Ivan Neto

SID: 862228249

EE120A Section 021

Lab 3 – Combinational Logic

Shreya Chitturi

Overview

Overview of what was done

- We implemented a BCD to 7 Segment LED display using common anodes. Common anodes are such that outputs are all connected to a power source. The change of an output from 1 to 0 causes the light to turn on, and thus an output = 0 means the light is on the 7 segment LED.

A brief summary of the results

- We implemented Behavioral Model to create this BCD to 7 Segment LED circuit. We also implemented a testbench to ensure that our behavioral model produced the correct outputs for the 7 segment LED display.

Conclusion from the results

- We found that all of the test cases were successful, and produced a waveform output of the simulation, which can be interpreted to find that all of the inputs produced the correct outputs.

New Concepts

- **BCD to 7 Segment LED display**
  - Way to decode a binary signal to turn on a 7 Segment LED display in different patterns.

- **Common Anode version**
  - Outputs are connected to constant power source
  - A current out of the output to ground turns the light on, thus output = 0 means the light is currently on in the segment.

- **Common Cathode version**
  - Outputs are connected to ground.
  - A current through the output turns the light on, thus output = 1 means the light is currently on in the segment.

Analysis

Procedure

1. Implement the behavioral model for the BCD to 7 Segment LED display.
2. Create a testbench.
3. Run the testbench simulation.
4. Analyze the output and determine whether the model works correctly.

Figures

All test cases

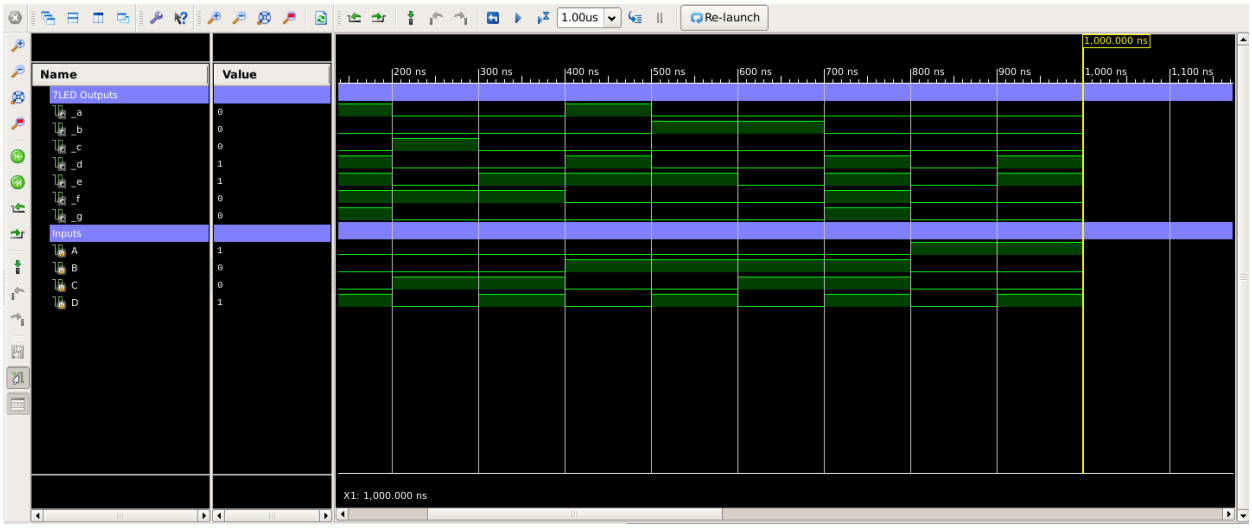


Figure 1: The waveform output of all of the testcases being run. The test cases were implemented separately from the design, and the correct outputs are produced.

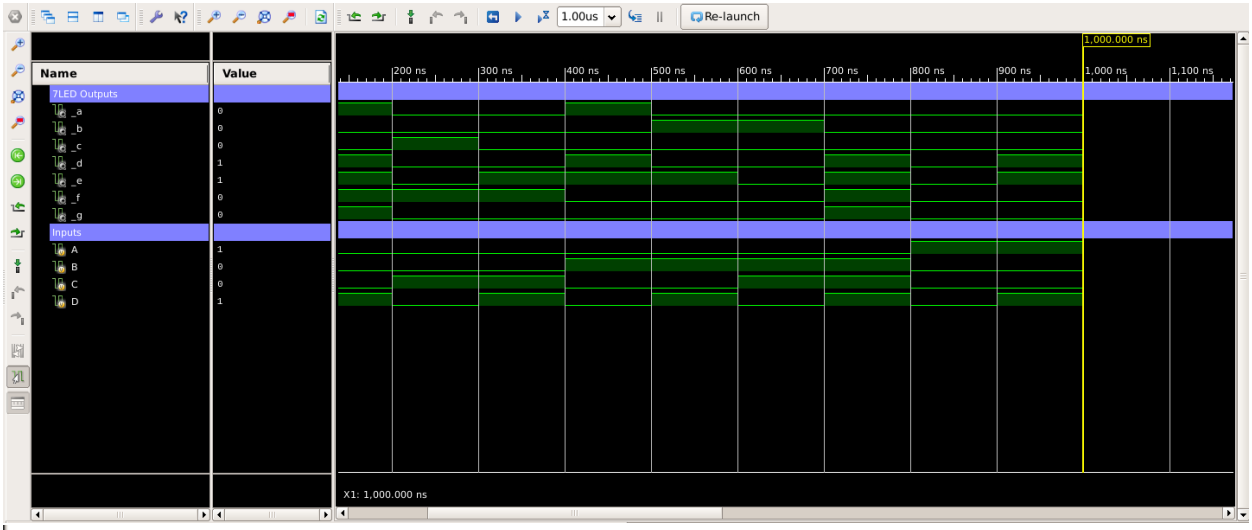
All test cases, terminal output

```
--------------------------------------------------------------------------------
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
TC00
  Test successful
TC01
  Test successful
TC02
  Test successful
TC03
  Test successful
TC04
  Test successful
TC05
  Test successful
TC06
  Test successful
TC07
  Test successful
TC08
  Test successful
TC09
  Test successful
ISim>
```

Here you can see all of the test case outputs in the terminal. Observe that all test cases pass the testbench successfully, which implies that our design is correctly implemented.

Records

Simulation

- 


- ```
  ---------------------------------------------------------------------
  This is a Full version of ISim.
  Time resolution is 1 ps
  Simulator is doing circuit initialization process.
  Finished circuit initialization process.
  TC00
    Test successful
  TC01
    Test successful
  TC02
    Test successful
  TC03
    Test successful
  TC04
    Test successful
  TC05
    Test successful
  TC06
    Test successful
  TC07
    Test successful
  TC08
    Test successful
  TC09
    Test successful
  ISim>
  ```

Code

Behavioral Model
  ```

```
module anode_7led (

   // Select lines
   input wire sw0,
   input wire sw1,
   input wire sw2,
   input wire sw3,

   // Segment outputs
   output reg a,
   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 segment signals 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 // Number 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 // Number 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 // Number 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 // Number 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 // Number 4
         a = 1'b1;
         b = 1'b0;
         c = 1'b0;
         d = 1'b1;
         e = 1'b1;
         f = 1'b0;
         g = 1'b0;
      end

```

```
86        4'b0101 : begin // Number 5
87           a = 1'b0;
88           b = 1'b1;
89           c = 1'b0;
90           d = 1'b0;
91           e = 1'b1;
92           f = 1'b0;
93           g = 1'b0;
94        end
95
96        4'b0110 : begin // Number 6
97           a = 1'b0;
98           b = 1'b1;
99           c = 1'b0;
100          d = 1'b0;
101          e = 1'b0;
102          f = 1'b0;
103          g = 1'b0;
104       end
105
```

```
106       4'b0111 : begin // Number 7
107          a = 1'b0;
108          b = 1'b0;
109          c = 1'b0;
110          d = 1'b1;
111          e = 1'b1;
112          f = 1'b1;
113          g = 1'b1;
114       end
115
116       4'b1000 : begin // Number 8
117          a = 1'b0;
118          b = 1'b0;
119          c = 1'b0;
120          d = 1'b0;
121          e = 1'b0;
122          f = 1'b0;
123          g = 1'b0;
124       end
```

```
126       4'b1001 : begin // Number 9
127          a = 1'b0;
128          b = 1'b0;
129          c = 1'b0;
130          d = 1'b1;
131          e = 1'b1;
132          f = 1'b0;
133          g = 1'b0;
134       end
135
136       endcase
137    end
138 endmodule
139
```

Testbench

```
142 module anode_7led_anode_7led_sch_tb();
143
144 // Inputs
145    reg A;
146    reg B;
147    reg C;
148    reg D;
149
150 // Output
151    wire _a;
152    wire _b;
153    wire _c;
154    wire _d;
155    wire _e;
156    wire _f;
157    wire _g;
```

```
160 // Instantiate the UUT
161    anode_7led UUT (
162        .sw3(A),
163        .sw2(B),
164        .sw1(C),
165        .sw0(D),
166        .a(_a),
167        .b(_b),
168        .c(_c),
169        .d(_d),
170        .e(_e),
171        .f(_f),
172        .g(_g)
173    );
174
175    initial begin
176
177        // Number 0
178        A = 0; B = 0; C = 0; D = 0;
179        #100;
180        $display("TC00");
181        if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b0000001 )
182            $display("  Test successful");
183        else
184            $display("  Result is wrong");
185
186        // Number 1
187        A = 0; B = 0; C = 0; D = 1;
188        #100;
189        $display("TC01");
190        if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b1001111 )
191            $display("  Test successful");
192        else
193            $display("  Result is wrong");
194
195        // Number 2
196        A = 0; B = 0; C = 1; D = 0;
197        #100;
198        $display("TC02");
199        if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b0010010 )
200            $display("  Test successful");
201        else
202            $display("  Result is wrong");
203
204        // Number 3
205        A = 0; B = 0; C = 1; D = 1;
206        #100;
207        $display("TC03");
208        if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b0000110 )
209            $display("  Test successful");
210        else
211            $display("  Result is wrong");
212
213        // Number 4
214        A = 0; B = 1; C = 0; D = 0;
215        #100;
216        $display("TC04");
217        if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b1001100 )
218            $display("  Test successful");
219        else
220            $display("  Result is wrong");
221
222        // Number 5
223        A = 0; B = 1; C = 0; D = 1;
224        #100;
225        $display("TC05");
226        if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b0100100 )
227            $display("  Test successful");
228        else
229            $display("  Result is wrong");
230
231        // Number 6
232        A = 0; B = 1; C = 1; D = 0;
233        #100;
234        $display("TC06");
235        if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b0100000 )
236            $display("  Test successful");
237        else
238            $display("  Result is wrong");
```

```
240         // Number 7
241         A = 0; B = 1; C = 1; D = 1;
242         #100;
243         $display("TC07");
244         if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b0001111 )
245            $display("  Test successful");
246         else
247            $display("  Result is wrong");
248
249         // Number 8
250         A = 1; B = 0; C = 0; D = 0;
251         #100;
252         $display("TC08");
253         if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b0000000 )
254            $display("  Test successful");
255         else
256            $display("  Result is wrong");
257
258         // Number 9
259         A = 1; B = 0; C = 0; D = 1;
260         #100;
261         $display("TC09");
262         if ( {_a, _b, _c, _d, _e, _f, _g} == 7'b0001100 )
263            $display("  Test successful");
264         else
265            $display("  Result is wrong");

267      end
268
269   endmodule
```

UCF

- We did not use UCF in this lab and thus we do not include that.

Discussion

Does the system work according to provided specifications?

- The system works according to provided specifications. We did not use a different behavioral model as the one provided.
- We implemented the provided behavioral model and created the rest of the test cases, from numbers 1 to 9. The correct outputs were produced and the testbench passed all test cases.

Problems and technical issues encountered

- A few syntactical issues were encountered, but by debugging the code we resolved all of the issues and were able to run the simulation successfully.

Ways to improve the system

- I would improve the system by removing the an0, an1, an2, and an3 outputs from the behavioral model, since those are not needed for a single common anode display. Since we were only working with one display, none of those inputs were needed and were discarded while writing the behavioral model code in Verilog.

Conclusion

- The purpose of this lab was to learn how to write Verilog code to create a model for BCD to 7 Segment LED display. In this case, we implemented a common anode version of the display.

- We created a behavioral model for our design. We further implemented a testbench and were able to pass all of the test cases, which implies that our design is correctly implemented. The waveform output demonstrates that each configuration of the select lines produces the desired outputs for each of the 7 segments.