United International University

Department of CSE

Course Code: CSE 1326

Course Name: Digital Logic Design Laboratory

Experiment no: 07

Experiment Name: Implementing different kinds of registers to accomplish different counters.

Submitted by:

Name: Talha Jubayer

Student ID: 0112410062

Submitted to:

Fahim Hafiz [Lecturer, Dept. of CSE]

Date of Performance: 15-01-2025

Date of Submission: 27-01-2025

Objective:

This experiment focuses on implementing various types of registers, each designed to perform specific tasks. These include creating standard registers, registers capable of performing left-right shifts, multifunctional registers, and registers built using multiple individual registers.

Apparatus/Instruments used in the Lab:

- Trainer Board
- Power supply
- NOT Gate (74LS04)
- AND Gate (74LS08)
- X-OR gate IC (74LS86)
- X-NOR gate IC (74LS266)
- NAND gate (74LS00)
- Dual D flip-flop IC (74AHC74)
- Dual JK flip-flop IC (74LS73)
- Logisim

Theory:

A counter is a digital device that records and sometimes displays the number of times a specific event or process has occurred, often synchronized with a clock signal. Counters are widely used in digital electronics for counting purposes. For instance, an UP counter increments its count with each rising edge of the clock signal. Beyond simple counting, counters can be designed to follow specific sequences, such as a custom pattern like 0, 1, 3, 2, and so on. These devices are typically constructed using flip-flops.

Counters serve various functions, such as acting as frequency dividers, where they divide the frequency of an input pulse waveform. They are classified as sequential

circuits capable of counting pulses in binary or BCD (Binary-Coded Decimal) formats. The primary characteristics of counters include timing, sequencing, and counting.

Counters operate in two main modes:

1. **Up Counter**: Increments the count.
2. **Down Counter**: Decrements the count.

- **Classifications:**
  i. **Ripple Counter:** In a ripple counter, a universal clock is not used. Instead, only the first flip-flop is directly driven by the main clock signal, while the clock input for the subsequent flip-flops is triggered by the output of the preceding flip-flop. This setup creates a cascading effect, as seen in the timing diagram:

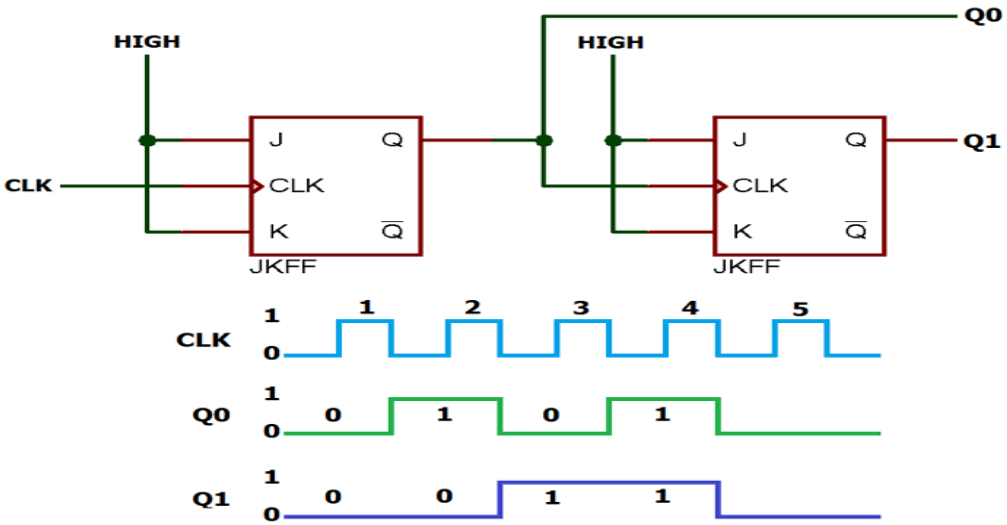


Figure 1 Timing diagram

Q0 changes state immediately upon the rising edge of the main clock pulse. Q1 changes state when it detects the rising edge of Q0 (since Q0 acts as the clock signal for the second flip-flop).
This pattern continues for Q2, Q3, and so on.
Due to this sequential propagation of changes, ripples are generated across the outputs (Q0, Q1, Q2, Q3), which is why this type of counter is also known as an asynchronous counter or serial counter. A ripple counter

consists of a series of flip-flops connected in a cascaded manner, where the output of one flip-flop serves as the clock input for the next.

ii. **Synchronous Counter:** A synchronous counter differs significantly from an asynchronous counter in its design and operation. In a synchronous counter, all flip-flops are driven by a single global clock signal. This means that all the flip-flops receive the clock pulse at the same time, causing their outputs to change simultaneously or in parallel. This synchronized operation eliminates the cascading effect present in asynchronous counters, where the output of one flip-flop serves as the clock input for the next.

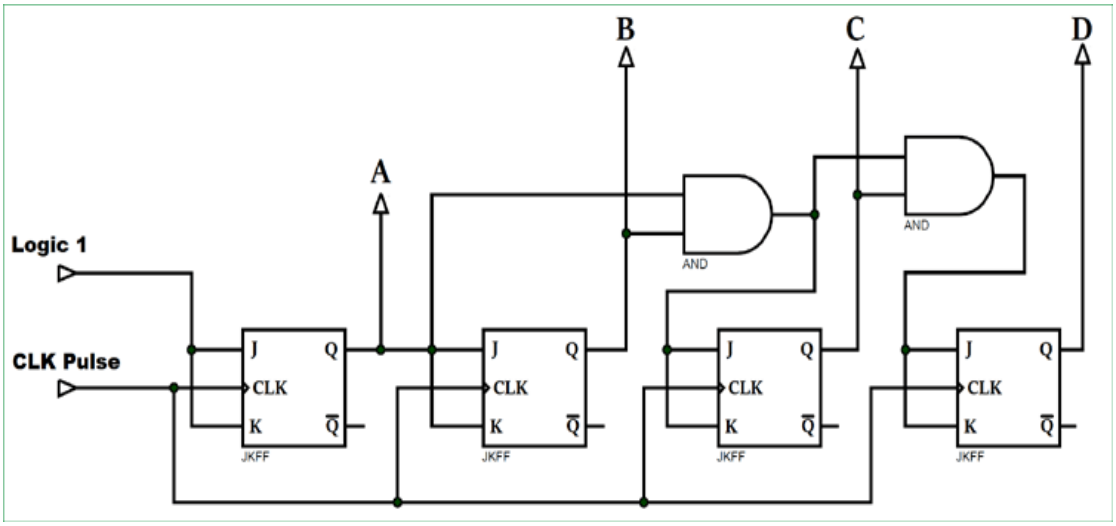


Figure 2 Synchronous Counter

One of the key advantages of a synchronous counter is its ability to operate at much higher frequencies compared to its asynchronous counterpart. In an asynchronous counter, the cumulative propagation delay increases with each additional flip-flop, as the clock signal must ripple through the chain of flip-flops sequentially. This cumulative delay limits the operating speed of the counter. In contrast, since all flip-flops in a synchronous counter are triggered by the same clock signal at the same time, there is no cumulative delay, allowing the counter to function more efficiently and at higher clock speeds.

Additionally, synchronous counters are often preferred in applications requiring precise and high-speed operations, as their parallel nature ensures accurate timing and output. This design makes synchronous counters particularly useful in digital systems where high-frequency operation and minimal delay are critical.

Due to their parallel operation, synchronous counters are also referred to as parallel counters. Their robust design and efficiency make them a popular choice in various digital circuits, especially those requiring rapid counting or sequencing.

From the circuit diagram, we see that the Q0 bit gives a response to each falling edge of the clock while Q1 is dependent on Q0, Q2 is dependent on (Q1, Q0) and Q3 is dependent on (Q2, Q1 & Q0).

iii. **Decade Counter:** A decade counter is a type of counter that cycles through ten distinct states before resetting to its initial state. A standard decade counter typically counts from 0 to 9, making it ideal for applications that require decimal counting. However, it's also possible to design decade counters that follow any ten states within the range of 0 to 15 (since a 4-bit counter can represent 16 unique states).

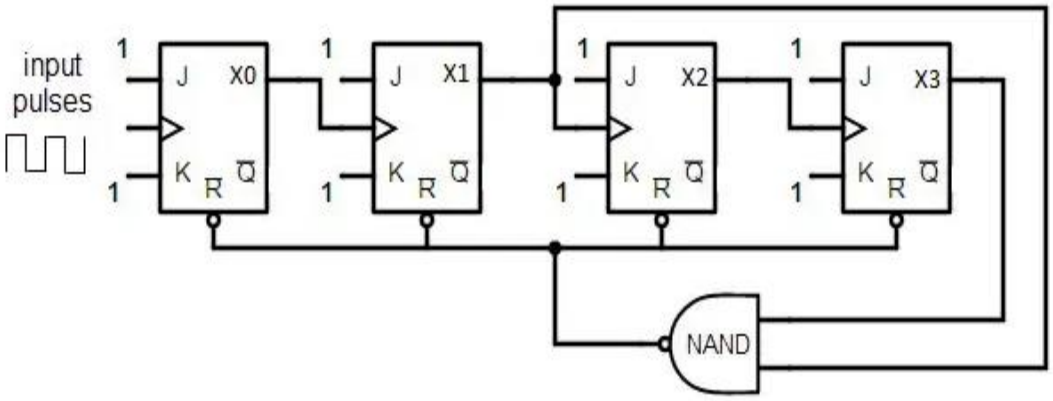
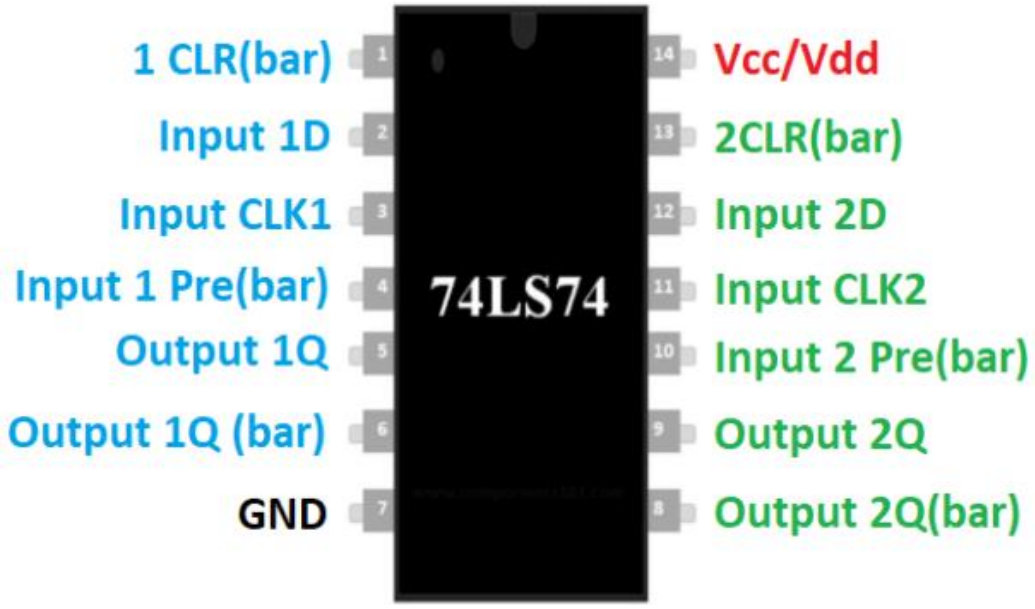


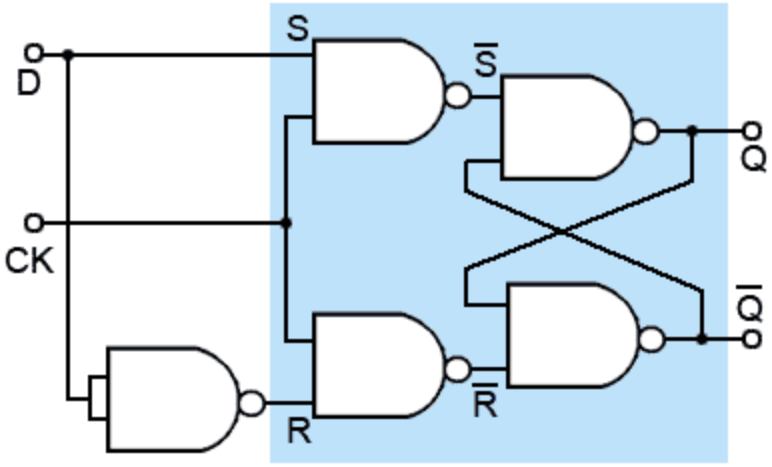
Figure 3 Decade Counter

Solution of the mentioned problems:

i) Pin Diagram of D Flip-Flop IC:

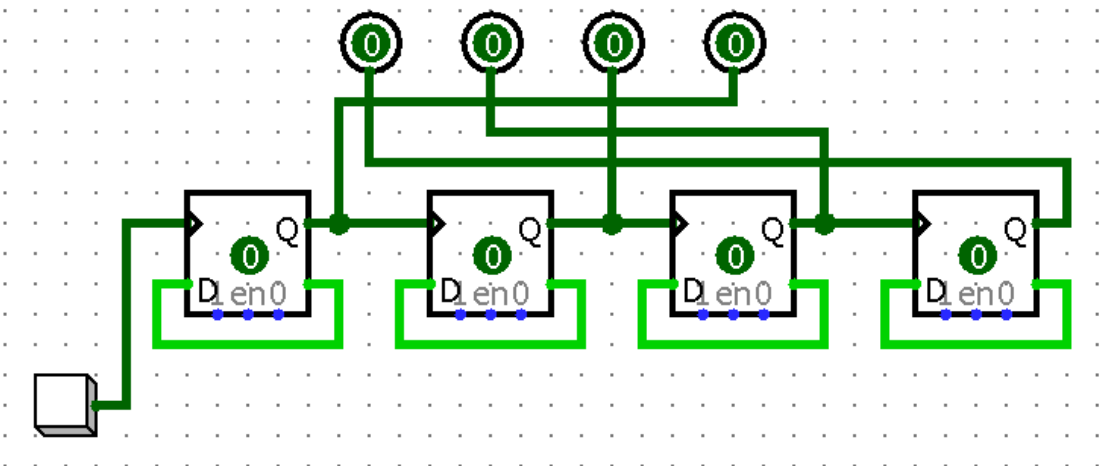


ii) Gate Diagram of D Flip-Flop IC:

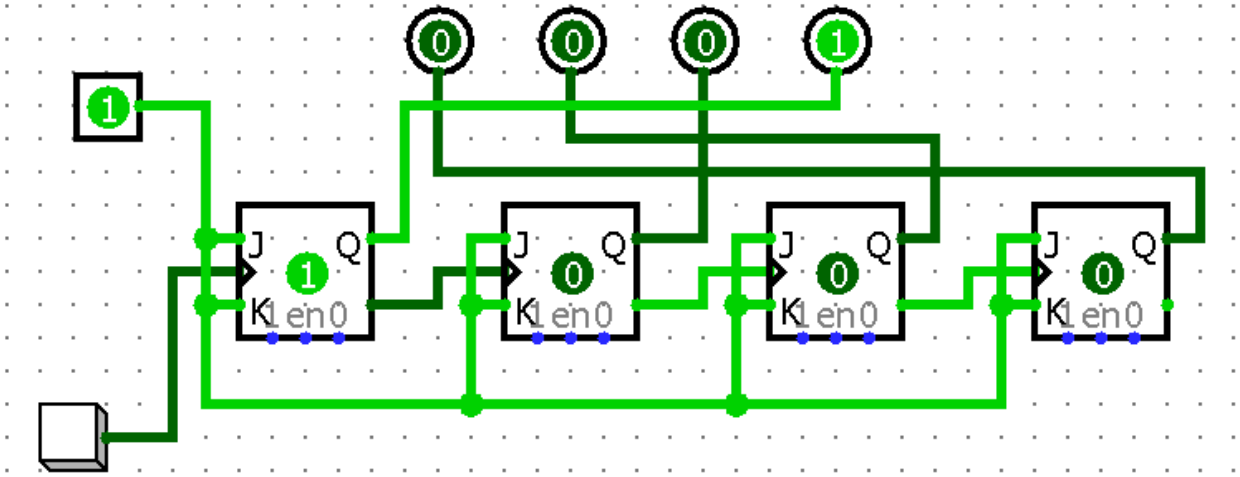


iii) Logic Diagram of 4-bit Downward Counter using:

a. +ve edge D flip-flop:



b. -ve edge JK Flip-Flop:



Discussion:

After conducting an experiment on ripple counters, I gained a deeper understanding of their functionality as a type of digital circuit capable of counting up or down based on the input signal. The counter operates using a series of flip-flops, each of which changes state with every clock pulse, enabling the counting process. Observing this behavior was particularly fascinating, as the clock signal ripples through the circuit, sequentially toggling each flip-flop.

During the experiment, I also identified some challenges associated with ripple counters, such as propagation delay and glitches. Propagation delay occurs because

the output of one flip-flop serves as the clock input for the next, causing a slight delay as the signal cascades through the circuit. This delay can impact the counter's accuracy, especially at higher frequencies. Glitches, or temporary erroneous outputs, can also arise from the sequential nature of the toggling. These issues highlighted the importance of careful design and testing to ensure reliable performance in practical applications.