Creeping Line Project

Task:

Implement a creeping line, running on four seven-segment LED displays. Following is the list of steps for the creeping line design:

- implement a decoder that converts a binary value that can be set with four switches to a hexadecimal value that is shown on one seven-segment LED display;
- increase the number of seven-segment LED displays to four, with each display being connected to a separate set of four switches (so each seven-segment LED display can be controlled independently);
- substitute switches for a 32-bit circular shift register (that shifts 4 bits at a time). Initialize the register with a value corresponding to e.g. student ID code. Shifting speed is not strictly defined, although the line should be readable. Optionally, the decoder can be changed to display other information besides hexadecimal numbers. If the modified decoder can output more than 16 characters, the shift register size should be changed accordingly.

Simulate and implement each design step on FPGA development board. Prepare and submit a written report and project achives for the lab.

Controlling the Seven-Segment LED Displays

Basys 3 FPGA board has four seven-segment LED displays. Each display consists of eight individual LED segments (Figure 1). In order to minimize the usage of FPGA's general purpose pins, all eight displays share eight common control inputs that light individual segments of the display, but each display has a separate enable input. When enable input of the seven-segment LED display is not active, the whole display is switched OFF. Note, that both control and enable inputs of the seven-segment LED displays are active LOW (e.g. the segment is turned OFF when the control input is driven HIGH).

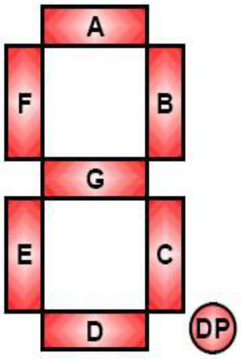

Figure 1: Seven-segment LED Display

For the first step it is required to control only one seven-segment LED display. The design structure should be similar to the one shown in Figure 2. The decoder drives segment control inputs and is connected directly to switches. The enable inputs are tied HIGH (turned OFF) except for one that is tied LOW (e.g. AN0 as shown in Figure 2). In this configuration the selected seven-segment LED display should show the HEX value corresponding to the binary value that is set with switches.

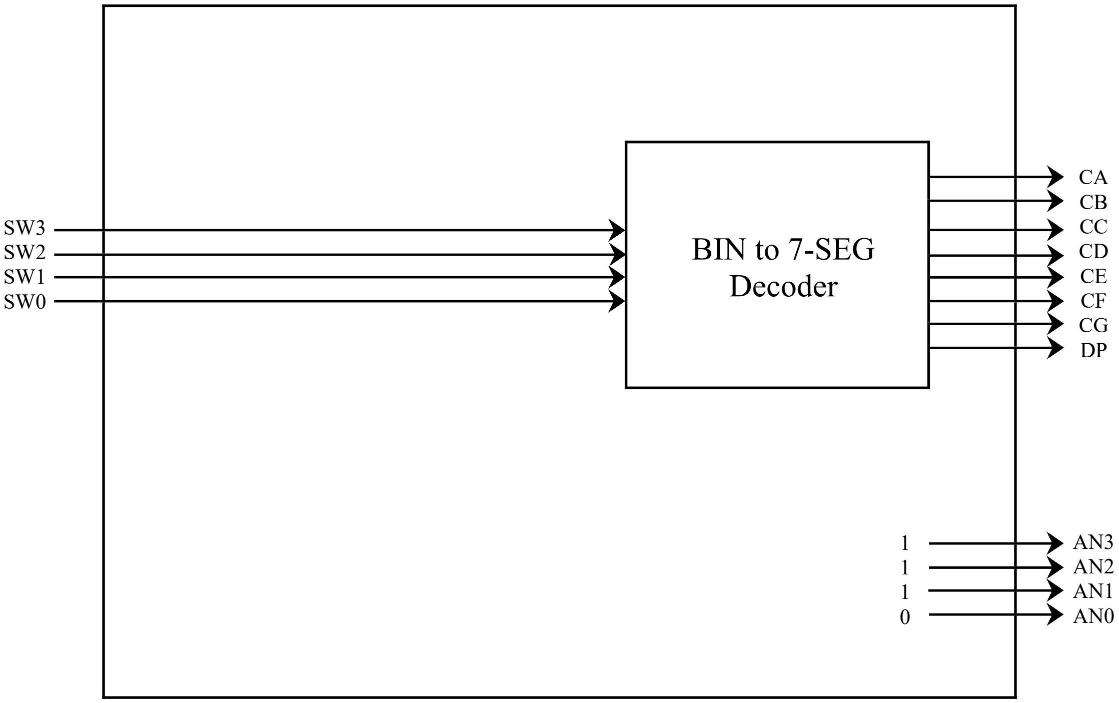

Figure 2: Design Structure for the First Step

To output data on several seven-segment LED displays, the control signals must be time multiplexed (Figure 3). Present the value to be displayed using the segment control inputs and select the corresponding seven-segment LED display by driving the associated enable input LOW (note, that other enable inputs are driven HIGH). After a certain period of time do the same for the next seven-segment LED display and so on. Through persistence of vision, the human brain perceives that all four characters appeared simultaneously. However, the switching frequency should not be too high, as it results in a constant glowing of all segments. Similarly, the switching frequency should not be too slow, as it leads to blinking of the displayed data. Further information can be found in section 8.1 (page 15) of the Basys 3 FPGA Board Reference Manual.

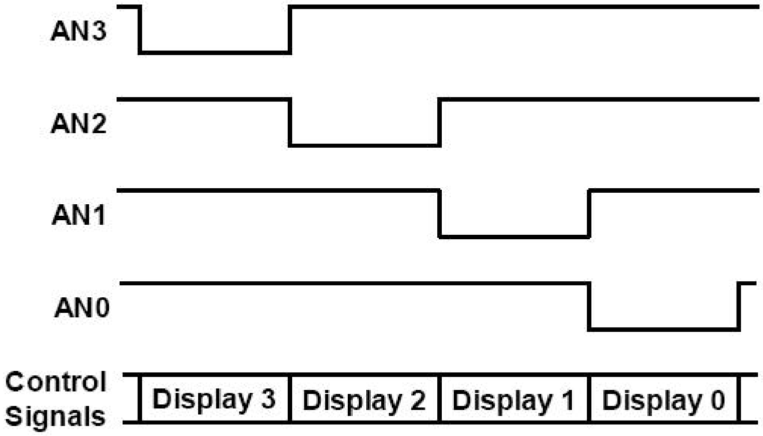

Figure 3: Time Multiplexed Data Output

For the second step it is required to control four seven-segment LED displays. The design structure should be similar to the one shown in Figure 4. The segment decoder is now being controlled with sixteen switches that are connected through a multiplexer. Note that the connectivity of switches is now shown with 4-bit buses. At any given time only one set of switches is controlling the segment decoder. The selection is done using a 2-bit counter. The counter value is also decoded to generate an appropriate value for enable inputs, just like it is shown in Figure 3. Note that the 2-bit counter is using a delay counter, since the display switching frequency should be far smaller than the frequency of the system clock.

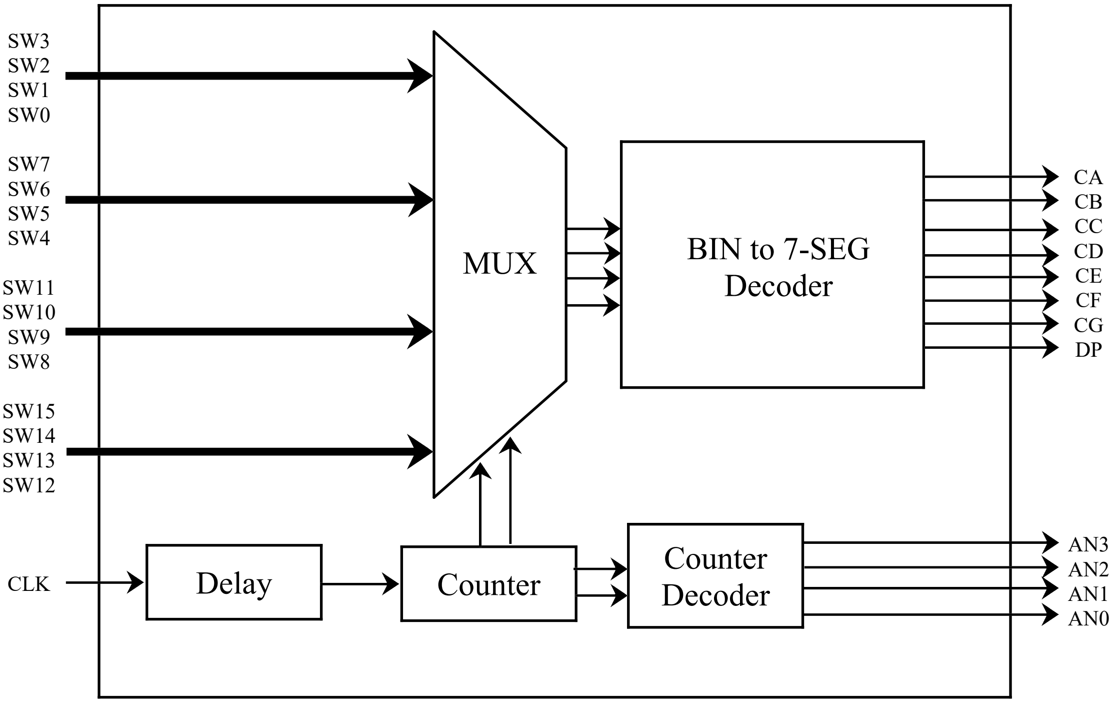

Figure 4: Design Structure for the Second Step

For the last step it is required to control all four seven-segment LED displays using 32-bit circular shift register. The design structure should be similar to the one shown in Figure 5.

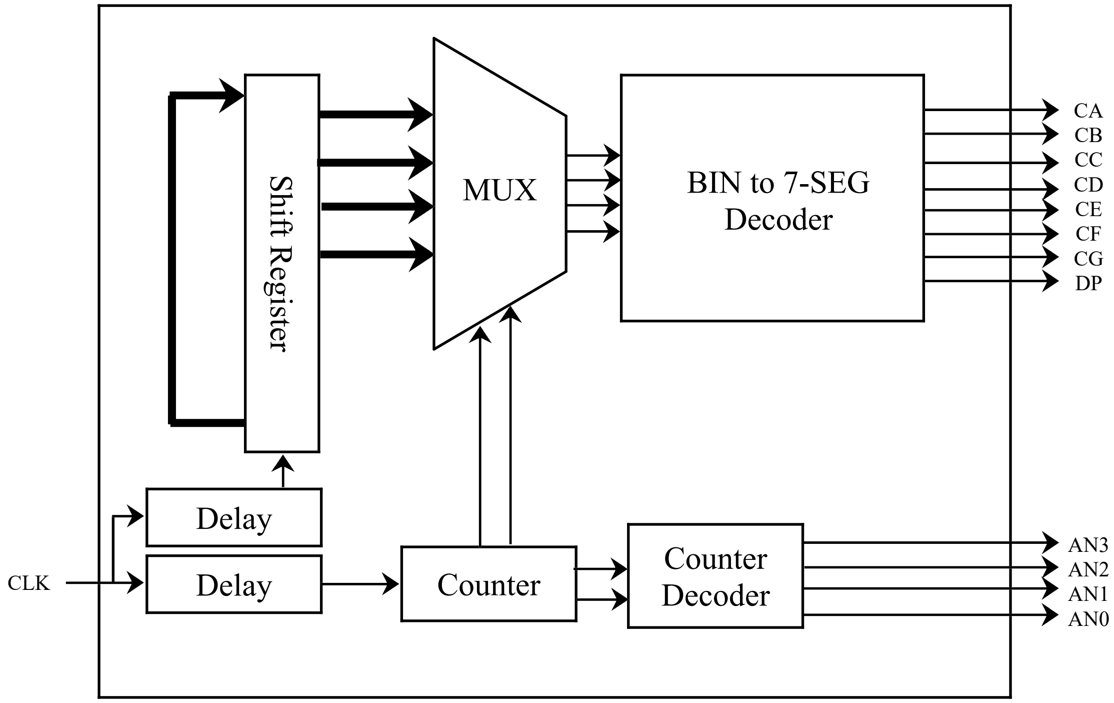

Figure 5: Design Structure for the Third Step

Note that in the final design the switches are no longer used, they are substituted with a 32-bit circular shift register (that shifts 4 bits at a time). The shift register should be initialized with a value that is going to be shown on four seven-segment LED displays (e.g. student ID code). Since the number of displays is four, only half of the characters that are stored in the shift register will be shown simultaneously. However, since the creeping line is constantly moving, the whole message will eventually be displayed. Note that shift register should have a separate delay counter since shifting frequency differs from refresh frequency.