# IMPLEMENTATION OF RANDOM NUMBER GENERATOR USING D FLIP-FLOPS IN FPGA

MARIKUNDAM HARSHITHA
marikundamdec@gmail.com
FWC22120 IITH-Future Wireless Communications Assignment-5

## Contents



## 1 Problem

(GATE2021-QP-EC)
Q.46 The propogation delay of the exclusive-OR(XOR) gate in the circuit in the figure is 3ns.The propogation delay of all the flip-flops is assumed to be zero.The clock(Clk) frequency provided to the circuit is 500MHz.

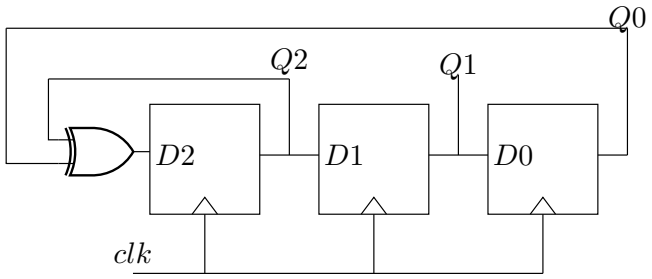


Figure 1: Circuit

Starting from the initial value of the flip-flop outputs $Q2Q1Q0 = 111$ with $D2 = 1$,the minimum number of triggering clock edges after which the flip-flop outputs $Q2Q1Q0$ becomes 1 0 0*(in integer)* is ___

## 2 Introduction

A random number generator using D flip-flops is a simple digital circuit that generates a sequence of random binary numbers.To implement this type of random number generator, we use a series of D flip-flops connected in a feedback

loop. The output of each flip-flop is fed back into the input of the next flip-flop,creating a circuit that generated a sequence of random binary values.

The feedback loop creates a delay in the circuit,which causes the circuit to exhibit unpredictable behavior.This unpredictable behavior results in a sequence of random binary values. The length of the delay can be adjusted to control the randomness of the output.

# 3 Components

| **Components** | **Value** | **Quantity** |
|---|---|---|
| Breadboard | | 1 |
| USB-C Cable | | 1 |
| Vaman | | 1 |
| Seven Segment Display | Common Anode | 1 |
| Decoder | 7447 | 1 |
| Flip Flop | 7474 | 2 |
| Jumper Wires | | 39 |

Table 1: Components

### 3.1 Seven Segment Display

The seven segment display has eight pins, *a,b,c,d,e,f,g* and *dot* that take an active LOW input,i.e. the LED will glow only if the input is connected to ground.Each of these pins is connected to an LED segment.The *dot* pin is reserved for the LED.

## 4 Setup

1. Connect the Vaman to the Laptop through USB.
2. There is a button and an LED to the left of the USB port on the Vaman.There is another button to the right of the LED.
3. Press the right button first and immediately press the left button.The LED will be blinking green.The Vaman is now in bootloader mode.

### 4.1 Steps for Implementation

1. Login to termux-ubuntu on the android device and execute the following commands:
   Make sure that the required installation and tool builds of pygmy-sdk had done prior executing below commands

```
proot-distro login debian
cd  /data/data/com.termux/files/home/
mkdir fpga
svn cohttps://github.com/srikanth9515/FWC/tree/main/fpga/codes
cd codes
ql$_$symbiflow -compile -src /data/data/com.termux/files/home/fpga/codes -d ql-eos-s3 -P PU64 -
    v helloworldfpga.v -t helloworldfpga -p quickfeather.pcf -dump binary
```

   This will generate **helloworldfpga.bin** file in codes directory transfer this bin file to laptop by executing the following command

```
scp /data/data/com.termux/files/home/fpga/codes/helloworldfpga.bin
    username$_$of$_$pc@IP$_$address:/home/username
```

   Make sure that the appropriate username,IP address of the Laptop is given in the above command.

2. Now execute the following commands on the Laptop terminal
   Make sure that required installation of programmer application had done prior executing below command

```
python3 /home/username/TinyFPGA-Programmer-Application/tinyfpga-programmer-gui.py --port /dev/
    ttyACM0 --appfpga /home/username/helloworldfpga.bin --mode fpga
```

3. After finishing the process of flashing with the programmer application press the button to the right of the USB port to reset. Vaman is now flashed with our source code

## 5 Implementation

A 7474 IC which has 14 pins and can store two seperate binary values.So we consider two IC's since we have three values and connect the D inputs of each flip-flop to the input signals of 7447 IC . Later interface 7447 IC to seven segment display for the output. The CLK input is used to trigger the flip-flop,and the Q output is used to read the stored value.When a positive edge is detected on the CLK input,the current value on the D input is stored in the flip-flop. The boolean expression of the D flip-flop is $Q(t+1) = D$

### 5.1 Truth table

| Present State | | | Flip-Flop input | | | Next State | | |
|---|---|---|---|---|---|---|---|---|
| Q2 | Q1 | Q0 | D2 | D1 | D0 | Q2' | Q1' | Q0' |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 2: Truth Table

### 5.2 K-map

Since $Q' = D$,we find the k-maps for D as outputs

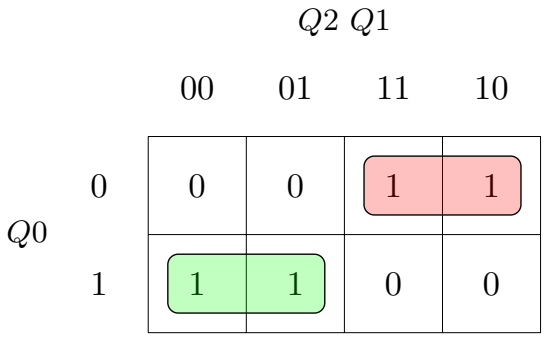


Figure 2: For D2

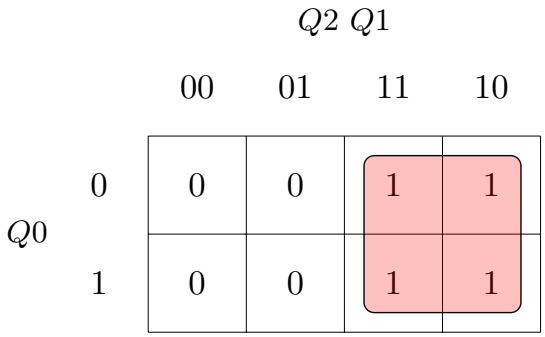


Figure 3: For D1

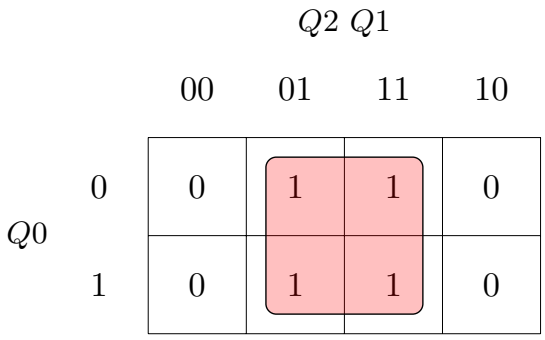


Figure 4: For D0

### 5.3 Boolean Equation

By solving the K-maps above we obtain as follows :

$$D2 = \overline{Q2}Q0 + \overline{Q0}Q2 \tag{1}$$
$$D1 = Q2 \tag{2}$$
$$D0 = Q1 \tag{3}$$

## 6 Hardware

1. Make the connections between the seven segment display and the 7447 IC as shown in Table3

| **7447** | $\overline{a}$ | $\overline{b}$ | $\overline{c}$ | $\overline{d}$ | $\overline{e}$ | $\overline{f}$ | $\overline{g}$ |
|---|---|---|---|---|---|---|---|
| **Display** | a | b | c | d | e | f | g |

Table 3: 7447

2. Connect the Vaman,7447 and the two 7474 ICs according to Table4

| | INPUT | | | OUTPUT | | | CLOCK | | 5V | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q0 | Q1 | Q2 | Q0' | Q1' | Q2' | | | | | | |
| Vaman | IO4 | IO5 | IO6 | IO1 | IO2 | IO3 | IO7 | | | | | |
| 7474 | 5 | 9 | | 2 | 12 | | CLK1 | CLK2 | 1 | 4 | 10 | 13 |
| 7474 | | | 5 | | | 2 | CLK1 | CLK2 | 1 | 4 | 10 | 13 |
| 7447 | | | | 7 | 1 | 2 | | | 16 | | | |

Table 4: Connections

3. Make the other D input pins of 7474 grounded and supply 5V and GND from the Vaman as well.
4. When the clock edge is trigerred we observe display of random numbers.

## 7 Software

Now write the following code and upload in vaman to see the results.

```
module logicex(
    input clk,
    input Q0,
    input Q1,
    input Q2,
    output reg D0,
    output reg D1,
    output reg D2
    );
```

```
    always @(*) begin
        D2 = (Q2&&(!Q1)) || (Q0&&Q2);
        D1 = (Q2);
        D0 = (Q1);
end
endmodule
```

---