LAB 3: Seven Segment Display on the FPGA using Chisel

This is a simple Chisel code to implement the interfacing with the provided FPGA board with front end written in Chisel. You need to use the switch buttons as inputs to provide decimal numbers from 0 to 9 and see the number being printed on the 7 segment display.

A simple chisel code is provided that maps the switches on the Nexys 4 ddr board to different decimal numbers from 0 to 9 and associates the 7-segment display with each input i.e. when the appropriate switch corresponding to 7 is pressed, 7 is displayed on the display. The constraint file named as nexys4_SevenSegDemo.xdc for this purpose is also provided. Use the appropriate **sbt** commands to get the Verilog file and then port it onto the provided FPGA board

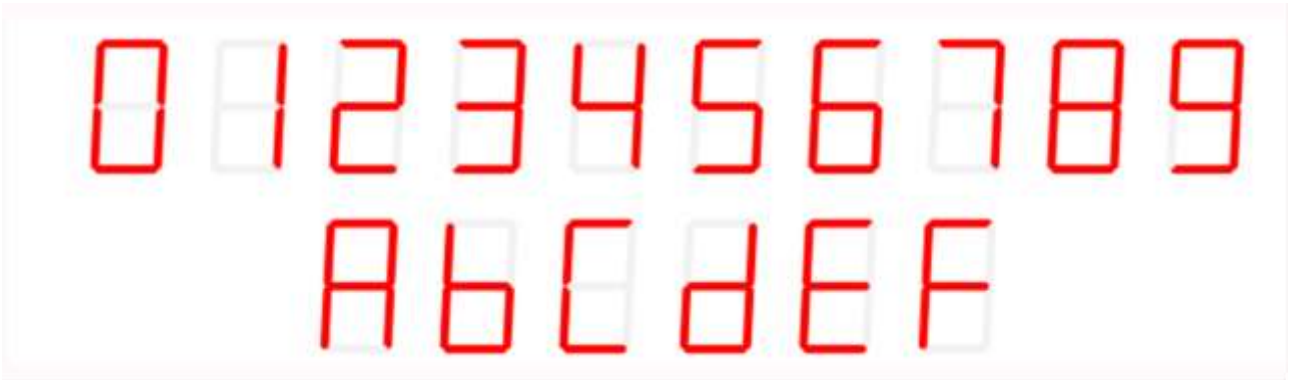
Tasks to be done:

SevenSegDecoder and **CountSevenSeg** chisel files are provided along with their xdc files. Your tasks are to complete the decoder code i.e. it will take the 4-bit binary input via switches and print the HEX equivalent on the 7-segment decoder; and also to complete the counter code i.e. implement a HEX up counter and print the values from 0 to F continuously on the display. The second part will use the same 7-segment decoder that is implemented in the first task. The first task can also be tested on the provided **SevenSegDemoTest** chisel testbench before it is ported onto the FPGA.

Some Background knowledge required for the lab:

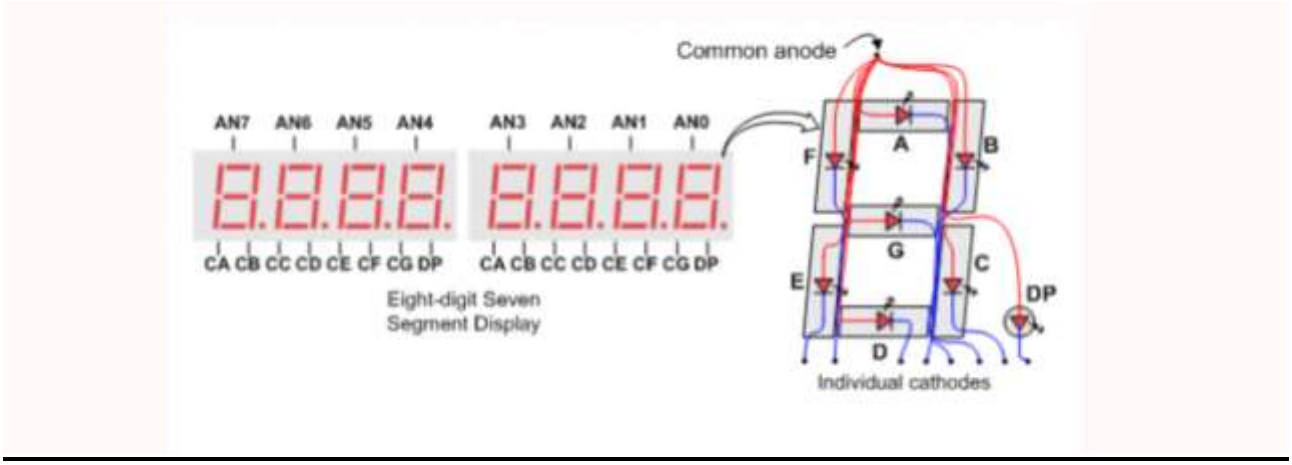
Seven Segment Display Interfacing with Artix7-DDR (PIN Assignment)

Let's refresh our memory quickly about the seven-segment LED display. Each digit is comprised of seven LEDs connected such that their anodes are all tied together (“common anode), and each LEDs cathode is independently controllable. So at a very basic level, knowing nothing else about their control circuit, we can expect to light up segments by turning the common anode high, and pulling low cathode pins that correspond to the LED segments we want to light up (and pulling high those that we don't by default).



There are eight digits of seven-segment display present. Each digit has a independently controllable anode pin. Each anode is driven on the low side of a PNP transistor, the base of which is attached directly to one of eight anode pins on the FPGA, labeled an[7:0]. In order to activate an anode (allowing current to flow), we'd therefore need to drive the corresponding anode an pin **low**. By

default pins from the output of an FPGA are low, and this is why in Lab 1, all the seven-segment digits were activated when we didn't even have to think about them.



Further complicating the situation, however is how **all** the seven segment displays (there are eight of them) are tied together. If we look at the documentation from Digilent it tells us that **all** of the "A" LEDs on all eight seven-segment displays are connected together, as are all the "B" pins and so on. So while each individual seven-segment digit is common anode, the displays as a whole have a common cathode scheme going on. Operation then becomes based on the following rules:

- If you want to drive a particular seven-segment digit, turn the anode of the LED high by pulling the base of the pnp BJT low (by setting the corresponding an pin low) This turns on the PNP transistor driving it, allowing current to flow.
- If you want to drive a particular seven-segment's segment, set its corresponding cathode pin **LOW** since this lets the current leaving the cathode go to ground.

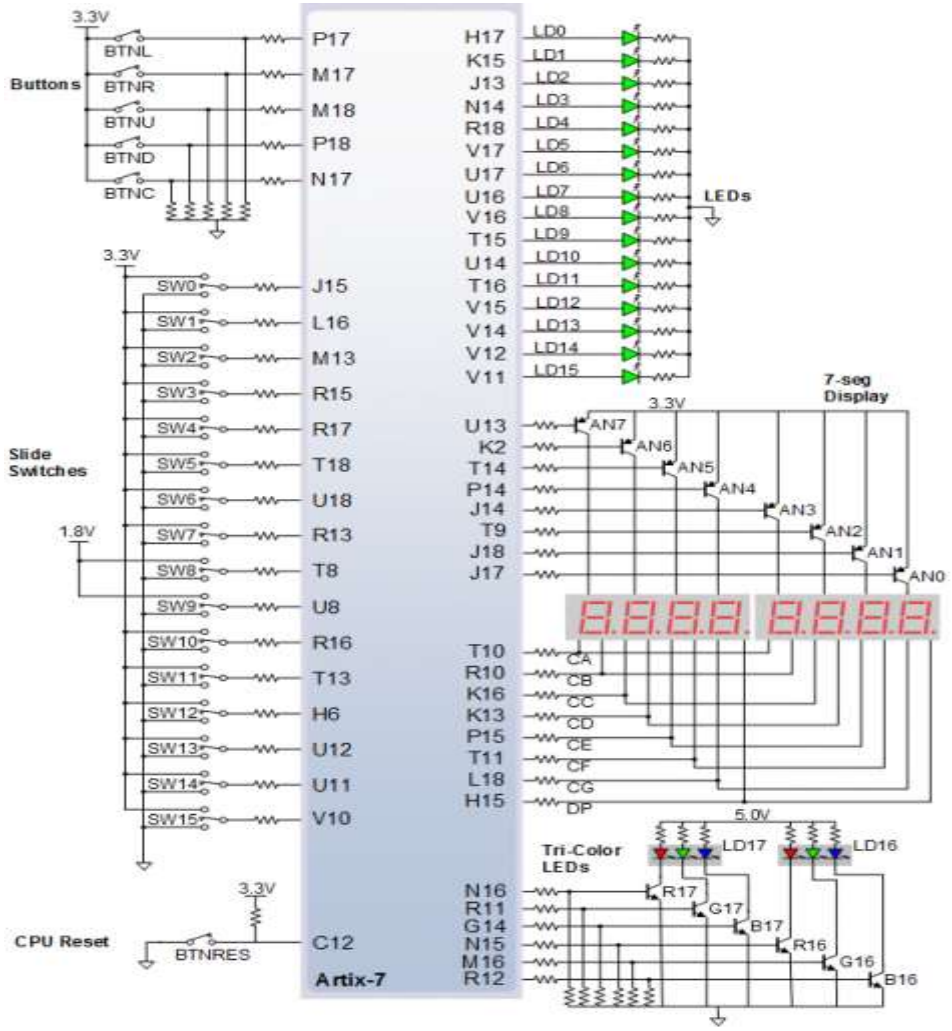


Figure 16. General Purpose I/O devices on the Nexys4 DDR.

Tasks:

1) To design a 2-to-4 decoder in Chisel using the swicthes as inputs and the decoded equivalent displayed on the 7 segment display. XDC file for the same is provided.

2) To use the 7 segment display table used in the above task and to create a HEX up counter with the counter output displayed on the 7 segment display. XDC file for the same is provided.