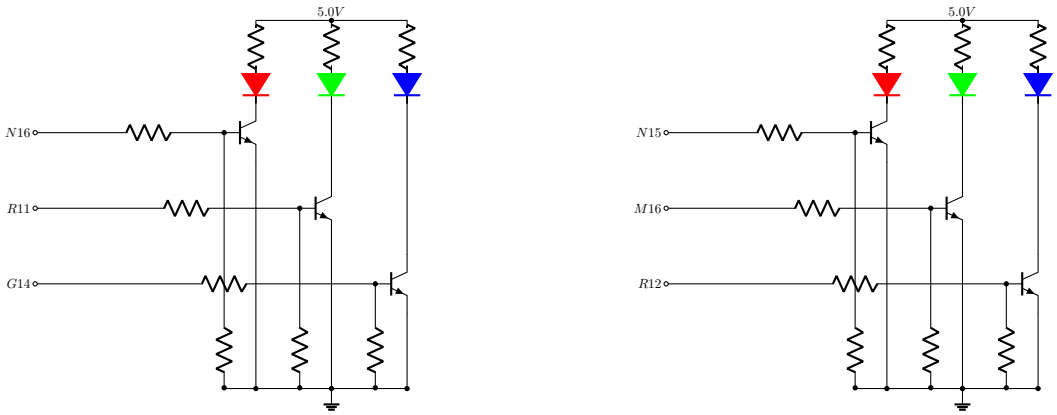
Experiment 4

Combinational Circuit Design using K-Maps

Overview of Tri-Color LEDs

For this lab, we will use the tri-color LEDs. Our board contains two tricolour LEDs. Each LED contains three smaller LEDs, i.e, one red, one green and one blue LED. The pin configurations of all the required components can be seen from the Nexys A7 reference manual in the **Basic I/O**.



(a) First RGB LED. (b) Second RGB LED.

Fig. 4.1: Configuration of tri-leds in the Nexys A7.

The Fig. 4.1 gives us the circuit diagram along with the pin configuration of the two LEDs on the FPGA board. Each LED of the RGB can be turned on by controlling the cathode of the respective LED. In order to turn on a LED, a high signal is required at the input which will be inverted using the transistor to turn on the LED. Hence, the transistor is going to act as a switch in the circuit.

Lab Task

For this lab, you are required to create a module containing two 2-bits inputs a and b such that we observe the following output on the tri-color LED.

$$F = \begin{cases} a > b & Cyan \\ a = b & Purple \\ a < b & Yellow \end{cases}$$

For this purpose, first develop the truth table according to the requirement and then get the minimized expression for the output using K-maps. Next, implement the design on Vivado. Also simulate your design on QuestaSim.

Deliverables

1. A report containing the following items:

(a) Truth table of the circuit.

(b) K-maps used to minimize the logic. Write down all prime and essential prime implicants of the function.

(c) Circuit diagram of your reduced design inferred from K-Maps.

(d) Circuit diagram inferred by the Xilinx Vivado. You can see the inferred circuit in "Schematic" under the process "Elaborated Design". Is it similar to the one you designed using K-maps?

(e) Maximum combinational delay in Synthesis: Read the report of your circuit and describe which path has the maximum combinational delay?

(f) Resource Utilization: Read the synthesis report and identify how many resources in the FPGA such as lookup tables (LUTs), input/output (IOs), etc., has been utilized.

2. System Verilog code for the circuit.

3. Synthesis of the circuit for the starter kit available in the lab. Tie inputs to the switches and output to the RGB LED available on the board.

4. Simulation of your design. Show it to the instructor.

Overview of Seven Segment Display

The Nexys-A7 board contains two four-digit common anode seven-segment LED displays, configured to operate like a single eight-digit display. All the nodes of a single seven-segment displays are tied together to form a common anode for a single seven-segment. All the eight seven-segments have their own common anode pin labeled from AN0 to AN7 as shown in Fig. 4.2. These common anode pins are active low, which means that in order to turn on all the seven-segments at the same time all the common anode pins must be set to zero. Similarly, all the cathodes of the seven-segment displays are tied together to form segment A to G.

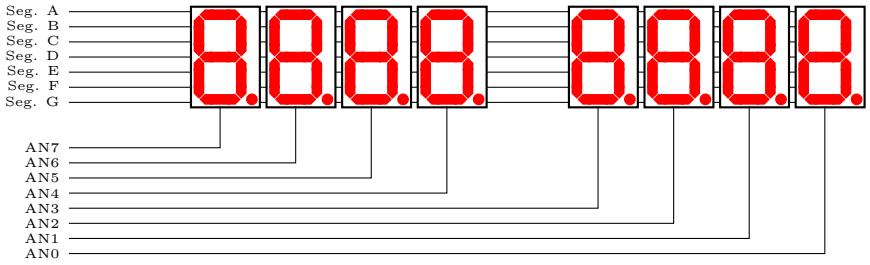


Fig. 4.2: Connections of the Seven-Segment Displays.

The anodes and cathodes of these seven-segment displays are connected to the labeled pin through the transistor circuit. The transistor circuit acts as an inverter, due to which these seven segments are active-low. In order to display a digit or a character, different combinations are applied to the segment pins. 4.1.

Character	Seg. A	Seg. B	Seg. C	Seg. D	Seg. E	Seg. F	Seg. G
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
B	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
D	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

Table 4.1: Combinations provided to Cathodes to display characters on Seven-Segment Display.

The common cathodes of the seven-segment display decide which seven-segment would be turned on for a specific task. For Example, if we want to display a character on the second seven-segment, the pin AN1 should be set to 0.

The FPGA pins assignment required for the seven segment display can be observed in table 4.2 for segment control. Similarly, for anode control we have the table 4.3.

Segment	FPGA Pin
A	T10
B	R10
C	K16
D	K13
E	P15
F	T11
G	L18
DP	H15

Table 4.2: FPGA pins Segment Control.

Anode Control	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
FPGA Pin	U13	K2	T14	P14	J14	T9	J18	J17

Table 4.3: FPGA pins Anode Control.

Task

You are required to build a circuit to display different characters (0 to F) on one of the eight seven-segment displays. The circuit has the modular diagram in Fig. 4.3.

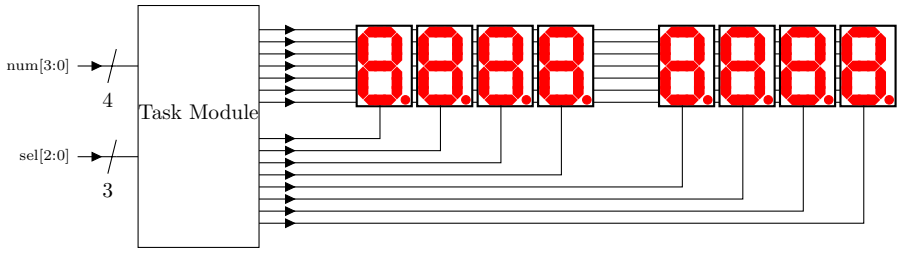


Fig. 4.3: Lab Task Modular Diagram.

The user should be able to display the character on any of the 7-segment displays as follows:

1. The user puts the binary equivalent of a hexadecimal between 0 and F on 4-bit num[3:0] bus.
2. The user selects which of the eight 7-segment displays by putting a binary equivalent of a number between 0 to 7 on sel[2:0] input. The eight seven segments are numbered from 0 to 7, the left-most segment being seven.

Fig. 4.4 explains the operations in the circuit in more detail. For example, when num bus has decimal 5 and the sel bus has 2 on it, the third display should show the number 5.

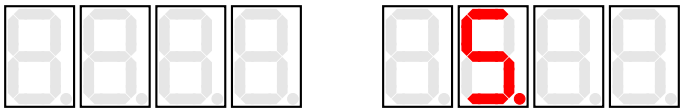


Fig. 4.4: Sample output for the Lab Task.

Deliverables

1. A report containing the following items:
   (a) Truth table of the circuit.
   (b) K-maps used to minimize the logic.
   (c) Circuit diagram of your reduced design inferred from K-Maps.
   (d) Circuit diagram inferred by the Xilinx Vivado. You can see the inferred circuit in "Schematic" under the process "Elaborated Design". Is it similar to the one you designed using K-maps?
   (e) Maximum combinational delay in Synthesis: Read the synthesis report of your circuit and describe which path has the maximum combinational delay?
   (f) Maximum combinational delay in Implementation: Read the post implementation static timing report and identify the path with the maximum combinational delay? Is the path same as the one in the synthesis report?
   (g) Resource Utilization Summary: Read the synthesis report and identify how many resources in the FPGA such as lookup tables (LUTs), input/output (IOs), etc., has been utilized.
2. System Verilog code for the circuit.
3. Simulation of your designed module on QuestaSim. Show the simulation to your instructor.
4. Synthesis of the circuit on NexysA7 Board available in the lab. Tie inputs to the switches and outputs to the seven-segment displays available on the board.

Collaboration between students is encouraged, but blind code sharing/copying is not allowed. If you are unable to explain anything in your code, it will be assumed you have copied it. So make sure you know everything you have written in your code. We are least concerned about how you have learnt something as long as you have learnt it well.

Acknowledgments

The manual has been written by Dr. Ubaid Ullah Fayyaz and Mr. Ali Imran.