EXPERIMENT – 7

RANDOM NUMBER GENERATOR FOR GAMING USING D- FLIPFLOP

Aim: To generate random number for gaming, using D flip-flops.

Components Required:

S. No	Component Name	Quantity
1	IC 7474	1
2	IC 7486	1
3	Breadboard	1
4	LEDs	4
5	Connecting Wires	Required number

Pre-lab:

1. **What is a D flip-flop, and how does it operate?**

   Explain the basic operation of a D flip-flop, including its inputs, outputs, and how the clock signal affects its behavior.

2. **How does a D flip-flop store a bit of information?**

   Discuss the concept of data storage in a flip-flop and the significance of the clock in updating and holding this information.

3. **What is a Linear Feedback Shift Register (LFSR)?**

   Define an LFSR and explain its structure and how it can generate sequences of numbers.

4. **How does feedback work in an LFSR, and why is it important?**

   Feedback in an LFSR involves combining outputs from specific taps using a logical operation and feeding this back into the input to generate a long sequence of pseudo-random numbers before repeating, crucially determining the sequence's quality and period.

Theory:

Creating a Random Number Generator (RNG) for gaming using D flip-flops typically involves employing a digital circuit design known as a Pseudo-Random Number Generator (PRNG). The theory behind using D flip-flops for this purpose revolves around the principles of digital logic design, the behavior of flip-flops, and the method for generating sequences that appear random.

D Flip-Flops: A D flip-flop is a type of digital memory circuit that has two stable states and can store a single bit of data. It has a data input (D), a clock input (CLK), and outputs Q and the inverse of Q (Q'). On the rising edge of the CLK signal (or falling edge, depending on the flip-flop's design), the value at the D input is captured and becomes the output at Q until the next clock cycle.

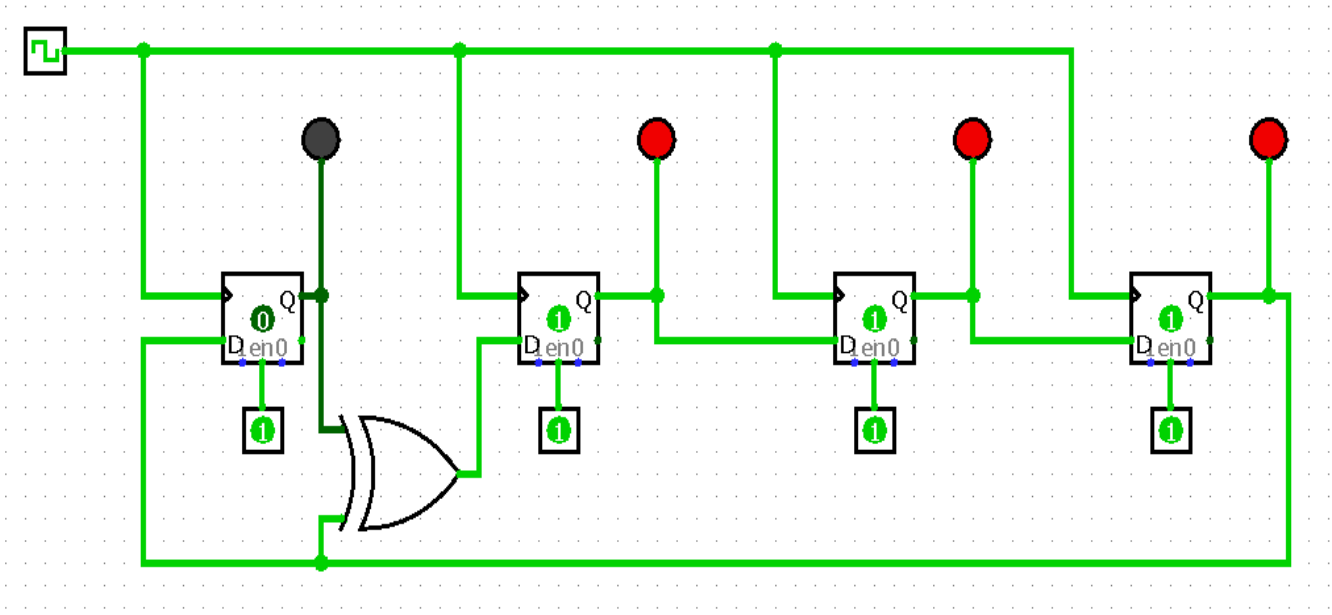
Generating Random Numbers

Pseudo-Random Number Generation: True randomness is difficult to achieve with digital circuits, which operate based on predefined rules and are inherently deterministic. Instead, circuits can generate pseudo-random numbers—sequences of numbers that appear random for practical purposes. The sequence is determined by the circuit's initial state (seed) and its structure but will repeat after a certain period.

Linear Feedback Shift Registers (LFSRs): One common approach to building PRNGs with D flip-flops is to create a Linear Feedback Shift Register. An LFSR is a shift register (a series of flip-flops arranged in a line) where the input to the first flip-flop is a linear function of the values of other flip-flops in the register. This is often accomplished by taking the XOR (exclusive OR) of the outputs of selected flip-flops (the taps) and feeding it back into the input of the first flip-flop in the sequence.

Feedback Mechanism: The choice of taps (which flip-flops' outputs are used for feedback) is crucial for the quality of the pseudo-random sequence. Proper tap selection can ensure a maximal length sequence, meaning the sequence will be as long as possible before repeating, given the number of flip-flops in the LFSR.

Circuit Diagram:



Procedure:

1. Arrange the four D flip-flops as illustrated in the provided circuit diagram.
2. Apply a common clock signal to all the flip-flops to synchronize their operation.
3. Build an XOR gate using the output (Q) of the first flip-flop and its input 'D', and connect this gate's output to the D input of the second flip-flop.
4. Initiate the clock signal (make sure any one flip-flop in state “1”) and observe the sequence of output bits from each flip-flop, which collectively represent a pseudo-random number.

Viva Questions and answers:

Q1: What makes a sequence of numbers pseudo-random?

A1: A sequence is considered pseudo-random if it meets two criteria: it appears to be random and unpredictable without knowledge of the internal state of the system generating it, and it is deterministic,

meaning it can be reproduced if the initial state of the system is known. In digital systems, pseudo-randomness is often a product of complex, deterministic processes.

Q2: How does a Linear Feedback Shift Register (LFSR) generate pseudo-random numbers?

A2: An LFSR generates pseudo-random numbers by shifting bits through a series of flip-flops and introducing nonlinearity through feedback that is a linear function (usually XOR) of the outputs of specific flip-flops (taps). The feedback modifies the input of the first flip-flop in the series, creating a sequence of bits that can appear random.

Q3: Why is the choice of taps important in an LFSR configuration?

A3: The choice of taps in an LFSR is crucial because it determines the sequence's period (how long it can run before repeating) and randomness quality. Properly chosen taps, based on mathematical criteria and polynomials, can maximize the LFSR's period, ensuring the generated sequence is as long and as random-looking as possible.

Q5: How can you ensure the randomness of an LFSR's output for gaming applications?

A5: To ensure the randomness of an LFSR's output for gaming, you can: Use taps that correspond to maximal-length polynomials.

Initialize the LFSR with a non-zero state (seed) to avoid generating a sequence of zeros.

Optionally, combine outputs from multiple stages of the LFSR or use more than one LFSR in parallel or series configurations to increase complexity.

Perform statistical tests (e.g., the Diehard tests) on the generated sequences to evaluate their randomness.

Result: The experiment successfully demonstrated the construction and operation of random number generator using D flip-flops.