Practicum #05 – Guideline Module

Introduction to Digital System – Odd Term 2016/2017

Name	Mochamad Aulia Akbar Praditomo
NPM	1606827145
Asistant Code	AMB

The Question part will have red color and your Answer will have green color. (Except for K-Map problem, you can color it by your own to show the selection rule).

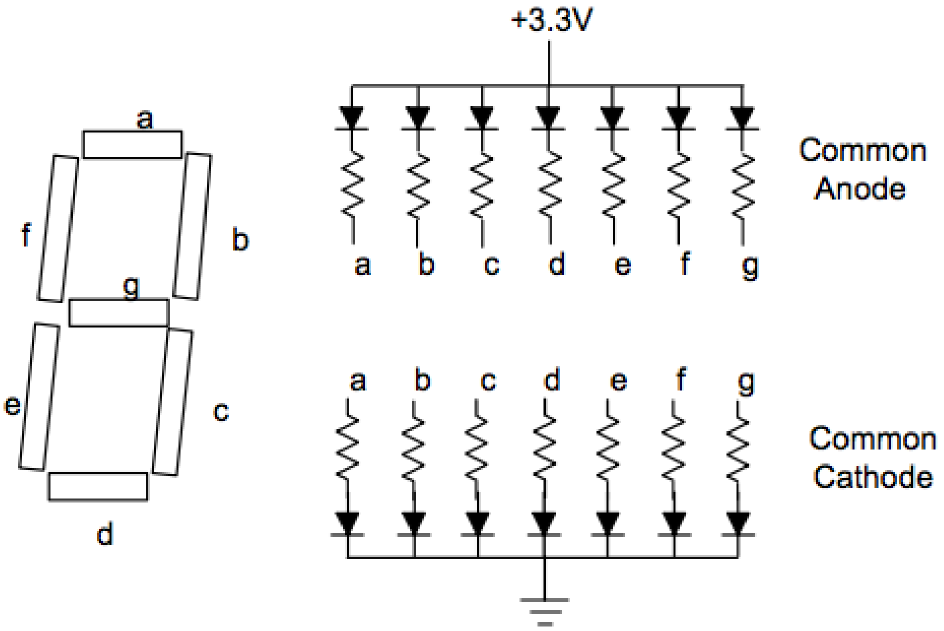


Image 1 7-Segment display contains seven light emitting diodes (LEDs)

In this tutorial, we will try to design a 7-segment decoder using Karnaugh Map and write a Verilog Program to implement the resulting logic equation. Seven LEDs in 7-segment display can be arranged in a pattern to form different digits as shown in Image 1. Digital watches that you wear use similar 7-segment display using liquid crystal rather than LEDs. Seven segment display come in two flavors: common anode and common cathode. A common anode 7-segment display has all of the anodes tied together while a common cathode 7-segment display has all the cathodes tied together as shown in Image 1.

The BASYS boards have four common anode 7-segment display, which meadns that all the anodes are tied together and connected through a pnp transistor to +3.3V. A different FPGA output pin is connected through a 100Ω current-limiting resistor to each of cathodes, a-g, plus a decimal point. **In the common anode case, an output 0 will turn on a segment and an output 1 will turn it off.**

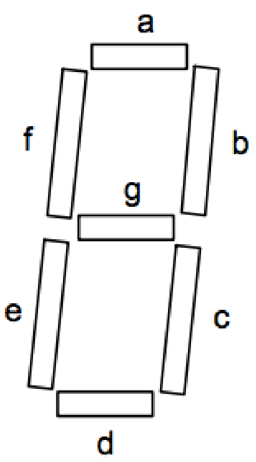
Image 2 7-segment display in BASYS

Complete the following table that shows output cathode values for each segment a-g nedded to display all hex values from 0-F

Number to Display (x)	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
b	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
d	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

1 = off

0 = on



The problem now is to design a hex to 7-segment decoder, called IDS_Lab05_hex7seg.v as shown in the following figure. The input is a 4-bit hex number, x[3:0], and the output are the 7-segment values a-g given the truth table above. We can make a Karnough Map for each segment and then write logic equations for the segments a-g.

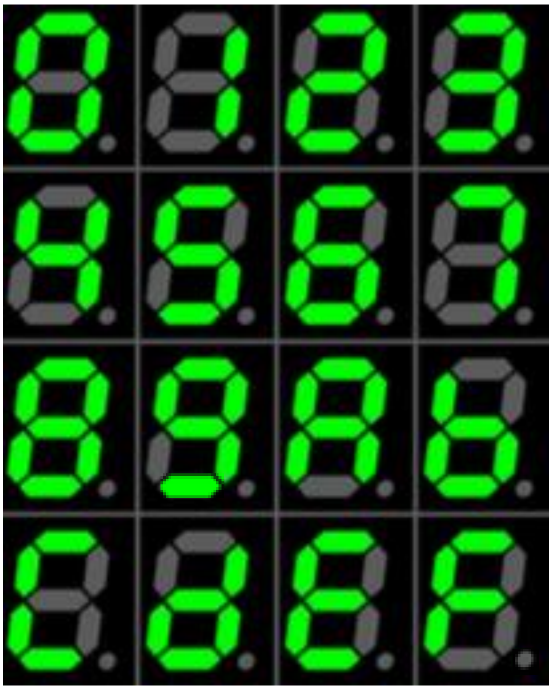


Image 3 Implementation of a-g

Complete the following Karnaugh Map for assignment of 7-segment display a-g.

x1x0 x3x2	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	0	1	0	0
10	0	0	1	0

a

x1x0 x3x2	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	1	0	1	1
10	0	0	1	0

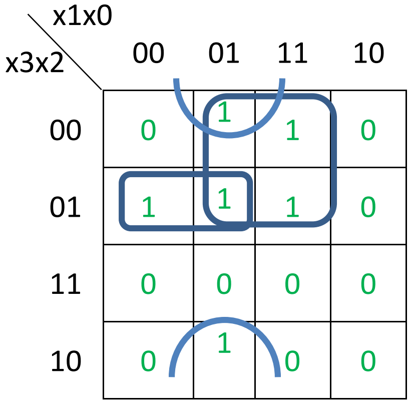
b

x1x0 x3x2	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	1	0	1	1
10	0	0	0	0

c

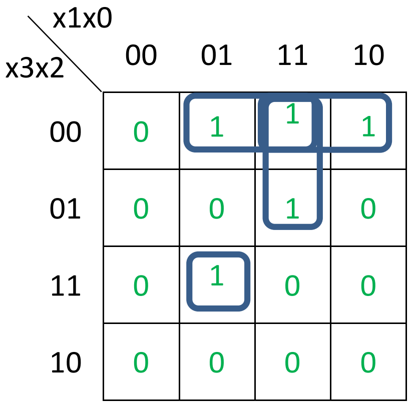
x1x0 x3x2	00	01	11	10
00	0	1	0	0
01	1	0	1	0
11	0	0	1	0
10	0	0	0	1

d



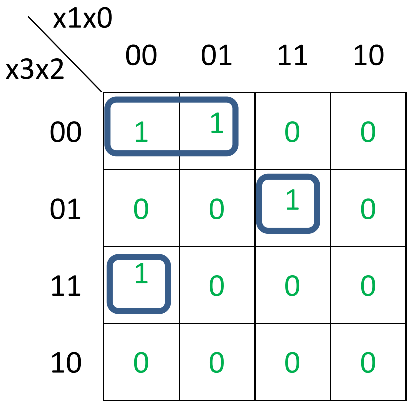
x1x0 x3x2	00	01	11	10
00	0	1	1	0
01	1	1	1	0
11	0	0	0	0
10	0	1	0	0

e



x1x0 x3x2	00	01	11	10
00	0	1	1	1
01	0	0	1	0
11	0	1	0	0
10	0	0	0	0

f



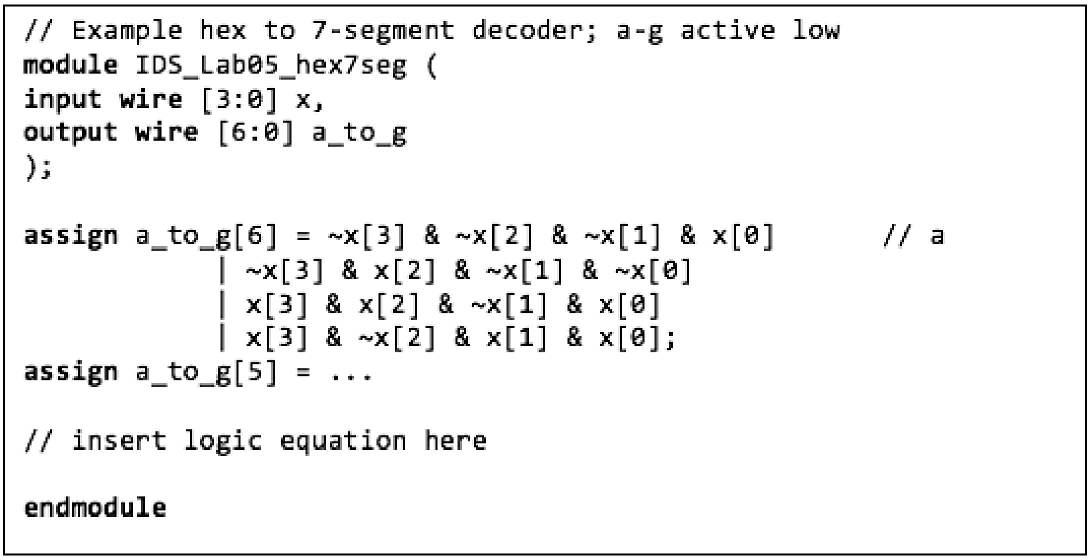
x1x0 x3x2	00	01	11	10
00	1	1	0	0
01	0	0	1	0
11	1	0	0	0
10	0	0	0	0

g

Fill the following equation for each assignment!

Assignment	Equation
a	(x0 & ~x1 & ~x2 & ~x3) | (~x0 & ~x1 & x2 & ~x3) | (x0 & ~x1 & x2 & x3) | (x0 & x1 & ~x2 & x3)
b	(x0 & ~x1 & x2 & ~x3) | (~x0 & x1 & x2) | (x0 & x1 & x3) | (~x0 & x2 & x3)
c	~x0 & x1 & ~x2 & ~x3 | x1 & x2 & x3 | ~x0 & x2 & x3
d	(x0 & ~x1 & ~x2 & ~x3) | (~x0 & ~x1 & x2 & ~x3) | (~x0 & x1 & ~x2 & x3) | (x0 & x1 & x2)
e	(~x3 & x0) | (~x3 & x2 & ~x1) | (~x2 & ~x1 & x0)
f	(x0 & ~x2 & ~x3) | (x1 & ~x2 & ~x3) | (x0 & x1 & ~x3) | (x0 & ~x1 & x2 & x3)
g	(~x1 & ~x2 & ~x3) | (x0 & x1 & x2 & ~x3) | (~x0 & ~x1 & x2 & x3)

Now you are ready to write the Verilog program for the 7-segment decoder. The following are the starter code that you can use!

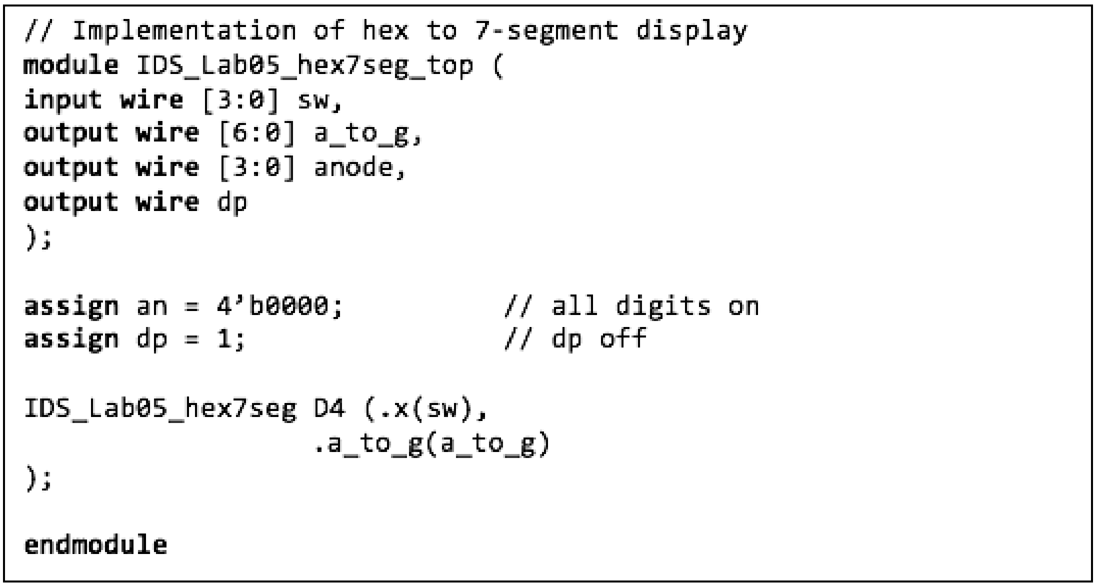


```
// Example hex to 7-segment decoder; a-g active low
module IDS_Lab05_hex7seg (
input wire [3:0] x,
output wire [6:0] a_to_g
);

assign a_to_g[6] = ~x[3] & ~x[2] & ~x[1] & x[0]      // a
             | ~x[3] & x[2] & ~x[1] & ~x[0]
             | x[3] & x[2] & ~x[1] & x[0]
             | x[3] & ~x[2] & x[1] & x[0];
assign a_to_g[5] = ...

// insert logic equation here

endmodule
```

Continue the implementation of Verilog module above, don’t forget to create the test fixture!

Make sure you have test your module and the following file already exist: IDS_Lab05_hex7seg.v and IDS_Lab05_hex7seg_test.v

Now complete your implementation by creating IDS_Lab05_hex7seg_top.v



```
// Implementation of hex to 7-segment display
module IDS_Lab05_hex7seg_top (
input wire [3:0] sw,
output wire [6:0] a_to_g,
output wire [3:0] anode,
output wire dp
);

assign an = 4’b0000;          // all digits on
assign dp = 1;                // dp off

IDS_Lab05_hex7seg D4 (.x(sw),
                  .a_to_g(a_to_g)
);

endmodule
```

Now, create Implementation Constraint File and Generate Programming File!

hint: on the left side of 7-segment display, there exist pin number for a to g, and bellow each segment, exists anode pin number. You can also find pin number for dot point (dp) there.

Look at your board and see the magic!