David Elliott

Evan Yeh

EE125 Homework #5

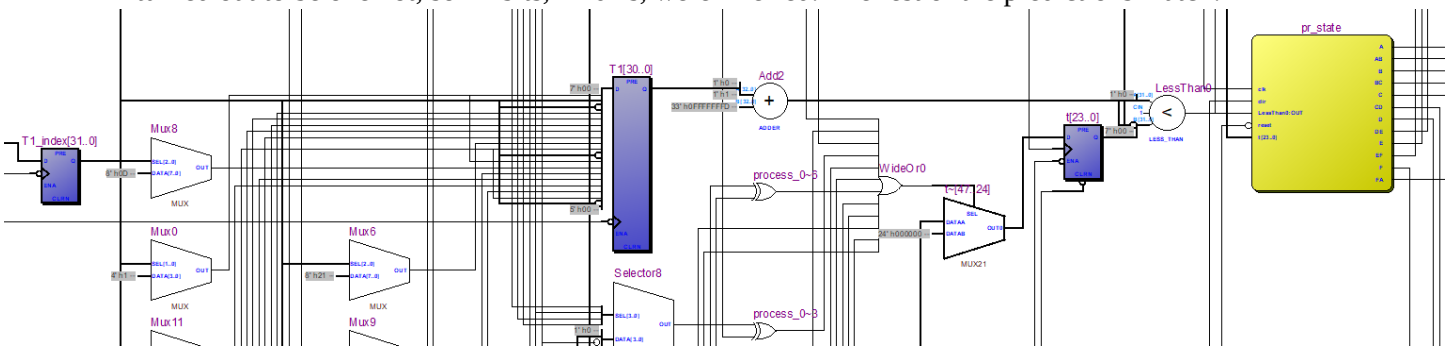
Exercise 9.6:

a.) Block Diagram:

b.) State transition diagrams:

c.) T1 is registered since there are different speeds for the light rotator. With the largest value of T1 as 12,500,000, T1 should be ceil(log(12500000)) = 24 bits long, inferring 24 dffs. t, which counts up to T1 will also have to hold a value as large as 12,500,000 (minus 1), requiring 24 bits, 24 dffs. T1_index is essentially a sequentially encoded state for the FSM of controlling the speed. There are 6 speeds to loop through, so T1_index goes from 0 to 5, requiring ceil(log(6)) = 3 bits, 3 dffs. Then assuming sequential encoding for the main FSM of light rotator states, there are 12 states, so ceil(log2(12)) = 4 bits are needed, 4 dffs. In total there should be 24 + 24 + 3 + 4 = 55 dffs.

d.) Looking at the RTL view, T1 ended up inferring 31 dffs, as did T1_index. The encoding for the FSM states turned out to be one hot, so 12 bits, 12 dffs, were inferred. The rest of the predictions match.




e.) Code:

```
001     ------------------------------------------------------------
002     library ieee;
003     use ieee.std_logic_1164.all;
004     ------------------------------------------------------------
005     entity light_rotator is
006         port (
007         stp, clk, rst, dir, spd: in std_logic;
008         ssd: out std_logic_vector(6 downto 0));
009     end entity;
010     ------------------------------------------------------------
011     architecture moore_fsm of light_rotator is
012
013     --FSM-related declarations:
014     type state is (A, AB, B, BC, C, CD, D, DE, E, EF, F, FA);
015     signal pr_state, nx_state: state;
016
017     --Timer-related declarations:
018     type int_array is array (integer range <>) of integer;
019     constant T1_TABLE: int_array(0 to 5) := (12_500_000, -- 250ms @fclk=50Mhz
020                                              9_000_000,  -- 180ms @fclk=50Mhz
021                                              6_500_000, -- 130ms @fclk=50Mhz
022                                              5_000_000, -- 100ms @fclk=50Mhz
023                                              3_500_000, --  70ms @fclk=50Mhz
024                                              2_000_000); -- 40ms @fclk=50Mhz
025     signal T1: natural := T1_TABLE(0); --120ms @ fclk=50MHz
026     constant T2: natural := 1_000_000; --20ms @ fclk=50MHz
027     constant tmax: natural := T1-1; --tmax ≥ max(T1,T2)-1
028     signal t: natural range 0 to tmax;
029     signal debounced: std_logic := '0'; --signal for debounced spd switch press
030     signal T1_index: integer := 0; --index for table of T1
031
032     begin
033         debouncer: entity work.switch_debouncer(switch_debouncer) port map(spd, debounced, clk);
034
035         --Timer (using strategy #1):
036         process (clk, rst, stp)
037             begin
038             if rst='0' then
039                 t <= 0;
040             elsif rising_edge(clk) and stp='0' then
041                 if pr_state /= nx_state then
```

```
042                 t <= 0;
043             elsif t /= tmax then
044                 t <= t + 1;
045             end if;
046         end if;
047     end process;
048 
049     --Speed controller
050     process (debounced, spd, clk)
051         begin
052         if falling_edge(debounced) then
053             T1_index <= T1_index + 1;
054             if T1_index = 5 then
055                 T1_index <= 0;
056             end if;
057             T1 <=T1_TABLE(T1_index);
058         end if;
059     end process;
060 
061     --FSM state register:
062     process (clk, rst)
063         begin
064         if rst='0' then
065             pr_state <= A;
066         elsif rising_edge(clk) then
067             pr_state <= nx_state;
068         end if;
069     end process;
070 
071     --FSM combinational logic:
072     process (all)
073         begin
074         case pr_state is
075             when A =>
076                 ssd <= "0111111";
077                 if dir = '1' then
078                     if t >= T1-1 then -- or t=T1-1
079                         nx_state <= AB;
080                     else
081                         nx_state <= A;
082                     end if;
083                 else
084                     if t >= T1-1 then -- or t=T1-1
085                         nx_state <= FA;
086                     else
087                         nx_state <= A;
088                     end if;
089                 end if;
090             when AB =>
091                 ssd <= "0011111";
092                 if dir = '1' then
093                     if t >= T2-1 then -- or t=T2-1
094                         nx_state <= B;
095                     else
096                         nx_state <= AB;
097                     end if;
098                 else
099                     if t >= T2-1 then -- or t=T2-1
100                         nx_state <= A;
101                     else
102                         nx_state <= AB;
103                     end if;
104                 end if;
105             when B =>
106                 ssd <= "1011111";
107                 if dir = '1' then
108                     if t >= T1-1 then -- or t=T1-1
109                         nx_state <= BC;
110                     else
111                         nx_state <= B;
```

```
                    end if;
                else
                    if t >= T1-1 then -- or t=T1-1
                        nx_state <= AB;
                    else
                        nx_state <= B;
                    end if;
                end if;
            when BC =>
                ssd <= "1001111";
                if dir = '1' then
                    if t >= T2-1 then -- or t=T2-1
                        nx_state <= C;
                    else
                        nx_state <= BC;
                    end if;
                else
                    if t >= T2-1 then -- or t=T2-1
                        nx_state <= B;
                    else
                        nx_state <= BC;
                    end if;
                end if;
            when C =>
                ssd <= "1101111";
                if dir = '1' then
                    if t >= T1-1 then -- or t=T1-1
                        nx_state <= CD;
                    else
                        nx_state <= C;
                    end if;
                else
                    if t >= T1-1 then -- or t=T1-1
                        nx_state <= BC;
                    else
                        nx_state <= C;
                    end if;
                end if;
            when CD =>
                ssd <= "1100111";
                if dir = '1' then
                    if t >= T2-1 then -- or t=T2-1
                        nx_state <= D;
                    else
                        nx_state <= CD;
                    end if;
                else
                    if t >= T2-1 then -- or t=T2-1
                        nx_state <= C;
                    else
                        nx_state <= CD;
                    end if;
                end if;
            when D =>
                ssd <= "1110111";
                if dir = '1' then
                    if t >= T1-1 then -- or t=T1-1
                        nx_state <= DE;
                    else
                        nx_state <= D;
                    end if;
                else
                    if t >= T1-1 then -- or t=T1-1
                        nx_state <= CD;
                    else
                        nx_state <= D;
                    end if;
                end if;
            when DE =>
                ssd <= "1110011";
```

```
182                     if dir = '1' then
183                         if t >= T2-1 then -- or t=T2-1
184                             nx_state <= E;
185                         else
186                             nx_state <= DE;
187                         end if;
188                     else
189                         if t >= T2-1 then -- or t=T2-1
190                             nx_state <= D;
191                         else
192                             nx_state <= DE;
193                         end if;
194                     end if;
195                 when E =>
196                     ssd <= "1111011";
197                     if dir = '1' then
198                         if t >= T1-1 then -- or t=T1-1
199                             nx_state <= EF;
200                         else
201                             nx_state <= E;
202                         end if;
203                     else
204                         if t >= T1-1 then -- or t=T1-1
205                             nx_state <= DE;
206                         else
207                             nx_state <= E;
208                         end if;
209                     end if;
210                 when EF =>
211                     ssd <= "1111001";
212                     if dir = '1' then
213                         if t >= T2-1 then -- or t=T2-1
214                             nx_state <= F;
215                         else
216                             nx_state <= EF;
217                         end if;
218                     else
219                         if t >= T2-1 then -- or t=T2-1
220                             nx_state <= E;
221                         else
222                             nx_state <= EF;
223                         end if;
224                     end if;
225                 when F =>
226                     ssd <= "1111101";
227                     if dir = '1' then
228                         if t >= T1-1 then -- or t=T1-1
229                             nx_state <= FA;
230                         else
231                             nx_state <= F;
232                         end if;
233                     else
234                         if t >= T1-1 then -- or t=T1-1
235                             nx_state <= EF;
236                         else
237                             nx_state <= F;
238                         end if;
239                     end if;
240                 when FA =>
241                     ssd <= "0111101";
242                     if dir = '1' then
243                         if t >= T2-1 then -- or t=T2-1
244                             nx_state <= A;
245                         else
246                             nx_state <= FA;
247                         end if;
248                     else
249                         if t >= T2-1 then -- or t=T2-1
250                             nx_state <= F;
251                         else
```

```
252                         nx_state <= FA;
253                     end if;
254                 end if;
255             end case;
256         end process;
257 
258     end architecture;
259     -------------------------------------------------------------
```