# Exp 6: BCD to 7-Segment Display Decoder

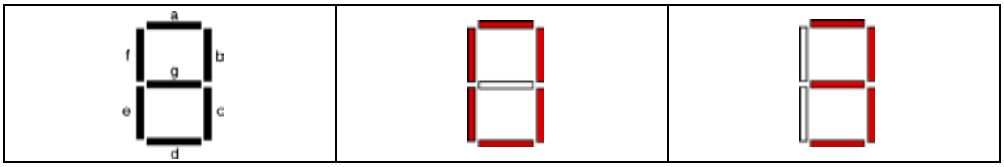
**Objectives:**

- To use Verilog to design and implement one of the basic digital logic circuits: the non-standard decoder
- To use a typical digital display device: the 7-segment display

**Background:** The 7-segment display is one of the most common display devices. These devices are generally used to display the full set of decimal numbers. These devices are also capable of displaying the full gamut of hex numbers although the case of the alpha characters is not consistent. The 7-segment displays used in CPE 133 are formed with seven LEDs (light emitting diodes). The LEDs are the small button-like colored devices that light up occasionally during your experimentation. The LEDs used in this lab activity have been elongated by using small stretching racks and configured in such a manner as to easily represent decimal numbers.

The LED is a two-terminal polarity sensitive device that turns-on when the voltage conditions across the two terminals are at the correct levels. The two LED terminals are referred to as the *anode* and the *cathode*. To make the LED emit light, you must provide a voltage rise from the anode to the cathode must be greater than 0.7 Volts. What this means to you in digital design-land is that you must either provide the LED with a "1" or a "0" depending on how the LED is wired into the circuit. To find out exactly how the LED is wired into the circuit, read the specification of the development board.

Representing individual decimal numbers is accomplished by turning on specific sets of the individual segments. Referencing the seven segments is done by assigning unique letters to each of the segments. Figure 13(a) shows the most common listing of these segments is. Using the 7-segment display to create the illusion of a '0' is shown in Figure 13(b) by lighting all the segments except segment 'g'. Figure 13(c) shows segments a, b, c, d, and g lit to simulate the number '3'. Making an LED emit light is a two-step process. You need to both turn-on the LED and also actuate the individual 7-segment display. Both of these actuation steps involve sending a logical '1' or '0' to the device. Consult the reference manual for the development board for the details (and then ask a bunch of questions).




(a) (b) (c)

**Figure 1: The ultra-amazing 7-segment display.**

**Assignment:** Design a BCD to 7-segment decoder using the awesome power of Verilog to model the circuit. Specifically, create the BCD to 7-segment decoder with the conditional operator (?) in Verilog (and without using an Always block). The 4-bit input of this circuit is a BCD number. The eight outputs are the various segments of the display (including the decimal point). Only use one 7-segment device in this experiment and make sure you turn off the unused displays. When the input BCD value exceeds the range of decimal digits, turn off the display.

| Input | Output |
|---|---|
| SW3,SW2,SW1,SW0 | CA,CB,CC,CD,CE,CF,CG,CDP |
| right-most four switches | right-most 7-segment display |

**Table 0.1: Input/Output specification for the BCD to 7-Segment decoder.**

**Questions:**

1. How many K-maps would you have needed if you implemented this design using equations for each of the segment outputs? How much longer would this design have taken you if you had to implement it using K-maps?

2. How many different letters of the English alphabet could be represented with a standard 7-segment display? Which letters could not be displayed? For this question, letters can be displayed in either lower or upper-case formats.

3. When representing the decimal digits, what is the least used segment?

4. Why would an engineer even care about the previous question?

# Exp 7: BCD to 7-Segment Decoder (Behavioral Model)

**Objectives:**

- To design and implement a non-standard decoder using a behavioral model (Always block)
- To see the ease at which extra features can be modeled using behavioral modeling

**Background:** The design of a BCD-to-7-segment decoder is not considered one of the more challenging feats in digital design. This experiment adds some features to the previously designed decoder in an attempt to make a boring experiment slightly more exciting.

The added features are not that amazing but there is a point. The general rule with HDLs is that if you can design something using a dataflow (RTL) model, they you should. If you have to put too much extra effort into designing a circuit using a dataflow (RTL) model, strongly consider switching to a behavioral model. So for this
experiment, you're basically going to re-implement the decoder in a previous experiment and include some added features.

**Assignment:** Design a BCD to 7-segment decoder using Verilog behavioral modeling (Always block). One of the inputs to this circuit is the BCD number. The eight outputs are the various segments of the display (including the decimal point). For this design, modify your decoder to include the alpha characters A-F. You design should work as follows.

- If BTN2 is pressed, no 7-segment display should be on.
- BTN1 & BTN0 determine which 7-segment display is actuated according to Table 0.1.

| BTN1-BTN0 | 7-segment display |
|---|---|
| "00" | Right-most |
| "01" | Second from right |
| "10" | Third from right |
| "11" | Left-most |

**Table 0.1: Individual 7-segment display output specification.**

**Questions:**

1. Think for a few minutes about how you would have implemented this project using a dataflow (RTL) model. Briefly describe whether you think the easier approach would be: dataflow (RTL) or behavioral.

2. Provide your well-supported opinion regarding the following statement: "Verilog behavioral models are much more powerful than Verilog dataflow (RTL) models."

# Exp 8: BCD to 7-Segment Decoder with Modifications

**Objectives:**

- To design and implement a non-standard decoder using a behavioral model
- To see the ease at which extra features can be modeled using behavioral modeling

**Background:** This experiment is a repeat of the previous experiment with a minor modification. The modification is that if BTN3 is pressed, the output displays the characters A-F when appropriate based on the inputs. If BTN3 is not pressed, the display blanks when $1010_2 - 1111_2$ is input. The entire design description of provided below; be sure to note that it is almost exactly the same as the previous experiments.

**Assignment:** Design a BCD to 7-segment decoder using the Verilog behavioral modeling. One of the inputs to this circuit is the BCD number. The eight outputs are the various segments of the display (including the decimal point). For this design, modify you decoder to include the alpha characters A-F. Your design should work as follows.

- If BTN3 is pressed, the lit 7-segment display should include hex letters a-f in the output.
- If BTN2 is pressed, no 7-segment display should be on.
- BTN1 & BTN0 determine which 7-segment display is lit according to Table 0.1.

| BTN1-BTN0 | 7-segment display |
|---|---|
| "00" | Right-most |
| "01" | Second from right |
| "10" | Third from right |
| "11" | Left-most |

**Table 0.1: Individual 7-segment display output specification.**

**Questions:**

1. There are many different ways to complete this design. My feeling with this type of design is to do it "the easiest way I can think of" and let the Verilog synthesizer sort out the details. In your own terse words, what does this statement mean? Also in your own terse words, list two ways you can tell whether this approach is "not too inefficient" or not.