Faculty of Engineering and Technology

Electrical and Computer Engineering Department

Digital Electronics And Computer Organization Laboratory

ENCS2110

Report #2

Experiment No. 5 – Sequential Logic Circuits

Prepared by :

Mohammed Jamil Saada **Number : 1221972**

Partner :

Laith Nimer **Number : 1213046**

Instructor : **Dr. Abualseoud Hanani**

Teaching assistant: **Eng. Hanan Awawdeh**

Section : **4**

Date : **11 / 04 / 2024**

Abstract

In this experiment, we will review our knowledge of digital systems and sequential logic circuits in general, and understanding the construction of digital latches, flip-flops, registers and counters especially. In this experiment we will construct SR latches with basic gates and ICs and its implementation and truth table. then, we will implement D latch which is development of the SR latch by eliminate the undefined condition in SR latch. After that, we will implement a D flip-flop using two D latches, in the flip-flop the output change occurs only at the clock edge while in the latches it occurs at the clock level. Then, we should construct registers. Digital systems use registers to hold binary entities. There is two type of shift registers: serial and parallel load. Finally, we will design digital counters which is a special-purpose register, there is two type of counters depends on having a common clock or not, Ripple and Synchronous counters. After we design these digital circuits practically, we can use their implementation to design more complex circuits such as n-bit register, n-digit counters, etc.

At the end of this lab, my theoretical background will be proved practically so, I will be able to construct, implement and design latch, flip-flop, register and counter digital circuits. In addition to construct more complex circuits using the small one.

Table of Contents

Table of Figures

List of Tables

1. Theory

1- Sequential Circuits

Sequential circuits are digital circuits that store and use the previous state information to determine their next state. Unlike combinational circuits, which only depend on the current input values to produce outputs, sequential circuits depend on both the current inputs and the previous state stored in memory elements. They are commonly used in digital systems to implement state machines, timers, counters, and memory elements and are essential components in digital systems design [1].

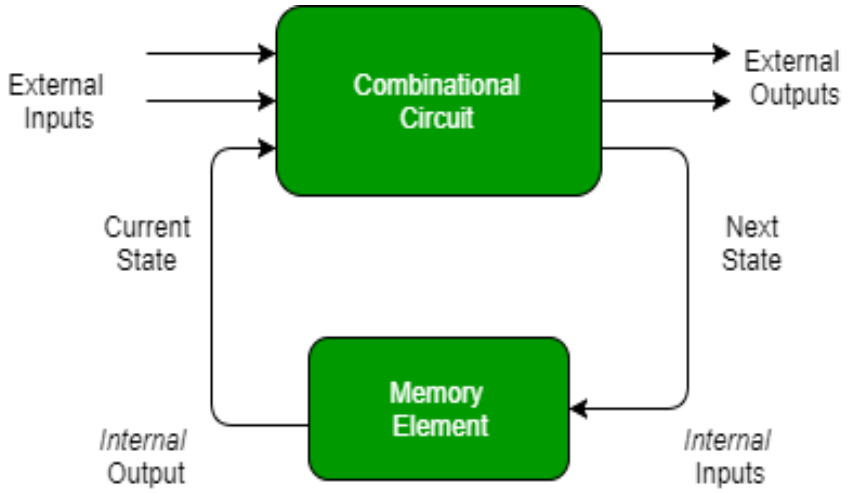

Figure 1 : Sequential Circuit

2- Latches

Latches are digital circuits that store a single bit of information and hold its value until it is updated by new input signals. They are used in digital systems as temporary storage elements to store binary information. Latches can be implemented using various digital logic gates, such as AND, OR, NOT, NAND, and NOR gates [2].

There are two types of latches :

1- The SR (Set-Reset) Latch

S-R (Set - Reset) Latches: S-R latches are the simplest form of latches and are implemented using two inputs: S (Set) and R (Reset). The S input sets the output to 1, while the R input resets the output to 0. When both S and R are at 1, the latch is said to be in an “undefined” state, the circuit in the Figure 2 below is SR Latch with NAND gates, note that this circuit is active low set / reset latch; that means the output Q goes to 1 when S (Set) input is 0 and goes to 0 when R (Reset) input is 0 [2].

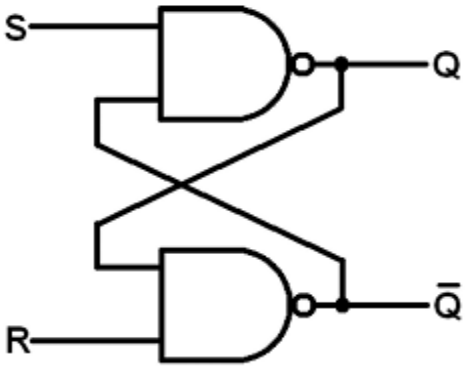

Figure 2 : SR Latch with NAND gate

INPUTS		OUTPUTS		State
S	R	Q	$\overline{Q}$	
1	0	0	1	RESET
1	1	0	1	No Change
0	1	1	0	SET
1	1	1	0	No Change
0	0	1	1	Invalid

Table 1 : SR Latch with NAND gate Truth table

2- The D Latch

D (Data) Latches : D latches are also known as transparent latches and are implemented using two inputs: D (Data) and a clock signal. The output of the latch follows the input at the D terminal as long as the clock signal is high. When the clock signal goes low, the output of the latch is stored and held until the clock become high. The D Latch was developed to eliminate the undefined condition of the indeterminate state in the SR latch [2].

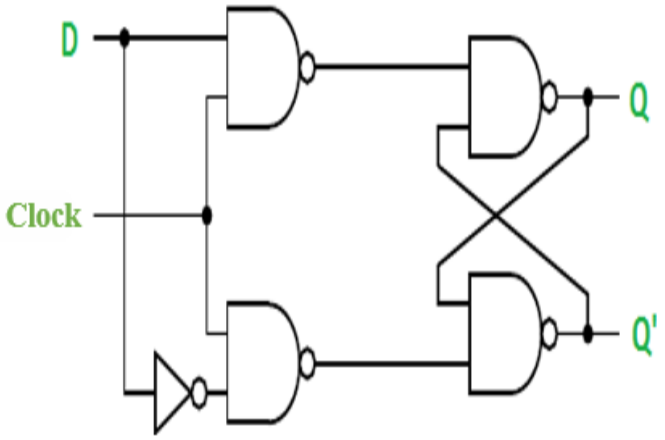

Figure 3 : D - Latch

INPUTS		OUTPUTS		State
Clock	D	Q_{n+1}	$\overline{Q}$	
0	X	Q_n	$\overline{Q}$	No Change
1	0	0	1	RESET
1	1	1	0	SET

Table 2 : D - Latch Truth table

3- Flip – Flops

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. Flip flop is popularly known as the basic digital memory circuit. It has its two states as logic 1 (High) and logic 0 (low) states. In the flip – flop the output change occurs only at the clock edge while in the latch it occurs at the clock level. A flip – flop can be implemented using two separate latches [3].

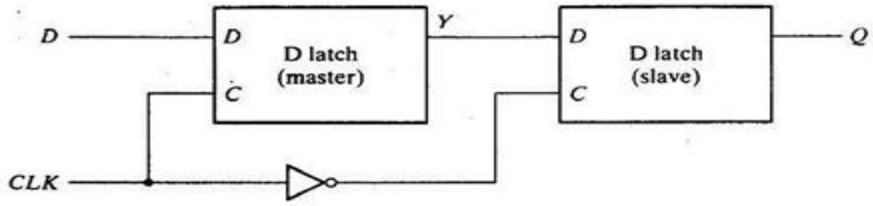

Figure 4 : D flip-flop implemented with two D latches

4- Registers

A register serves as a quick memory for accepting, storing, and sending data and instructions that the CPU will need right away. A register is a collection of flip-flops, Single bit digital data is stored using flip-flops. By combining many flip-flops, the storage capacity can be extended to accommodate a huge number of bits. We must utilize an n-bit register with n flip flops if we wish to store an n-bit word [4].

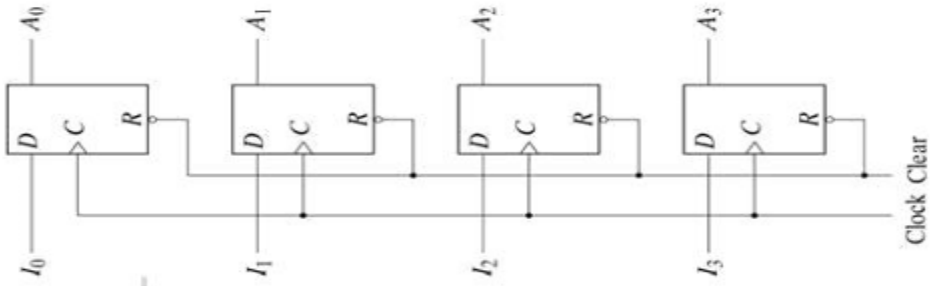

Figure 5 : 4-bit Register

Shift Register is a group of flip flops used to store multiple bits of data. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses [5].

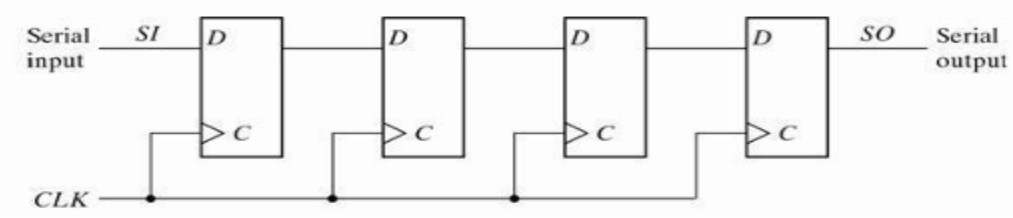

Figure 6 : 4-bit shift - right register

5- Counters

The counter is a special – purpose register, it is a register that goes through a prescribed sequence of states. The counters are classified into two categories : Ripple and Synchronous counters.

- Ripple Counters :

  is a cascaded arrangement of flip-flops where the output of one flip-flop drives the clock input of the following flip-flop. The number of flip flops in the cascaded arrangement depends upon the number of different logic states that it goes through before it repeats the sequence a parameter known as the modulus of the counter [6].

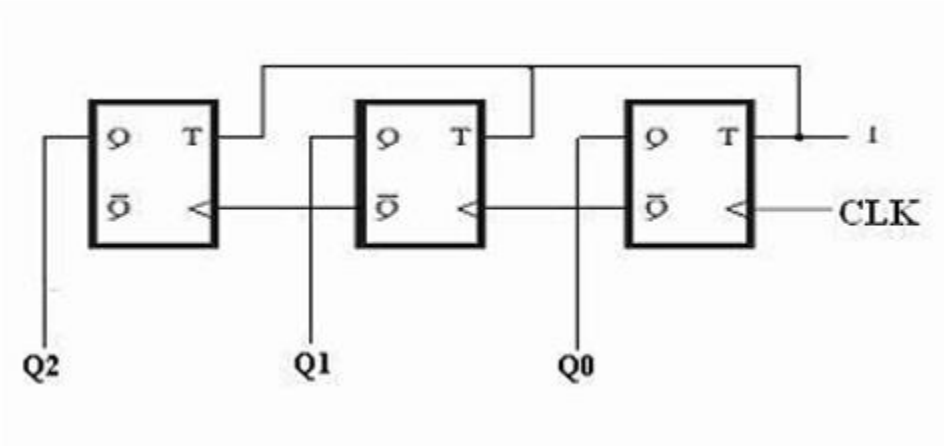

Figure 7 : 3-bit ripple counter

- Synchronous Counters :

  Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop. It is also called as parallel counter [7].

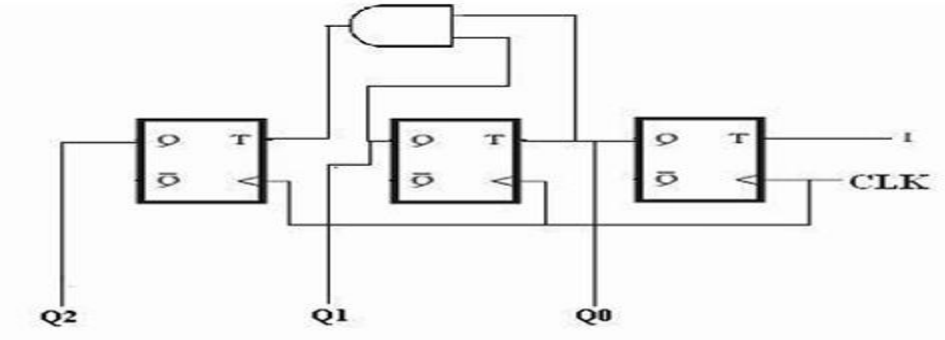

Figure 8 : 3-bit synchronous counter

2. Procedure and Discussion

2.1 Latches and Flip – Flops

2.1.1 Constructing SR latch with Basic Logic Gates

We connected the Circuit as it shown in the Figure 2.1 using IT-3008 module.

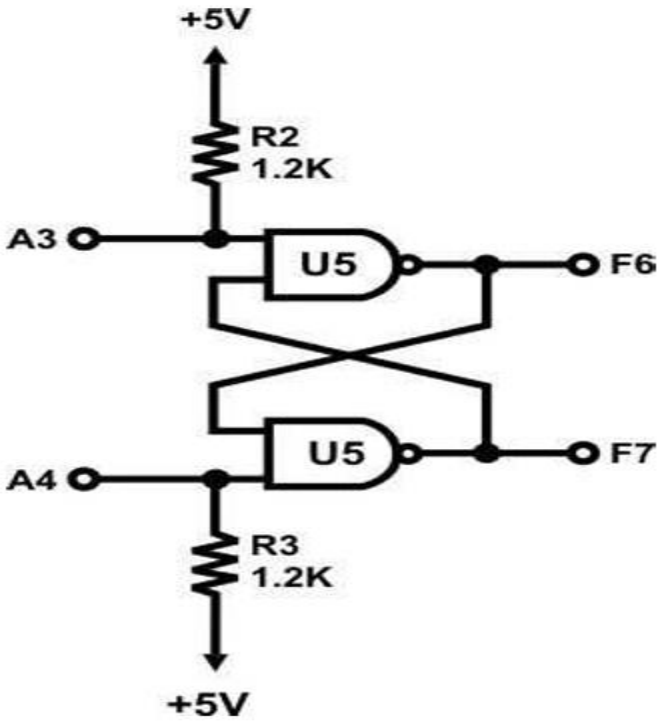

Figure 2. 1 IT-3008 SR latch block (Source: Lab Manual)

a) set module IT-3008 block SR latch. We will use U5 to construct the SR latch.
b) We connected the +5V of module IT-3008 to the +5V output of the fixed power supply IT-3000 and do the same for GND.
c) Then, We connected the inputs A3 and A4 to Data Switches SW1 and SW2 respectively. After that we connected the outputs F6 and F7 to Logic Indicators (LEDs) L1 and L2 respectively.
d) The results are shown in Table 2.1 :

INPUTS		OUTPUTS	
A3	A4	F6	F7
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	1

Table 2. 1 : Results of SR Latch

e) The Circuit connected as following Figure 2.2 :

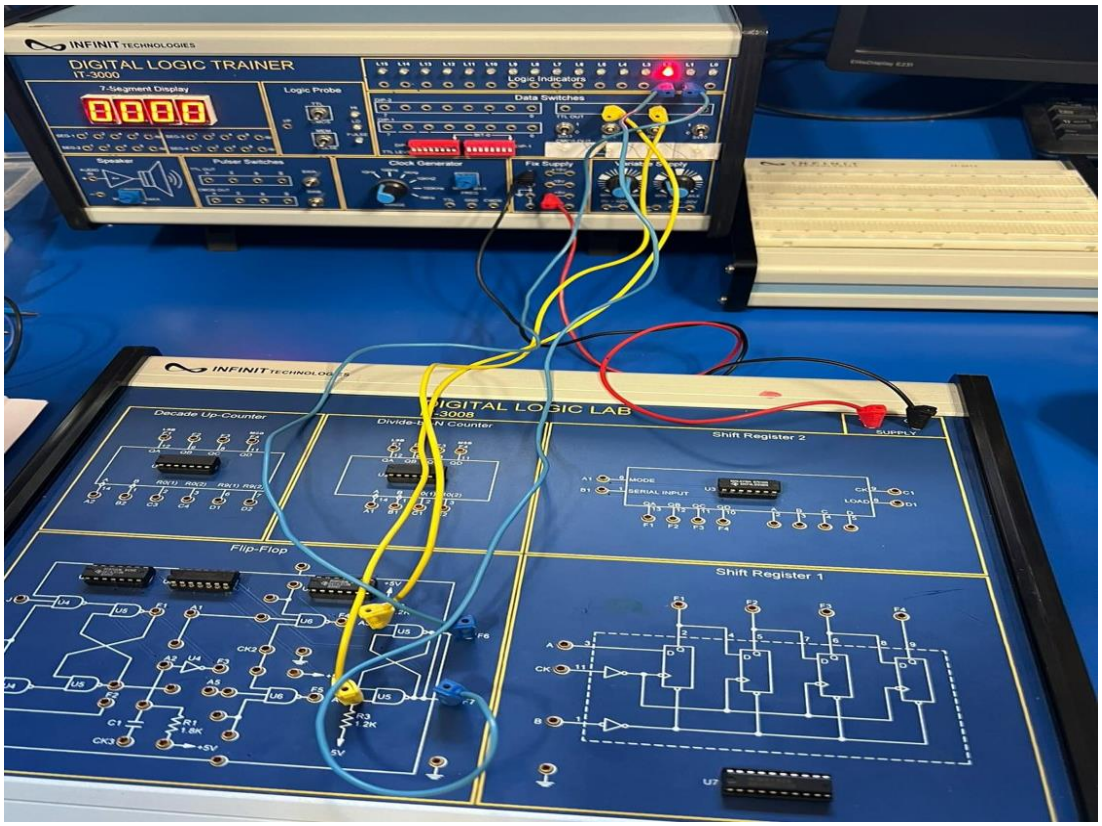

Figure 2. 2 SR latch circuit connection

Discussion :

The sequential circuit that shown above is SR latch circuit. It is one type of memory. It has 4 states, one of them is indeterminate state, as we show in the truth table of this circuit we have 2 inputs S (A3) -SET- and R (A4) -RESET- and this circuit work in active low mode; that means when S = 0 , the output goes to 1, when R = 0, the output goes to 0, when S = R = 1, the output doesn't change, and when S = R = 0, this is indeterminate state (Invalid) because both outputs Q and Q' will be 1 and this is contradiction because it's impossible to be Q = Q' = 1.

2.1.2 Constructing SR latch with control input

We connected the Circuit as it shown in the Figure 2.3 using IT-3008 module.

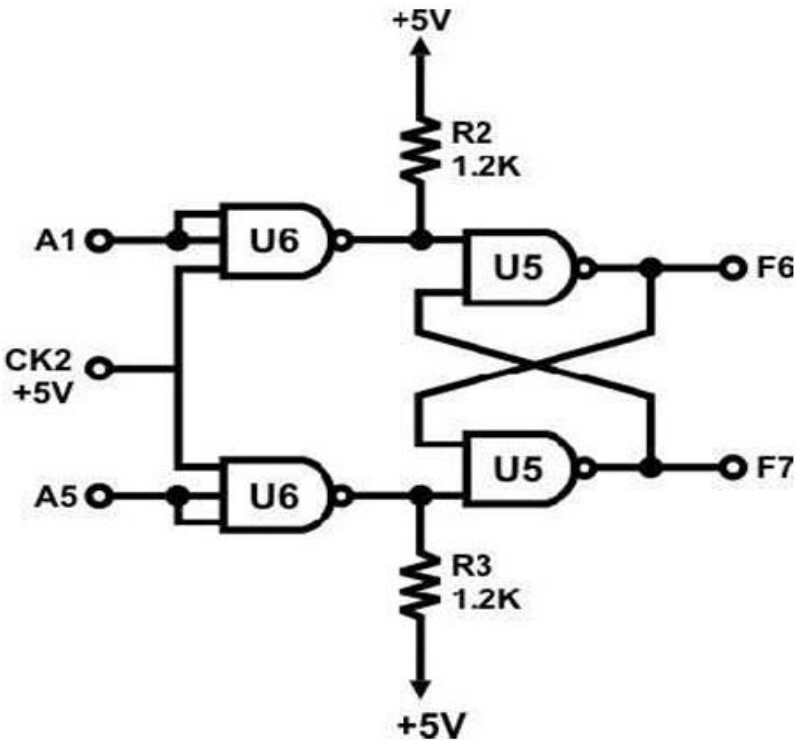

Figure 2. 3 : SR latch with control input (Source: Lab Manual)

a) We connected inputs A1 and A5 to Data Switches SW1 and SW2 respectively.

b) By following the inputs sequence, the results are shown in Table 2.2:

INPUTS		OUTPUTS	
A1	**A5**	**F6**	**F7**
0	**0**	**0**	**1**
0	**1**	**0**	**1**
1	**0**	**1**	**0**
1	**1**	**1**	**1**

Table 2. 2 : Results of SR latch circuit with control input

c) The circuit connected as following Figure 2.4

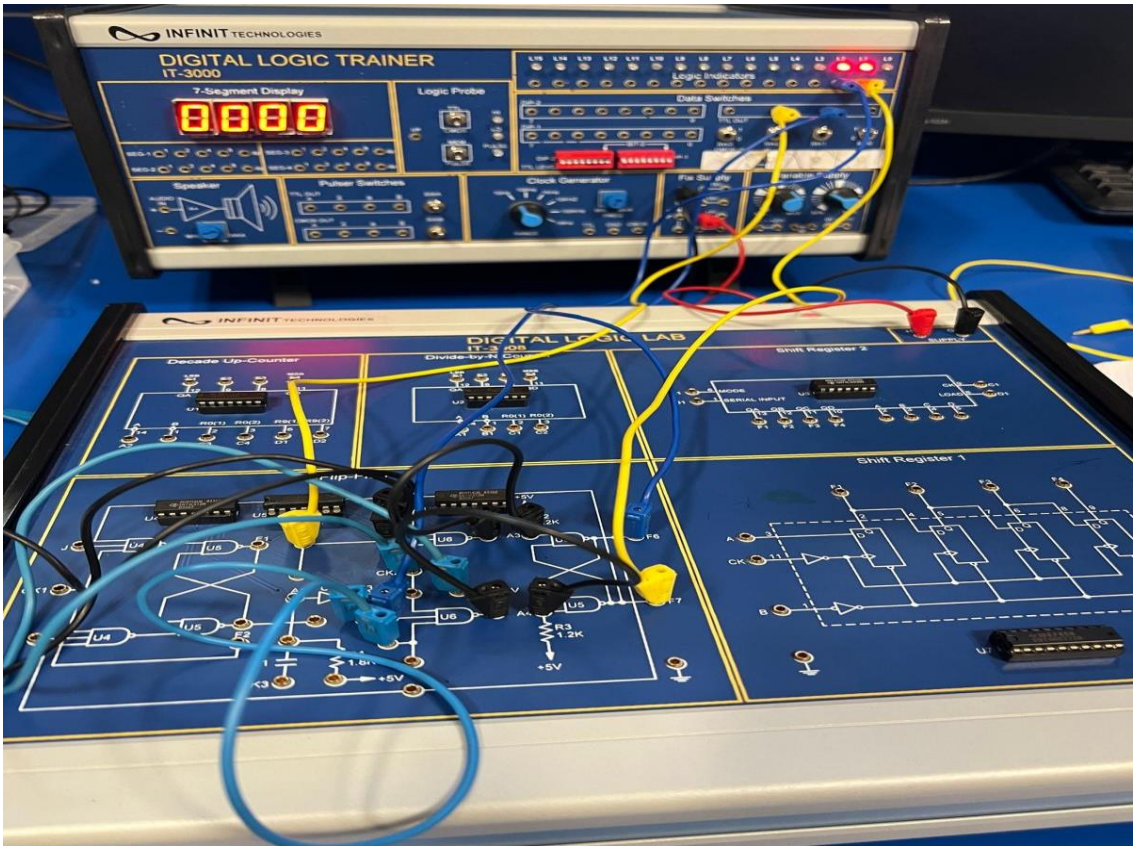

Figure 2. 4 : SR latch with control input connection

Discussion :

In this section we design SR latch with control input. This circuit has two inputs A1 (S) -SET- and A5 (R) -RESET- and this inputs work in active high voltage mode; that means when A1 (S) = 1 , and A5 (R) = 0, the output Q goes to 1 [SET], when A1 (S) = 0, and A5 (R) = 1, the output Q goes to 0 [RESET], when both inputs A1 (S) = A5 (R) = 0 the output won't change, and when the both inputs A1 (S) = A5 (R) = 1, the output will be invalid (indeterminate) because the output Q and its complement will be 1 and this is impossible.

2.1.3 Constructing D latch with SR latch

We connected the Circuit as it shown in the Figure 2.5 using IT-3008 module.

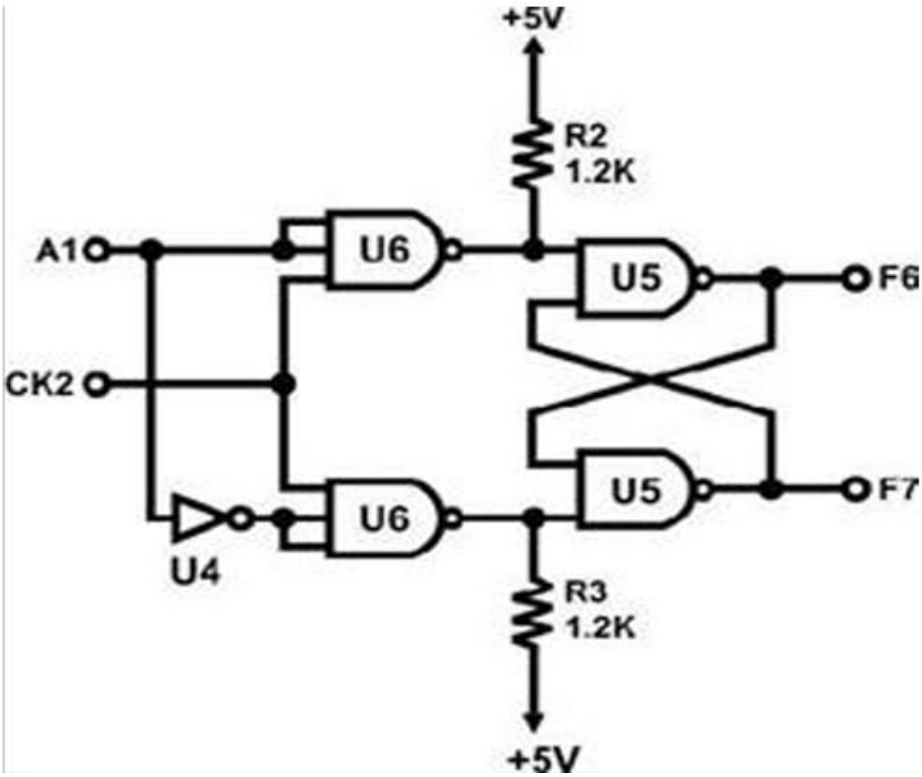

Figure 2. 5 : D Latch (Source : Lab Manual)

a) We connected inputs A1 to Data Switch SW1, and clock CK2 to Switch SWA A, then connected the output F6 to logic indicator L1.

b) By following the inputs sequence, the results are shown in Table 2.3:

INPUTS		OUTPUT
CK2	A1	F6
0	0	1
0	1	1
⎍	0	0
⎍	1	1

Table 2. 3 : Results of D latch circuit

c) The circuit connecting as following Figure 2.6

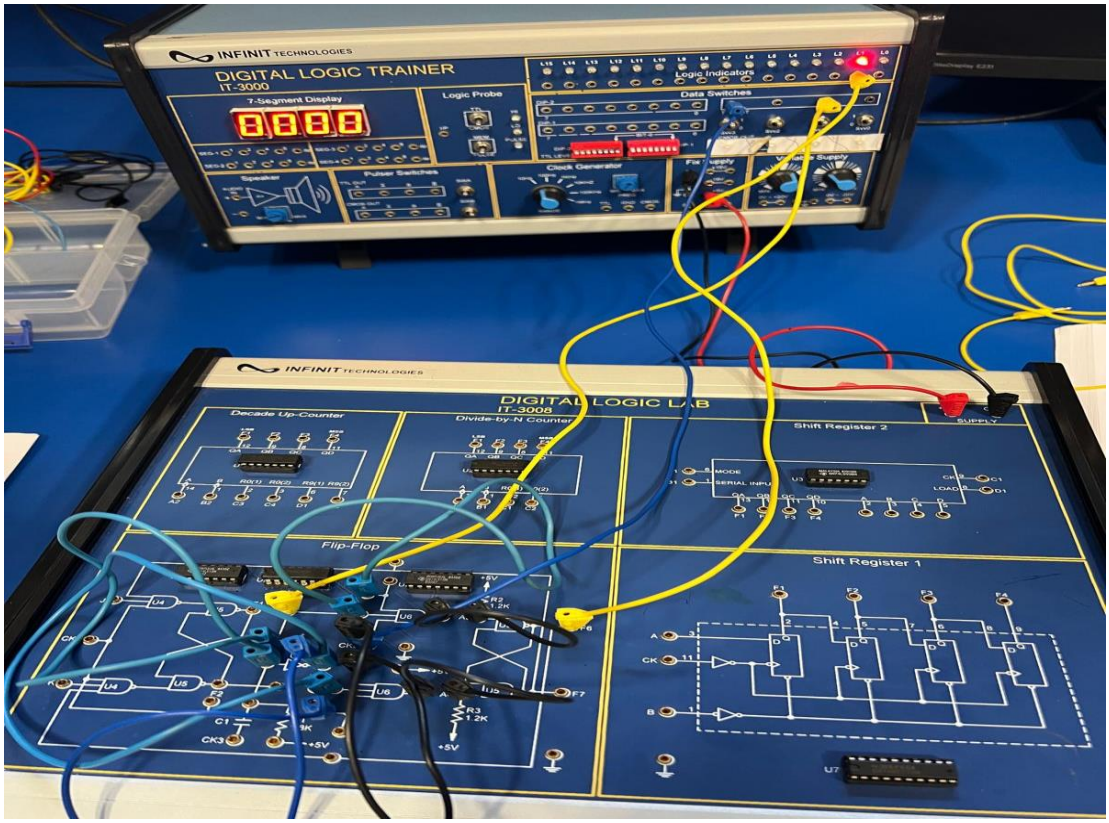

Figure 2. 6 : D latch circuit connection

Discussion :

In this section, we implemented D latch using SR latch with control input, by connected the two inputs of SR latch with one input, so connect S with new input (D) and connect R with its complement (D'). So D latch has two inputs D and clock, when clock = 0, the output Q won't change. But when clock = 1, the output depends on value of input D; that means when clock = 1 and D = 0 the output Q goes to 0, and when clock = 1 and D = 1 the output Q goes to 1. This latch solve the SR latch problem when the two inputs = 1, so in D latch we don't have this case, then there is no indeterminate state.

2.1.4 Constructing JK latch with SR latch

We connected the Circuit as it shown in the Figure 2.7 using IT-3008 module.

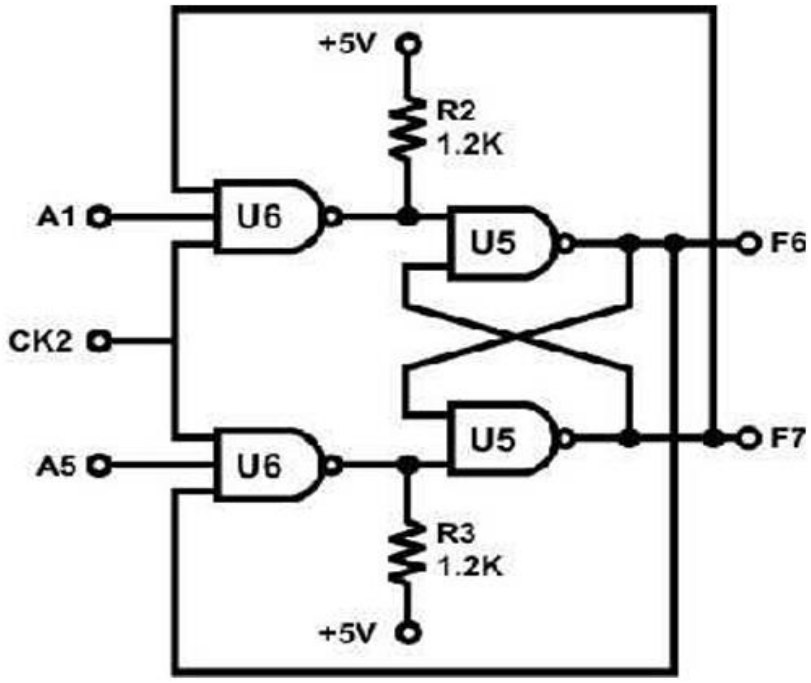

Figure 2. 7 : JK latch (Source : Lab Manual)

a) We connected clock CK2 to Data Switch SWB B, then connected the inputs A1 and A5 to Data Switches SW0 and SW1 respectively. After that, we connected the output F6 to logic indicator L1.

b) By following the inputs sequence, the results are shown in Table 2.4:

INPUTS			OUTPUT	STATUS
CK2	A1	A5	F6	
⎍	1	0	1	SET
⎍	0	0	1	No Change
⎍	1	1	0	Toggles
⎍	1	0	1	SET
⎍	0	0	1	No Change
⎍	0	1	0	RESET
⎍	1	1	1	Toggle

Table 2. 4 : Results of JK Flip-flop

c) The circuit connecting as following Figure 2.8

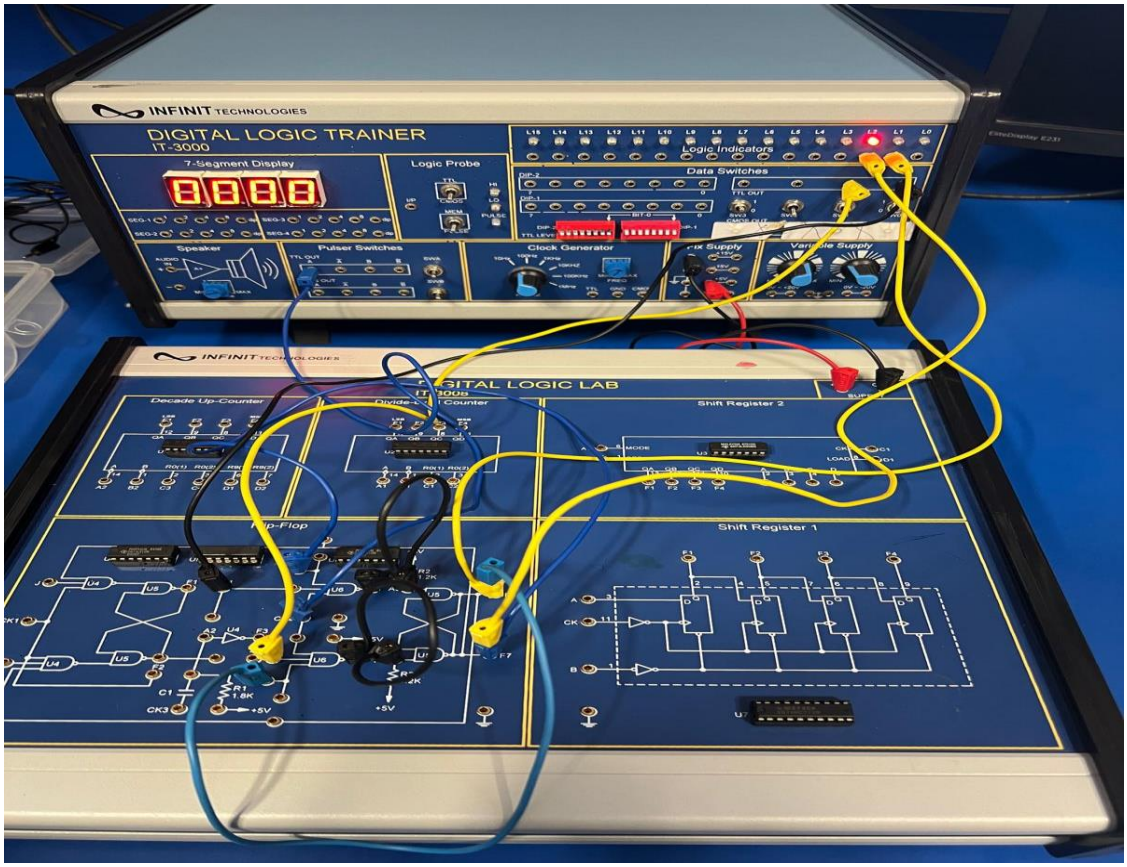

Figure 2. 8 : JK Flip-flop circuit connection

Discussion :

As we show in the circuit above, we implemented the JK flip-flop. This circuit has four cases depends on the value of J and K. in the circuit above A1 is J and A5 is K. when J = 1 and K = 0 the output Q goes to 1 [SET], when J = 0 and K = 1 the output Q goes to 0 [RESET], when J = K = 1 the output doesn't change, and when J = K = 1 the output Q goes to it's complement [Toggles].

2.1.5 Constructing JK Flip – flop with master – slave SR latches

We connected the Circuit as it shown in the Figure 2.9 using IT-3008 module.

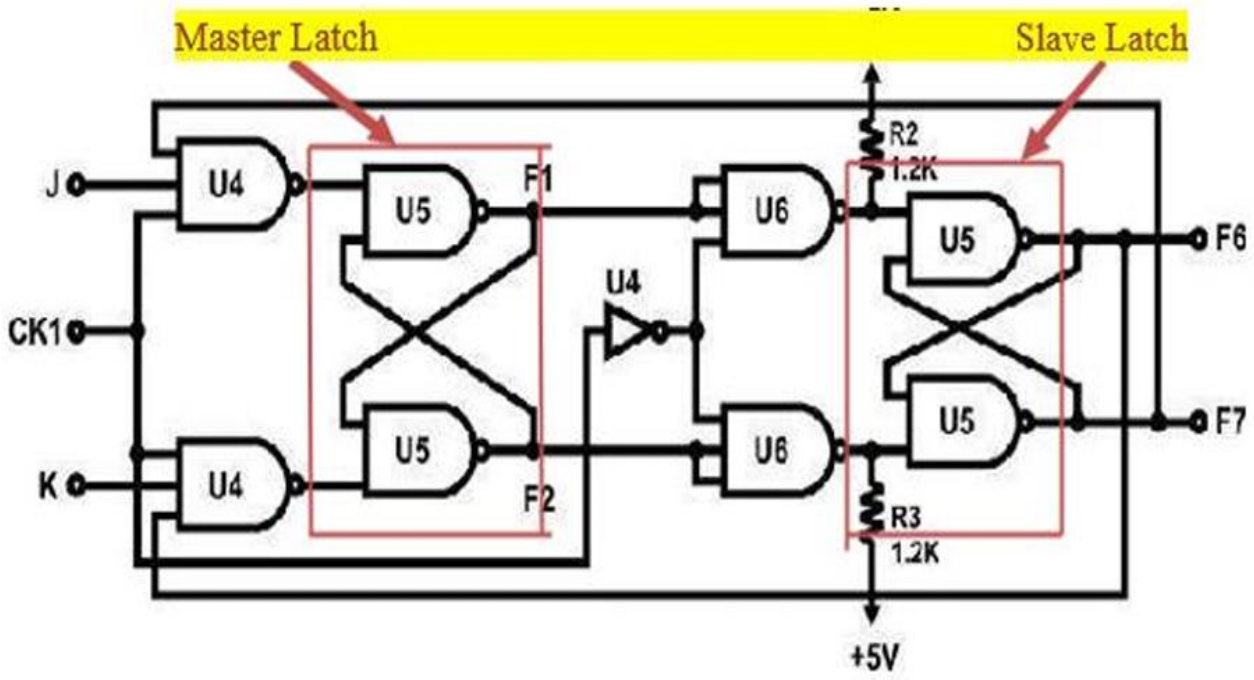

Figure 2. 9 : JK Flip – flop with master-slave SR latches (Source : Lab Manual)

a) First, we connected clock CK1 to Data Switch SWA A, then connected the inputs J and K with the Data Switches SW1 and SW0 respectively.

b) After that, we connected the outputs F1, F2, F6 and F7 to logic indicators (LEDs) L3, L2, L1 and L0 respectively.

c) By following the inputs sequence, the results are shown in Table 2.5:

INPUTS			OUTPUTS			
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 2. 5 : Results of JK Flip-flop with master-slave SR latches

d) The circuit connecting as following Figure 2.10

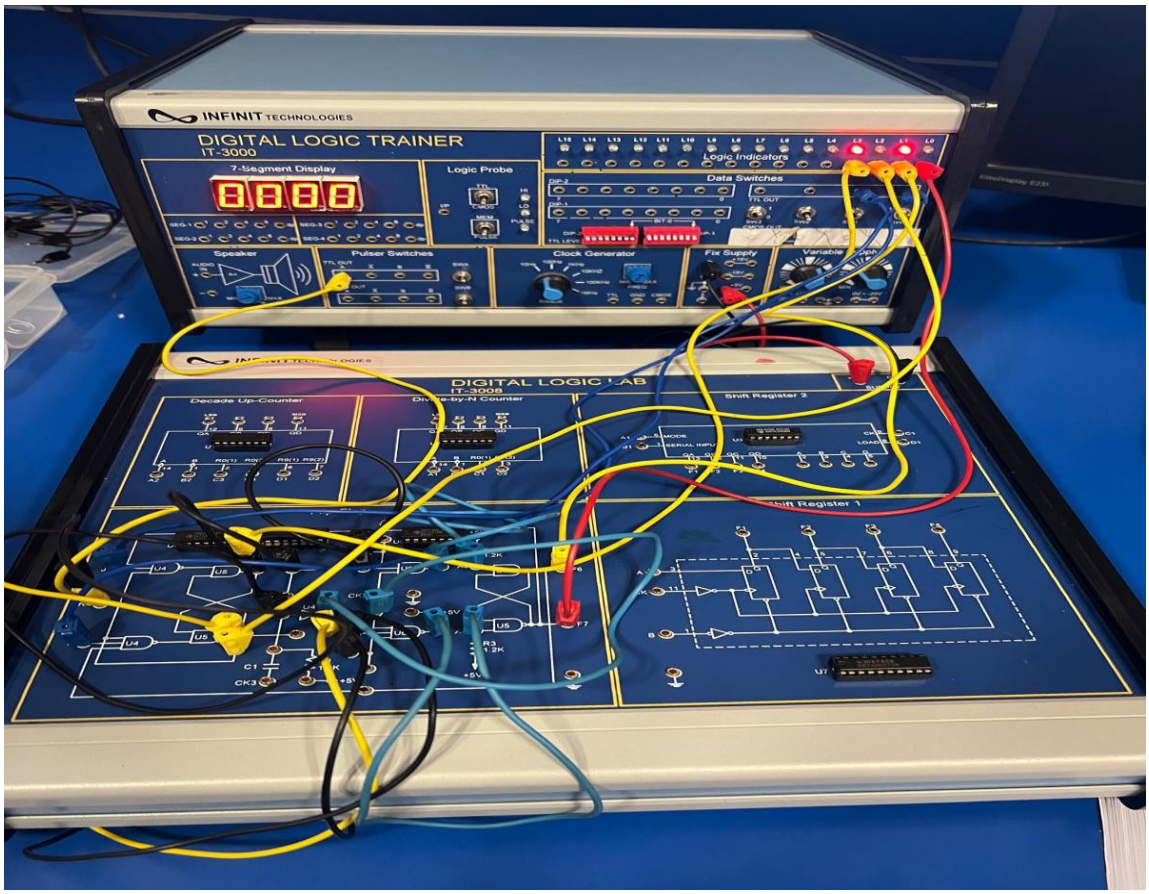

Figure 2. 10 : JK Flip-flop with master-slave SR latches connection

Discussion :

In this section, we implemented JK Flip – flop with master – slave SR latches, this implementation of flip – flops cancels all timing problems by using two SR flip – flops connected. So, the first flip flop acts as the “Master” circuit, that triggers on the leading edge of the clock pulse, while the other acts as the “Slave” circuit, which triggers on the failing edge of the clock pulse. The results of these two sections [the Master section and the Slave section] being enabled during opposite half – cycles of the clock signals.

2.2 Registers

2.2.1 Constructing Shift Register with D Flip - Flops

We connected the Circuit as it shown in the Figure 2.11 using IT-3008 module.

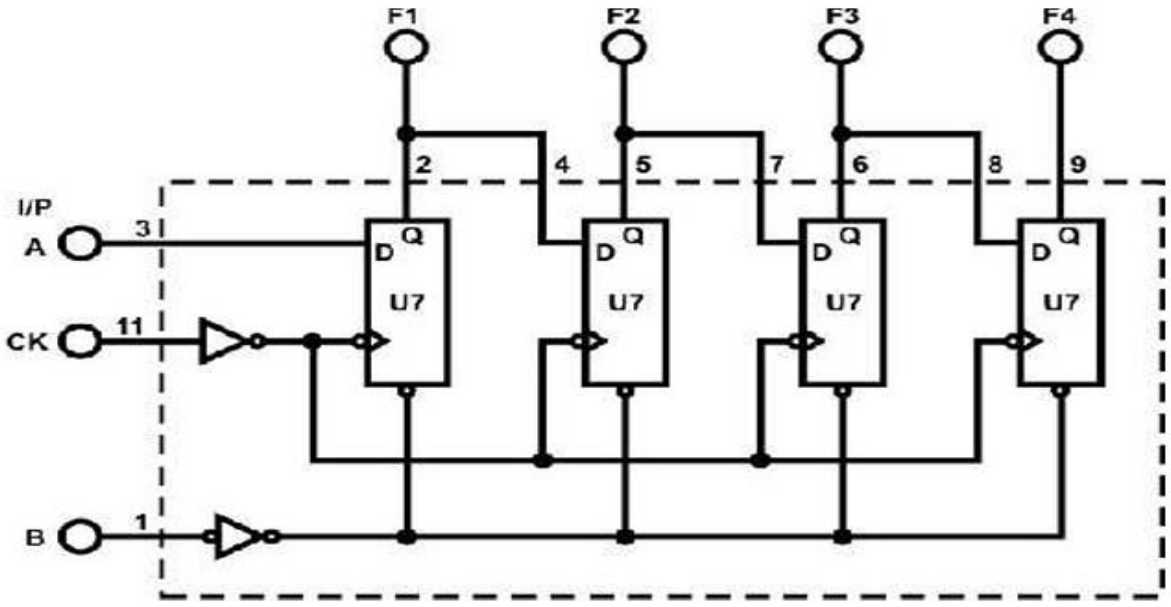

Figure 2. 11 : Shift - Right Register (Source : Lab Manual)

a) We connected the +5V of module IT-3008 to the +5V output of the fixed power supply, and do the same for the Ground (GND).

b) Then, we connected B (clear) to Data Switch SW0, A (I/P) to another Data Switch SW1, after that we connected the clock signals CK to Data Switch SWA A.

c) Then, We connected the outputs F1, F2, F3 and F4 to logic indicators (LEDs) L1, L2, L3 and L4 respectively.

d) We set SW0 to '0' to clear B, then set it to '1'.

e) Then, the results are shown in the Table 2.6 below :

INPUTS		OUTPUTS			
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

Table 2. 6 : Results of Shift – Right Register

f) The circuit connecting as following Figure 2.12

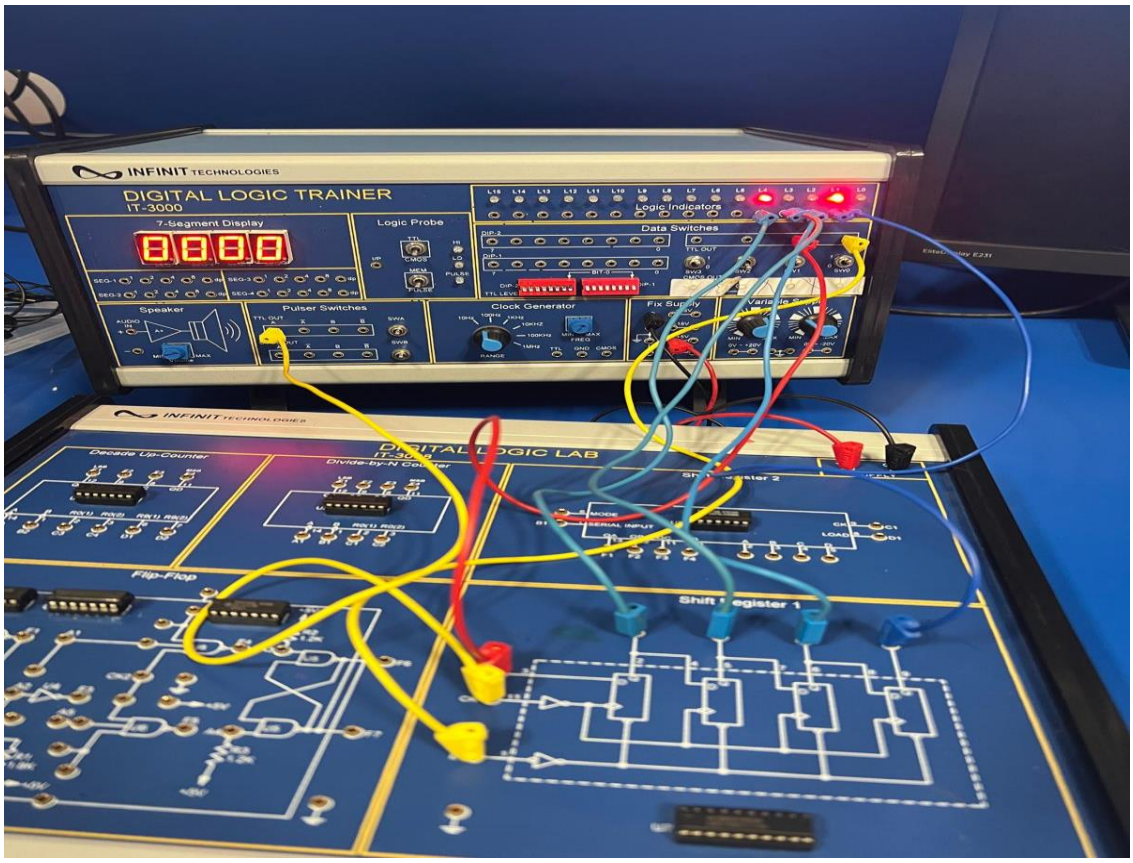

Figure 2. 12 : Shift – Right Register Circuit Connection

Discussion :

After we connected the Shift – Right Register circuit, we show how it works, so we use the input B to clear the Data that stored in the register and reset it to ‘0000’, after that we enter a value in input A to store it as the most significant bit in the register and throw (ignore) the least significant bit; that means when the register reset to ‘0000’ and we enter ‘1’ in the input A, then the register stored ‘1000’, then if we enter another ‘1’ in input A, the register will stored ‘1100’ and so on.

2.2.2 4-Bit Register with serial and parallel load

We connected the Circuit as it shown in the Figure 2.13 using IT-3008 module.

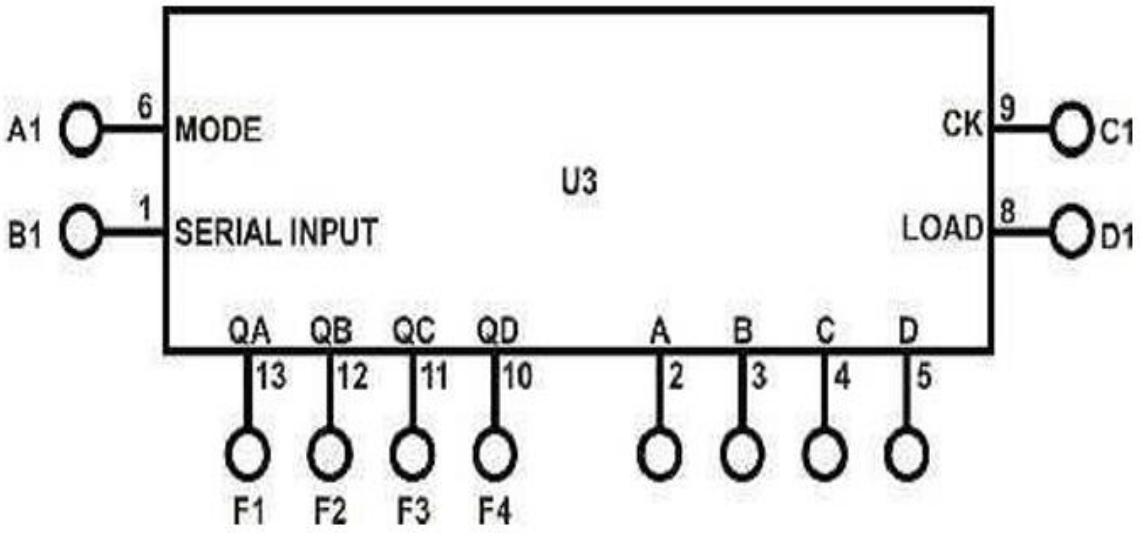

Figure 2. 13 : Shift-Register with serial and parallel load (Source : Lab Manual)

a) Firstly, we connected the inputs A, B, C and D to Data Switches SW0, SW1, SW2 and SW3 respectively.

b) Secondly, connected the outputs F1, F2, F3 and F4 to logic indicators (LEDs) L0, L1, L2 and L3 respectively.

c) Then, connected B1 (I/P) to DIP Switch DIP2.0 and connect A1 (MODE) to DIP2.1

d) After that, connect clock CK (C1) to the clock generator TTL level output at 1Hz and change data at B1 with DIP2.0 - We enter the sequence 1001 for B1.

e) Finaly, the results are shown in the Table 2.7 below :

INPUTS		OUTPUTS			
A1	**C1**	**L3**	**L2**	**L1**	**L0**
0	⎍	**0**	**0**	**0**	**1**
0	⎍	**0**	**0**	**1**	**0**
0	⎍	**0**	**1**	**0**	**0**
1	⎍	**0**	**1**	**0**	**0**

Table 2. 7 : Results of Shift-Register with serial load

f) The circuit with serial load (A1 – MODE = ‘0’) connecting as following Figure 2.14

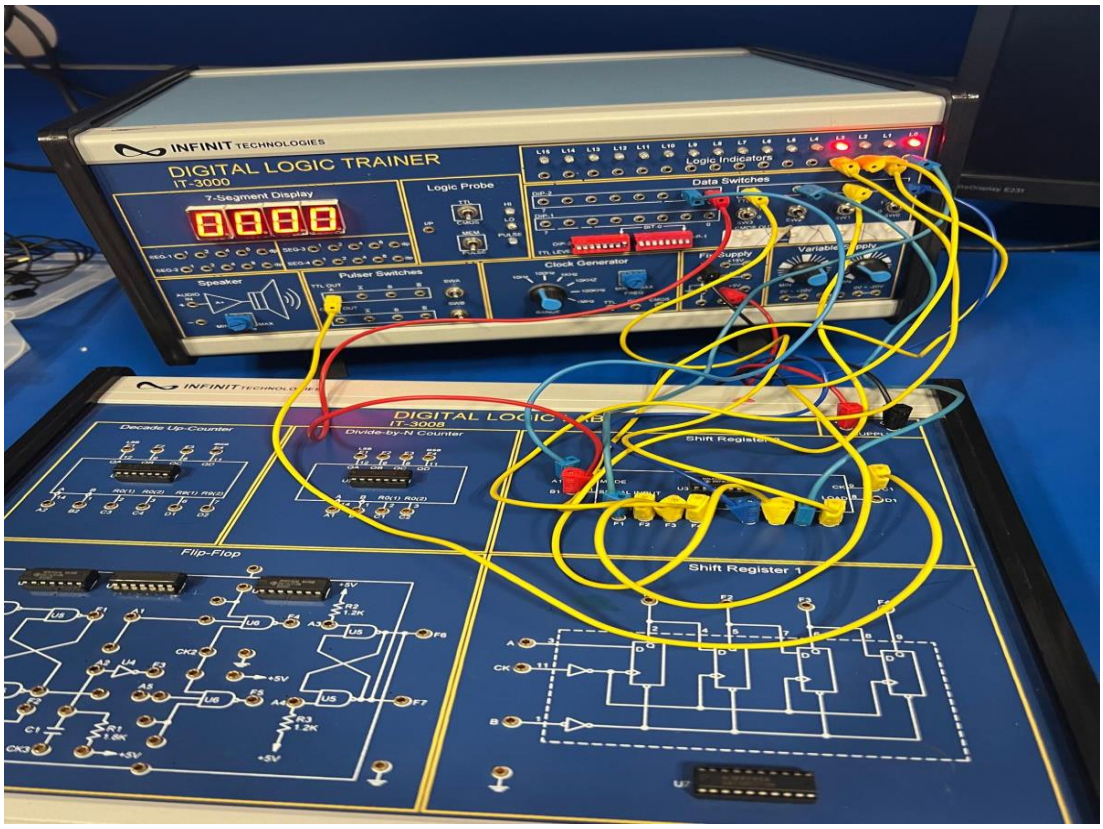

Figure 2. 14 : Shift-Register with serial load circuit connection

g) We connected LOAD (D1) to the clock generator TTL level output at 1Hz. Then we set A1 (MODE) to ‘1’ , then the results are shown in the Table 2.8 below :

INPUTS					OUTPUTS			
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**	**0**

Table 2. 8 : Results of Shift-Register with parallel load

h) The circuit with parallel load (A1 – MODE = ‘1’) connecting as following Figure 2.15

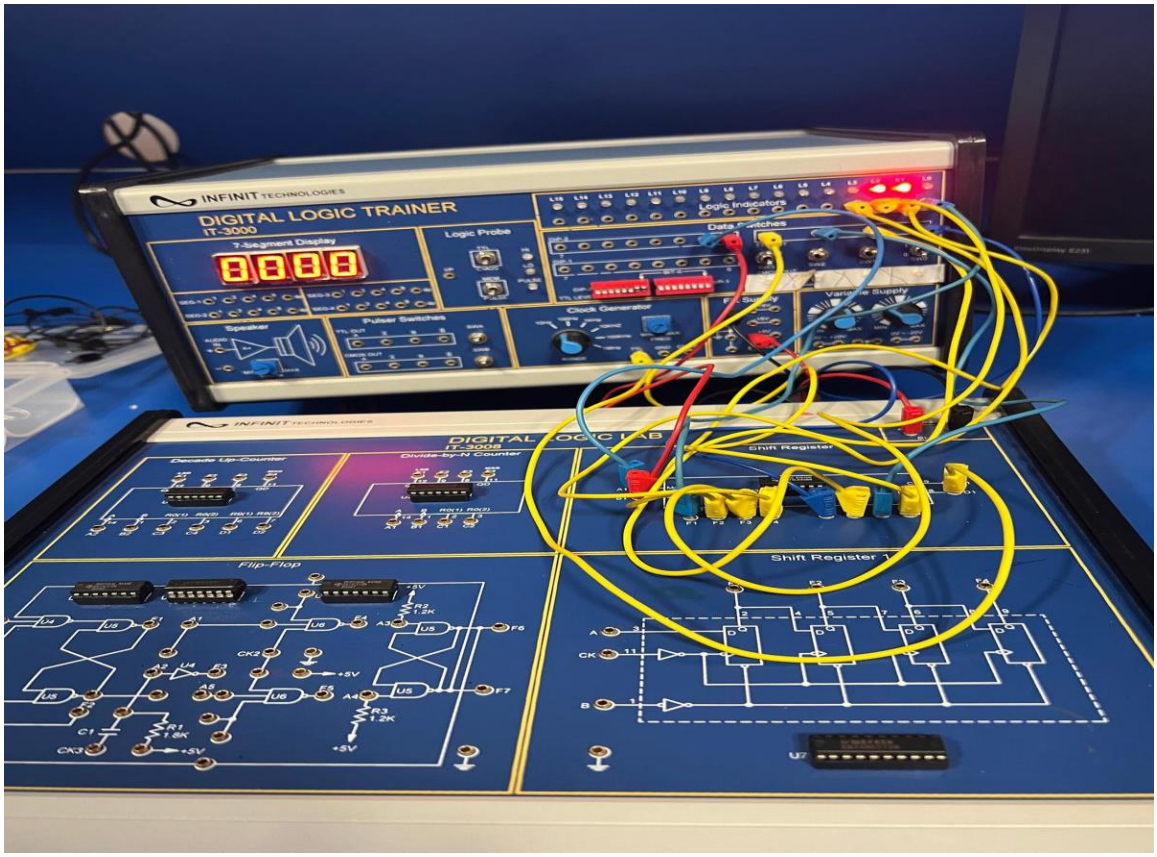

Figure 2. 15 : Shift-Register with parallel load circuit connection

Discussion :

In this section we construct a Shift – Register circuit, that has two MODE; Serial data load and Parallel data load. We can control with MODE by the input A1; when A1 = ‘0’ the MODE is serial load, so the clock C1 is activate and take the data from B1 input as serial data load, while when A1 = ‘1’ the MODE is parallel load, so the clock D1 is activate and take the inputs from (DCBA) as parallel data load, then stored this data in the register.

2.3 Counters

2.3.1 2-Bit Synchronous Counter

We connected the Circuit as it shown in the Figure 2.16 using IT-3007 module.

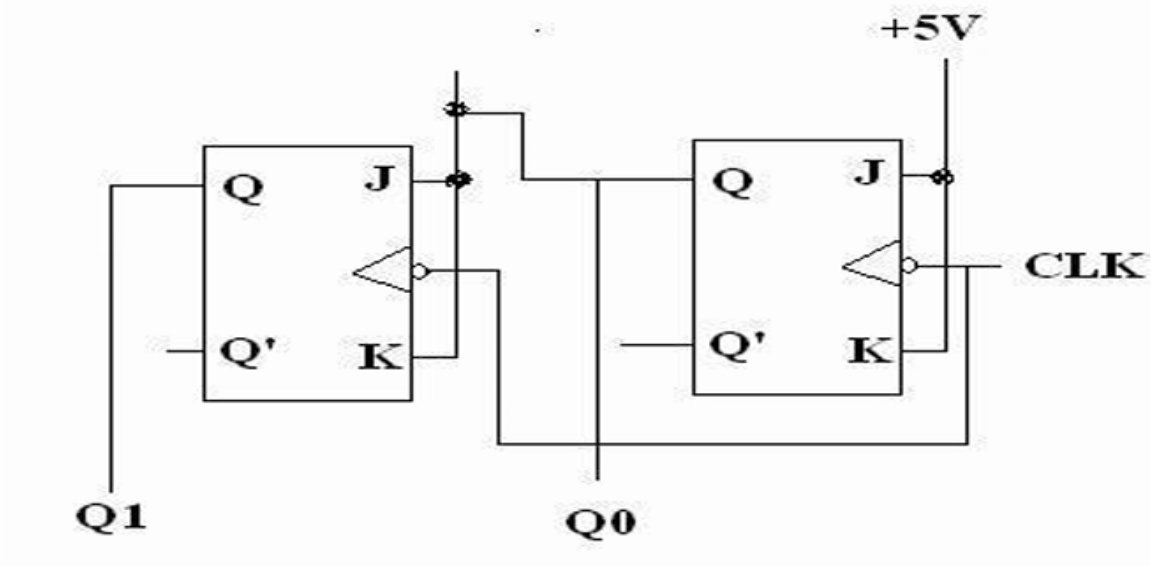

Figure 2. 16 : 2-Bit Synchronous Counter (Source : Lab Manual)

a) First, we connected +5V of module IT-3007 to the +5V output of fixed power supply and did the same for the GND.

b) We connected the clock CLK input to the pulser Switch SWA

c) Then, connected the counter outputs Q1 and Q0 to logic indicator (LEDs) L1 and L2 respectively.

d) By applying the clock pulses to CLK, then the results as shown in Table 2.9 below :

INPUT	OUTPUTS	
CLK	Q1	Q0
⎍	0	0
⎍	0	1
⎍	1	0
⎍	1	1
⎍	0	0
⎍	0	1
⎍	1	0
⎍	1	1

Table 2. 9 : Results of 2-Bit Synchronous Counter in binary

e) The Circuit with binary outputs connecting as the Figure 2.17 bellow :

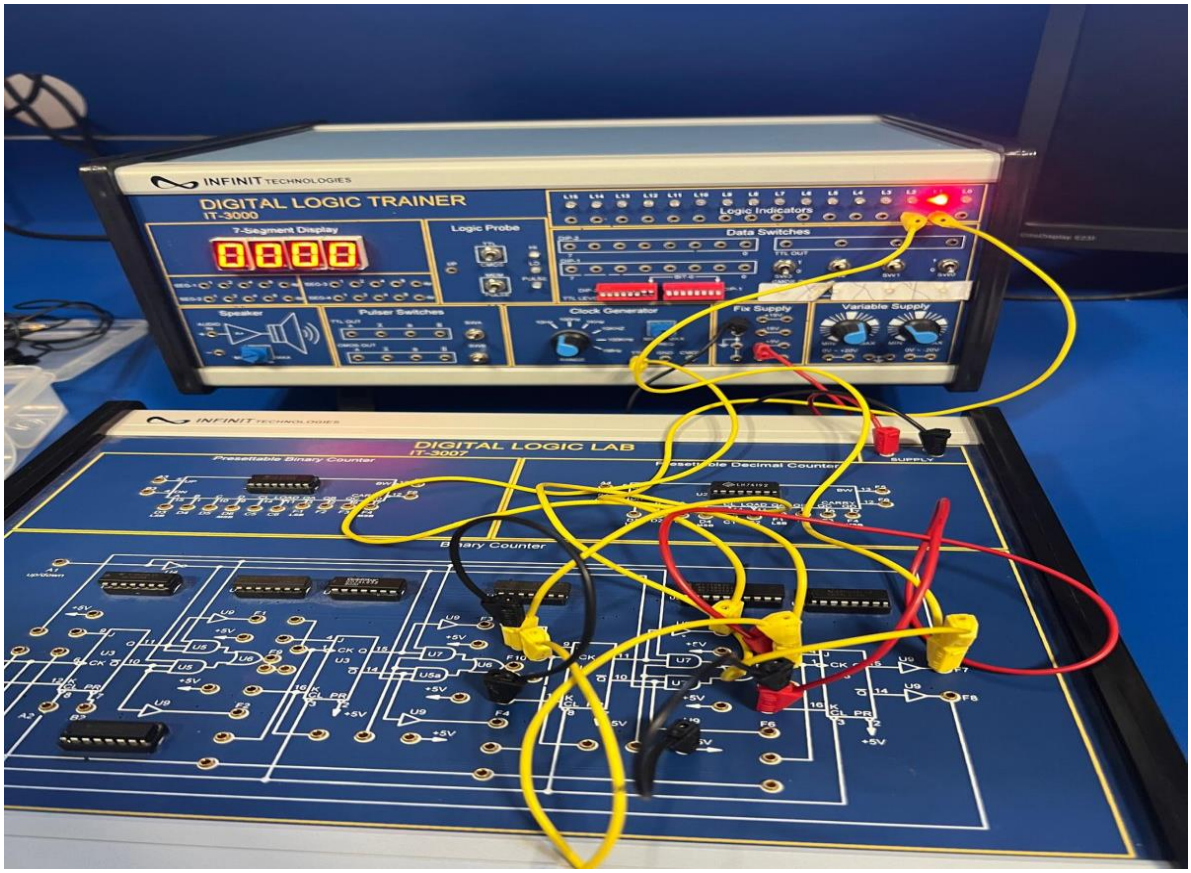

Figure 2. 17 : 2-Bit Synchronous (binary outputs) Circuit Connection

f) Finaly, we connected the outputs Q1 and Q0 to seven segment display, then the results in decimal as shown in Table 2.10 below :

INPUT	OUTPUTS
CLK	D
⎍	0
⎍	1
⎍	2
⎍	3
⎍	0
⎍	1
⎍	2
⎍	3

Table 2. 10 : Results of 2-Bit Synchronous Counter in decimal

g) The Circuit with decimal outputs connecting as the Figure 2.18 bellow :

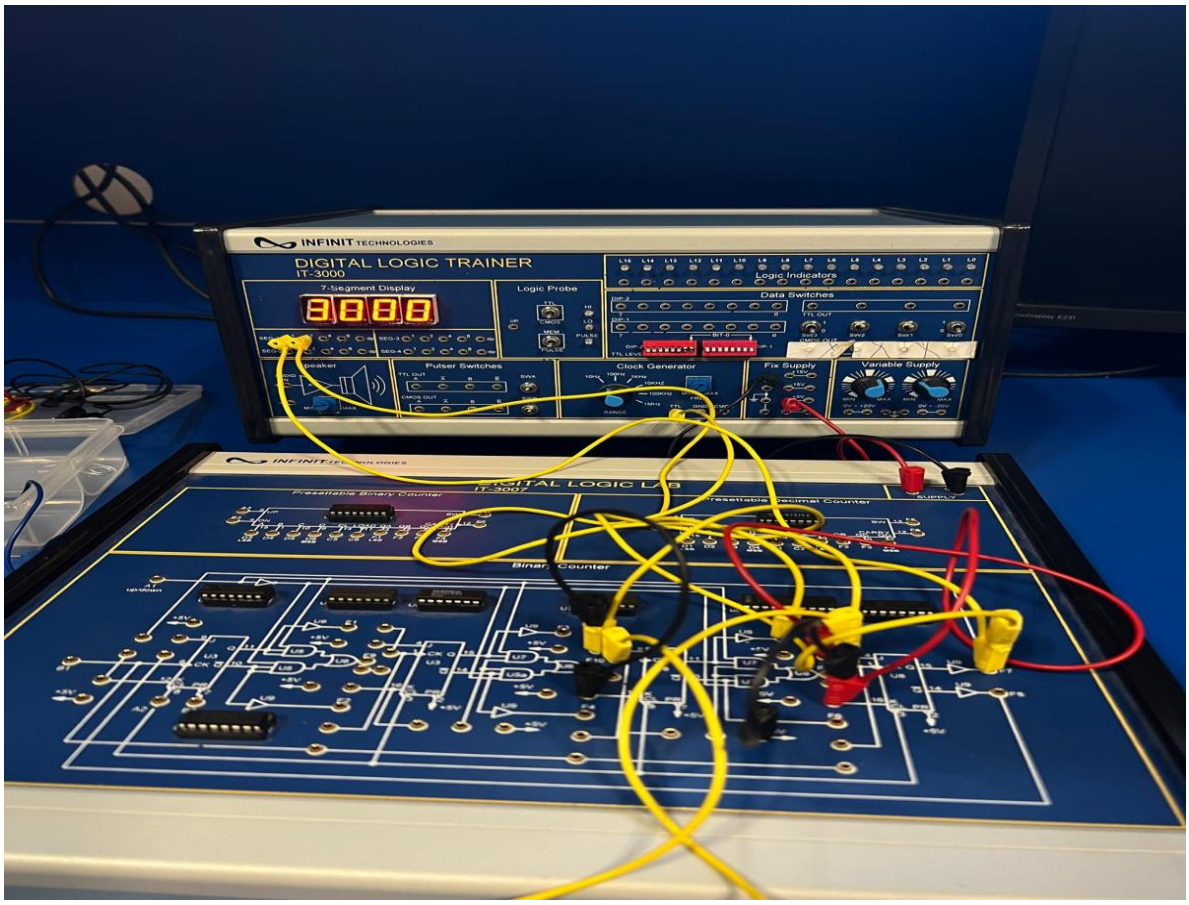

Figure 2. 18 : 2-Bit Synchronous (decimal outputs) Circuit Connection

Discussion :

As we show in the Circuit above, we implement 2-Bit Synchronous counter, with clock CLK, so each clock pulse the counter count up by '1', this counter is 2-bit counter so it counts from 00 to 11 in binary. We connect this circuit outputs once to logic indicators (LEDs) and show the outputs in binary from 00 to 11, then we connect the outputs to seven segment display and show the output in decimal, so we show that this circuit count from 0 to 3 in decimal.

2.3.2 3-Bit (divide-by-eight) Ripple Counter

We connected the Circuit as it shown in the Figure 2.19 using IT-3007 module.

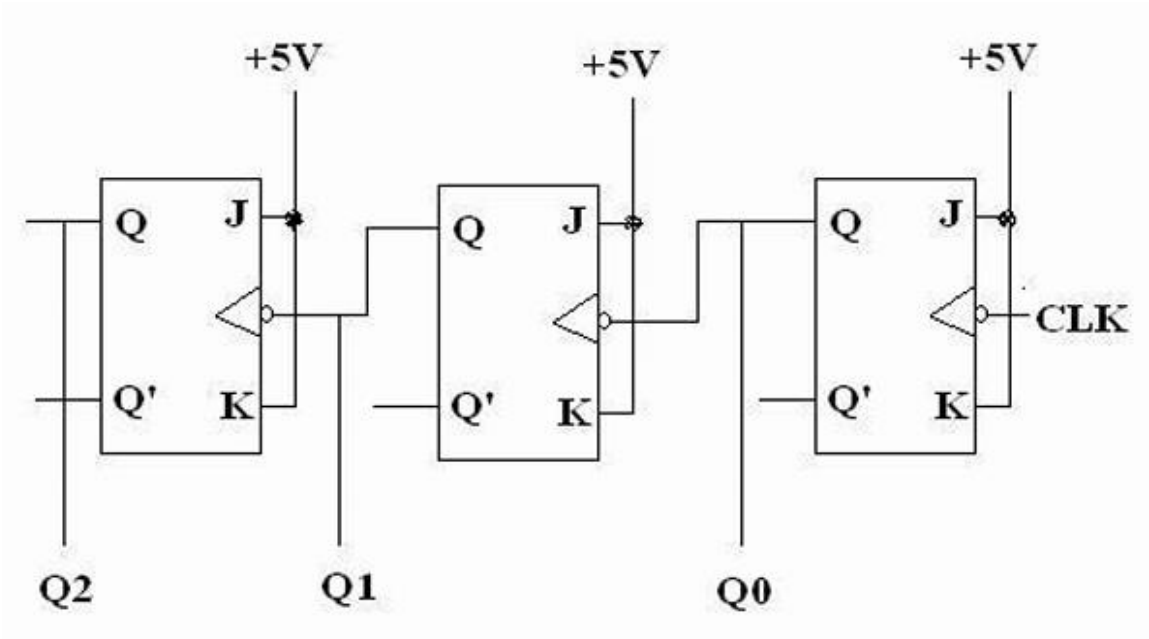

Figure 2. 19 : 3-bit Ripple Counter (Source : Lab Manual)

a) Firstly, we connected the clock CLK input to pulser switch.

b) Then, we connected the counter outputs Q2,Q1 and Q0 to logic indicators (LEDs) to show the results in binary.

c) By applying the clock pulses to CLK, then the results as shown in Table 2.11 below :

INPUT	OUTPUTS		
CLK	Q2	Q1	Q0
⎍	0	0	0
⎍	0	0	1
⎍	0	1	0
⎍	0	1	1
⎍	1	0	0
⎍	1	0	1
⎍	1	1	0
⎍	1	1	1
⎍	0	0	0
⎍	0	0	1

Table 2. 11 : Results of 3-bit Ripple Counter in binary

d) The Circuit with binary outputs connecting as the Figure 2.20 bellow :

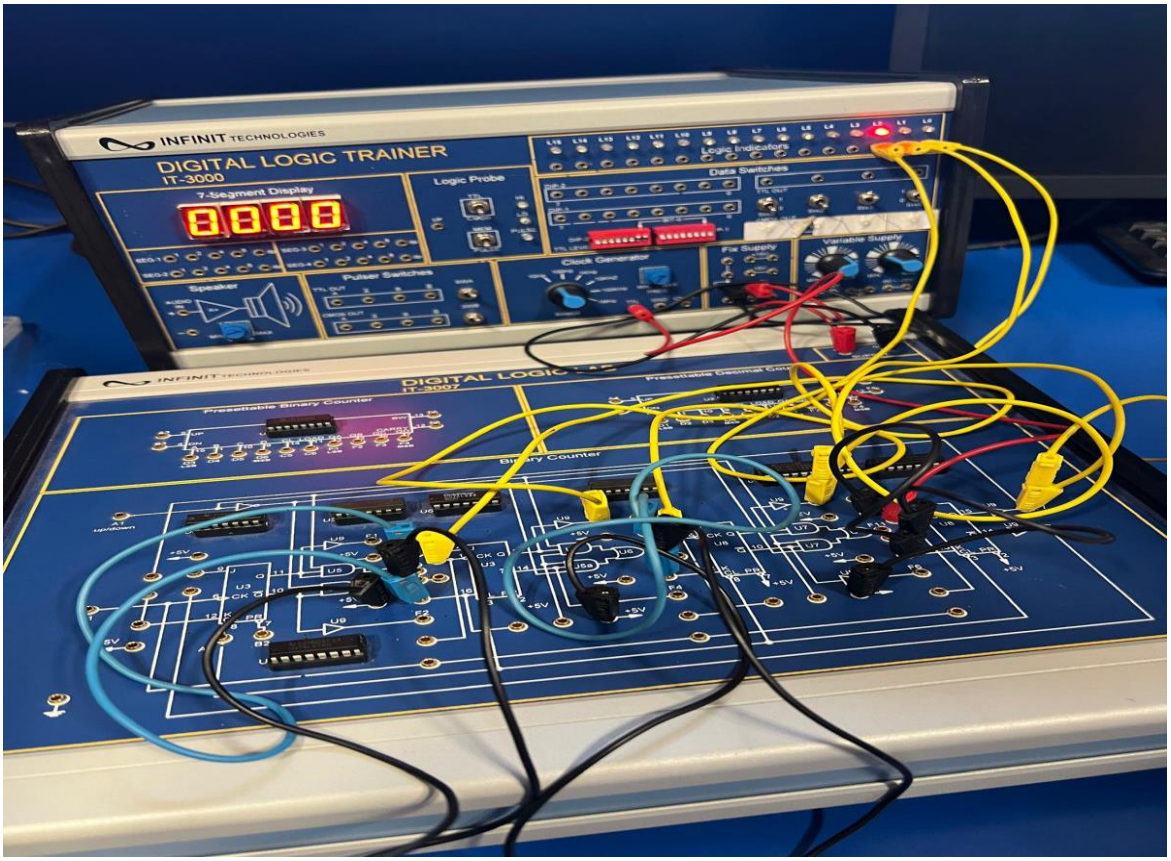

Figure 2. 20 : 3-Bit Ripple counter (binary outputs) Circuit Connection

e) Finaly, we connected the outputs Q2, Q1 and Q0 to seven segment display, then the results in decimal as shown in Table 2.12 below :

INPUT	OUTPUT
CLK	D
⎍	0
⎍	1
⎍	2
⎍	3
⎍	4
⎍	5
⎍	6
⎍	7
⎍	0
⎍	1

Table 2. 12 : Results of 3-bit Ripple Counter in decimal

a) The Circuit with decimal outputs connecting as the Figure 2.21 bellow :

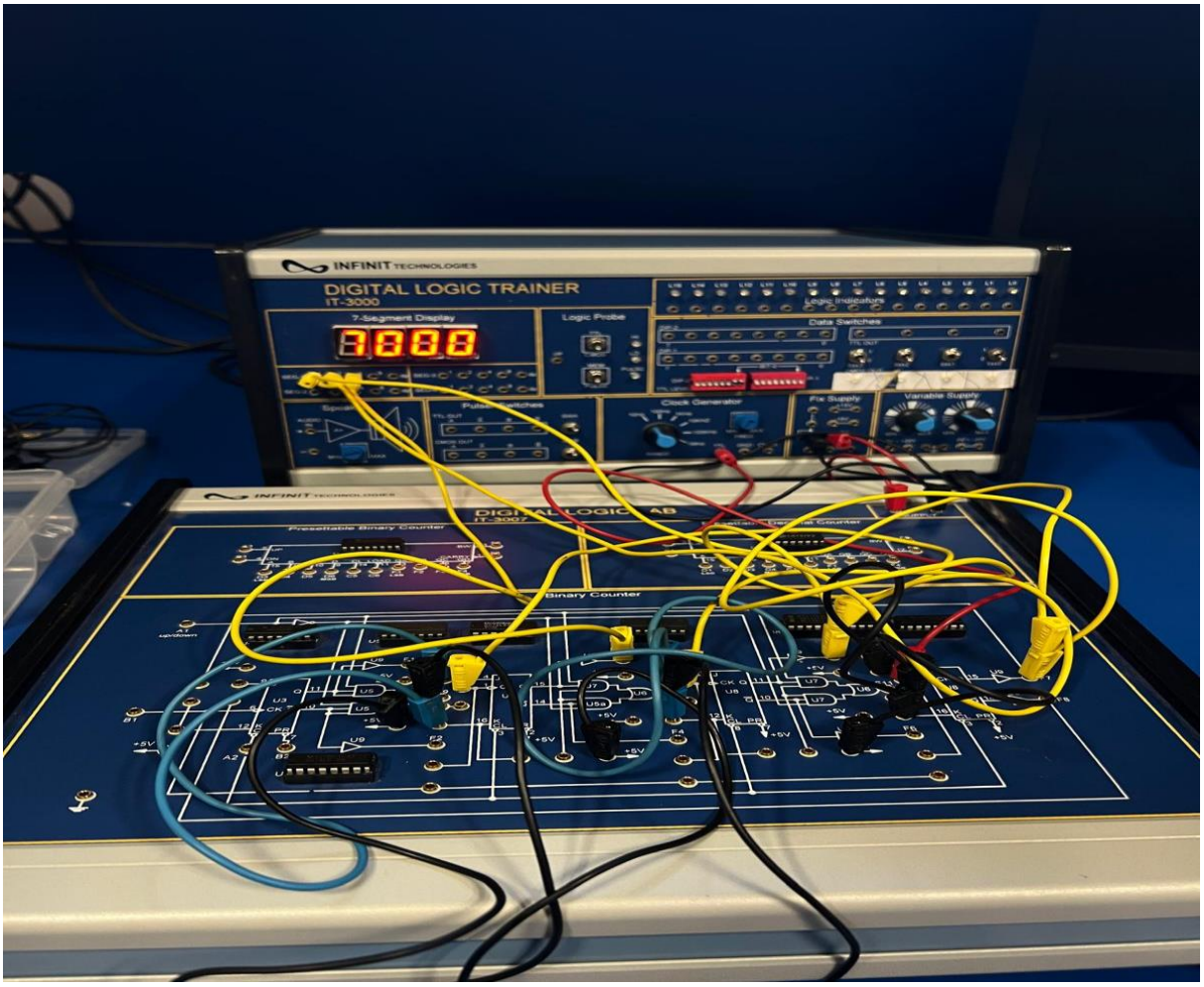

Figure 2. 21 : 3-Bit Ripple counter (decimla outputs) Circuit Connection

Discussion :

As we show in the Circuit above, we implement 3-Bit Ripple counter, with clock CLK, so each clock pulse the counter count up by '1', this counter is 3-bit counter so it counts from 000 to 111 in binary. We connect this circuit outputs once to logic indicators (LEDs) and show the outputs in binary from 000 to 111, then we connect the outputs to seven segment display and show the output in decimal, so we show that this circuit count from 0 to 7 in decimal.

2.3.3 BCD Counter

We connected the Circuit as it shown in the Figure 2.22 using BCD Counter block (IC 7490) on IT-3008 module.

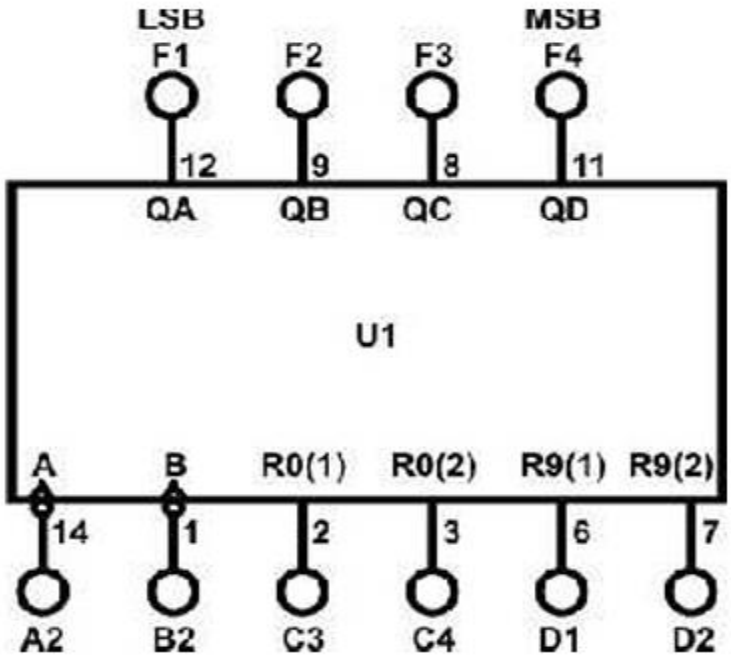

Figure 2. 22 : IC 7490 BCD Counter (Source : Lab Manual)

This IC is connected using four flip – flops as the following Figure 2.23

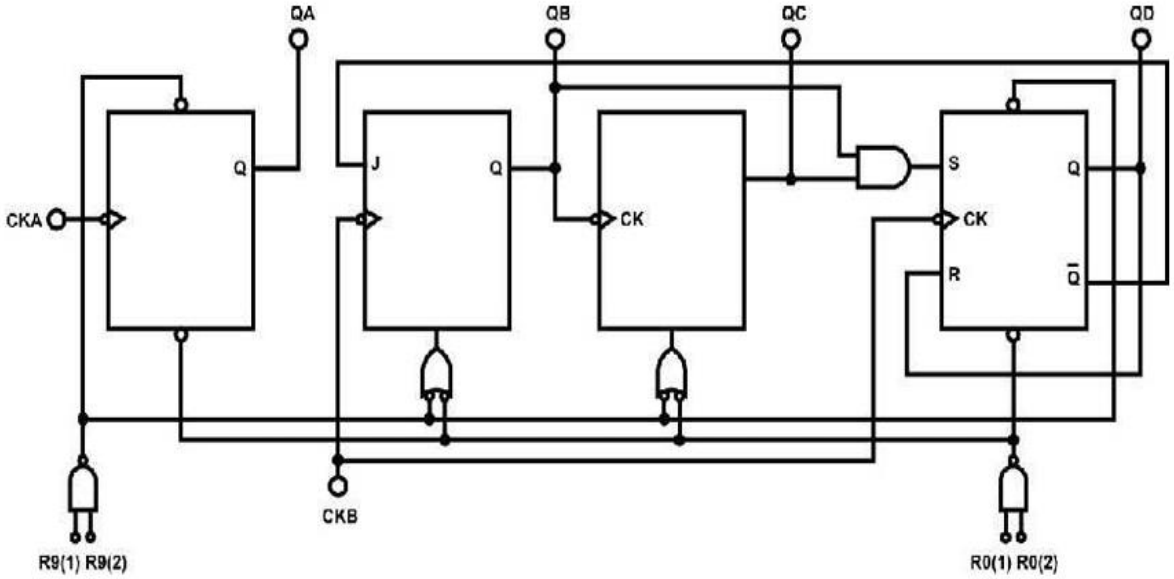

Figure 2. 23 : IC 7490 BCD Counter Implementation (Source : Lab Manual)

a) Firstly, connecting +5V of module IT-3008 to the +5V output of fixed power supply and doing the same for GND.

b) Secondly, connect the inputs C3 and C4 to Data Switches SW0 and SW1 and connect D1 and D2 to switches SW2 and SW3.

c) Then, connect the outputs F1…F4 to logic indicators (LEDs) L1…L4 and connect A2 to SWA A output.

d) Finally, we connected F1 to B2, and set C3, C4, D1 and D2 to ground.

e) The circuit connecting as the following Figure 2.24 :

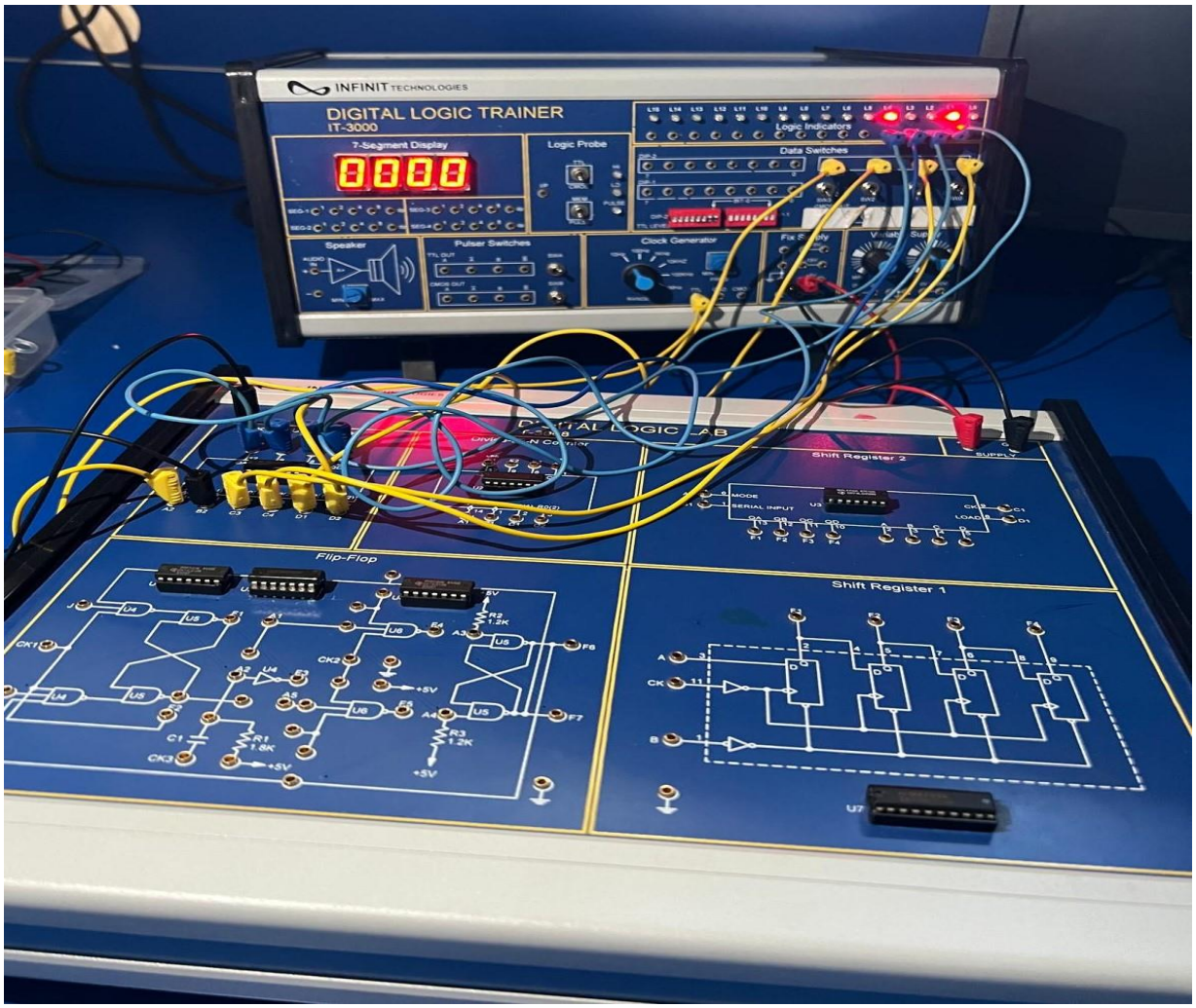

Figure 2. 24 : BCD counter circuit connection

Discussion :

In this section, we implement BCD counter and show it's result in binary by logic indicators (LEDs), so this circuit count from 0000 to 1001 – BCD – , and we can modify on this circuit by change R0(1) and R0(2) to reset the counter on the digit that I want.

3. Discussion

Answer the following questions :

1- Although latches are useful for storing binary information, they are rarely used in sequential circuit design, why ?

Latches are level triggered, which means that the output changes whenever the input changes, which can cause glitches, while flip-flops more suitable for circuits because it's edge triggered, so the output depends on a clock to be changed.

2- What is the disadvantages of the SR flip – flop ?

We have a problem in the SR latch, when both inputs equal '1' [S = R = 1] in active high latches or both equal '0' [S = R = 0] in active low latches, in this case both Q and Q' are the same [Q = Q'], which impossible, so it gives a wrong answer, this case known as indeterminate status.

3- What is the difference between "Synchronous" and "Ripple" counters ?

In the synchronous counter all flip – flops have the same clock [common clock], that means they all change their states simultaneously at each edge of the clock, so the delay in this counter is small, while the ripple counters don't have common clock, only the first flip – flop has an external clock input and the output of each flip – flop is the clock input for the next one, this make this counter slower because this counter has big delay.

Conclusion

After completing this experiment all the objectives are obtained. Now, I can construct sequential digital circuits and implement them by using basic gates or ICs. Such as latches, first we implement SR latch, then use its implementation to implement more complex circuits like flip – flops. For Register, we construct flip – flop circuits then using them to implement a register. To construct a counter circuit, we use flip – flops and clock pulses to transform from state to another. In synchronous counter the clock input is common for all flip – flops, while in ripple counter the first flip – flop has an external clock input and the output from the flip – flop will be the clock input for the next one.

After we construct the circuits and trace the results, we noticed that the results are similar to theoretical results, so we construct them correctly and there is no problem or any issue in constructing the circuits.

References

- Manual for Digital Electronics and Computer Organization Lab, 2024, Birzeit University.

[1] : GeeksforGeeks : Sequential Circuits in digital logic, Retrieved April 5, 2024 from https://www.geeksforgeeks.org/introduction-of-sequential-circuits/

[2] : GeeksforGeeks : Latches in digital logic, Retrieved April 5, 2024 from https://www.geeksforgeeks.org/latches-in-digital-logic/

[3] : GeeksforGeeks : flip-flops in digital logic, Retrieved April 5, 2024 from https://www.geeksforgeeks.org/flip-flop-types-their-conversion-and-applications/

[4] : GeeksforGeeks : General Purpose Registers in digital logic, Retrieved April 6, 2024 from https://www.geeksforgeeks.org/general-purpose-registers/?ref=lbp

[5] : GeeksforGeeks : Shift Registers in digital logic, Retrieved April 6, 2024 from https://www.geeksforgeeks.org/shift-registers-in-digital-logic/?ref=lbp

[6] : GeeksforGeeks : Ripple Counters in digital logic, Retrieved April 6, 2024 from https://www.geeksforgeeks.org/ripple-counter-in-digital-logic/?ref=lbp

[7] : GeeksforGeeks : Synchronous Counters in digital logic, Retrieved April 6, 2024 from https://www.geeksforgeeks.org/counters-in-digital-logic/