ELECTRICAL & COMPUTER ENGINEERING
School of Engineering

EGRE 365 – Digital Systems

Homework 4

Name: Luis Barquero

Major: Computer Engineering

Date: 10/31/17

Honor Pledge: *I have neither given nor received any unauthorized help on this lab.*
Signed:
_____*Luis Barquero*_____

In this Homework, the T-Bird example was modified to include the brake light functionality, alongside the Left and Right turn signal, and the Hazard functionality.

To fully implement the brake light functionality, the following changes were made to the original state transition graph:

Idle (000_000): If the hazard or brakes are on, then the system will transition to LR3, turning on all six lights. Similarly, if the left and right turns are both on, then the system will also enter the LR3 state.

Otherwise, if the left turn signal is activated, then the system will transition to L1, which will turn the first light on the left side, while having the right-side lights off since no brakes have been implemented. Similarly, if the right signal is activated, the system will enter R1, which will have the first light on the right-side on, and the left-side lights off since no brakes were activated.

LR3 (111_111): When the system is the LR3 state, all six lights are on. While in this state, if the brakes have been activated and no turn signal (left = '0' and right = '0') has been detected, then system will remain in the present state.

Otherwise, if the left turn signal is on and the brakes are on, the system enters L4, which will reset left-side lights, while maintaining the right-side lights on, because of the brakes. Similarly, if the right turn signal is on and the brakes are on, the system transitions to R4, which resets the right-side lights and keeps the left-side lights on.

Finally, if the system is LR3 because the hazard lights have been activated, the system will revert back to idle, to simulate the flashing of all six lights.

L1 (001_000): In this state, the first light on the left-side is on, while all the lights on the right-side are off. When the system is in L1, if the hazard lights have been activated, the system will transition to LR3. Otherwise, if no hazard lights have been detected, then the system enters L2 state, which turns the first two lights on the left-side on, and the right-side lights remain off.

Finally, if while in this state the brakes are activated, the system enters L6 state (011_111), which turns on the first two lights on the left-side, but since brakes = '1', all the lights on the right-side will turn on.

L2 (011_000): In this state, the first two lights on the right-side are on, while the lights on the right-side are off. While in this state, if hazard or brakes = '1', the system immediately enters LR3 state. Otherwise, if hazard and brakes = '0', the system transitions to L3 state (111_000), which turns on all lights on the left-side, and the lights on the right-side remain off.

L3 (111_000): In this state, all lights on the left-side are on, and the lights on the right-side are all off. While in this state, if the brakes = '1', the system enters L4 state (000_111), which resets the left-side lights while maintaining the right-side lights all on.

If hazard = ‘1’ or brakes = ‘1’, then the system enters LR3 (111_111), which will turn all six lights on. Otherwise, the system transitions the Idle state (000_000).

L4 (000_111): In this state, the left-side lights have been reset, while the right-side lights are all on since the brakes are still on. While in this state, if brakes = ‘0’, the system enters L1 (001_000). Otherwise, if brakes = ‘1’, the system transitions to L5 (001_111), which turns the first light on the left-side and all three lights on the right-side.

L5 (001_111): L5 will implement the brakes and left-turn signal by having all three lights on the right-side and the first light on the left-side. While in this state, if brakes = ‘0’, the system enters L2 (011_000). On the other hand, if brakes = ‘1’, the system transitions to L6 (011_111), which turns the first two lights on the left-side and all three lights on the right-side.

L6 (011_111): L6 will have the first two lights on the left-side on and all three lights on the right-side on. While in this state, if brakes = ‘0’, the system enters L3 (111_000). Otherwise, if brakes = ‘1’, the system transitions to LR3 (111_111), turning all six lights on.

R1 (000_100): R1 has the first light on the right-side on, and since brakes = ‘0’, the left-side lights are all off. While in this state, if hazards = ‘1’, the system transitions to LR3 (111_111), turning all six lights on. If brakes or hazards = ‘0’, then the system enters R2 (000_110), which turns on the first two lights on the right-side. Finally, if brakes = ‘1’, the system transitions to R6 (111_110), which turns on the first two lights on the right-side, maintaining the three lights on the right-side.

R2 (000_110): R2 has the first two lights on the right-side on, and since brakes = ‘0’, the left-side lights are off. Once again, if hazards or brakes = ‘1’, the system transitions to LR3 (111_111), turning all six lights on. Otherwise, if hazards or brakes = ‘0’, the system will transition to R3 (000_111), turning on all three lights on the right side, and no lights on the left-side.

R3 (000_111): R3 will turn on all three lights on the right-side, and since brakes = ‘0’, left-side lights are off. While in this state, if either hazards or brakes = ‘1’, then system enters LR3 (111_111), turning on all six lights. If, at this state, brakes = ‘1’, the system transitions to R4 (111_000), which resets the right-turn lights. Otherwise, the system goes to idle.

R4 (111_000): R4 will reset the left-turn lights after the brakes have been implemented. While in this state, if brakes = ‘0’, the system enters R1 (000_100), which turns off the left-side lights and turns on the first light on the right-side. Otherwise, if the brakes are still on, then the system enters R5 (111_100), which turns on the first light on the right-side.

R5 (111_100): R5 will turn on the first light on the right-side, and since brakes = ‘1’, all lights on the left-side are on. In this state, if the brakes = ‘0’, then the system will enter R2 (000_110), which will turn off the lights on the left-side and will turn on the first two lights on the right-side. Otherwise, if the brakes = ‘1’, then the system enters R6 (111_110), which turns on the second light on the right-side.

R6 (111_110): Final state is R6, which turns on the first two lights on the right-side and since brakes = ‘1’, all lights on the right-side are on. While in this state, if brakes = ‘0’, the system will enter R3 (000_111), turning off all lights on left-side and turning on all lights on the right-side. Otherwise, if brakes = ‘1’, system enters LR3 (111_111), turning on all six lights.

To following is the State Transition Graph illustrating all 14 states:

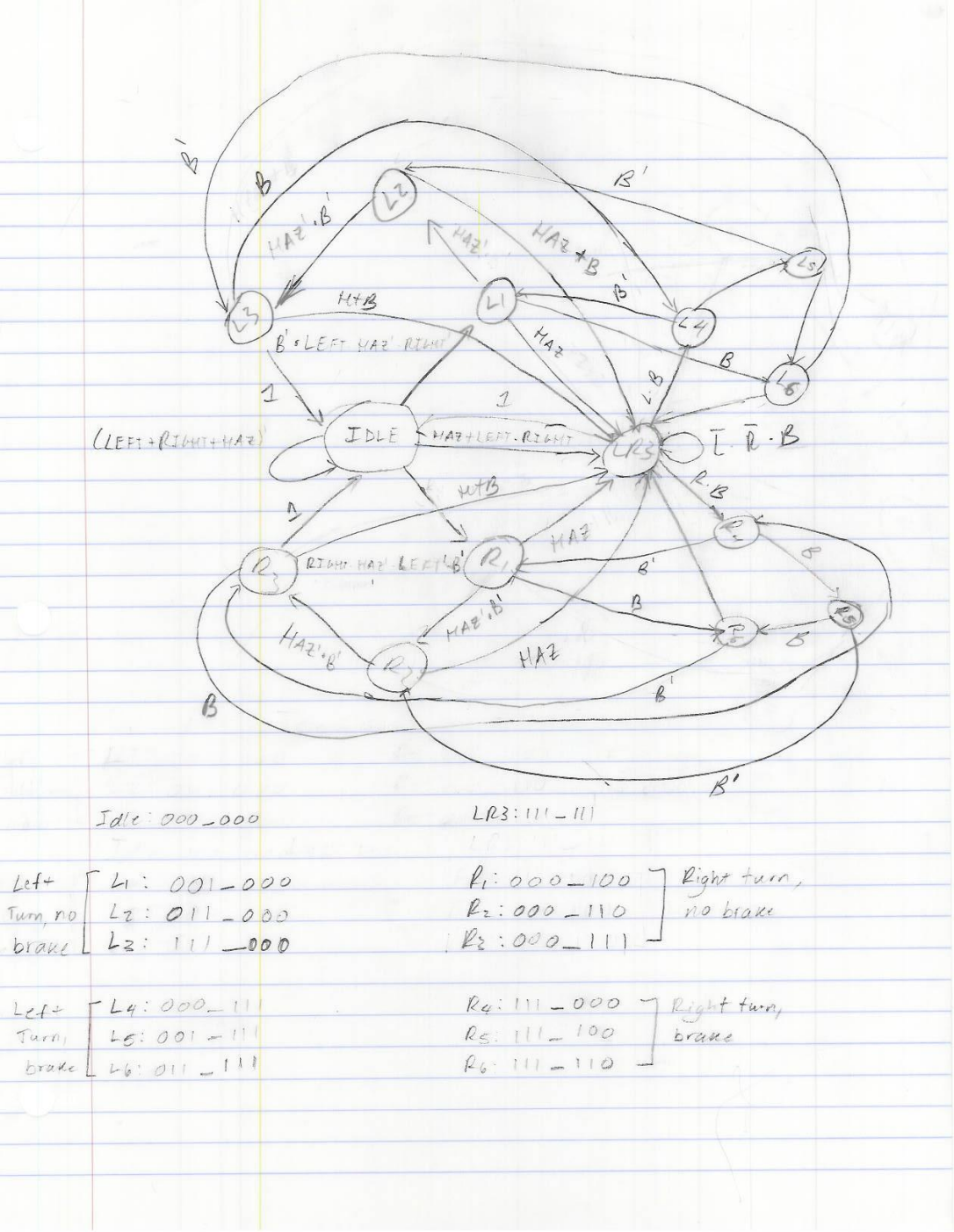


Figure 1 – Figure 1 contains the State Transition Graph for the T-Bird.

The T-Bird testbench was also modified to verify that the system transitions to the correct states. First, the system will have the left turn signal activated and then the right signal. Then, the hazards will be activated. From there, the system will turn on the left turn will be activated, and after 20 seconds, the brakes will be activated. Next, the brakes will turn off and after 20 seconds, the left turn also turns on, at which point the right turn signal is activated. After 20 seconds, the brakes turn on and after 60 ns, they will be turn off. After 20 seconds, the right turn signals turns off.

Appendix A
VHDL Code

```
--Programmed by: Luis Barquero
--Purpose: This program will take the T-Bird's light example, and implement the brakes.
     --For example, when the brakes are activated, all six lights are on.
     --When the left or turn signal are on, the lights the lights in the direction of the turn
     -- function as before ? blinking sequentially, the lights opposite the tail lights
     --are all on continuously.

library IEEE;
use IEEE.std_logic_1164.all;

ENTITY tbird_lc is
  PORT(clk           : IN  std_logic;
       rst           : IN  std_logic;
       left          : IN std_logic;
       right         : IN std_logic;
       haz           : in std_logic;
       brakes        : in std_logic;
       left_tail_lt  : OUT std_logic_vector(3 downto 1);
       right_tail_lt : OUT std_logic_vector(1 to 3));
END tbird_lc;

ARCHITECTURE behavior OF tbird_lc IS

  TYPE state_type IS (IDLE, LR3, L1, L2, L3, L4, L5, L6, R1, R2, R3, R4, R5, R6);
  SIGNAL present_state, next_state : state_type;
  CONSTANT leftoff : std_logic_vector(3 downto 1) := "000";
  CONSTANT left1on : std_logic_vector(3 downto 1) := "001";
  CONSTANT left2on : std_logic_vector(3 downto 1) := "011";
  CONSTANT left3on : std_logic_vector(3 downto 1) := "111";
  CONSTANT rightoff : std_logic_vector(3 downto 1) := "000";
  CONSTANT right1on : std_logic_vector(3 downto 1) := "100";
  CONSTANT right2on : std_logic_vector(3 downto 1) := "110";
  CONSTANT right3on : std_logic_vector(3 downto 1) := "111";

BEGIN

 clocked : PROCESS(clk,rst)
   BEGIN
     IF(rst='1') THEN
       present_state <= idle;
    ELSIF(rising_edge(clk)) THEN
      present_state <= next_state;
    END IF;
 END PROCESS clocked;

 nextstate : PROCESS(present_state,left,right,haz, brakes)
  BEGIN
     CASE present_state IS
       WHEN idle => --(000_000). Idle will have all lights turned off.
         IF(haz = '1'OR (left = '1' AND right = '1') OR brakes = '1') THEN
           next_state <= LR3;
         ELSIF(left = '1') THEN
           next_state <= L1;
         ELSIF(right = '1') THEN
           next_state <= R1;
         ELSE
           next_state <= idle;
         END IF;

       WHEN LR3 => --(111_111). LR3 will have all six lights(all 3 from left side and all 3 from right side).
     IF(left = '1' AND brakes = '1') THEN
       next_state <= L4;
     ELSIF (right = '1' AND brakes = '1') THEN
        next_state <= R4;
       ELSIF ((left = '0') AND (right = '0') AND (brakes = '1')) THEN
          next_state <= present_state;
     ELSE
```

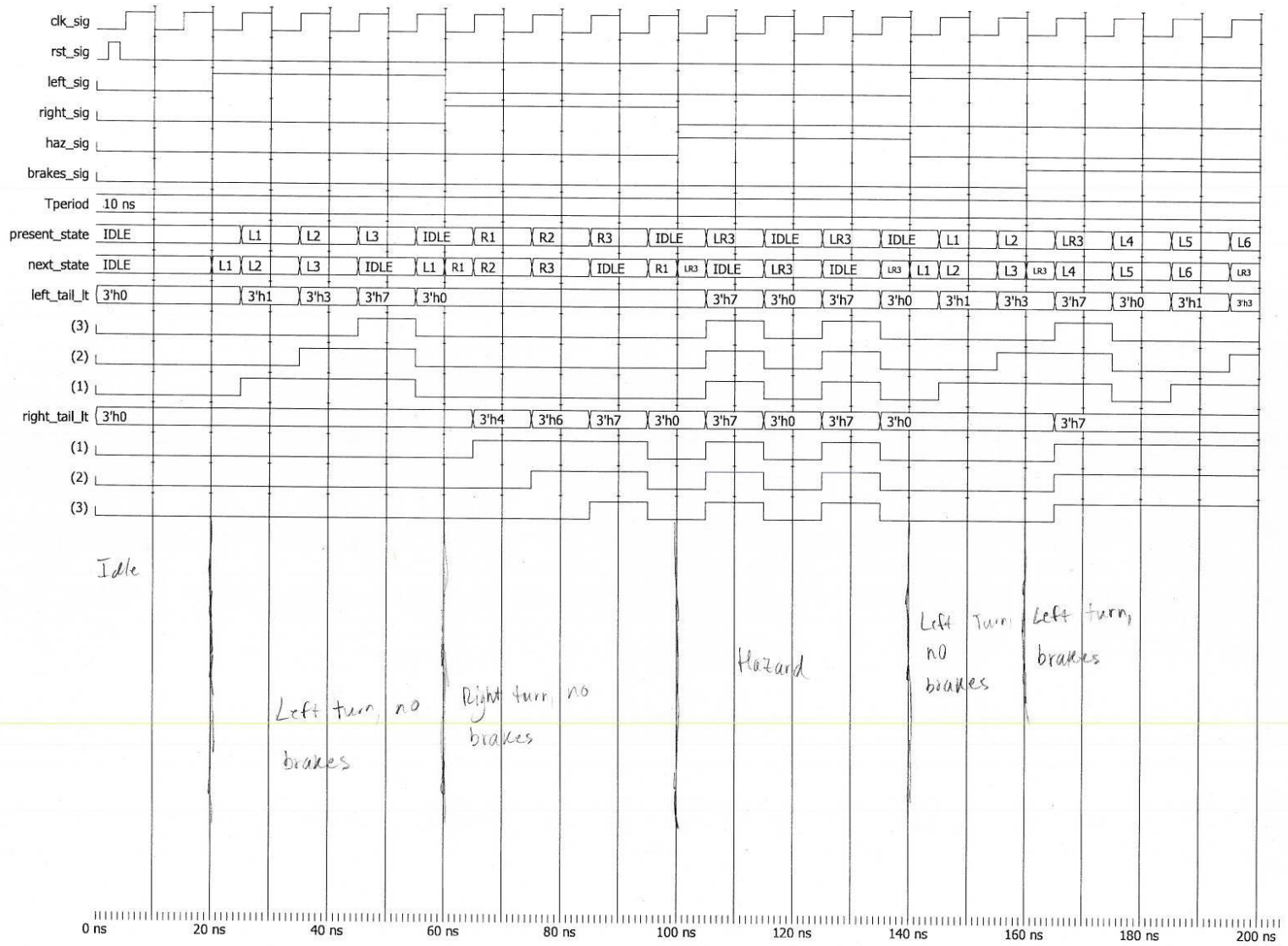
```
68            next_state <= idle;
69            END IF;
70
71          WHEN L1 => --(001_000). L1 will implement the first left light, with the right
            lights off because of no brakes.
72            IF(haz = '1') THEN
73              next_state <= LR3;
74            ELSIF (brakes = '1') THEN
75          next_state <= L6;
76            ELSIF (haz = '0') THEN
77          next_state <= L2;
78            END IF;
79
80          WHEN L2 => --(011_000). L2 will implement the first two left light, with the
            right lights off because of no brakes.
81            IF(haz = '1' OR brakes = '1') THEN
82              next_state <= LR3;
83            ELSIF (haz = '0' AND brakes = '0') THEN
84              next_state <= L3;
85            END IF;
86
87          WHEN L3 => --(111_000). L3 will have all 3 lights from left side, with the right
            side lights off because of no brakes.
88            IF(brakes = '1') THEN
89               next_state <= L4;
90            ELSIF (haz = '1' OR brakes = '1') THEN
91               next_state <= LR3;
92            ELSE
93               next_state <= idle;
94          END IF;
95
96          WHEN L4 => -- (000_111) Left and Brakes. L4 will reset the left turn lights
            after all three have been lit when the brake has been activated.
97         IF (brakes = '0') THEN
98            next_state <= L1;
99         ELSIF (brakes = '1') THEN
100           next_state <= L5;
101         END IF;
102
103         WHEN L5 => --(001_111) Left and Brakes. L5 will have the first left light on and
            all the right lights on, because of the brakes.
104       IF (brakes = '0') THEN
105              next_state <= L2;
106        ELSIF (brakes = '1') THEN
107        next_state <= L6;
108        END IF;
109
110       WHEN L6 => --(011_111) Left and Brakes. L6 will have the firt two left lights on
          and all the right lights on, because of the brakes.
111        IF (brakes = '0') THEN
112           next_state <= L3;
113        ELSIF (brakes = '1') THEN
114           next_state <= LR3;
115            END IF;
116
117         WHEN R1 => --(000_100). R1 will implement the first right light, with the left
            lights off because of no brakes.
118           IF(haz = '1') THEN
119             next_state <= LR3;
120           ELSIF (haz = '0' AND brakes = '0') THEN
121             next_state <= R2;
122        ELSIF (brakes = '1') THEN
123             next_state <= R6;
124           END IF;
125
126         WHEN R2 => --(000_110). R2 will implement the first two right lights, with the
            left lights off because of no brakes.
127           IF(haz = '1' or brakes = '1') THEN
128             next_state <= LR3;
```

```
129             ELSIF (haz = '0' AND brakes = '0') THEN
130               next_state <= R3;
131             END IF;
132
133           WHEN R3 => --(000_111). R3 will have all 3 lights from right side, with the left
              side lights off because of no brakes.
134         IF (haz = '1' OR brakes = '1') THEN
135            next_state <= LR3;
136         ELSIF (brakes = '1') THEN
137            next_state <= R4;
138         ELSE
139                next_state <= idle;
140           END IF;
141
142        WHEN R4 => -- (111_000) Right and Brakes. R4 will reset the right turn lights after
           all three have been lit when the brake has been activated.
143         IF (brakes = '0') THEN
144            next_state <= R1;
145         ELSIF (brakes = '1') THEN
146            next_state <= R5;
147           END IF;
148
149        WHEN R5 => --(111_100) Right and Brakes. L5 will have the first right light on and
           all the left lights on, because of the brakes.
150        IF (brakes = '0') THEN
151                next_state <= R2;
152            ELSIF (brakes = '1') THEN
153            next_state <= R6;
154        END IF;
155
156        WHEN R6 => --(111_110) Right and Brakes. L5 will have the first two right light on
           and all the left lights on, because of the brakes.
157        IF (brakes = '0') THEN
158            next_state <= R3;
159        ELSIF (brakes = '1') THEN
160            next_state <= LR3;
161        END IF;
162
163        END CASE;
164      END PROCESS nextstate;
165
166      output : PROCESS(present_state)
167       BEGIN
168         CASE present_state IS
169           WHEN idle =>  --(000_000)
170             left_tail_lt <= leftoff;
171             right_tail_lt <= rightoff;
172
173           WHEN LR3 => --(111_111)
174             left_tail_lt <= left3on;
175             right_tail_lt <= right3on;
176
177           WHEN L1 => --(001_000)
178             left_tail_lt <= left1on;
179             right_tail_lt <= rightoff;
180
181           WHEN L2 => --(011_000)
182             left_tail_lt <= left2on;
183             right_tail_lt <= rightoff;
184
185           WHEN L3 => --(111_000)
186             left_tail_lt <= left3on;
187             right_tail_lt <= rightoff;
188
189          WHEN L4 => --(000_111)
190         left_tail_lt <= leftoff;
191         right_tail_lt <= right3on;
192
193          WHEN L5 => --(001_111)
```

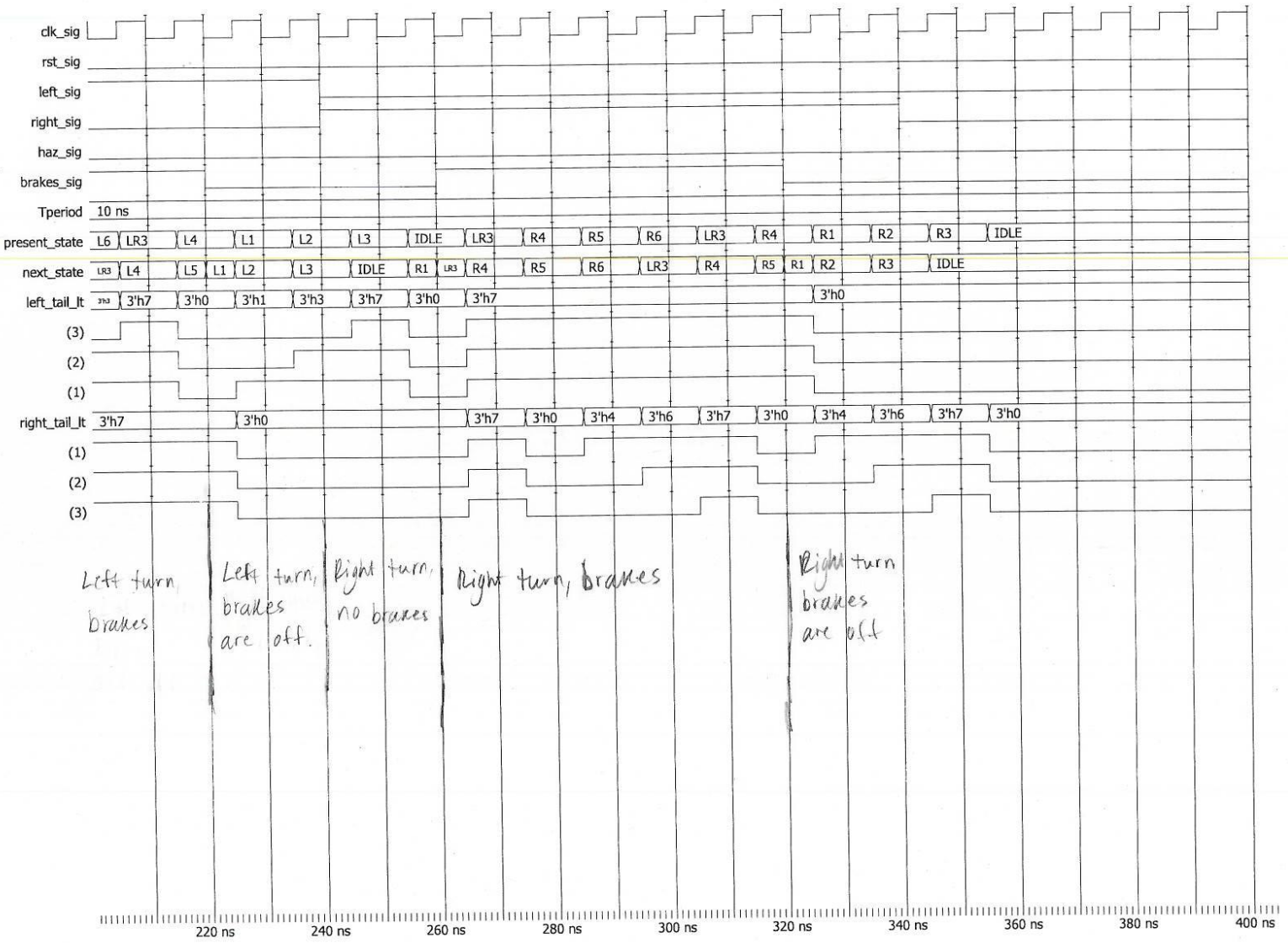
```
194         left_tail_lt <= left1on;
195         right_tail_lt <= right3on;
196 
197          WHEN L6 => --(011_111)
198         left_tail_lt <= left2on;
199             right_tail_lt <= right3on;
200 
201           WHEN R1 => --(000_100)
202             left_tail_lt <= leftoff;
203             right_tail_lt <= right1on;
204 
205           WHEN R2 => --(000_110)
206             left_tail_lt <= leftoff;
207             right_tail_lt <= right2on;
208 
209           WHEN R3 => --(000_111)
210             left_tail_lt <= leftoff;
211             right_tail_lt <= right3on;
212 
213           WHEN R4 => --(111_000)
214         left_tail_lt <= left3on;
215         right_tail_lt <= rightoff;
216 
217          WHEN R5 => --(111_100)
218         left_tail_lt <= left3on;
219         right_tail_lt <= right1on;
220 
221          WHEN R6 => --(111_110)
222         left_tail_lt <= left3on;
223             right_tail_lt <= right2on;
224 
225 
226         END CASE;
227      END PROCESS output;
228 
229    END ARCHITECTURE behavior;
```

```
--Programmed by: Luis Barquero
--Purpose: Testbench will simulate the T-Bird light example, except there is the inclusion of the brakes
   --The different states will be: Idle, Left Turn(no brakes), Right Turn(no brakes), Hazards, Left Turn(with brakes),
   --                              Right Turn(with brakes).

library ieee;
use ieee.std_logic_1164.all;

entity tbird_testbench is
end tbird_testbench;

architecture behavior of tbird_testbench is

  signal clk_sig : std_logic := '0';
  signal rst_sig : std_logic := '0';
  signal left_sig,right_sig,haz_sig,brakes_sig : std_logic;
  constant Tperiod : time := 10 ns;

  begin

    process(clk_sig)
      begin
        clk_sig <= not clk_sig after Tperiod/2;
    end process;

  rst_sig <= '0', '1' after 2 ns, '0' after 4 ns;        --Reset Signal

--Left will be on from 20 to 60 ns(no brake), then it will be on from 140-240 (with brakes from 160-220).

  left_sig <= '0', '1' after 20 ns, '0' after 60 ns, '1' after 140 ns, '0' after 240 ns;


--Right will be on from 60 to 100 ns(no brake), then it will be on from 240-340 (with brakes from 260-320).

  right_sig <= '0', '1' after 60 ns, '0' after 100 ns, '1' after 240 ns, '0' after 340 ns ;


--Hazard will be on from 100 to 140 ns.

  haz_sig <= '0', '1' after 100 ns, '0' after 140 ns;


--Brakes will be on from 160 to 220 ns(for left turn with brakes), then it will be on from 260-320 (for right turn with brakes).

  brakes_sig <= '0', '1' after 160 ns, '0' after 220 ns, '1' after 260 ns, '0' after 320 ns;


    -- this is the component instantiation for the
    -- DUT - the device we are testing
    DUT : entity work.tbird_lc(behavior)
      port map(clk => clk_sig, rst => rst_sig,
               left => left_sig, right => right_sig,
               haz => haz_sig, brakes => brakes_sig);


end behavior;
```

Appendix B
Simulation Output Waveforms



Entity:tbird_testbench Architecture:behavior Date: Tue Oct 31 14:17:32 AKDT 2017 Row: 1 Page: 1



Entity:tbird_testbench Architecture:behavior Date: Tue Oct 31 14:17:32 AKDT 2017 Row: 2 Page: 2