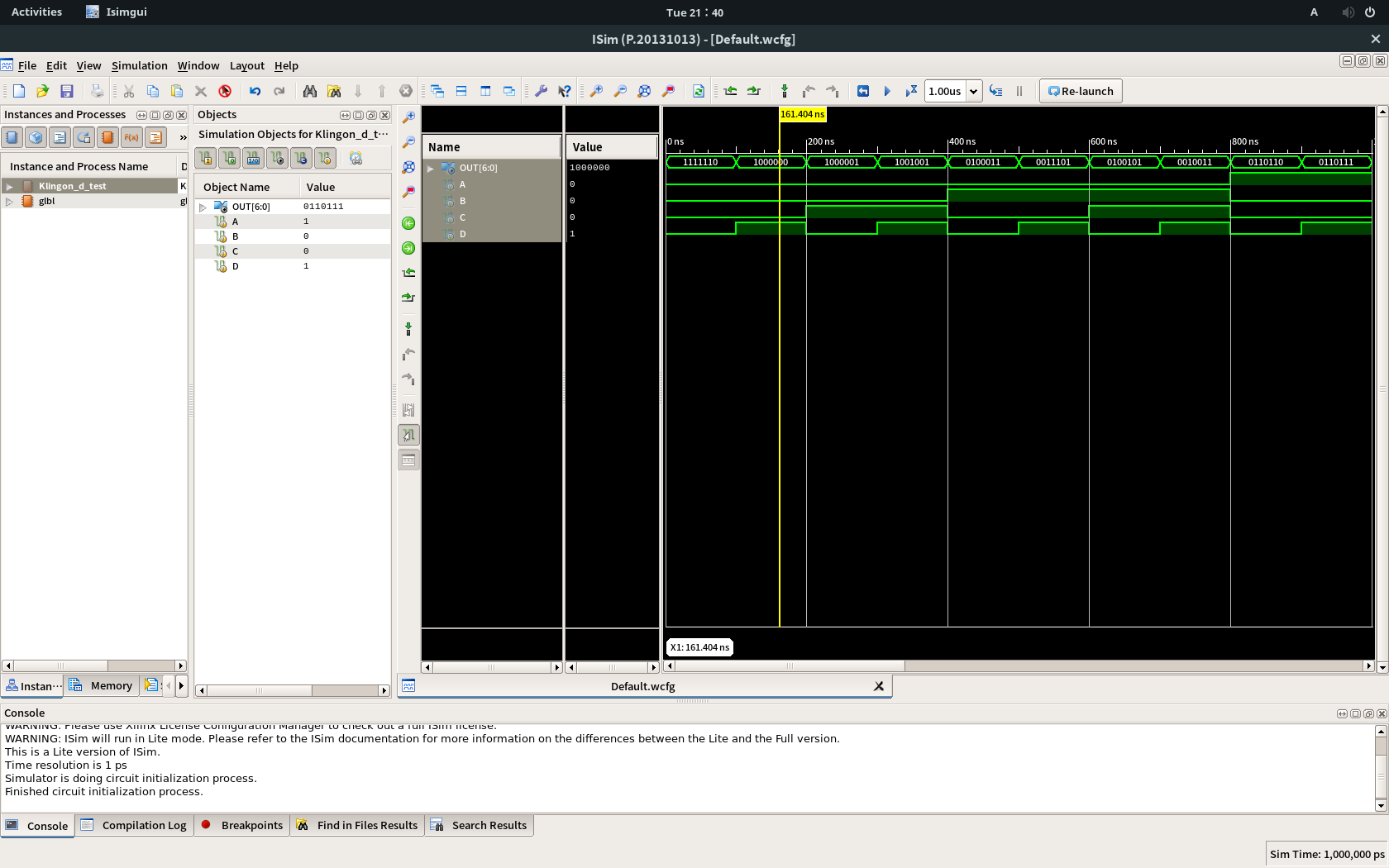
(in == 4'b1000) ? 7'b0110110 :

(in == 4'b1001) ? 7'b0110111 :

7'b0000000;

endmodule

사진:



1-(3). In behavioral description

코드:

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date: 21:41:54 04/10/2018

// Design Name:

// Module Name: Klingon\_b

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module Klingon\_b(

input A,

input B,

input C,

input D,

output [6:0] OUT

);

wire [3:0] in;

reg [6:0] t;

assign in = {A, B, C, D};

assign OUT = t;

always@(in)

begin

case(in)

4'b0000 : t = 7'b1111110;

4'b0001 : t = 7'b1000000;

4'b0010 : t = 7'b1000001;

4'b0011 : t = 7'b1001001;

4'b0100 : t = 7'b0100011;

4'b0101 : t = 7'b0011101;

4'b0110 : t = 7'b0100101;

4'b0111 : t = 7'b0010011;

4'b1000 : t = 7'b0110110;

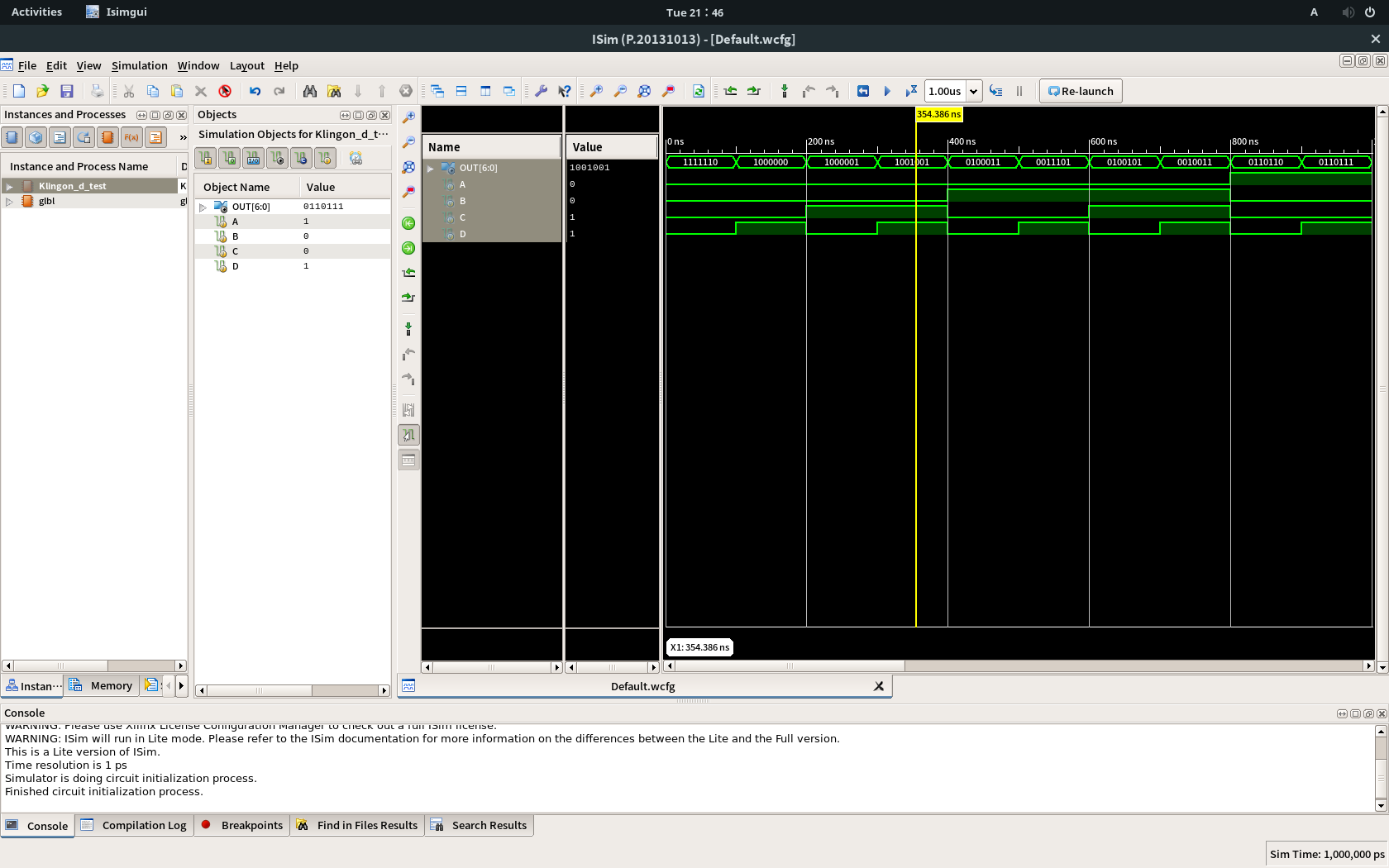
4'b1001 : t = 7'b0110111;

endcase

end

endmodule

사진:



부록: BCD to 7-segment decoder와 Klingon number system의 테스트코드

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date: 20:06:12 04/10/2018

// Design Name: BCDto7\_s

// Module Name: /tmp/tmp.avGN9aNacr/BCDto7/BCDto7\_test.v

// Project Name: BCDto7

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: BCDto7\_s

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module BCDto7\_test;

// Inputs

reg A;

reg B;

reg C;

reg D;

// Outputs

wire [6:0] out;

// Instantiate the Unit Under Test (UUT)

BCDto7\_s uut (

.A(A),

.B(B),

.C(C),

.D(D),

.out(out)

);

initial begin

// Initialize Inputs

A = 0;

B = 0;

C = 0;

D = 0;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

A = 0; B = 0; C = 0; D = 1;

#100 A = 0; B = 0; C = 1; D = 0;

#100 A = 0; B = 0; C = 1; D = 1;

#100 A = 0; B = 1; C = 0; D = 0;

#100 A = 0; B = 1; C = 0; D = 1;

#100 A = 0; B = 1; C = 1; D = 0;

#100 A = 0; B = 1; C = 1; D = 1;

#100 A = 1; B = 0; C = 0; D = 0;

#100 A = 1; B = 0; C = 0; D = 1;

end

endmodule