Lab # 9
An Alarm Clock – Phase 3

Objectives

1. An application of MUXs.
2. Time-multiplexing of the four 7-segment displays to display time.

Overview

In the previous lab we used the 7-segment display to display a single digit. Power dissipation of the 7-segment display is usually fairly considerable. To save on power, typically 7-segment displays are not enabled continuously but are rather alternately enabled and disabled with such 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 7-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 7 inputs that control the illumination of the 7 segments are common to all the four digits. The 4 digits (7-seg displays) however, have *independent enable inputs* (En_0, En_1, En_2, En_3). Figure 9.1 shows this arrangement. Thus, the 4 digits receive the same 7 control inputs, however, only digits which are enabled will display the corresponding pattern.

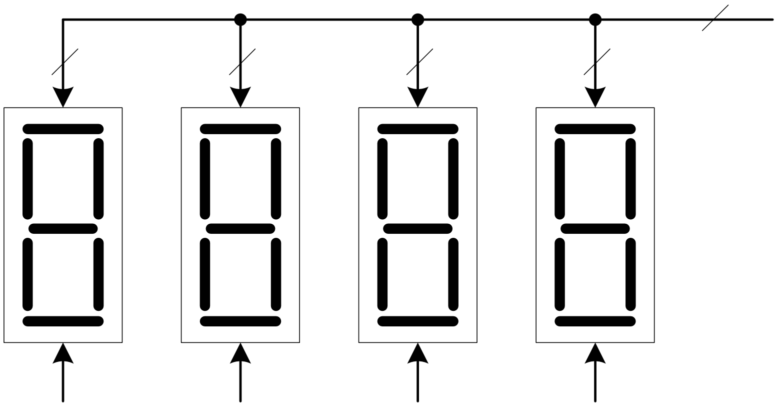

Figure 9.1

The 4 display digits will be used as follows:

- The 2 leftmost digits will display the hour H_1H_0 (00 – 23) where the H_1 digit is enabled by the En3 signal and the H_0 digit is enabled by En2 signal. 7
- The 2 rightmost digits will display the minutes M_1M_0 (00 – 59) where the M_1 digit is enabled by the En2 signal and the M_0 digit is enabled by En0 signal.

Design Specifications

In this lab, you are required to display hours and minutes digits on the 4 7-segments in a circular fashion. This implies that the activation signals for the segments (En0-En3) would be enabled in a circular manner. This ring type activation can be achieved with the use of a *2x4 decoder* with its input coming from a *2-bit binary counter*. The clock frequency of the counter (call it REF_RATE) should be high enough such that human eye will not observe the round-robin enabling scenario, i.e. the human eye should have the illusion that all 4 digits are continuously displayed without flicker!!!

The overall block diagram of the circuit is shown in Figure 9.2.

Illustrative Example

Consider the case of the current time being 09:34. To display this on the four display digits, the following steps are taken:

1. Let the count of the counter be 00, this causes En0 =1, while En1=En2=En3=0. At the same time the 00 count passes the rightmost minute's digit (4) to the input of the BCD-to-7 segment decoder. This causes the M_0 display digit to show the digit 4.
2. After one clock cycle, the count becomes 01. This cause En1 =1, while En0=En2=En3=0. At the same time, the 01 count passes the second minute's digit (3) to the input of the BCD-to-7 segment decoder. This causes the M_1 display digit to show the digit 3. En_3
3. After another clock cycle, the count becomes 10. This cause En2 =1, while En0=En2=En3=0. At the same time the 10 count passes the least significant

hour's digit (9) to the input of the BCD-to-7 segment decoder. This causes H_0 display digit to show the digit 9.

4. After another clock cycle, the count becomes 11. This cause En3 =1, while En0=En1=En2=0. At the same time the 11 count passes the most significant hour's digit (0) to the input of the BCD-to-7 segment decoder. This causes the M_1 display digit to show the digit 0.
5. The process repeats every 4 clock cycles causing each 7-segment display digit to be enabled for one out of the 4 clock periods.

Obviously, when **MODE**=0, the current time is displayed, and when MODE=1, the alarm time is displayed instead.

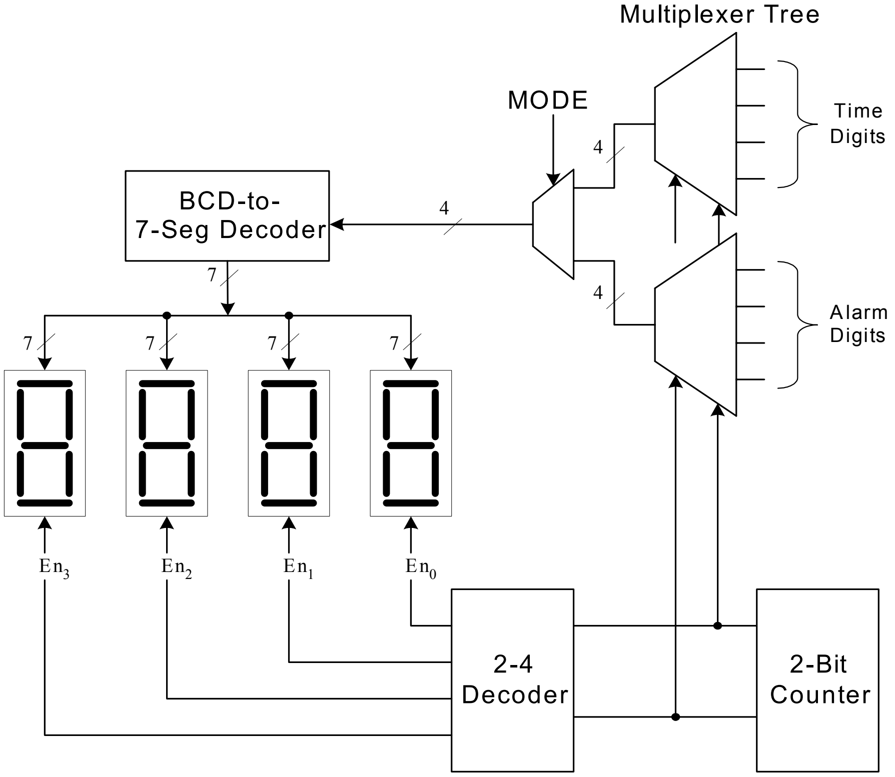

Figure 9.2

Pre-Lab

- In this lab, you will utilize the BCD-to-7-segment display decoder macro; so make sure you bring it with you.
- Schematic of Lab# 7 should be brought to this lab, as you are going to build upon it.
- Review the operation of decoders, and multiplexers.
- Design the detailed tree MUX required for the above diagram. Decide which library modules you need for this circuit.
- Make sure you understand the block diagram of Figure 9.2.

- If the counter outputs are designated C1C0, you are required to sketch the waveforms of C0, C1, and the En0 signals for 4 clock cycles. Use figure 9.3 for this purpose assuming that C1C0=00 after first clock pulse.
- Assuming the counter input clock to have a period **T**, from the timing diagram of Figure 9.3 determine the following:
  a) The period for which each display digit is enabled each 4 clock cycles and the period for which it is OFF.
  b) 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 (call it REF_RATE) that will achieve this.

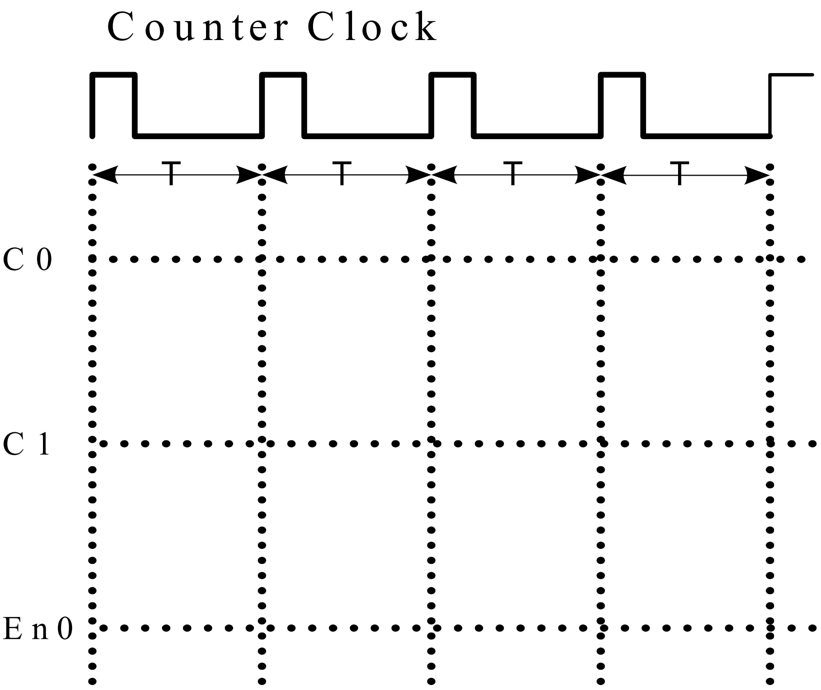

Figure 9.3: Timing Diagram

- Determine (by calculation) which output of the clock divider (Refer to exp#6), when provided with the 25.175 MHz input clock, will satisfy the required minimum REF_RATE.
- Read (and understand) the remainder of this document.

In-Lab

1. Open the schematic of lab # 7. Add the round-robin enabling logic of the four display digits. A block diagram of this logic is shown in Figure 9.4, where En0, En1, En2, and En3 are the "*__Enable__*" inputs of the four 7-segment display digits.

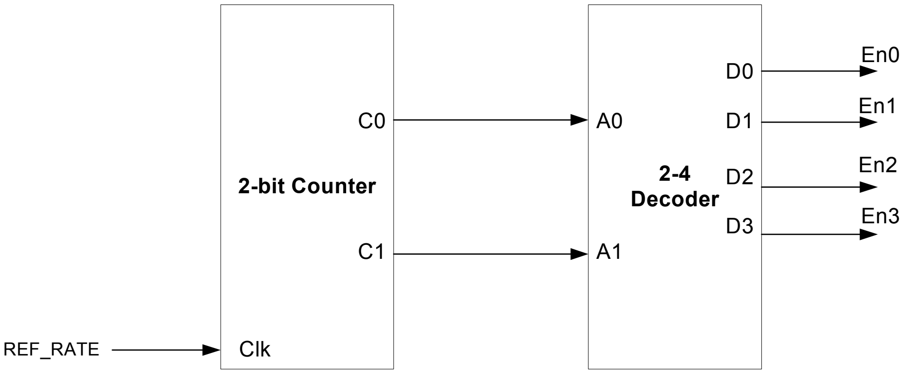

Figure 9.4: Round-robin enabling logic of the four 7-segment display digits.

Table 9.1: The input of the BCD-to-7-segment display decoder is selected through a multiplexer-tree using MODE, C0 and C1.

MODE	C1	C0	What should feed the BCD-to-7 segment display decoder	Comment
0	0	0	En0 is enabled ➔ Ones digit of the minutes (M_0).	**Current time is displayed** **(since mode=0)**
		1	En1 is enabled ➔ Tens digit of the minutes (M_1).	
	1	0	En2 is enabled ➔ Ones digit of the hours (H_0).	
		1	En3 is enabled ➔ Tens digit of the hours (H_1)	
1	0	0	En0 is enabled ➔ Ones digit of the minutes (M_0).	**Alarm time is displayed** **(since mode=1)**
		1	En1 is enabled ➔ Tens digit of the minutes (M_1).	
	1	0	En2 is enabled ➔ Ones digit of the hours (H_0).	
		1	En3 is enabled ➔ Tens digit of the hours (H_1)	

2. Depending on the values of C0, C1 (which are the outputs of the 2-bit counter in Figure 9.4) and the MODE input, one can determine what to pass as input to the BCD-to-7-segment-display decoder. Table 9.1 lists all possible values of the three variables and tells what action to take. For example, if C1C0=00 & MODE=0, En0 (which enables the rightmost digit M_0) is enabled, and the ones digit of the minutes counter of the current time should feed the BCD-to-7-segment-display decoder. A close look at Table 9.1 will make us conclude the following:

    a. If MODE=0, a digit from the current time is displayed. Otherwise, a digit from the alarm time is displayed.

    b. If C1=0, the digit to be displayed is from the minutes counter. Otherwise, it is from the hours counter.

    c. If C0=0, ones (least significant) digit is selected for display. Otherwise, the tens (most significant) digit is selected.

    This behavior can easily be achieved by utilizing Quadruple 2-1 MUXs. Before we show how this is possible, let's have the some notations. Let

    $\mathbf{H_{00}}$ be the ones digit of the hours counter in the current time,

H_{01} be the tens digit of the hours counter in the current time,

M_{00} be the ones digit of the minutes counter in the current time,

M_{01} be the tens digit of the minutes counter in the current time,

H_{10} be the ones digit of the hours counter in the alarm time,

H_{11} be the tens digit of the hours counter in the alarm time,

M_{10} be the ones digit of the minutes counter in the alarm time and,

M_{11} be the tens digit of the minutes counter in the alarm time.

Note that each one of the above symbols is 4 bits wide, and this is the reason behind using Quad MUXs. Figure 9.5 shows one possible configuration of how Quad MUXs may be utilized to build the multiplexer tree. Note that all lines in this figure are 4 bits wide. Exceptions are the selection lines, which are 1 bit lines. **The 2-1 Quad MUX is available in the Spartan library as X74_157**.

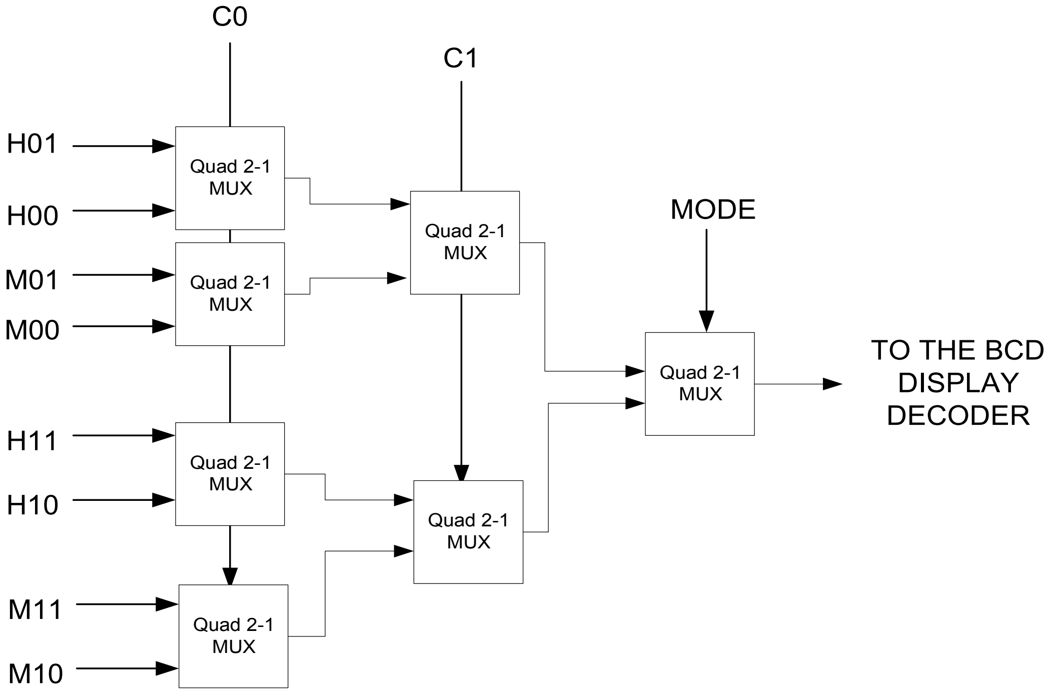

Figure 9.5: Multiplexing Logic

3. Implement & download your design.
4. Verify it is functioning and demonstrate to your instructor.