

## EXPERIMENT 6

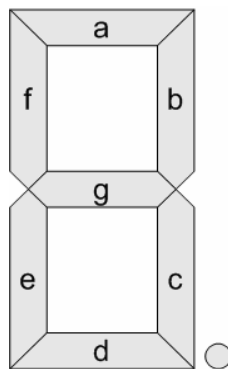
### 6. Seven-Segment Display Decoder.

#### 6.1 Objectives

- Practice using the K-map simplification method
- Designing a seven-segment display decoder.
- Introduction to time multiplexing concept.

#### 6.2 Overview

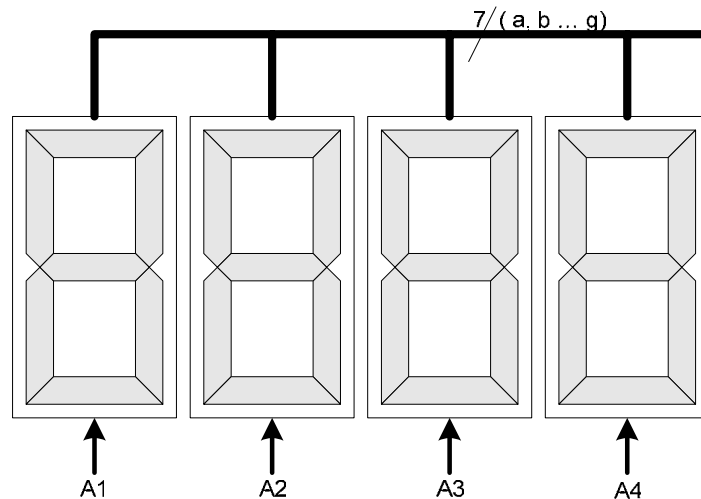
A seven-segment, like the one shown in Figure 6.1, is a common output device used in many digital applications like timers. The seven-segment is so called because of the seven LEDs, or segments, it is composed of. There are two types of seven segment displays, common anode and a common cathode seven-segment display. In either case, one end, either positive or negative, of all the seven segments are connected together to a common pin. The seven-segment in which positive end of all the segments are joined together to a common terminal is referred as a common anode seven segment. On the contrary, common-cathode seven-segment display has negative terminals of its seven-segments connected to a common node. To light a segment in the seven-segment display, the common terminal is first connected to logic high for common anode or to logic low for common cathode. The other ends for individual segments are then connected to logic low for common anode and logic high for a common cathode type seven-segment display.



**Figure 6.1: Seven-Segment Display.**

The Digilab provides four common-anode type seven-segment displays. The common anode for the first seven-segment display is called A1; A2 for the second display, and so on. The segments of each display are called *a*, *b*, up to *g* as shown in Figure 6.1. In order to reduce the number of connections needed on the board, the cathode pins from each display have been connected together to form *seven common terminals*, called *a*, *b*, *c*, *d*, *e*, *f* and *g*, corresponding to the seven-segments as shown in

Figure 6.2. Thus to illuminate a segment of a particular display, e.g. segment *e* of the third display, one needs to apply a “1” (High or V<sub>dd</sub>) to anode A3 (to first select the segment A3) and a “0” (GND or Low) to cathode node *e* (to light segment *e*).



**Figure 6.2: Four Seven-Segment Displays with common segments inputs.**

The pin connection for seven-segment display available on the FPGA board is given in **Error! Reference source not found.** of the lab guide.

Power dissipation of the seven-segment display is usually fairly considerable. To save on power, typically seven-segment displays are not enabled continuously but are rather alternately enabled and disabled with a frequency such that the *human eye* will have the illusion that the display is continuous. The human eye and brain retain a visual impression for about 1/30th of a second. (The exact time depends on the brightness of the image.) This ability to retain an image is known as ***persistence of vision***. As a conclusion from this fact, a display should not be disabled for more than 1/30th of a second; otherwise human eyes will notice the discrete behavior.

If two or more distinct digits are to be displayed, then *time-multiplexing* is employed. In the context of this experiment, time multiplexing simply means that every one of the four seven-segment displays is enabled for a fixed amount of time, then disabled and another one is enabled for the same amount of time, and this process continues in a circular manner.

The seven inputs that control the illumination of the seven segments are common to all the four digits. The 4 digits (7-seg displays) however, have ***independent enable inputs*** (A1, A2, A3, A4). Figure 6.2 shows this arrangement. Thus, the 4 digits receive the same 7 control inputs. However, only digits which are enabled will display the corresponding pattern.

### 6.3 Design Specifications

You need to design decoder that accepts 4-bit code (one hexadecimal digit) and outputs the seven segments states that correspond to the supplied hexadecimal number.

A seven-segment display decoder is a combinational circuit that accepts a 4-bit input and generates appropriate outputs for enabling/disabling each of the 7 segments so as to display the corresponding decimal digit. The decoder has seven outputs (*a*, *b*, *c*, *d*, *e*, *f*, and *g*) each output controls the corresponding segment in the seven-segment display as shown in Figure 6.1.

For example, for a zero to be displayed, all segments should be enabled (connected to GND) with the exception of segment-*g*, which should be disabled (connected to Vdd).

Furthermore, you need to find the frequency at which when you time-multiplex the four seven-segment displays you eye will see it as if it is always on.

## 6.4 Pre-Lab

A partially filled truth table for the circuit is given in Table 6.1. Complete the table by filling in the values for the seven outputs so that the circuit will display the correct decimal digit when the input lines supply the corresponding Hex code as shown in Figure 6.3.

It is important to notice that **for a segment to be ON, it should be connected to the GND.**

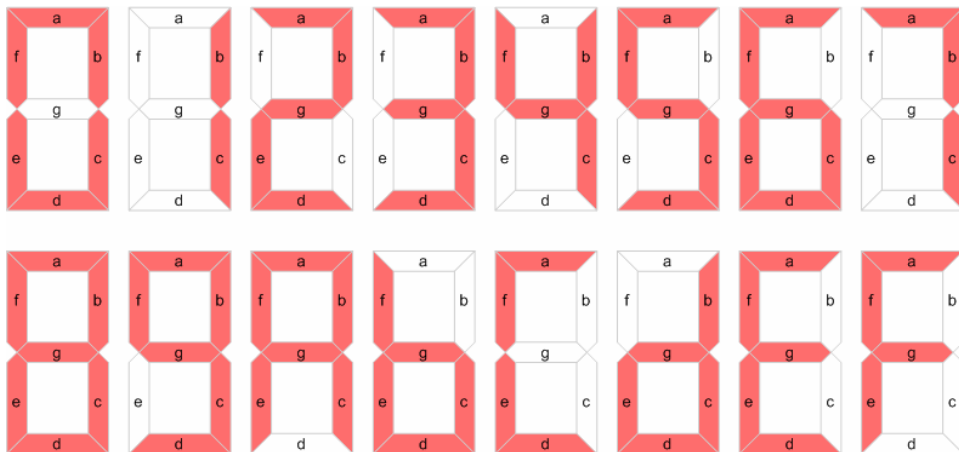


Figure 6.3: Hex Digits on the Seven-Segment Display.

Table 6.1: Truth table Seven-Segment Display Decoder.

Decimal digit	Inputs				Outputs						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0						
1	0	0	0	1	1						
2	0	0	1	0	0						
3	0	0	1	1	0						
4	0	1	0	0	1						
5	0	1	0	1	0						
6	0	1	1	0	0						
7	0	1	1	1	0						
8	1	0	0	0	0						
9	1	0	0	1	0						
10	1	0	1	0	0						
11	1	0	1	1	1						
12	1	1	0	0	0						
13	1	1	0	1	1						
14	1	1	1	0	0						
15	1	1	1	1	0						

Use K-Maps to simplify each segment function. You should come to the lab with seven simplified Boolean equation (one for every segment).

## 6.5 In-Lab

### 6.5.1 Hex to Seven-Segment Decoder

1. In a new schematic sheet, build a macro for the above Hex. to seven-segment display decoder. The macro should accept a Hex. digit (4 bits) as input. Constraint the outputs of this macro to the seven-segments directly.
2. Connect four switches (SW5, SW6, SW7, and SW8) to the four inputs of your decoder inputs (D, C, B, and A) respectively.
3. Constrain the anode of the seven-segment display you have chosen to the Vdd, and constrain the anodes of the other to GND.
4. Download your design.
5. Verify its functionality by trying all input combinations (0, 1, 2 ... 14, 15).
6. Demonstrate your work to the lab instructor.

### 6.5.2 Time Multiplexing the Four Seven-Segment Displays.

7. Use the macro you made in the steps 1 and 2.
8. Connect the four anodes to a two-to-four decoder (D2-4E) as shown in Figure 6.4.
9. Connect the two inputs of the decoder to a two-bit counter (CB2CE) as shown in Figure 6.4.
10. Connect the clock of the counter to the 1 Hz clock you made before.
11. Download your design. Note that the Hex value switch between the four seven-segment displays.
12. Increase the clock frequency and download the design again.
13. Repeat step 12 till the four seven-segment displays appear to be all ON all the time. (What is this frequency? calculate it and call it REF\_RATE.).

## 6.6 Post-Lab

- Document your design by providing a diagram of the Hex to seven-segment display decoder.
- If the counter outputs are designated Q1Q0, you are required to sketch the waveforms of Q0, Q1, and D0-D3 of the decoder for 4 clock cycles. Use Figure 6.5 for this purpose assuming that Q1Q0=00 after first clock pulse.
- Assuming the counter input clock to have a period **T**, from the timing diagram of Figure 6.5 determine the following:
  - The period for which each display digit is enabled each 4 clock cycles and the period for which it is OFF.
  - Given that each display digit should not be OFF for more than 1/30th of a second to retain a continuous image of all 4 digits, compute the minimum counter clock frequency (called REF\_RATE) that will achieve this.
- Determine (by calculation) which output of the clock divider (Refer to exp#5), when provided with the 25.175 MHz input clock, will satisfy the required minimum REF\_RATE.

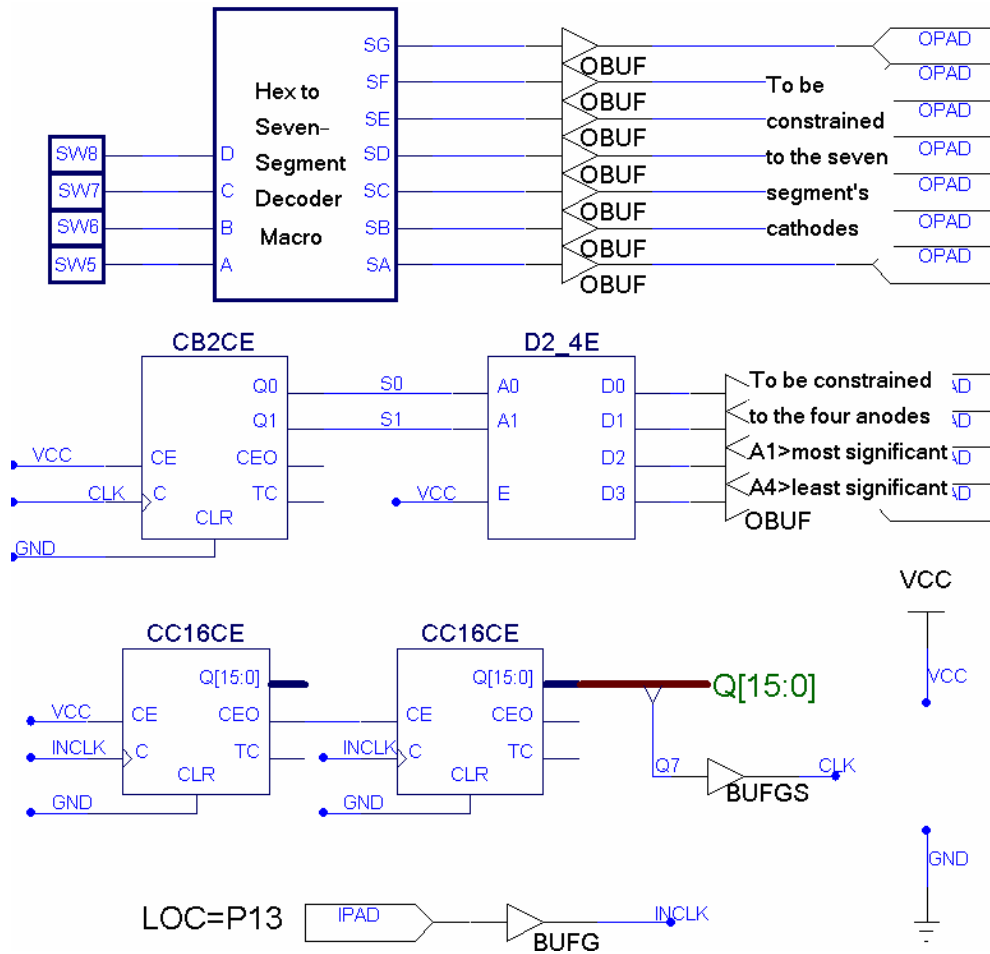


Figure 6.4: Time Multiplexing the Same Digit on Four 7-Sement Displays

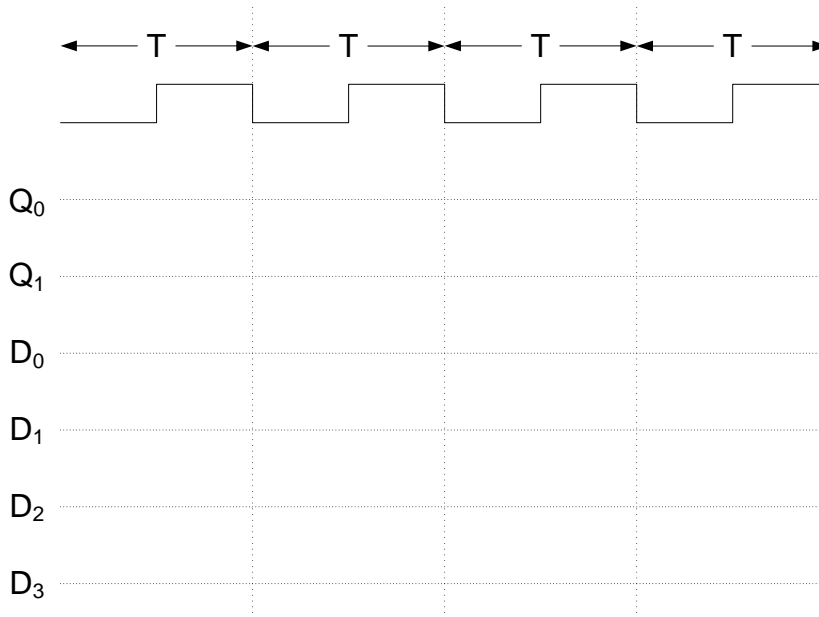


Figure 6.5: Timing Diagram.