جامعة بيرزيت
BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

Digital Electronics & Computer Organization Laboratory (ENCS2110)

"Report 3"

Sequential Logic Circuits.

Prepared by:

Name: Eman Asfour **Number**: 1200206

Partners:

Name: Worood Fathi **Number**: 1210412

Name: Rand Awadallah **Number**: 1211963

Instructor: Dr. Jamal Seyam

Assistant: Eng.Haleema Hmedan

Section: 4

Date: 26 / 8 / 2023

Place: Masri 107

1. Abstract

The experiment is aimed at understanding the distinctions between sequential and combinational logic circuits, as well as the uses of various memory units. Additionally to research various flip-flops' applications and operating principles. Moreover, it is important to comprehend how counters work and how to build them using JK flip-flops. Finally To study the synchronous and asynchronous counters.

Table of Contents

Table of Figures

Table of Table

2. Theory

— Sequential Circuits

Sequential circuits, as you correctly pointed out, are two-valued networks where the outputs depend not only on the current inputs but also on the previous inputs and the internal state of the circuit. These circuits have memory elements like flip-flops, which store information and allow the circuit to maintain a state. Sequential circuits have a concept of time or sequencing, as the outputs change over time based on the input history and the current state of the circuit. Examples of sequential circuits include flip-flop-based memory units (RAM), counters, registers, and more complex systems like microprocessors.

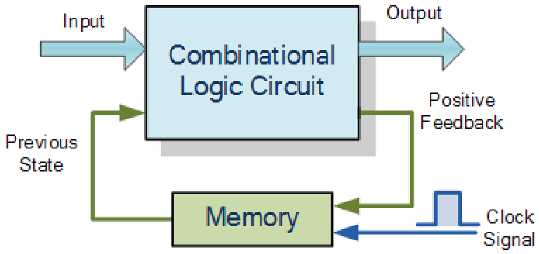


Figure 1: Sequential Circuit Block Diagram [1]

— Latches:

One class of flip-flops includes latches. The lack of controlled timing for output changes is what distinguishes this class. Although latches are useful for designing asynchronous sequential circuits and storing binary information, they are not practical for use in sequential circuits that are synchronous.

— The SR (Set-Reset) Latch

There are two cross-coupled NOR or NAND gates in the circuit.

a) SR latch with NAND gates

Figure 2 depicts the one with NAND gates. As shown in Table 5.1, this circuit is an active low set/reset latch, meaning that the output Q changes to 1 when the S (set) input is 0, and to 0 when the R (reset) input is 0. When both inputs are equal to 0 at once, the condition is undefined.

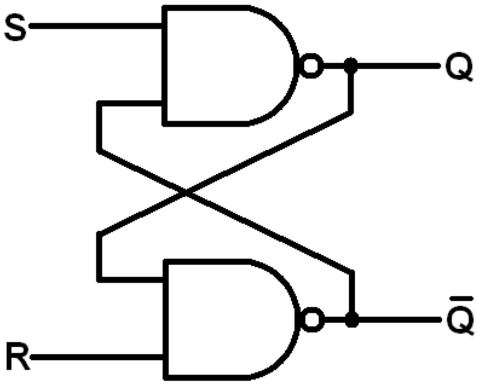


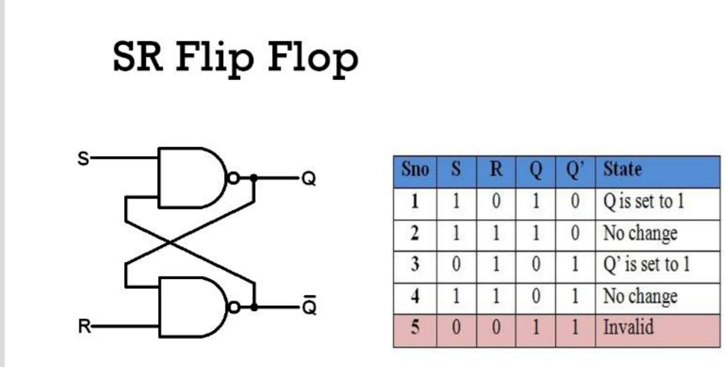
Figure 2: SR latch with NAND gate [2]

INPUT		OUTPUT		State
S	R	Q	¯Q	
1	0	0	1	SET
1	1	0	1	No Change/ Memory
0	1	1	0	RESET
1	1	1	0	No Change/ Memory
0	0	1	1	Invalid

Table 1: Table 1:SR latch with NAND gate Truth table

b) SR latch with NOR gates:

An SR latch, also known as a Set-Reset latch, is a fundamental digital logic circuit used to store one bit of information. It has two inputs: Set (S) and Reset (R), and two outputs: Q and Q'. Here's how you can implement an SR latch using NOR gates:



Sno	S	R	Q	Q’	State
1	1	0	1	0	Q is set to 1
2	1	1	1	0	No change
3	0	1	0	1	Q’ is set to 1
4	1	1	0	1	No change
5	0	0	1	1	Invalid

Figure 3: SR latch with NOR gates

c) RS latch with control input:

Figure 4 depicts the RS latch with control input C. If C=0, only the R and S values are affected by the output Q. The circuit will operate normally if C = 1, as shown in Table 2.

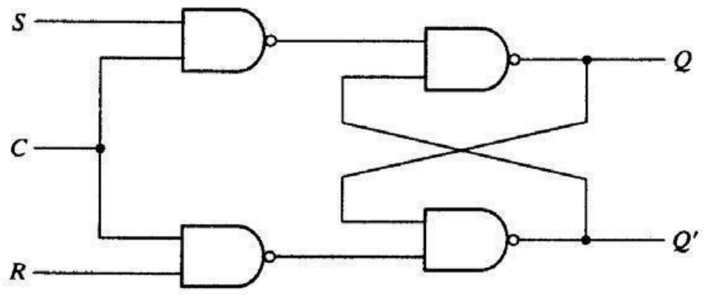


Figure 4: RS latch with control input. [3]

INPUT			OUTPUT		State
C	S	R	Q_{n+1}	Q◌́	
0	X	X	Q_n	Q◌́	No Change/ Memory
1	0	0	Q_n	Q◌́	No Change/ Memory
1	0	1	0	1	RESET
1	1	0	1	0	SET
1	1	1	0	0	Indeterminate

Table 2: RS latch with control truth table

— Flip-flop

“Flip-flop is a circuit that maintains a state until directed by input to change the state. A basic flip-flop can be constructed using four-NAND or four-NOR gates”. [4]

Both flip-flops and latches are components of digital storage, but they function differently and have different applications. Edge-triggered flip-flops offer steady, synchronous behavior ideal for precise timing in sequential circuits like registers and memory. Latches, on the other hand, are level-sensitive and used in asynchronous designs where timing is less important, though they might use more power and be more prone to errors. The decision between them is based on the particular specifications of the digital system that is being designed.

It is possible to implement a flip-flop using two distinct latches. Two D latches are used to implement the D flip flop in Figure 5.

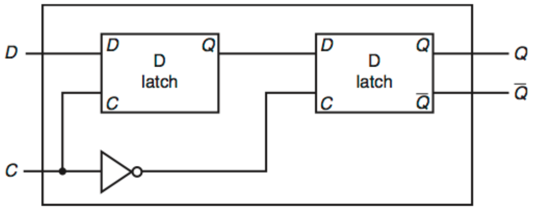
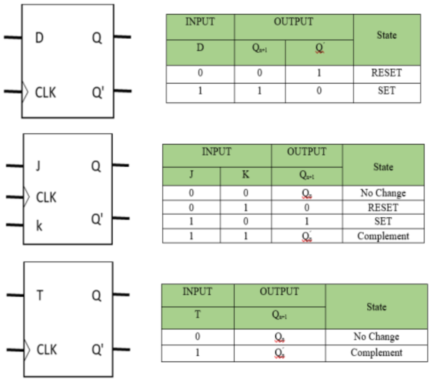


Figure 5: D flip flop implemented with two D latches. [5]

These flip flops are shown in Figure 6 along with their function tables.



INPUT	OUTPUT		State
D	Q_{n+1}	Q'	
0	0	1	RESET
1	1	0	SET

INPUT		OUTPUT	State
J	K	Q_{n+1}	
0	0	Q_n	No Change
0	1	0	RESET
1	0	1	SET
1	1	Q_n'	Complement

INPUT	OUTPUT	State
T	Q_{n+1}	
0	Q_n	No Change
1	Q_n'	Complement

Figure 6: D, JK, and T flip flops [3].

— Registers

Registers are frequently used in digital systems to store binary data. An N-bit register is made up of N separate flip-flops. A register is essentially a collection of flip-flops. You mentioned a straightforward 4-bit register in your case that was implemented using D flip-flops. These flip-flops can be reset simultaneously and share a common clock signal.

This configuration enables the digital system to synchronize data manipulation and storage, ensuring that every bit in the register is updated simultaneously on the rising or falling edge of the clock signal. In addition, a simultaneous reset makes sure that every flip-flop is reset simultaneously to a known state, which is necessary for initializing the register's data. [6]

In conclusion, a 4-bit register implemented with D flip-flops is a typical building block in digital systems, enabling synchronized binary data storage and manipulation with the option to reset all bits simultaneously when necessary.

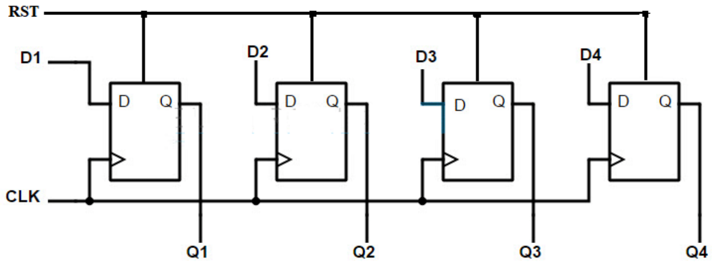


Figure 7: 4-bit Register. [7]

A collection of flip-flops connected in a manner such that the input of one flip-flop becomes the output of the next forms a shift register. The 4-bit shift-right register is shown in Figure 8.

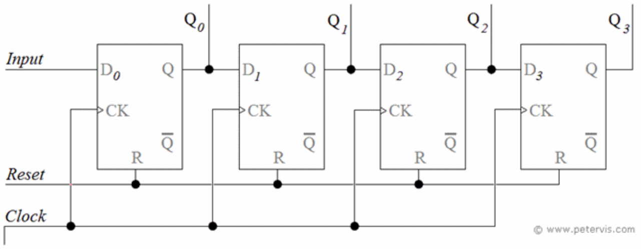
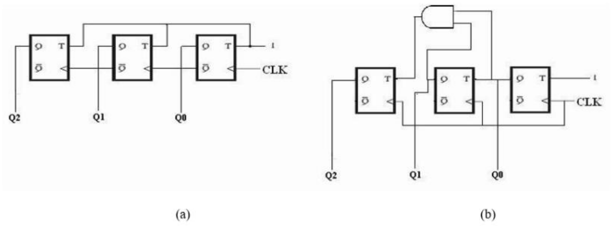


Figure 8: 4-bit shift- right register. [8]

— Counters

The counter is a special-purpose register that transitions through a set of predetermined states. Ripple counters and Synchronous counters are the two divisions of the counters. There is no common clock in ripple counters; instead, other flip-flops are triggered by the flip-flop output transition. All flip flops in synchronous counters share the same clock. 3-bit ripple and synchronous counters are displayed in Figure 9.



(a)

(b)

Figure 9: (a) 3-bit ripple counters, (b) 3-bit synchronous counter

3. Procedure

❖ Latches and Flip flops

— Constructing RS latch with Basic Logic Gates

The circuit was set up using the IT-3008 module. We connected the +5V output of a fixed power supply to the +5V input of the IT-3008 module, and we also connected the GND (Ground) output of the fixed power supply to the GND input of the IT-3008 module. Inputs A3 and A4 were linked to Switches SW1 and SW2, respectively. Additionally, Outputs F6 and F7 were connected to Logic Indicators L1 and L2..

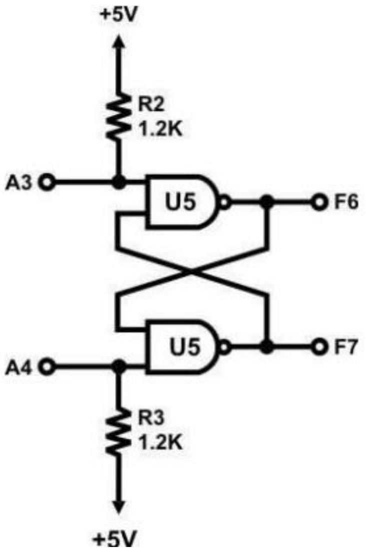


Figure 11: wiring diagram of RS latch.

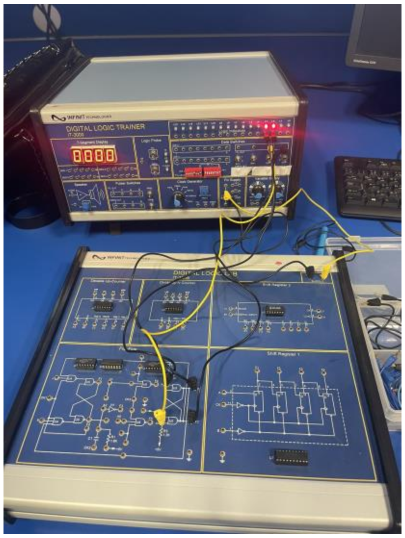


Figure 10: The Circuit when inputs A3& A4=0

INPUT	OUTPUT

A3	A4	F6	F7
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	0

Table 3: Data for RS latch

When both R and S inputs are set to 0, the RS latch retains its previous state. This is indicated by the 0-0 input combination, resulting in 0-0 outputs (A3 and F6 retain their values) When R is set to 0 and S is set to 1, the RS latch is set or turned ON. In the table, this is represented by the 0-1 input combination, which results in the Q output (F7) being set to 1 while its complement (Q', or A4) is set to 0. This means the latch is in a state where it stores a "1." .Conversely, when R is set to 1 and S is set to 0, the RS latch is reset or turned OFF. This corresponds to the 1-0 input combination, leading to the Q output (F7) being reset to 0, while its complement (Q' or A4) is set to 1. In this state, the latch stores a "0."If both R and S are set to 1 simultaneously (1-1 input combination), the RS latch enters an undefined or "forbidden" state, which results in unpredictable behavior. It's important to avoid this condition in practical applications

— Constructing RS latch with control input

The IT-3008 module was utilized to establish a connection within the circuit, facilitating the interface. Inputs A1 and A5 were linked to physical switches, SW1 and SW2, respectively. These switches were employed to emulate different logic levels (0 or 1) on the circuit inputs. The input sequence, as outlined in Table 4, was then passively observed, with no active changes to the switch positions. The circuit's behavior was analyzed based on this input sequence, focusing on its response and operation without actively toggling the switches.

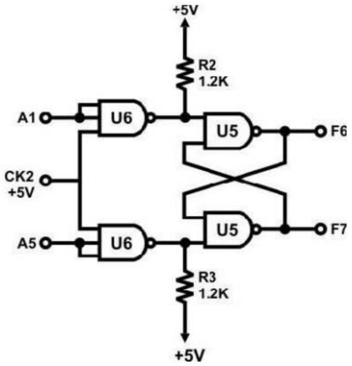


Figure 13:RS Latch with control input

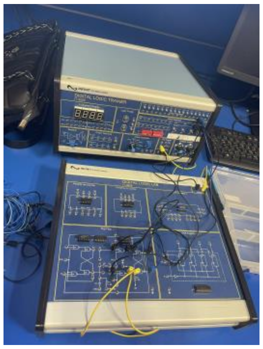


Figure 12: The circuit when input is 11 (but the power is off)

INPUT		OUTPUT	
A1	**A5**	**F6**	**F7**
0	**0**	**0**	**1**
0	**1**	**0**	**1**
1	**0**	**1**	**0**
1	**1**	**1**	**1**

Table 4:RS Latch with control input

The provided table details how an RS latch behaves when it has a third control input (A1) in addition to the usual Set (S) and Reset (R) inputs. The RS latch responds to the Set and Reset inputs normally when A1 is set to 0. A conditional behavior is introduced when A1 is set to 1; if Set (A5) is 0, the latch maintains its current state; if Set is 1, the latch sets itself. This control input makes the RS latch more adaptable to various applications by allowing it to react differently depending on the state of A1.

— Constructing D latch with RS latch

The circuit was constructed using the IT-3008 module, which played a critical role in interfacing various inputs and outputs. Specific connections included A1 to SW1, CK2 to SWA, and F6 to L1, facilitating control and observation. The circuit's behavior was studied by passively observing its responses to sequences detailed in Table 5, without active switching.

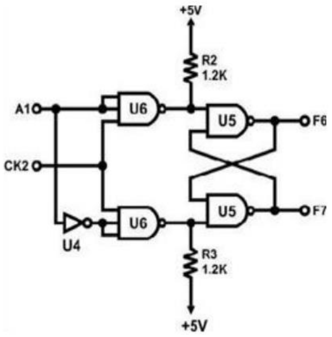


Figure 15: D Latch.

Figure 14 : The circuit connection

INPUT		OUTPUT
CK2	**A1**	**F6**
0	**0**	**0**
0	**1**	**0**
⎍	**0**	**0**
⎍	**1**	**1**

Table 5:D Latch Data

The D latch operates based on the rising edge (positive edge) of the clock signal (CK2). Here's an analysis of the data for the D latch: When CK2 is 0 (not at the rising edge): Regardless of the value of A1, the output F6 remains at 0. This is because D latches are level-sensitive, and their outputs only change on the rising edge of the clock. On the positive edge of CK2: When A1 is 0, the output F6 remains at 0. This behavior corresponds to holding or maintaining the previous state. When A1 is 1, the output F6 becomes 1. This behavior represents setting the latch to the logic high state. The D latch is essentially a data latch that captures the input (A1) value and stores it when the clock signal (CK2) transitions from 0 to 1 (rising edge). It holds the stored value until the next rising edge. When A1 is 0 at the rising edge, it retains the previous value, and when A1 is 1 at the rising edge, it updates the latch to the new value.

— Constructing JK latch with RS latch

This experimental section used U6 (7442) from Block Decoder 2 of Module IT-3004. Data Switches SW0, SW1, SW2, and SW3 were each connected to Inputs A, B, C, and D, respectively. Ten outputs were also connected to their corresponding Indicators L0 to L9. The input and output logic states were observed for the output states at L0 to L9 and recorded in the provided table.

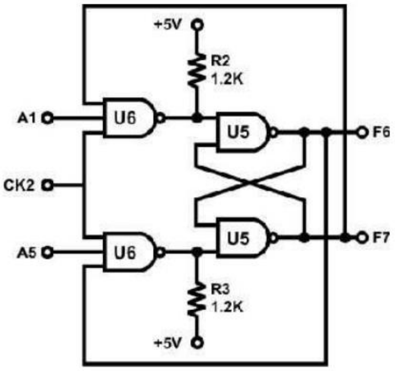


Figure 17: JK Latch.

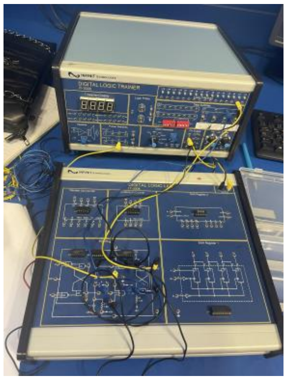


Figure 16: The Circuit Connection

INPUT			OUTPUT	Status
CK2	**A1**	**A5**	**F6**	
⎍	**1**	**0**	**1**	Set
⎍	**0**	**0**	**1**	No change
⎍	**1**	**1**	**0**	Reset
⎍	**1**	**0**	**1**	Set
⎍	**0**	**0**	**1**	No change
⎍	**0**	**1**	**0**	Reset
⎍	**1**	**1**	**1**	Toggle

Table 6:JK Latch.

The "Status" column reflects the behavior of the JK latch at each clock transition based on the inputs and the latch's internal state. The JK latch can be in one of three states: "Set," "Reset," or "Toggle," depending on the inputs and the previous state.

— Constructing JK Flip-flop with master- slave RS latches

The master-slave flip-flop eliminates all timing issues by connecting two SR flip-flops. The first flip-flop serves as the "Master" circuit and triggers on the clock pulse's leading edge, while the second flip-flop serves as the "Slave" circuit and triggers on the clock pulse's falling edge. As a result, the two sections—the master section and the slave section—are turned on during the clock signal's opposing half-cycles. Let's complete this part by taking these actions.

The circuit was built using the IT-3008 module, which facilitated various connections and interfaces. CK1 was connected to SWA A output, while J was linked to SW1, and K was connected to SW0. The outputs F1, F2, F6, and F7 were connected to LEDs or indicators labeled L3, L2, L1, and L0, respectively.

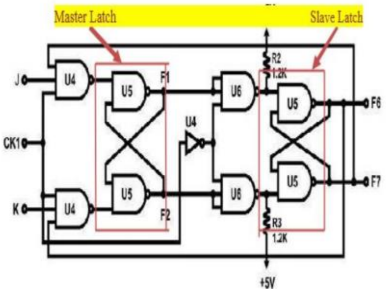


Figure 19: 2-to-1 Multiplexer.

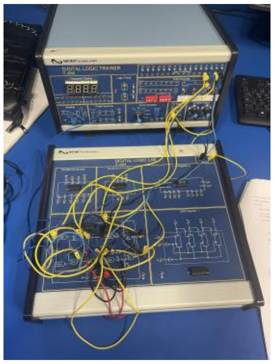


Figure 18:The Circuit Connection

INPUT			OUTPUT			
CK2	**K**	**J**	**F1**	**F2**	**F6**	**F7**
⎍	**0**	**0**	**0**	**1**	**0**	**1**
⎍	**0**	**1**	**1**	**0**	**1**	**0**
⎍	**1**	**0**	**0**	**1**	**0**	**1**
⎍	**1**	**1**	**1**	**0**	**1**	**0**
⎍	**1**	**1**	**0**	**1**	**0**	**1**

Table 7: JK Flip-Flop.

The provided data consists of an input signal labeled "CK2" and corresponding output values (F1, F2, F6, F7) for different states of CK2 (0 or 1). The data suggests that when CK2 is 0, F1 and F6 are 0, and F2 and F7 are 1, while when CK2 is 1, F1 and F6 are 1, and F2 and F7 are 0. However, the specific logic or context behind these relationships is not clear from the data alone. To fully understand the significance of this data, additional information about the purpose and meaning of CK2 and the associated outputs is needed.

❖ Registers

— Constructing Shift Register with D Flip-Flops

The circuit was constructed using a Block Shift Register 1 module from IT-3008. The connections were made as follows: B (clear) to SW0, A (input) to SW1, CK to SWA A output, and F1, F2, F3, and F4 to indicators labeled L1, L2, L3, and L4, respectively. Initially, SW0 was set to "0" to clear the B input, followed by setting SW0 to "1."

To observe the behavior of the circuit in a passive manner, the following input sequence for A (input) was followed: At A = "1," a clock signal was sent in from SWA.

At A = "0," another clock signal was sent in from SWA.

At A = "1," once again, a clock signal was sent in from SWA.

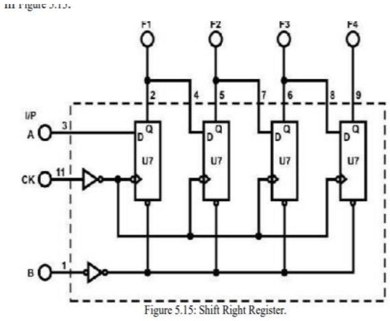


Figure 5.15: Shift Right Register.

Figure 21: Shift Right Register

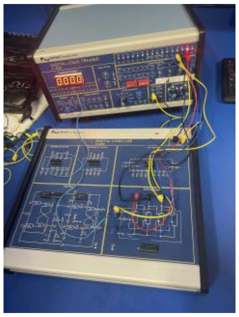
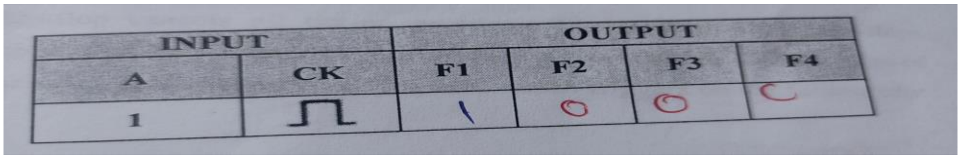
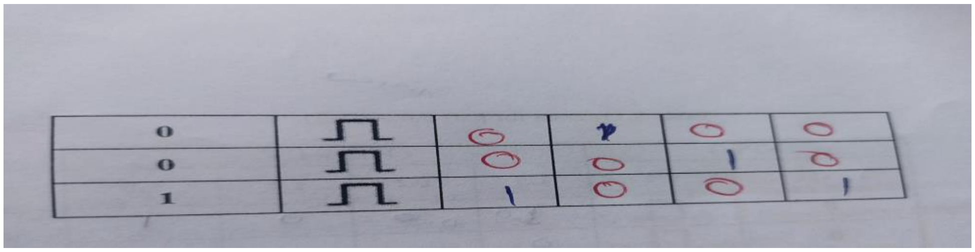


Figure 20: The circuit when B is change to 1 and A is 1





INPUT		OUTPUT			
A	CK	F1	F2	F3	F4
1	⎍	1	0	0	0
0	⎍	0	1	0	0
0	⎍	0	0	1	0
1	⎍	1	0	0	1

Figure 22: The Data

A register with D flip-flops is a crucial component in digital electronics for storing binary data, and it can be designed to support parallel or serial loading, depending on the specific application's requirements.

— 4-Bit Shift Register with serial and parallel load

The configuration of the IT-3008 was employed, incorporating a 4-Bit Shift Register with both serial and parallel synchronous operating modes. Connections were established as follows: Inputs A, B, C, and D were linked to switches SW0, SW1, SW2, and SW3, respectively, while Outputs F1, F2, F3, and F4 were connected to LEDs L0, L1, L2, and L3. Serial Input B1 was connected to DIP2.0, and MODE A1 was connected to DIP2.1. The clock input, CK (C1), was connected to a clock generator generating a TTL-level 1Hz output, with data at Input, LOAD (D1), was connected to the same 1Hz clock generator. A1 was set to "1" to enable serial mode operation, allowing for the adherence to the input sequences for A, B, C, and D from Table 8, and subsequent observations and recordings of the outputs (F1, F2, F3, and F4) were executed with meticulous precision.

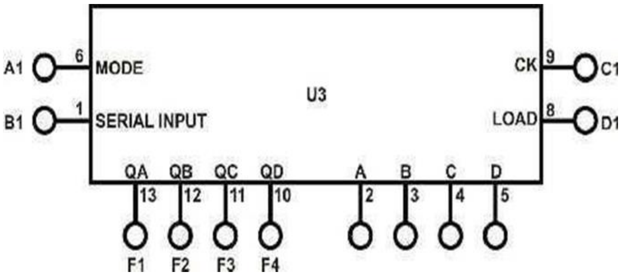


Figure 24:shift register with serial and parallel load

Figure 23: The circuit Connection

INPUT		OUTPUT			
A1	**C1**	**L3**	**L2**	**L1**	**L0**
0	⎍	**0**	**0**	**0**	**1**
0	⎍	**0**	**0**	**1**	**1**

0	⎍	**0**	**1**	**1**	**1**
1	⎍	**0**	**1**	**1**	**1**

Table 8:shift register with serial and parallel load

The provided data appears to relate to a shift register with the potential for both serial and parallel loading. It consists of two input signals, A1 and C1, and four output signals, L0, L1, L2, and L3. When both A1 and C1 are set to 0, the shift register appears to be in a parallel load mode, where data is loaded simultaneously into the register's outputs. However, the exact details of this parallel loading operation, such as the source of the data and the clocking mechanism, are not specified in the given data. More context and information about the behavior of A1 and C1, as well as the complete operation of the shift register, would be necessary for a comprehensive understanding of its functionality.

Input					**Output**			
D1	**D**	**C**	**B**	**A**	**L3**	**L2**	**L1**	**L0**
⎍	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**
⎍	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**
⎍	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**0**
⎍	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**1**
⎍	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**1**

Table 9: Shift register with serial and parallel load B

The provided data corresponds to a shift register capable of both parallel data loading and serial shifting. The inputs D1, D, C, B, and A represent the data loaded into the register, while the outputs L3, L2, L1, and L0 reflect the state of the register after each input change.

Initially set to all zeros, indicating a reset state, subsequent changes in the input values demonstrate parallel data loading operations, with the register rapidly updating its content to reflect the input values. However, the specific clocking or control mechanism responsible for the serial shifting operation is not explicitly detailed in the data. To fully grasp the shift register's behavior, further information regarding the clocking and control signals is necessary.

❖ Counters

— 2-bit Synchronous Counter

The IT-3007 module was employed for the implementation of a 2-bit synchronous counter. This involved connecting the module's +5V input to the +5V output of a fixed power supply and its GND to the GND output of the same power supply. The CLK input was linked to a pulse switch designated as SWA. Outputs Q1 and Q0 were connected to indicator lamps denoted as L1 and L2, respectively, for the purpose of monitoring the binary outputs. Clock pulses were applied to the CLK input, and the resulting binary outputs were duly observed and recorded. Furthermore, the counter outputs Q1 and Q0 were utilized to drive a seven-segment display, enabling the binary values to be prominently displayed.

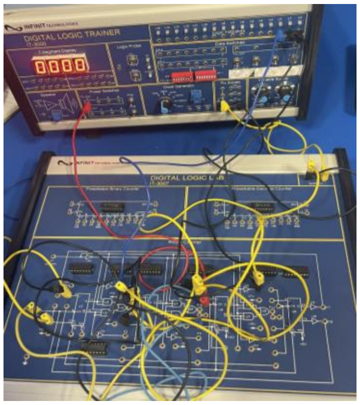
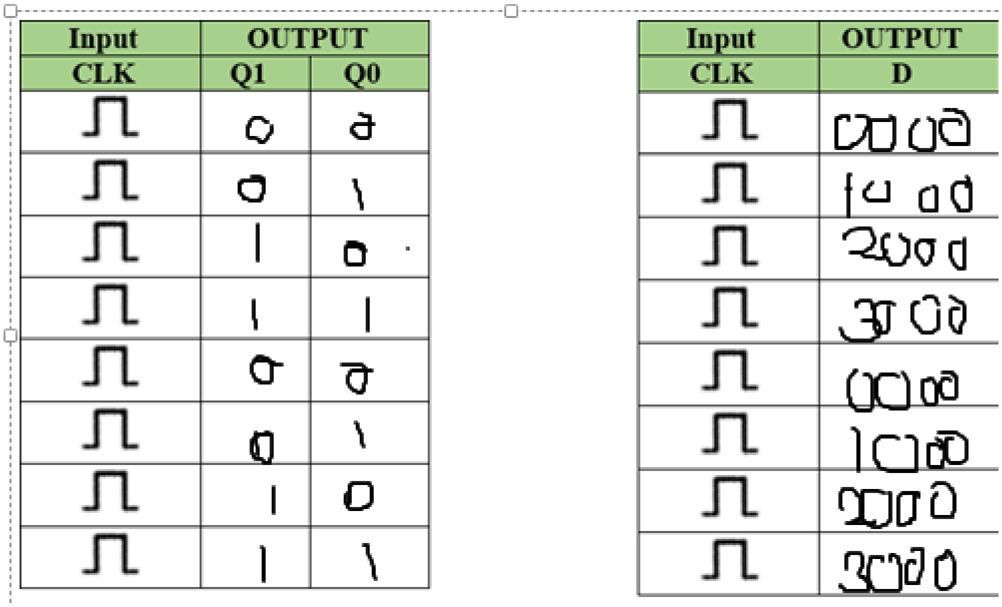


Figure 25:2-bit Synchronous Counter



Input	OUTPUT	
CLK	Q1	Q0
⎍	0	0
⎍	0	1
⎍	1	0
⎍	1	1
⎍	0	0
⎍	0	1
⎍	1	0
⎍	1	1

Input	OUTPUT
CLK	D
⎍	0000
⎍	1000
⎍	2000
⎍	3000
⎍	0000
⎍	1000
⎍	2000
⎍	3000

Table 10:2-bit Synchronous Counter

— 3-bit (divide-by-eight) Ripple Counter

The implementation of a 3-bit (divide by eight) Ripple counter, as depicted in Figure 5.18, was executed using the IT-3007 module. Connections were established by linking the CLK input to a pulser switch. Furthermore, the counter outputs Q2, Q1, and Q0 were connected to indicator lamps for the purpose of monitoring their respective states. Clock pulses were applied to the CLK input to initiate its operation. Subsequently, the counter outputs Q2, Q1, and Q0 were directed to a seven-segment display, enabling the counter's binary values to be prominently displayed.

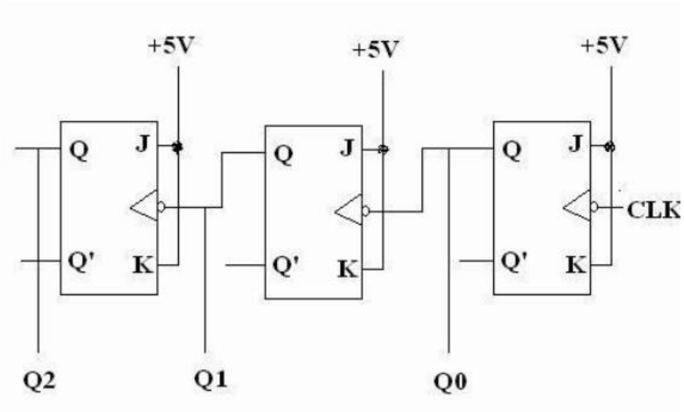


Figure 26: 3-bit Ripple Counter

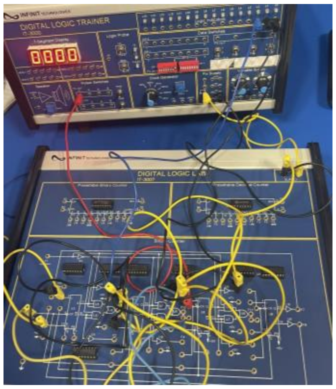
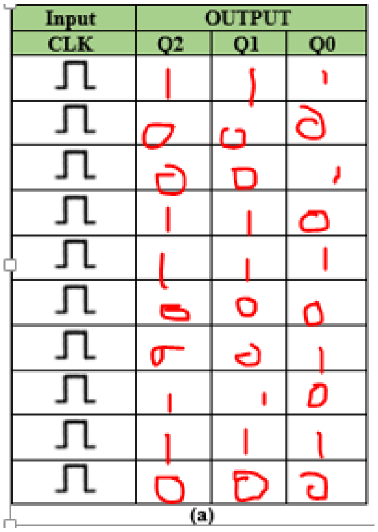


Figure 27: The Circuit Connection



Input	OUTPUT		
CLK	Q2	Q1	Q0
⎍	1	1	1
⎍	0	0	0
⎍	0	0	1
⎍	1	1	0
⎍	1	1	1
⎍	0	0	0
⎍	0	0	1
⎍	1	1	0
⎍	1	1	1
⎍	0	0	0

(a)

Table 11:3-bit (divide-by-eight) Ripple Counter

A 3-bit ripple counter, also known as a divide-by-eight counter, is a digital circuit used in electronics to count binary numbers. It consists of three flip-flops connected in a cascading fashion, where the output of each flip-flop serves as the clock input for the next one. When the counter is triggered, it starts from 000 and progresses sequentially through binary numbers up to 111. Since it has three flip-flops, it can represent eight unique states (2^3 = 8), hence the name "divide-by-eight." Each clock pulse causes the counter to advance by one count, making it suitable for various applications like frequency division, time delay generation, or addressing multiple memory locations in microcontrollers and digital systems. However, it's important to note that the ripple counter suffers from propagation delays due to its cascaded design, which can limit its use in high-speed applications.

4. Conclusion

Through this experiment, we've achieved our primary objectives comprehensively. We've not only mastered the art of rectifying connection errors but also fine-tuned our teamwork skills. Moreover, we've gained a deep understanding of the disparities between combinational and sequential logic circuits and have explored the practical applications of diverse memory units. Hands-on circuit construction and connection have solidified our knowledge. Furthermore, we've delved into the nuances of synchronous and asynchronous counters. This experiment has also equipped us with the valuable skill of constructing counters using JK flip-flops.

5. References

[1] [Online]. Available: [1] https://www.electronics-tutorials.ws/sequential/seq_1.html.

Accessed on 24-08-2023 at 12:01AM.

[2] [Online]. Available: [2] https://electronics.stackexchange.com/questions/163164/sr-flip-flop-nor-or-nand.

Accessed on 20-08-2023 at 10:01PM.

[3] [Online]. Available: [3] https://ritaj.birzeit.edu/bzu-msgs/attach/2357758/Digital+lab+manual+2023.pdf.

Accessed on 23-08-2023 at 02:01AM.

[4] [Online]. Available: [4] https://www.geeksforgeeks.org/flip-flop-types-their-conversion-and-applications/.

Accessed on 29-08-2023 at 12:36 AM.

[5] [Online]. Available: [6] https://curtsinger.cs.grinnell.edu/teaching/2018F/CSC211/labs/sequential-logic/.

Accessed on 23-08-2023 at 5:01PM.

[6] [Online]. Available: [6] https://datatrained.com/post/register-in-digital-electronics/.

Accessed on 20-08-2023 at 6:01PM.

[7] [Online]. Available: [7] https://www.researchgate.net/figure/Block-diagram-of-4-bit-PIPO-Shift-register-using-D-Flip-Flop_fig3_344366614.

Accessed on 20-08-2023at 5:50P M.

[8] [Online]. Available: [8] https://www.bing.com/images/search?view=detailV2&ccid=suynipkC&id=C332E97B415623A9E378BAA58164DAA725F07C10&thid=OIP.suynipkC-gXD-ou7MOpoWgHaC2&mediaurl=https%3a%2f%2fwww.petervis.com%2fdictionary-of-digital-terms%2f4-bit-shift-register%2f4-bit-shift-r.

Accessed on 21-08-2023 at 11:030 PM.

[9] [Online]. Available: [1] https://www.tutorialspoint.com/digital_circuits/digital_circuits_decoders.htm.