Diane Shan

862148900

EE120A Section 23

Jack Huang

Lab 3 – Programming Combinatorial Logic Using EDA Playground

**Overview**

In this lab, we did a group design of a programming combinatorial logic for a BCD to 7 Segment LED Display. We built a basic project with behavior design, using switches to control the LED display. We created a test bench to test the design and made sure it passed the test cases. After building the design and the test bench, we found that using the BCD to 7 Segment LED display, we could display all the numbers from 0 to 9.

**New Concepts**

BCD – binary coded decimal, numbers are made up using just 4 data bits, similar to hex and only range from 0 to 9 and A to F.

7 Segment LED Display – provide a very convenient way of displaying information or digital data in the form of numbers, letters or even alpha-numerical characters.

Digital Decoder – device which converts one digital format into another one of the most commonly used devices for doing this is BCD to 7-Segment Display Decoder

**Analysis**

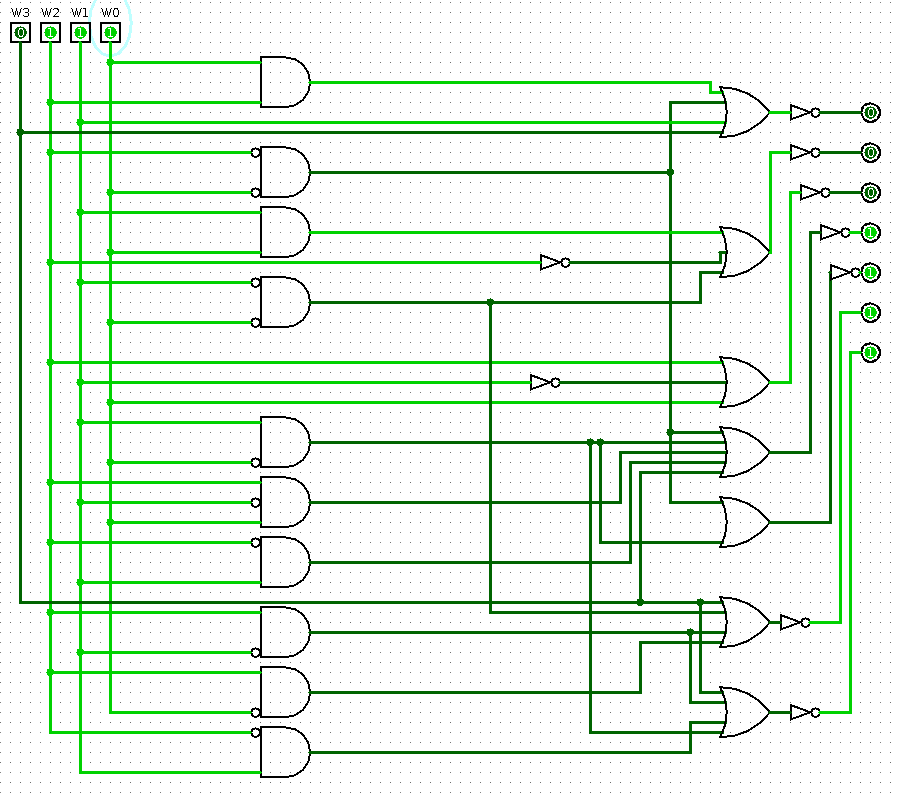
*Procedure*:

1. Read and learn about the 7-segment LED display
2. Implement gate level schematics of a BCD-to-7seg decoder so that switches SW[3:0] control the decimal number displayed on right-most LED display
3. Implement structural or behavioral description of BCD-to-7seg decoder
4. Create a test bench to test the decoder behavioral model

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SW3** | **SW2** | **SW1** | **SW0** | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **Display** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 3 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 5 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 6 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 9 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Invalid |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | Invalid |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | Invalid |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | Invalid |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | Invalid |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | Invalid |

**Records**

*Logic Circuit:*

****

design.sv

module bcdto7led\_bh (

input wire sw0 , // Switches

input wire sw1 ,

input wire sw2 ,

input wire sw3 ,

output reg a , // LED segments

output reg b ,

output reg c ,

output reg d ,

output reg e ,

output reg f ,

output reg g

);

// Internal wire

wire [3:0] bundle ;

assign bundle = {sw3,sw2,sw1,sw0 } ;

always @(\*) begin

// Setting the segments signals (Initialize all to off/1)

a = 1'b1 ;

b = 1'b1 ;

c = 1'b1 ;

d = 1'b1 ;

e = 1'b1 ;

f = 1'b1 ;

g = 1'b1 ;

case ( bundle )

4'b0000 : begin // 0

a = 1'b0 ;

b = 1'b0 ;

c = 1'b0 ;

d = 1'b0 ;

e = 1'b0 ;

f = 1'b0 ;

g = 1'b1 ;

end

4'b0001 : begin // 1

a = 1'b1 ;

b = 1'b0 ;

c = 1'b0 ;

d = 1'b1 ;

e = 1'b1 ;

f = 1'b1 ;

g = 1'b1 ;

end

4'b0010 : begin // 2

a = 1'b0 ;

b = 1'b0 ;

c = 1'b1 ;

d = 1'b0 ;

e = 1'b0 ;

f = 1'b1 ;

g = 1'b0 ;

end

4'b0011 : begin // 3

a = 1'b0 ;

b = 1'b0 ;

c = 1'b0 ;

d = 1'b0 ;

e = 1'b1 ;

f = 1'b1 ;

g = 1'b0 ;

end

4'b0100 : begin // 4

a = 1'b1 ;

b = 1'b0 ;

c = 1'b0 ;

d = 1'b1 ;

e = 1'b1 ;

f = 1'b0 ;

g = 1'b0 ;

end

4'b0101 : begin // 5

a = 1'b0 ;

b = 1'b1 ;

c = 1'b0 ;

d = 1'b0 ;

e = 1'b1 ;

f = 1'b0 ;

g = 1'b0 ;

end

4'b0110 : begin // 6

a = 1'b0 ;

b = 1'b1 ;

c = 1'b0 ;

d = 1'b0 ;

e = 1'b0 ;

f = 1'b0 ;

g = 1'b0 ;

end

4'b0111 : begin // 7

a = 1'b0 ;

b = 1'b0 ;

c = 1'b0 ;

d = 1'b1 ;

e = 1'b1 ;

f = 1'b1 ;

g = 1'b1 ;

end

4'b1000 : begin // 8

a = 1'b0 ;

b = 1'b0 ;

c = 1'b0 ;

d = 1'b0 ;

e = 1'b0 ;

f = 1'b0 ;

g = 1'b0 ;

end

4'b1001 : begin // 9

a = 1'b0 ;

b = 1'b0 ;

c = 1'b0 ;

d = 1'b0 ;

e = 1'b1 ;

f = 1'b0 ;

g = 1'b0 ;

end

endcase

end

endmodule

testbench.sv

module bcdtoled\_tb;

// Inputs

reg sw0;

reg sw1;

reg sw2;

reg sw3;

// Outputs

wire a;

wire b;

wire c;

wire d;

wire e;

wire f;

wire g;

// Instantiate the Unit Under Test (UUT)

bcdto7led\_bh uut (

.sw0(sw0),

.sw1(sw1),

.sw2(sw2),

.sw3(sw3),

.a(a),

.b(b),

.c(c),

.d(d),

.e(e),

.f(f),

.g(g)

);

initial begin

$dumpfile("dump.vcd"); $dumpvars;

// Initialize Inputs

sw3 = 0; sw2 = 0; sw1 = 0; sw0 = 0;

#100;

$display("TC10");

if ( {a,b,c,d,e,f,g} != 7'b0000001 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g});

// Your test cases go here (9 left)

sw3 = 0; sw2 = 0; sw1 = 0; sw0 = 1;

#100;

$display("TC11");

if ( {a,b,c,d,e,f,g} != 7'b1001111 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //1

sw3 = 0; sw2 = 0; sw1 = 1; sw0 = 0;

#100;

$display("TC12");

if ( {a,b,c,d,e,f,g} != 7'b0010010 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //2

sw3 = 0; sw2 = 0; sw1 = 1; sw0 = 1;

#100;

$display("TC13");

if ( {a,b,c,d,e,f,g} != 7'b0000110 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //3

sw3 = 0; sw2 = 1; sw1 = 0; sw0 = 0;

#100;

$display("TC14");

if ( {a,b,c,d,e,f,g} != 7'b1001100 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //4

sw3 = 0; sw2 = 1; sw1 = 0; sw0 = 1;

#100;

$display("TC15");

if ( {a,b,c,d,e,f,g} != 7'b0100100 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //5

sw3 = 0; sw2 = 1; sw1 = 1; sw0 = 0;

#100;

$display("TC16");

if ( {a,b,c,d,e,f,g} != 7'b0100000 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //6

sw3 = 0; sw2 = 1; sw1 = 1; sw0 = 1;

#100;

$display("TC17");

if ( {a,b,c,d,e,f,g} != 7'b0001111 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //7

sw3 = 1; sw2 = 0; sw1 = 0; sw0 = 0;

#100;

$display("TC18");

if ( {a,b,c,d,e,f,g} != 7'b0000000 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //8

sw3 = 1; sw2 = 0; sw1 = 0; sw0 = 1;

#100;

$display("TC19");

if ( {a,b,c,d,e,f,g} != 7'b0000100 ) $display ("Result is wrong %b", {a,b,c,d,e,f,g}); //9

end

endmodule

*Simulation Waveform:*

**A picture containing text, electronics, computer

Description automatically generated**

**Discussion**

The system works according to the provided specifications. After testing the design extensively with the testbench, we found that it passed all the test cases. The display displays the correct numbers on the LED display as specified. It displays the numbers from 0 to 9. There were no problems encountered that resulted in system redesign, modifications, or otherwise. Some possible ways to improve the system would be to use a different display that would also display letters so that there wouldn’t be invalid switch combinations.

**Conclusion**

For this lab, we got familiar with synthesis and implementation of combinatorial logic applications using the EDA Playground. We also learned about the 7-segment LED display and LED decoder. We built a basic project using behavior design and we got more familiar with how to implement a behavioral design. We also learned about how switched can control the output. We got more practice designing test benches for our designs and making sure our code passes the test cases. The results were that we were able to display the numbers 0 to 9 in the 7-segment LED display. This is what we were aiming for so we were successful and we were also able to learn about these displays and what they are all about.

**Questions**

No questions in this lab