Indian Institute of Technology Delhi

COL 215 - Digital Logic and System Design

Assignment 1

Displaying 16-bit input in hexadecimal format on a 7-segment display using timing circuit

Aryan Sharma, Garv Nagori
2021CS10553, 2021CS10549

Contents

1 Introduction

The objective of this assignment is to display a 16-bit input hexadecimal number on four 7-segment displays, the 16 input bits are made up of 4 groups of 4 bit numbers, where each of these 4 numbers are to be converted to their corresponding hexadecimal representation. We are going to acheive the same on a **BASYS3 FPGA board** by programming it(program is written in **VHDL on vivado(editor)**).

2 Strategy

We are going to implement the timing circuit below

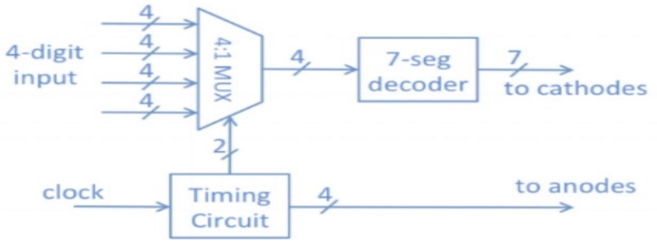


Figure 1: Timing Circuit For Display

The reason behind using a timing circuit is that all the cathodes of the four 7-segment displays available on the BASYS3 FPGA board have a common point of contact(this can be seen in Figure 2), thus if we keep all the anodes in **ACTIVE** state, then we can display only one number on all the four displays, to overcome this shortcoming, we use the **Clock(timing circuit)** instead, the timing circuit outputs to selection which go into the **MUX(4:1)** to select which anode to transmit to the **7-segment decoder**, thus this design keeps only one of the 4 displays switched on at a time but the frequency of the timing is circuit is of the order of milliseconds, thus to the viewer it appears as if all the 4 digits are simultaneously being displayed.
Thus we created three components in our implementation namely, **MUX(4:1)**, **Clock(the timing circuit)**, **Decoder(7 segment decoder)**.

2.1 Clock

This component takes as input the clk signal from the ports in the BASYS3 board and the ouputs of this component are two selection bits s1 and s2. In order to display all the 4 hexa decimal digits simultaneously, we need to refresh each digit within 1-16 ms, to acheive this time we use the internal clock of the BASYS3 FPGA board, we maintain an integer type signal count to count the falling edges of the clock signal from the BASYS3 board, by checking the value of the signal count using if/else statements we output selections bits s1 and s2(as output signals) We break 10 ms into following ranges and ouput the 4 possible combinations of s1 ans s2 in each of these 4 possibilities, for aur program, the signal(of integer type) count acts as time. To implement the same we define a sensitivity list (clk).
(a) 0ms ¡= time ¡= 2.5ms we output s1 ¡= '1' and s2 ¡= '1'.
(b) 2.5ms ¡= time ¡= 5ms we output s1 ¡= '1' and s2 ¡= '0'.
(c) 5ms ¡= time ¡= 7.5ms we output s1 ¡= '0' and s2 ¡= '1'.
(d) 7.5 ¡= time ¡= 10ms we output s1 ¡= '0' and s2 ¡= '0'.
This cycle keeps repeating as the variable count is set to 1 after each cycle.

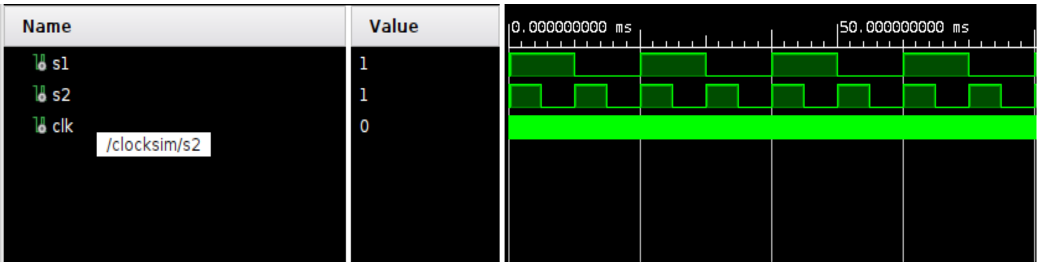
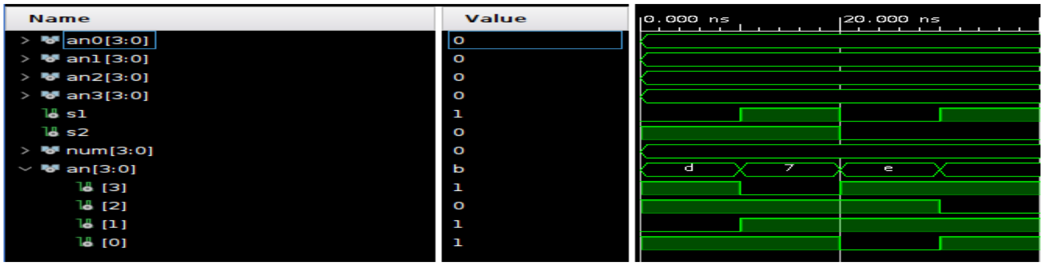


Figure 2: Simulation Of clock, s1 and s2 signals

2.2 MUX

The input terms for the MUX are 4 std_logic_vector(3 downto 0)(namely an0,an1,an2,an3) and two selection bits out of which depending on the input from the timing circuit we select the vector which we wish to turn on depending on the input of selection bits which we receive from the Clock(timing circuit), the selection bits used are s1 and s2,The outputs of the MUX are two std_logic_vector(3 downto 0) (namely num and an), the vectors an0,an1,an2,an3 signify the 4 bits used to display the number on each of the display(from left to right in the order 3 to 0), the vector an

represent which of the 4 input vectors will be transmitted which depends on the state of the selection bits, whereas the vector num represents the state of the anode pins on the BASYS3 board. We define a process with sensitivity list (s1,s2,an0,an1,an2,an3).
We use the following selection scheme, when
(a) both s1 and s2 are ACTIVE, an3 is transmitted and the vector num is set to ”0111”.
(b) only s1 is ACTIVE, an2 is transmitted and the vector num is set to ”1011”.
(c) only s2 is ACTIVE, an1 is transmitted and the vector num is set to ”1101”.
(d) both s1 and s2 are INACTIVE, an0 is transmitted and the vector num is set to ”1110”.

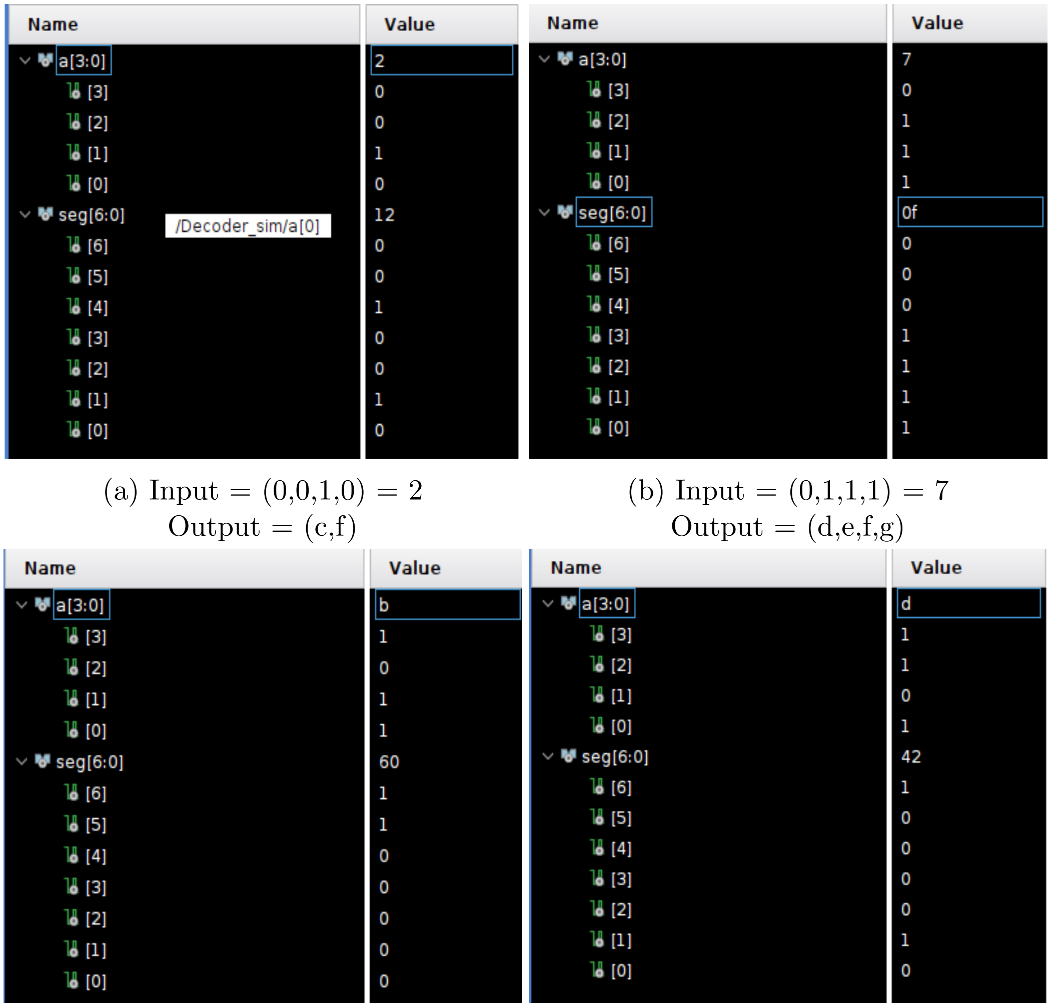


Name	Value
an0[3:0]	0
an1[3:0]	0
an2[3:0]	0
an3[3:0]	0
s1	1
s2	0
num[3:0]	0
an[3:0]	b
[3]	1
[2]	0
[1]	1
[0]	1

Figure 3: Simulation of MUX, selecting appropriate 4-bit input vector

2.3 Decoder

This component takes as input an std_logic_vector(3 downto 0) namely input this represents the current hexadecimal 4 - bit input we receive as the output of the MUX. The decoder maps this vector to another vector named seg(std_logic_vector(6 downto 0)), this represents the input for the 7 segments of the display unit. We use the function designed in previous Software Assignment opt_function_reduce to find the minimised boolean expression from the minterms, we made the minterm form of the boolean function by manually writing out the truth table for each of the segment. We assigned the 7 positions of the seg array the corresponding values by using the minimsed boolean functions.



Name	Value
a[3:0]	2
[3]	0
[2]	0
[1]	1
[0]	0
seg[6:0]	12
[6]	0
[5]	0
[4]	1
[3]	0
[2]	0
[1]	1
[0]	0

(a) Input = (0,0,1,0) = 2
Output = (c,f)

Name	Value
a[3:0]	7
[3]	0
[2]	1
[1]	1
[0]	1
seg[6:0]	0f
[6]	0
[5]	0
[4]	0
[3]	1
[2]	1
[1]	1
[0]	1

(b) Input = (0,1,1,1) = 7
Output = (d,e,f,g)

Name	Value
a[3:0]	b
[3]	1
[2]	0
[1]	1
[0]	1
seg[6:0]	60
[6]	1
[5]	1
[4]	0
[3]	0
[2]	0
[1]	0
[0]	0

(c) Input = (1,0,1,1) = 11(c)
Output = (a,b)

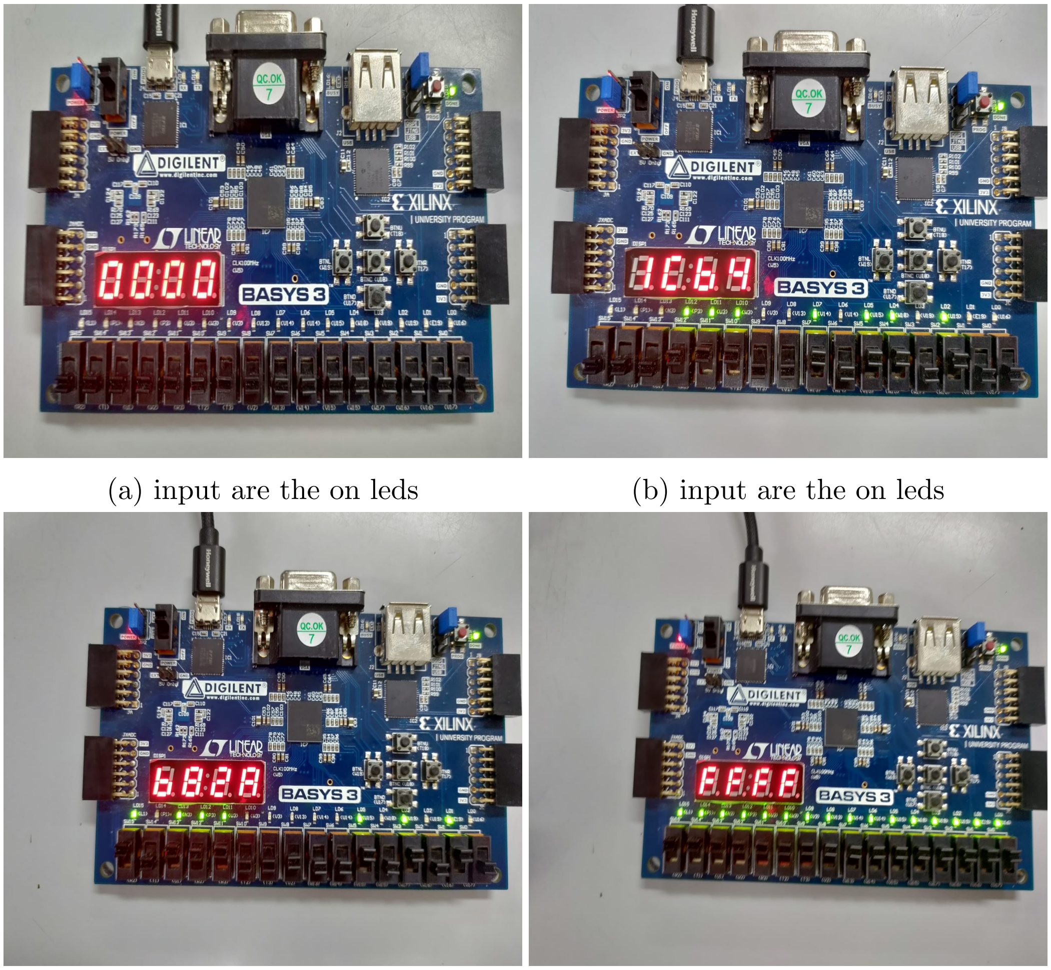
Name	Value
a[3:0]	d
[3]	1
[2]	1
[1]	0
[0]	1
seg[6:0]	42
[6]	1
[5]	0
[4]	0
[3]	0
[2]	0
[1]	1
[0]	0

(d) Input = (1,1,0,1) = 13(d)
Output = (a,f)

Figure 4: Simulation of the Decoder, corresponding to different numbers

3 Main Component

We use the above three discussed components together to finally implement the timing circuit, In this we take as input 4 std_logic_vector(3 downto 0) namely an0,an1,an2,an3 and the clk signal(from the internal clock of the BASYS3 board). Two temporary signals are created to store the value of output of the component clock, these valiues are then passed into the MUX along with the vectors an0,an1,an2,an3, another temporary antemp and temp are also created to map the output signals of the component MUX. The signal temp is then mapped to the input of Decoder, the corresponding output is the 7 segment segjents being assigned corresponding signals, the anodes are assigned the values of antmep. This completes our timing circuit implementation. As a good representative measure we have also mapped all the leds corresponding to each of 16-bit input switches.



(a) input are the on leds

(b) input are the on leds

(c) input are the on leds

(d) input are the on leds

Figure 5: Pictures Of BASYS3 board, the on led represents that the corresponding switches are on

4 Conclusion

In this Assignment we used VHDL to create a four digit seven segment display which would display decimal/hexadecimal numbers inputted in binary using the sixteen switches given on the BASYS board. A decoder was implemented to convert four bits into the corresponding seven segments on the board. A MUX to select which number would be displayed and a Clock circuit to control the selection bits of the MUX.