Students: Kevin (Kaiwen) Xue - 20814292, Anni Wang - 20849760

BME 393L: Pre-Lab 3

Question-1

The VHDL code that was given (seven_segment.vhd) was completed for the remaining cases/characters.

seven_segment.vhd

```
-- This code is half-written. To be completed by students.
-- The goal is showing the value of data_in input on the segments_out output.
--    ┌---a---┐
--    f       b
--    |       |
--    ├---g---┤
--    |       |
--    e       c
--    |       |
--    └---d---┘
-- Each segment of a, b, c, d, e, f, and g can be turned ON or OFF independently.
-- The following line is from page 32 of the C5G_User_Manual document available on FPGA_Datasheets folder on LEARN.
-- "Applying a low logic level to a segment will light it up and applying a high logic level turns it off."

 library ieee;
 use ieee.std_logic_1164.all;
 use ieee.numeric_std.all;

entity seven_segment is port(
         SW:      in std_logic_vector(4 downto 0); -- The 4 bit data to be displayed
         HEX0:    out std_logic_vector(6 downto 0) ); -- 7 bits out to a 7-segment display
end entity seven_segment;

architecture behavioral of seven_segment is
signal data_in: std_logic_vector(3 downto 0);   -- The 4 bit data to be displayed
signal blanking: std_logic;                     -- Blank the output if this is input is set to HIGH
signal segments_out: std_logic_vector(6 downto 0); -- 7 bits out to a 7-segment display
begin
      data_in(3 downto 0) <= SW(3 downto 0);
      blanking <= Sw(4);
      with blanking & data_in select      -- seven-segment bits are 'a' to 'g'. The LSB is 'a' and MSB is 'g'
                        --   gfedcba
      segments_out(6 downto 0) <=   "1000000" when "00000",    -- 0
                           "1111001" when "00001",    -- 1
                           "0100100" when "00010",    -- 2
                           "0110000" when "00011",    -- 3
                           "0011001" when "00100",    -- 4
                           "0010010" when "00101",    -- 5
                           "0000010" when "00110",    -- 6
                           "1111000" when "00111",    -- 7
                           "0000000" when "01000",    -- 8
                           "0010000" when "01001",    -- 9
                           "0001000" when "01010",    -- a
                           "0000011" when "01011",    -- b
                           "1000110" when "01100",    -- c
                           "0100001" when "01101",    -- d
                           "0000110" when "01110",    -- e
                           "0001110" when "01111",    -- f
                           "0000000" when others;  -- "1xxxx", blank

      HEX0 <= segments_out;


end architecture behavioral;

```

Figure 1: VHDL code modelling the entity sitting between the input and the 7-segment display

Question-2

A truth table was generated given the problem description, and was used to derive K-maps and boolean equations.

Table 1: Truth table for the circuit to be designed for the motor control of a simplified elevator

A	B	Y	Z	ENABLE	DISABLE
0	0	X	X	0	X
X	X	0	0	0	X
0	1	0	1	0	X
0	1	1	X	1	1
1	0	0	1	1	0
1	0	1	0	0	X
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	X

Here, the first and second rows of **Table 1** (not counting the header row) accounts for the safety considerations, where an input wire could be potentially broken, and the input being registered as a LOW. It ensures that the ENABLE output is always LOW in those cases.

YZ \ AB	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	1	0	1
10	0	1	1	0

Figure 2: K-map of output, ENABLE

From the K-map in **Figure 2**, the following boolean equation can be generated:

$ENABLE = A\overline{B}Z + A\overline{Y}Z + \overline{A}BY + BY\overline{Z} = AZ(\overline{B} + \overline{Y}) + BY(\overline{A} + \overline{Z})$

By using Theorem T8, the number of total gates required for the $ENABLE$ output have been reduced, thus, improving the safety of the system.

AB / YZ	00	01	11	10
00	X	X	X	X
01	X	X	0	0
11	X	1	X	1
10	X	1	0	X

Figure 3: K-map of output, DIRECTION

From the K-map in **Figure 3**, the following boolean equation can be generated:
$DIRECTION = \overline{A} + YZ$

Using the reduced boolean expressions for ENABLE and DIRECTION, the following circuit (**Figure 4**) was created for the elevator system.

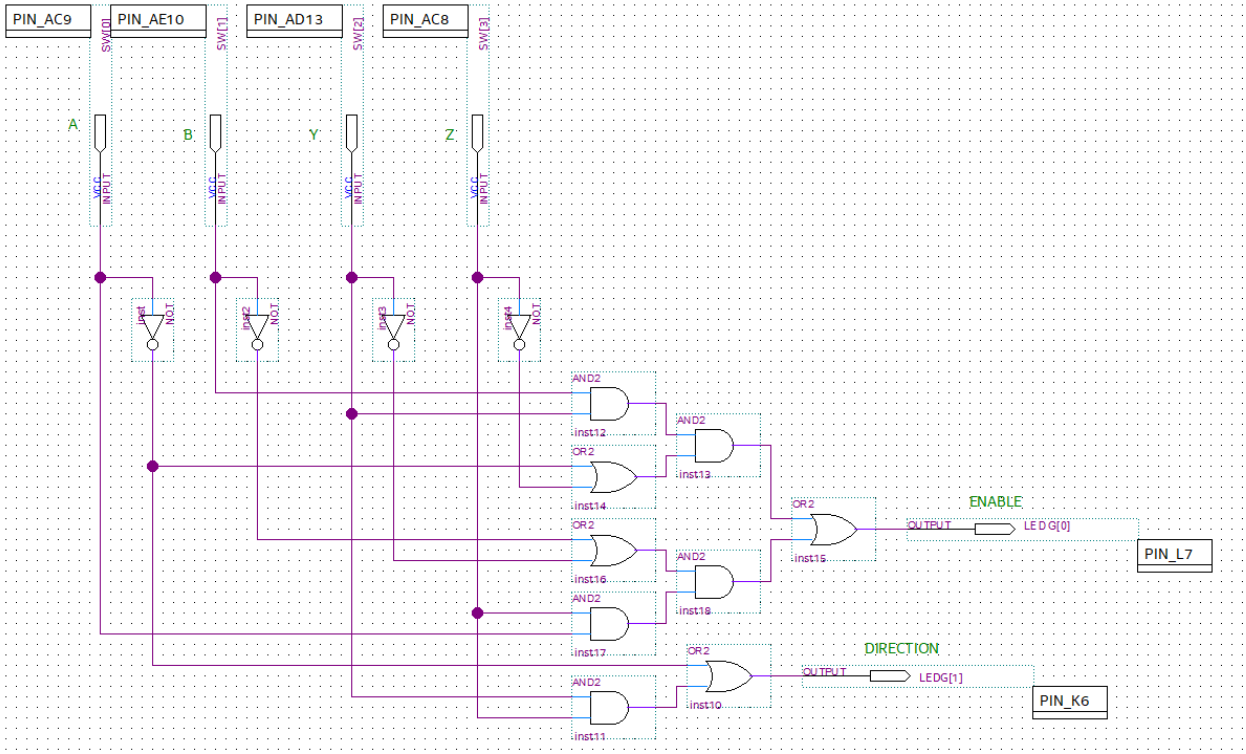

Figure 4: Schematic entry of the designed circuit using primitive gates.

The circuit was then simulated, and the results shown in **Figure 5** matched the truth table (**Table 1**), which was as expected.

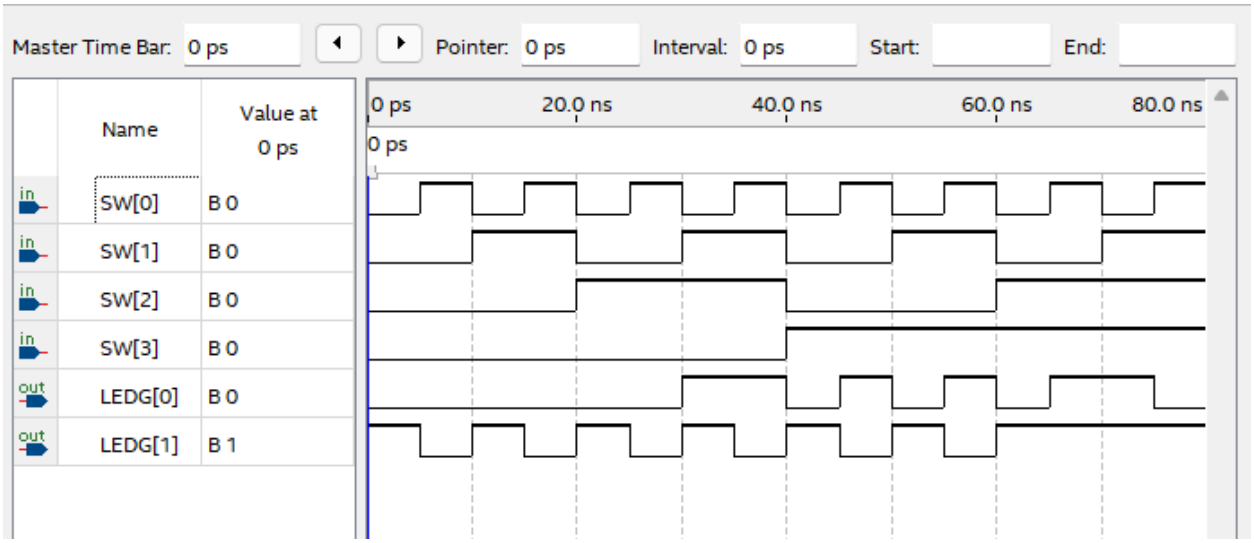

Figure 5: Simulation results of the circuit shown in **Figure 4**.