Department of Design Engineering and Mathematics

Middlesex University

Digital Dice

Inas El-Aroussi

PDE2431 Analogue and Digital Electronics

BEng Electronics Engineering

January 2021

1. Introduction

The aim was to design the digital dice using the engineering design skills and knowledge acquired in the previous blocks and in other modules. Here we are going to replicate the way some dice is rolled and rest at random face on top in digital way with LEDs light a random number. With this it can get easy to play games with dice as you can not cheat with this dice and it want to get lost while playing. For this challenge we are going to use NI Multisim, digilent Nexys 4(PLD), LEDs, wire, Power supply, logic gates, flipflops, clock, integrated circuits(74LS48), 7-segment display and digital constants.

2. Theoretical analyse

2.1. Dice pattern truth table

On having a close look, we got to know that all face on dice can be represented with basic 4 unique patterns. The 4 of that basic can be seen in the image 2

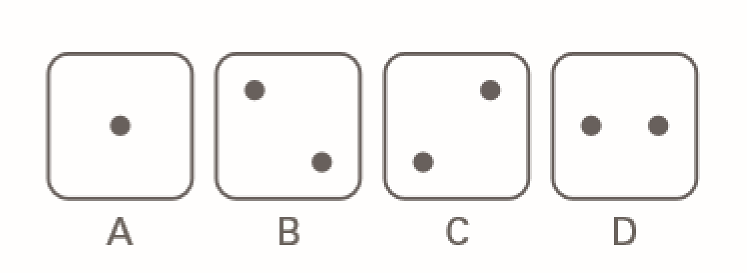


Figure 1 Basic pattern

So, with the help of this I created the following table for combination of basic pattern for all the faces on dice.

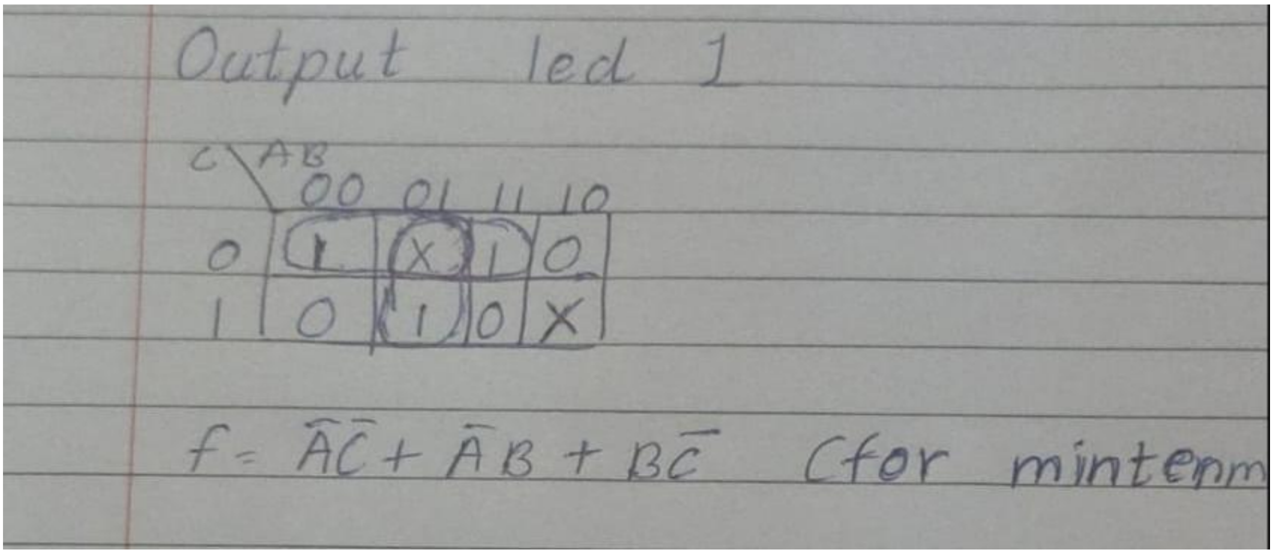
Die Face	A	B	C	D
1	1	0	0	0

2	0	1	0	0
3	1	1	0	0
4	0	1	1	0
5	1	1	1	0
6	0	1	1	1

On looking at output of counter it gets easy to consider that as input for the logic circuit and then desire the combination of basic pattern for all the faces on dice.

Dice face	Input A (Q1)	Input B (Q2)	Input C (Q3)	Output A	Output B	Output C	Output D
1	0	0	0	1	0	0	0
2	1	0	0	0	1	0	0
3	1	1	0	1	1	0	0
4	1	1	1	0	1	1	0
5	0	1	1	1	1	1	0
6	0	0	1	0	1	1	1

On analysing the table and there is a unique pattern for input and output. We can say that the Led 1 or output A needs to be needs to be on at alternate numbers. And from the table we can make a K-map and find the equation. As shown in figure 2. Similarly, we can find the equations for output B (led2 and 7), output C (led3 and 6) and output D (led 4 and 5) As shown in figure 3,4 and 5. From the equations in the figure we can determine the



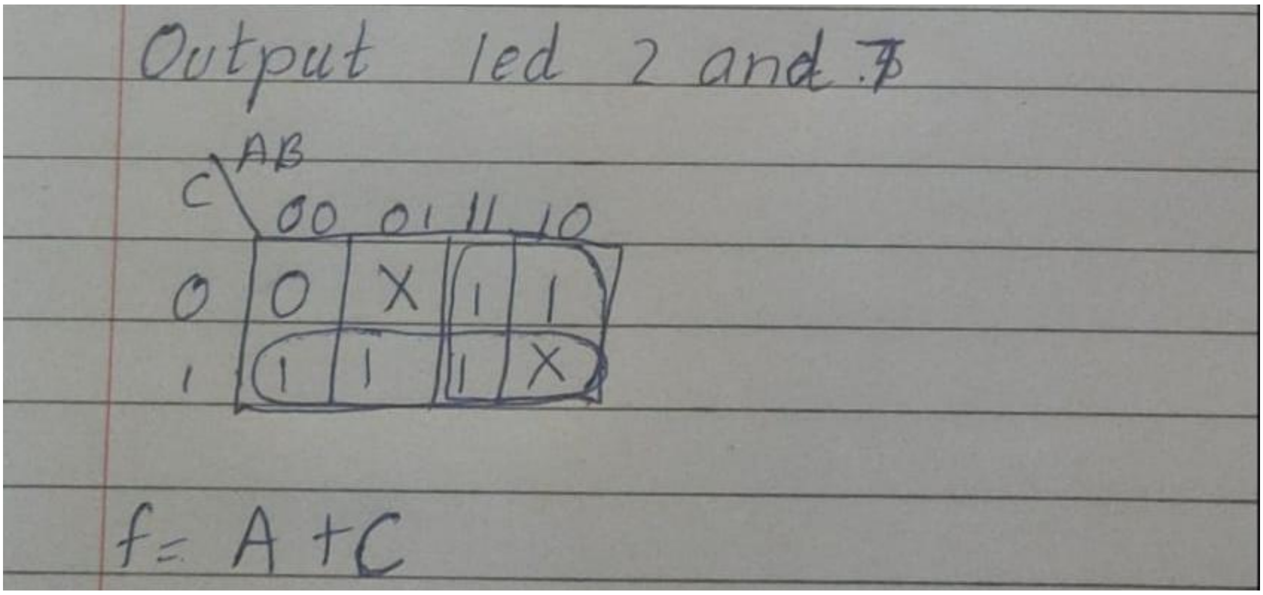
Output led 1

C\AB	00	01	11	10
0	1	X	1	0
1	0	1	0	X

$f = \bar{A}\bar{C} + \bar{A}B + B\bar{C}$ (for minterm

Figure 2 Equation for led 1 from K-map

The equation in figure 2 we can say that we can have a combination of AND gate and OR gates but on having close look we can use a XNOR logic gate with 3 inputs (A, B, C) so the led 1 will be ON and OFF for every alternate number.



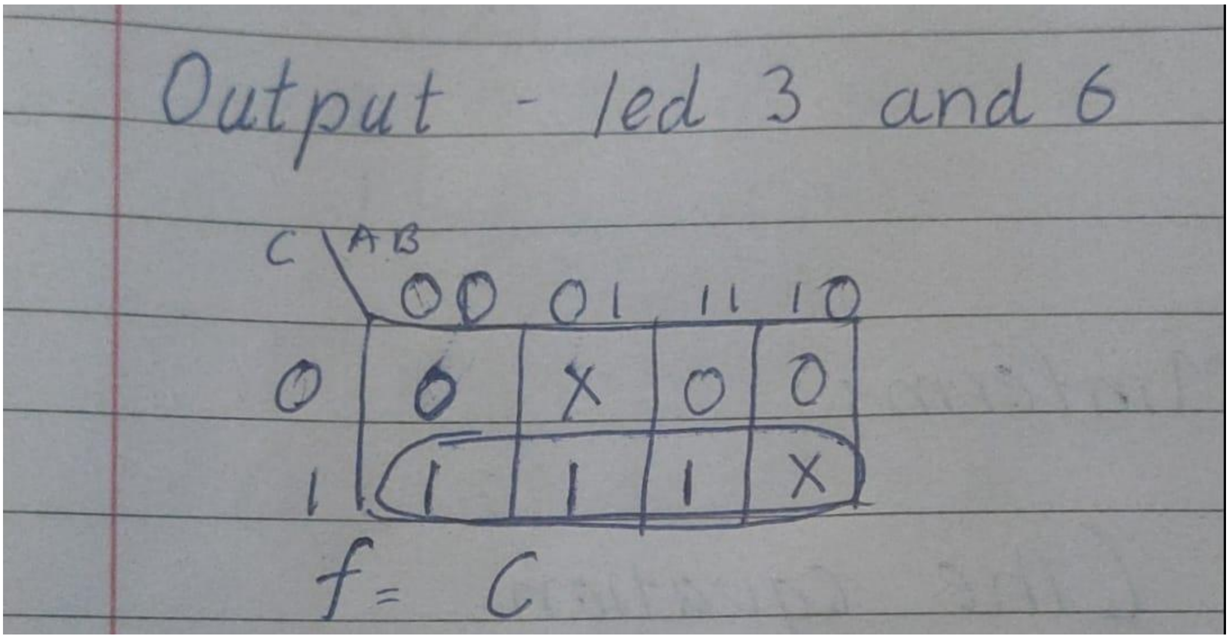
Output led 2 and 7

C\AB	00	01	11	10
0	0	X	1	1
1	1	1	1	X

$f = A + C$

Figure 3 Equation for led 2and 7 from K-map

The equation in figure 3 we can say that we can have a simple OR gate with input A and C for the led 2 and 7.



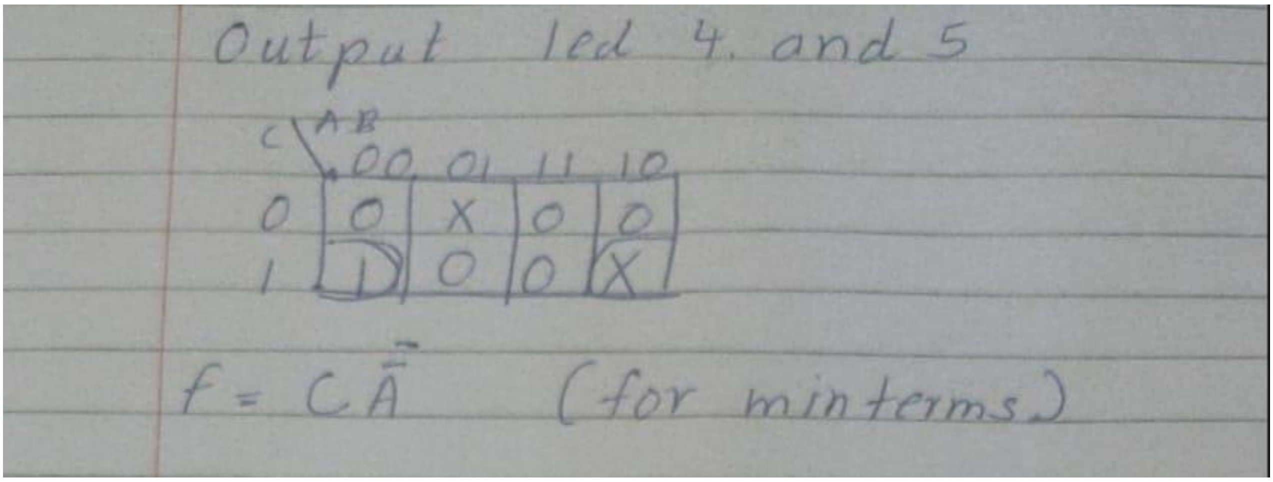
Output - led 3 and 6

C\AB	00	01	11	10
0	0	X	0	0
1	1	1	1	X

f = C

Figure 4 Equation for led 3 and 6 from K-map

The equation in figure 4 we can say that we can directly connect the led 3 and 6 to C input or use a buffer.



Output led 4. and 5

C\AB	00	01	11	10
0	0	X	0	0
1	1	0	0	X

$f = C.\bar{A}$ (for minterms)

Figure 5Equation for led 4 and 5 from K-map

from the equation in figure 5 we can say that we can have a combination of NOT gate and a AND gate with B input for NOT gate and its output is feed to OR gate with C.

2.2. Stimulating BCD to 7-Segment Display Decoder

On studying the datasheets of the two main groups of decoders we can say that they both are have same basic work as they have 4 inputs and 7 outputs. The difference between them is that the other 3 pins (lamp test, blinking input and ripple blinking input) have a difference in 74LS47 they are active low but in 74LS48 they are active high. And the output in 74LS47 is active high but in 74LS47 they are active low.

So 74LS47 works with common anode 7-segment display where 74LS48 works with common cathode 7-segment display.

In this project we are using the decoder to show the numbers on 7-segment display to which is same as number shown on dice and used to cheek that the stimulation is working in write order.

2.2.1. Moduleo-n counter.

In this step we design a counter that give 6 unique state output which repeat in same sequence. For this we are going to use a switched tail ring counter with 3D-flipflop to build a modulo-6 counter. We will connect all flipflop with same clock input. And an inverted output of the last flipflop is connected back to the input of the first flipflop and non-inverting outputs of first and second flipflop is connected to the input of second and last flipflop, respectively.

2.3. FPGA

Now to run the circuit with the peripherals on the board we need to use FPGA (field programable gate array). It is a type of Programable Logic Device. And this helps us to convert the circuit design to a real circuit. For this we are using digilent Nexys 4. On having look at the datasheet, the three are a few buttons which can be used to roll the dice and Pmod ports to connect the LEDs and other external circuit. We will be using the Pmod JA port to setup the LEDs and button D to run the clock.

3. Implementation

3.1. Logic circuit

In Multisim open the new blank design.

Place the following components

Component	Quantity
Interactive_digital_constant	3
AND gate	1
OR gate	1
XNOR gate(3-input)	1
NOT gate	1
Probe	7

Turn all key of digital constant to A, B, C. which represent as input A, B, C.

And then rearrange the components and wire it as shown in figure 6

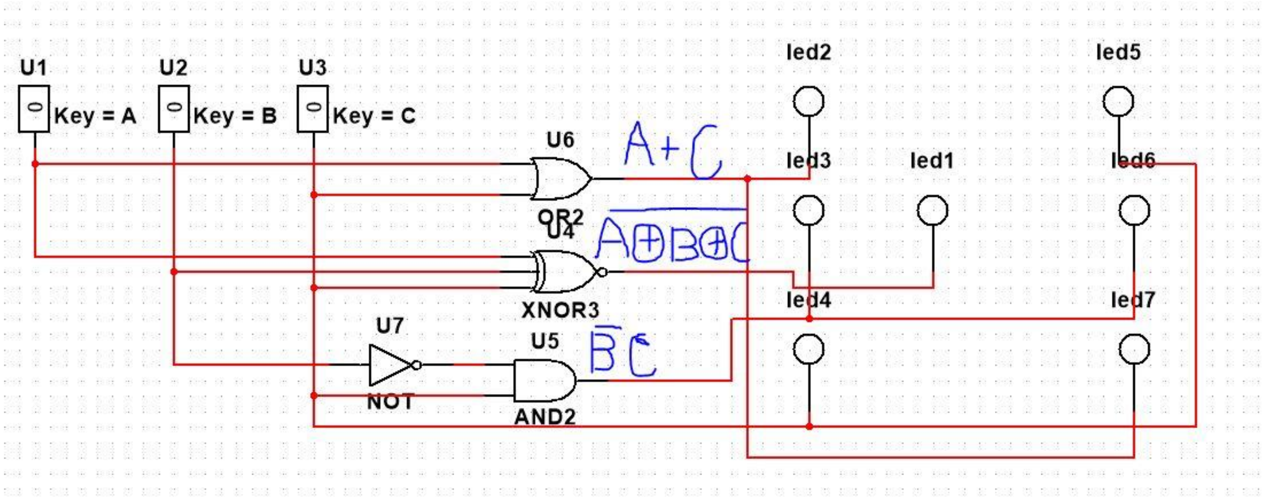


Figure 6 Logic circuit for all side of dice.

Run the stimulation and try changing the inputs and notice the change in LEDs.

And save the file and name it as ‘Logic circuit’

3.2. Stimulating BCD to 7-Segment Display Decoder

In Multisim open new Design
Place the following components

Component	Quantity
Interactive_digital_constant	4
Digital constant	1
7-segment display with common cathode	1
7Line_Isolated	1
ground	1

Turn all key of digital constant to A, B, C, D.

And then rearrange the components and wire it as shown in figure7

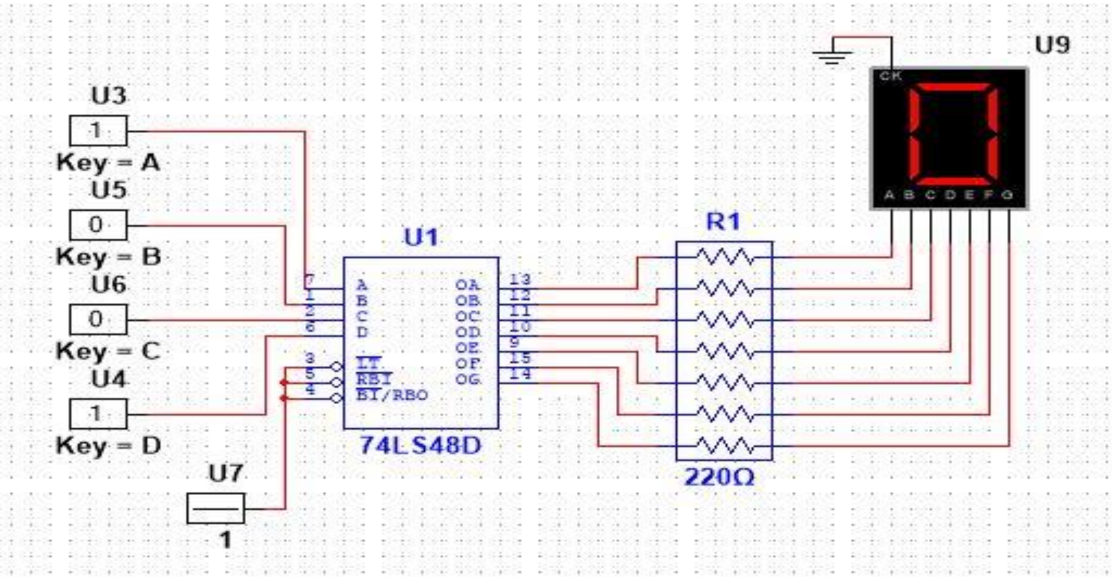


Figure 7 7-segment decoding circuit
Run the stimulation and try changing the inputs and notice the change in 7-segment display.

Save the file as 7-Segment display decoder.

3.3. Moduleo-n counter

In Multisim open new Design
Place the components as shown in table

Component	Quantity
Interactive_digital_constant	1
Digital clock	1
AND gate	1
D_FF	3
probe	3

Change the frequency of the clock to 10 Hz.
And then rearrange the components and wire it as shown in figure8

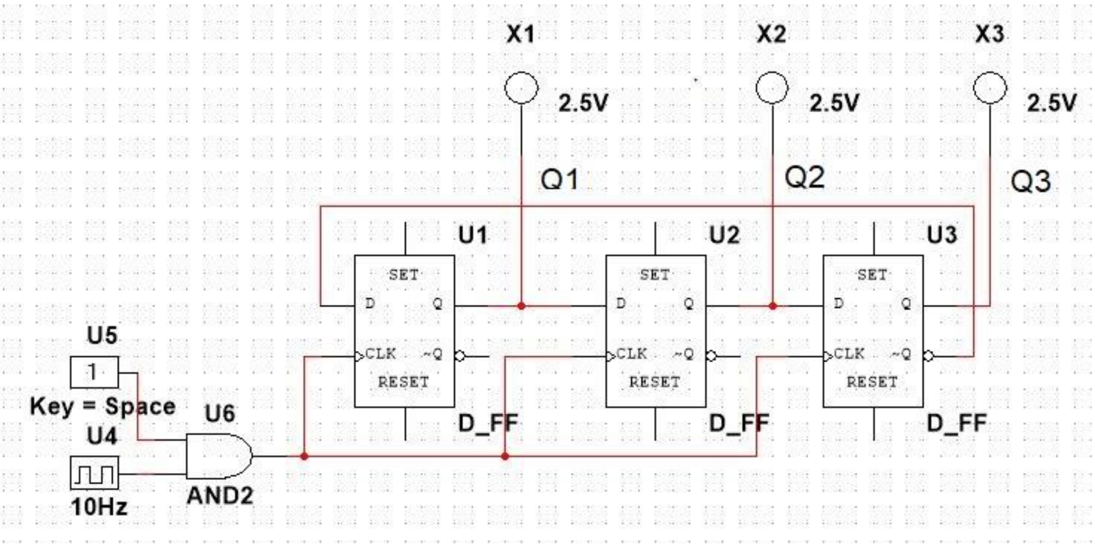


Figure 8Moduleo-6 counter

Run the stimulation and try changing the input with Space and notice the change in probe LEDs
Save the file as Moduleo-6 counter.

3.4. Connecting all the circuits

In Multisim open new Design and name it as Digital dice
Also, open moduleo-6 counter and logic circuit design

Now copy the moduleo-6 counter design and paste it in the digital dice design

Then change the frequency of the clock to 5kHz and delete all the three probes.

Now copy the logic circuit and paste it in the digital dice design.

Now select the all the logic gates of logic circuits and press ctrl+shift+b to make it a subcircuit. And name it as ‘ledon’.

Remove all the interactive probes that are connected to subcircuit block. And connect the inputs of the flipflop. As shown in the figure9

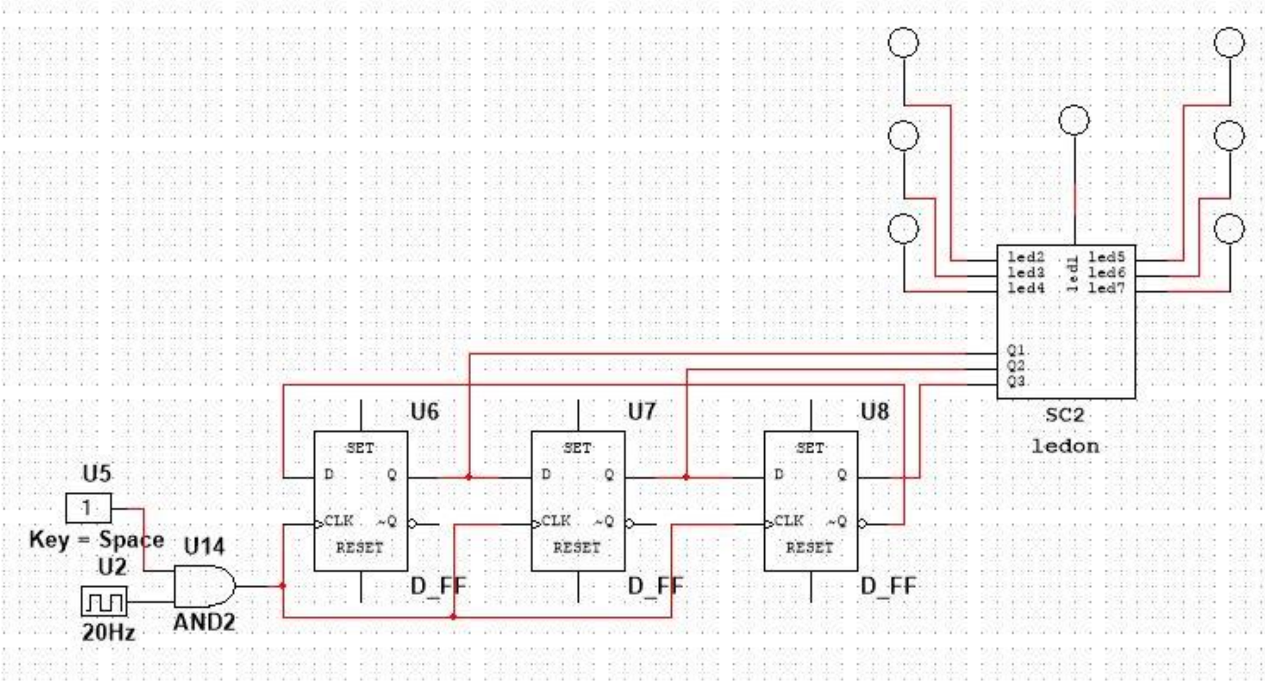


Figure 9 Final Design for digital dice

Open the subcircuit and open the hierarchical Connectors of input and change the direction to input for all three inputs.

In the subcircuit open the hierarchical Connectors of LEDs and change the direction to output for all seven LEDs. As shown in figure 10

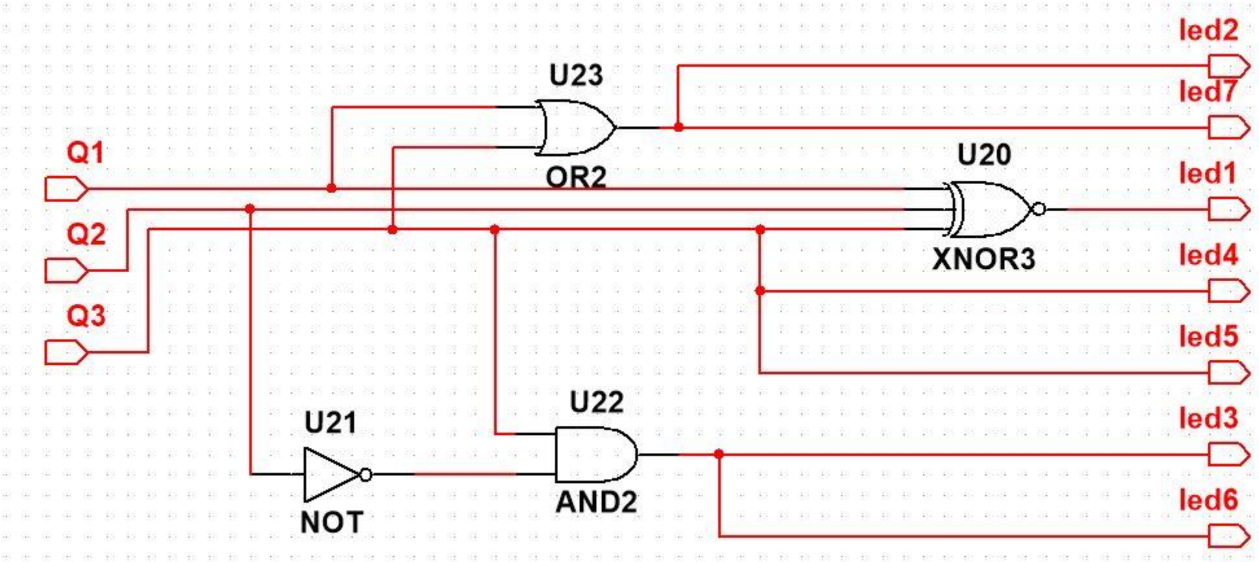


Figure 10 subcircuit for final design

3.5. FPGA

Open a new PLD design in digilent Nexys 4 and name it as Digital dice.

Select the connectors JA0 to JA6 and BtnD

Add the components shown in the list

Component	Quantity
Digital clock	1
D flipflop	3
Xnor gate (3 inputs)	1
Or gate	1
And gate	1
Not gate	1

Set all port mode to output (except BtnD).

And then rearrange the components and wire it as shown in figure 11

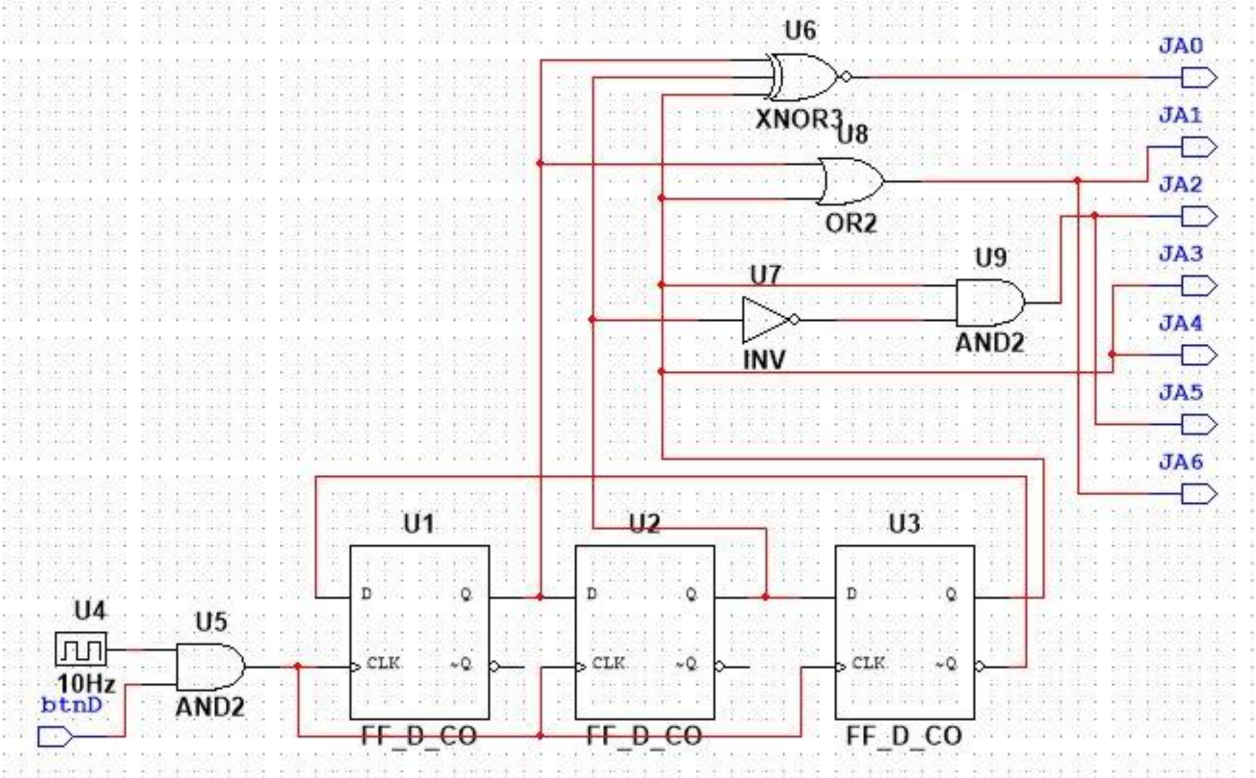


Figure 11 PLD design

Now export the file to PLD

Built the external circuit with 7 LEDs in 'H' shape.

Power up the PLD and try running the design with the help of button and observe the change in LEDs pattern

4. Results

Let us have a look at the outcome of all the steps

4.1. Logic circuit

So here we had designed a Logic circuit with the help of the Gates form truth table. So, on running the stimulation the LEDs are turning in the desired pattern which is like the table

Output A is on for every alternate case (1,3 and 5)

Output B is on in all the case except the first case (1).

Output C is on for last the case (4, 5 and 6)

Output D is on for only the last case (6)

With the help of the logic gate and the K-map it was easy to find out the logic circuit for all the desired output. And, was not required to reduce the equation.

4.2. Stimulation of the 7-segment display

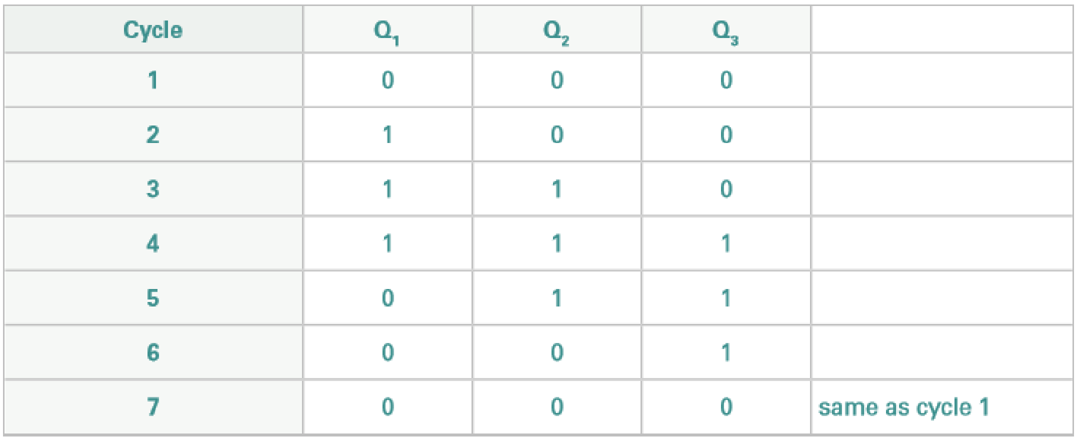
Here we try to Understand the use of the seven segment display decoders. We used 2 main families of IC (74LS47 AND 74LS48) We understood that they both are way too different but work quite similar. And we had look to the datasheet and found the information of both and decided to use 74LS48 to decode. On running the circuit and changing the inputs with interactive constants the output shows the corresponding numbers on 7-segment display. Here we need to make sure that the 7-segment display have common cathode to run the stimulation.

Input A	Input B	Input C	Input D	Output on Display
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
1	1	0	0	3
0	0	1	0	4

1	0	1	0	5
0	1	1	0	6
1	1	1	0	7
0	0	0	1	8

4.3. Moduleo-6 counter

In this section we had a look at a different type of counter than we used in class. The main difference between then was sync. As Moduleo-6 counter in Synchronous counter. So due to this we also need to wire them in different ways and have different way of outputs as well. And on running the stimulation we got the output as shown in table



Cycle	Q_1	Q_2	Q_3	
1	0	0	0	
2	1	0	0	
3	1	1	0	
4	1	1	1	
5	0	1	1	
6	0	0	1	
7	0	0	0	same as cycle 1

4.4. Compiling all together

In this section we had connected the counter to the logic circuit and made a stimulation for the digital dice. Here we also changed the frequency of the digital dice to 5kHz and it is to fast for eyes to read as shown in figure12

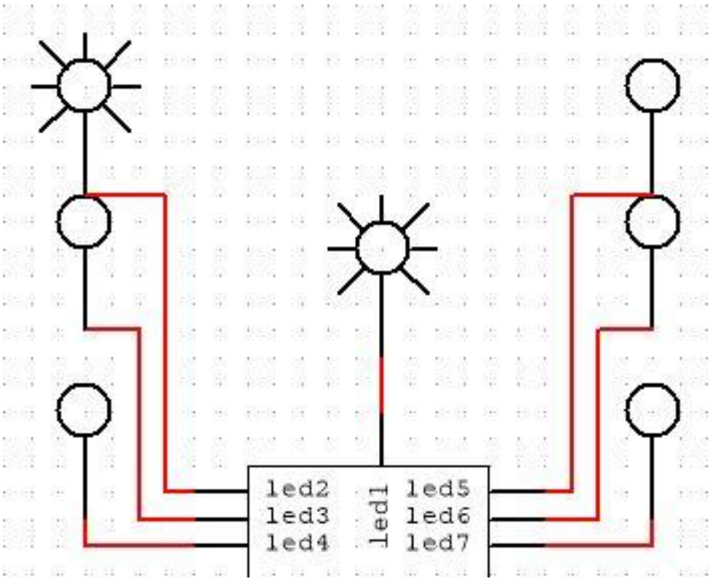


Figure 12 rolling dice in stimulation

4.5. FPGA

In this optional section we got the design exported to PLD, but we need to make a few changes to export to PLD (like change interactive digital constant to button and LEDs to Pmod outputs etc). We also built an external circuit for LEDs.

5. Conclusion

We created a digital die by using logic gates along with counter. A total of 4 basic pattens were used in different combinations to create all the 6 possible outcomes. During solving the challenge, I got to learn about how to use logic gates in practical. I also learn about different types of decoders, counters, PLD and clocks. The error is that the clock would not give perfect 5kHz frequency so that we cannot predict to outcome, but it will be enough fast that eyes cannot read the numbers. That make this capable of replacing the physical dice.