REPORT FOR COL215 HARDWARE ASSIGNMENT-1

Objective:

To design a combinational circuit that takes a hexadecimal digit in its 4-bit binary form as input from the slider switch and produces a 7-bit output for the cathode of the 7-segment display of Basys3 FPGA board. Extending the design to create a timing circuit that takes four such digits (16-bit input) and drives all 4-displays for displaying the 4-digits together.

Procedure:

The FPGA board has four 7-segment displays each of which has a separate anode. The 7 LEDs of each display connect to 7 common cathodes. So, we cannot provide different digits to each display simultaneously. Thus, we make use of the on-board clock to make them turn on sequentially. This is done in the order of milliseconds making it appear as if they are simultaneously on to the eye.

Step-1:

We take a 16-bit input corresponding to the 4 hexadecimal digits as a vector. The on-board clock has a 10ns. In other words, there is a rising edge after every 10ns. We maintain one counter to keep track of the number of such cycles of 10ns. Further, we maintain another counter to keep track of the anode we want to keep ACTIVE ('0'). When the first counter hits 400000 (which represents 4 ms), we reset it to zero and change the second counter which indicates a display shift. The 2^{nd} counter takes values from 0 to 3 cyclically.

Step-2:

It is this 2^{nd} counter that decides the ACTIVE anode and the 4 bits corresponding to it using two variables 's0' and 's1'. These variables are decided by the value of the 2^{nd} counter, and we have used a VHDL process to carry out the same. We then apply a MUX-like logic on the i^{th} bits of each digit one-by-one to get the needed 4-bits.

Step-3:

The below table shows the 16 hexadecimal digits mapped to the corresponding LEDs. The cathode associated with the LED must be zero for it to be on.

Digit	LED-A	LED-B	LED-C	LED-D	LED-E	LED-F	LED-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

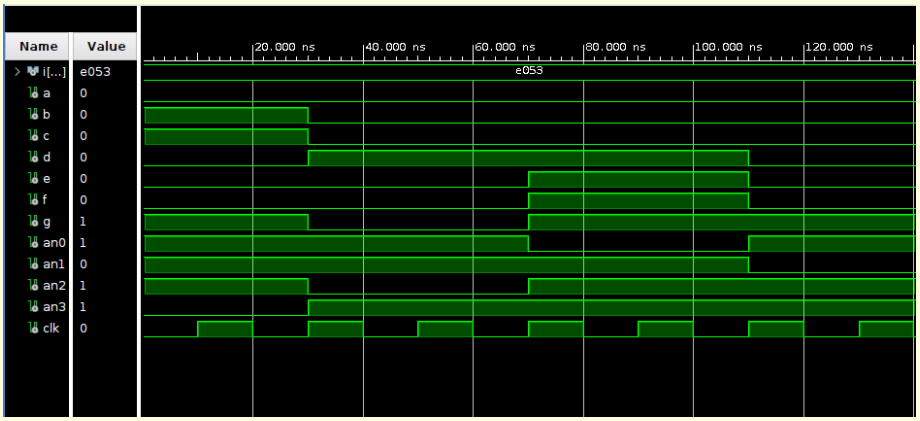
For a 4-bit input (digit), the table gives us the mapping for 7 functions which give us the outputs A,B.C,D,E,F,G corresponding to the 7 LEDs which can be minimized by the Kmap method to get the reduced function with least number of literals.

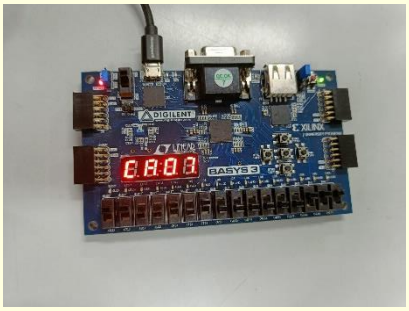
Lab Work:

Simulation Snapshots with Explanation:

1.

The input is 1100101000000111 (CA07). We can clearly observe in the simulation that the anodes (an0 to an3) are changing periodically with only one anode ACTIVE ('0') at a time. Note that we have changed the frequency of the clock for the simulation.

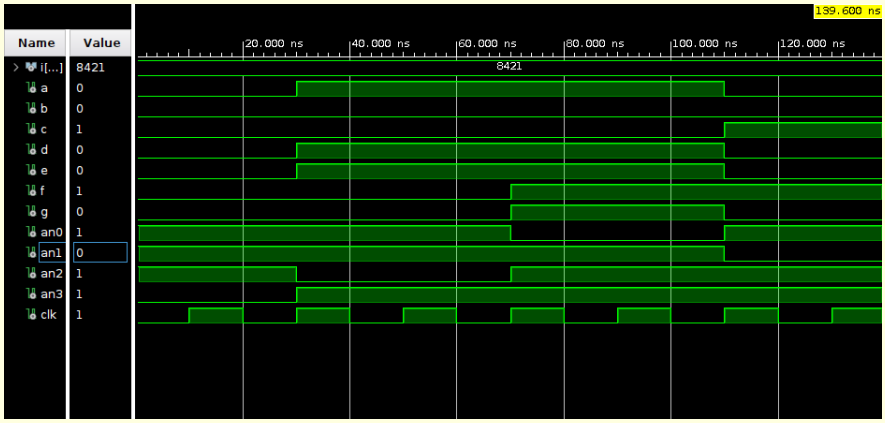


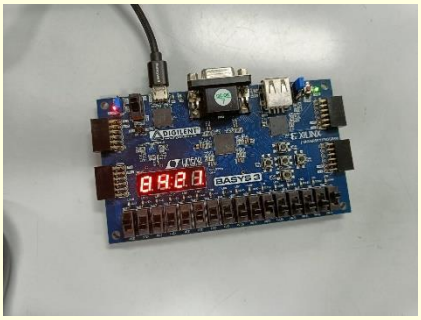


(Note: e053 (1110000001010011) is nothing but the reversed bit order notation of CA07. It is shown because we used a vector to store the number CA07.)

2.

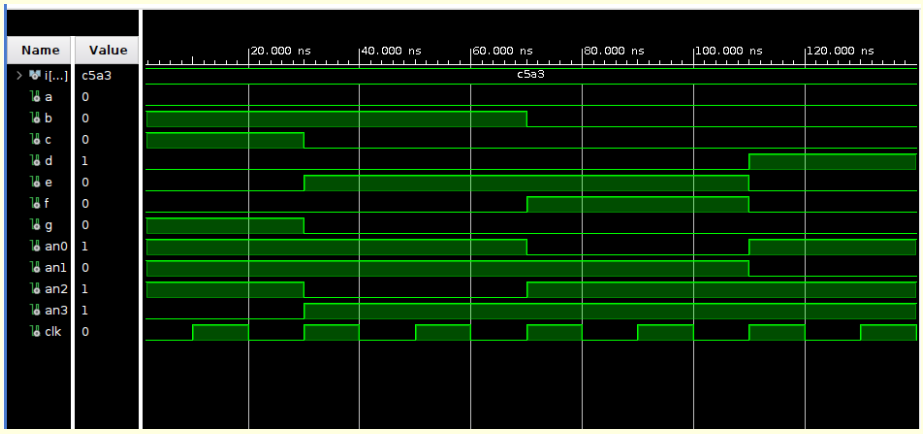
The input is 1000010000100001 (8421).

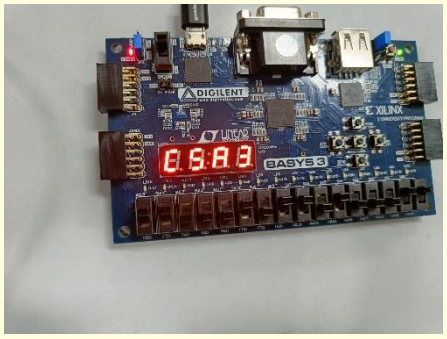




(Note: 8421 is symmetric in bits. So, the same number appears in the simulation screen.

3.
The input is 1100010110100011 (C5A3).





(Note: C5A3 is symmetric in bits. So, the same number appears in the simulation screen.)

Synthesis Report:

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	18	0	0	20800	0.09
LUT as Logic	18	0	0	20800	0.09
LUT as Memory	0	0	0	9600	0.00
Slice Registers	21	0	0	41600	0.05
Register as Flip Flop	21	0	0	41600	0.05
Register as Latch	0	0	0	41600	0.00
F7 Muxes	0	0	0	16300	0.00
F8 Muxes	0	0	0	8150	0.00

Site Type	Used	Fixed	Prohibited	Available	Util%
Block RAM Tile	0	0	0	50	0.00
RAMB36/FIFO*	0	0	0	50	0.00
RAMB18	0	0	0	100	0.00

Site Type	Used	Fixed	Prohibited	Available	Util%
DSPs	0	0	0	90	0.00

We can clearly see the number of used LUTs, BRAMs (Block RAM), DSPs and Flip-Flops. This is the synthesis utilization report. The number of LUTs further decreased to 17 post-implementation.