

**Electrical Engineering Department**

**ENCS211**

**Digital Lab**

**Experiment #5**

**Sequential Logic Circuits**

**Prepared By :**

**Ahmad Zahran 1142583**

**Instructure : Dr.** [**Wasel Ghanem**](https://ritaj.birzeit.edu/bzu-msgs/type?mttid=104&classid=109130)

***Abstract:***

***Aims of the experiment:***

1. To understand the differences between combinational and sequential logic circuits and the applications of various memory units.
2. To study the operating principles and applications of various flip-flops.
3. To understand the operating principles of counters and how to construct counters with JK flip-flops.
4. To study asynchronous and synchronous counters.

***Theory:***

In some experiments it is need to use a result from the previous one. In combinational circuits there is no feedback so the output depends on the inputs only meaning that the previous state isn’t effect on the next state results.

Other type of circuits is a sequential circuit which is designed to be used as a memory element .Those circuits gives feedback such that the next output results depends on the inputs and the previous state.

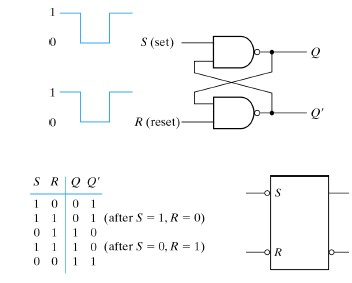
**Sequential circuits types:**

1. **Latch:**

 Is a [circuit](https://en.wikipedia.org/wiki/Electronic_circuit) that has two stable states and can be used to store state information.

1. **RS Latch:**

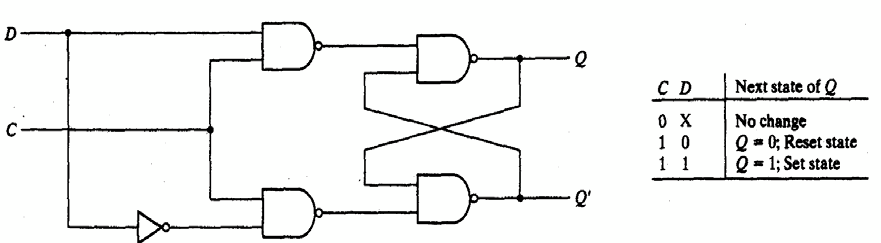
It is a circuit with two cross-coupled NOR or NAND gates. It has two inputs, S and R. S is called set and R is called reset. he output of the S-R latch depends on current as well as previous inputs or state, and its state can change as soon as its inputs change. The circuit and the truth table of RS latch is shown below.



**Figure.1: SR latch with NAND gate**

1. **D Latch:**

The D latch was developed to eliminate the undefined condition of the indeterminate state in the RS latch. The D latch and its state table is shown below .



**Figure.2: D-Latch**

1. **Flip-Flops:**

Flip-flop is a Memory element that storage binary information which is actually a latch with a control input (controls when to get the output).

Types of Flip flops:

1. D Flip Flop:

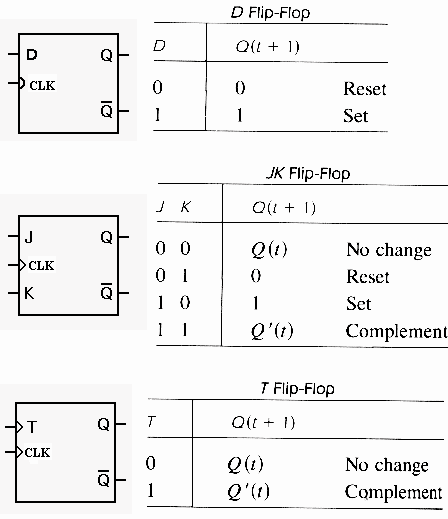
The output is the same as the input (D=0 →Q=0, D=1→Q= 1).

1. JK Flip Flop:

The output has 4-cases when JK =00 no change in output, when JK =01 reset, JK =10 → (set) and the last when 1JK=11 →(complement of the present state ) .

1. T Flip Flop:

when T = 0 → Qnext = Qpresent, T=1 → Qnext= (Qpesent).



**Figure.3: D, JK, and T flip flops**

1. **Registers:**

A register is a collection of flip-flops that have a common clock and all reset at the same time; any N-bit register consists of N flip-flops. Digital systems use registers to hold binary entities. Shift register is a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop.

1. **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. In ripple counters, there is no common clock; the flip-flop output transition serves as a source for triggering other flip-flops. In synchronous counters, all flip flops receive a common clock.

***Procedure* *And* *Discussion:***

1. **Latches and Flip flops:**
2. **Constructing RS latch with Basic Logic Gates:**

The KL-26006 Module was set on the KL-22001 and block (c) was located



Figure 4: NAND RS flip-flop

Then we follow the input as next. We found the output as shown in the table:

|  |  |
| --- | --- |
| S R | Q Q' |
| 0 0 | 1 1 |
| 0 1 | 1 0 |
| 1 0 | 0 1 |
| 1 1 | 0 1 |

Table(1):truth table of active low RS latch

1. **Constructing RS latch with control input:**

The connections was completed by referring to wiring diagram



Figure5: RS Latch with control input

Then we follow the input as next. We found the output as shown in the table:

|  |  |
| --- | --- |
| S R | Q Q' |
| 0 0 | 1 0 |
| 0 1 | 0 1 |
| 1 0 | 1 0 |
| 1 1 | 1 1 |

Table(2): truth table of RS latch with control input

1. **Constructing D latch with RS latch:**

This connection in fig (6) change the RS latch to D latch was tested and had the result in table (3).

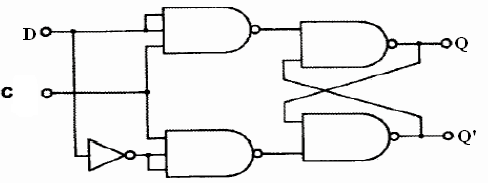
****

Figure.6: D Latch

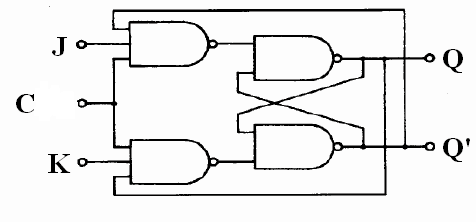
|  |  |
| --- | --- |
| CK2 D | Q Q' |
| 0 0 | 0 1 |
| 0 1 | 0 1 |
| 1 0 | 0 1 |
| 1 1 | 1 0 |

Table (3): truth table for D latch circuit

The value of Q =D only when clock pulse is positive. Otherwise Q doesn’t change.

1. **Constructing JK latch with RS latch:**

We connected the circuit of fig.7. And follow the input sequence of Table.4.

****

**Figure.7: JK Latch**

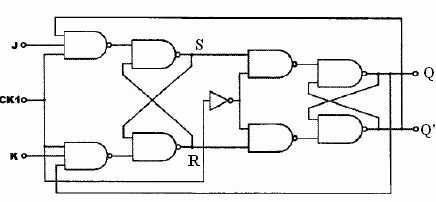
|  |  |
| --- | --- |
| J K | Q Q' |
| 0 0 | 1 0 |
| 0 1 | 0 1 |
| 1 0 | 1 0 |
| 1 1 | 0 1 |

Table(4): truth table for JK Latch

This type of latch (JK) works as SR latch in No change state JK = 00, Reset state JK = 01 & Set state JK = 10. But at state JK = 11 it complements the previous value of Q & so Q'. As shown in the table.

1. **Constructing JK Flip-flop with master- slave RS latches:**

We connected the circuit of fig.8. And follow the input sequence of Table.5



**Figure.8: JK Flip-Flop**

|  |  |
| --- | --- |
| Clk K J | S R Q Q' -> S R Q Q' |
| 0 0 | 1 0 1 0 -> 1 0 1 0 |
| 0 1 | 1 0 1 0 -> 1 0 1 0 |
| 1 0 | 1 0 1 0 -> 0 1 0 1 |
| 1 1 | 0 1 0 1 -> 1 0 0 1 |
| 1 1 | 1 0 0 1 -> 1 0 1 0 |

Table (5)

1. **Registers:**
2. **Constructing Shift Register with D Flip-Flops:**

Circuit below was constructed as required.

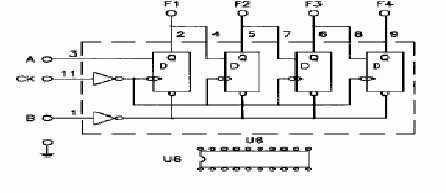


Figure.9: Shift Right Register

SW0 was set to 0 to clear the register, and then it was kept on 1. SW1 was set to 1 and then three clock pulses were applied and the outputs sequence was:

1. Before clock: F1F2F3F4 = 0000
2. After clock1: F1F2F3F4 = 1000
3. After clock2: F1F2F3F4 = 1100
4. After clock3: F1F2F3F4 = 1110
5. After clock4: F1F2F3F4 = 1111

To load the value 1011 on the register, the following steps were applied:

1. Before clock: F1F2F3F4 = 0000
2. SW1 = 1, apply clock pulse and the output is: F1F2F3F4 = 1000
3. SW1 = 1, apply clock pulse and the output is: F1F2F3F4 = 1100
4. SW1 = 0, apply clock pulse and the output is: F1F2F3F4 = 0110
5. SW1 = 1, apply clock pulse and the output is: F1F2F3F4 = 1011
6. **4-Bit Shift Register with serial and parallel load:**

The figure below shows the chip (IC7495) pins and what each pin does:

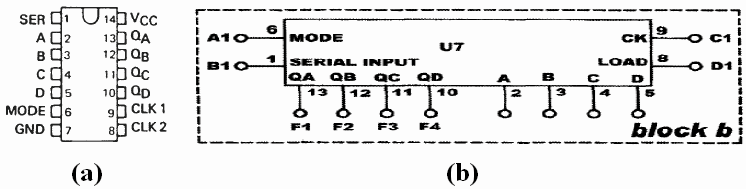


Figure.10: IC7495 shift register with serial and parallel load

**1. Shift- right serial- input:**

To enable the shift-right operation which is controlled by CK C1 (pin 9), the MODE control input A1 (pin 6) must be 0. In this mode CK depends only on the SERIAL INPUT B1 (pin 1) to shift the inputs A-D to the right.

B1 (serial input) was connected to SW0, A1 to GND, inputs A-D to SW4-SW1, and outputs QA-QD to L3-L0 and CK to pulser switch.

The value 1101 was loaded to the register (ABCD = 1101) and then four clock pulses were applied and resulted in the following outputs sequence:

1. After clock1: QAQBQCQD = 0110
2. After clock2: QAQBQCQD = 0011
3. After clock3: QAQBQCQD = 0001
4. After clock4: QAQBQCQD = 0000

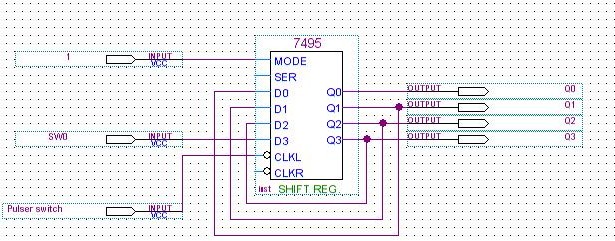
**2. Parallel- load Register:**

To enable the parallel load operation which is controlled by LOAD D1 (pin 8), the MODE control input A1 (pin 6) must be 1. In this mode LOAD depends only on the parallel inputs A-D to load them into the parallel outputs QA-QD. A1 was connected to +5VCC, inputs A-D to SW4-SW1, outputs QA-QD to L3-L0 and LOAD to pulser switch.

When we applied a clock pulse the value on the output was the same as the value loaded to the inputs. For example, to load the value 1101 we set this value on the inputs ABCD = 1101, and then we applied a clock pulse which resulted in the output QAQBQCQD = 1101.

**Task 1:**

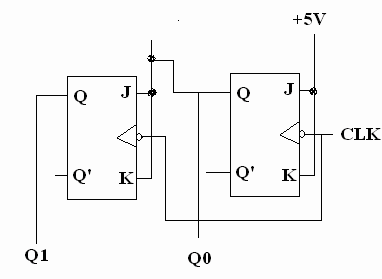
The load mode is used to get the serial shift-left. MODE = 1 and the pulser switch is connected to CLKL (LOAD), then outputs Q1Q2Q3 are connected to inputs D0D1D2 and input D3 (serves as the serial input) is connected to SW0.



Fig(11)

1. **Counters:**
2. **2-bit Synchronous Counter:**

We connected the circuit of fig.12. And follow the input sequence of Table.6

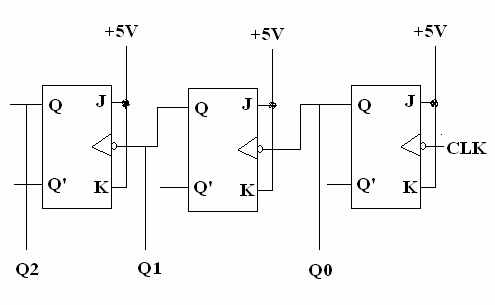
 Figure.12: 2-bit Synchronous Counter

|  |  |  |  |
| --- | --- | --- | --- |
| Clk | Q1 | Q0 | D1 |
|  | 0 | 0 | 0 |
|  | 0 | 1 | 1 |
|  | 1 | 0 | 2 |
|  | 1 | 1 | 3 |
|  | 0 | 0 | 0 |
|  | 0 | 1 | 1 |
|  | 1 | 0 | 2 |
|  | 1 | 1 | 3 |
|  | 0 | 0 | 0 |
|  | 0 | 1 | 1 |

Table(6)

**B) 3-bit (divide-by-eight) Ripple Counter:**

We connected the circuit of fig.13. And follow the input sequence of Table.7

Figure.13: 3-bit Ripple Counter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLK | Q2 | Q1 | Q0 | D1 |
|  | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 1 | 1 |
|  | 0 | 1 | 0 | 2 |
|  | 0 | 1 | 1 | 3 |
|  | 1 | 0 | 0 | 4 |
|  | 1 | 0 | 1 | 5 |
|  | 1 | 1 | 0 | 6 |
|  | 1 | 1 | 1 | 7 |
|  | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 1 | 1 |

Table(7)

**C) BCD Counter**

We constructed the circuit shown in the fig(13); we set RO1 and RO2 to 0 volt

Then we connected A-D to logic indicators to see the output results:



Figure.13: IC 7490 BCD Counter

We followed the output sequence. We noticed that this chip counts from 0000 to 1001 and after 1001 it resets the counter to 0000.

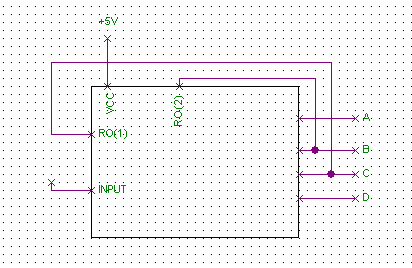
**D) Divide-by-8 counter using BCD chip counter:**

This circuit counts from 0 to 111 so the same procedure to construct the BCD counter was followed here except that RO(2) was connected to +5VCC and D was connected to RO(1) since the binary value of 8 is 1000.

**Task 3:**

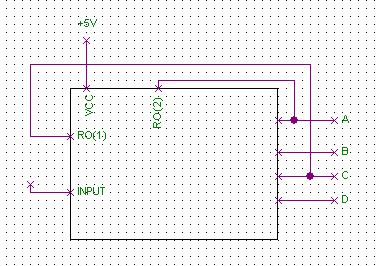
Change the connection of counter in Figure .13to count from:

0-to-5: RO(1) is connected to C and RO(2) is connected to B (6 = 0110).



Fig(14)

0-to-4: RO(1) is connected to C and RO(2) is connected to A (5 = 0101).

****

Fig(15)

***Conclusions:***

In this experiment, we learned that latches depend on clock level and so the output changes can’t be controlled and that there is an indeterminate state in the SR latch, therefore latches are rarely used in sequential circuits design. Flip-flops were the solution to such a problem since they depend on clock edge.

We note that the characteristic of flip flop is applied in the table of present and next state.  
**D**

*Qnext = Q*

**T**

*Qnext = TQ' + T'Q*

**JK**

*Qnext =JQ' + K'Q*

We also learned how to use registers (collection of flip-flops) to load inputs into the outputs and to shift inputs (right or left) each time a clock pulse is applied.

Finally, we used registers as counters for different sequences. In each sequence the idea was what to connect to the inputs of each flip-flop, for example, to construct a 2-bit synchronous counter we had to use two JK flip-flops with J and K of the first connected to +5VCC and J and K of the second connected to the output of the first and, of course, the two flip-flops were connected to the same clock. Another idea was how to construct a circuit that counts from, for example, 0 to 5; the solution was to reset this circuit when the input reaches 6 and this is done by connecting the 1’s of 6 (0110) to two inputs of a NAND gate that resets the circuit.

**References**

1. <https://www.tutorialspoint.com/computer_logical_organization/sequential_circuits.htm>
2. <https://en.wikipedia.org/wiki/Flip-flop_(electronics)#Simple_set-reset_latches>
3. <http://www.asic-world.com/digital/seq3.html>
4. <http://www.electronics-tutorials.ws/sequential/seq_1.html>
5. <https://en.wikipedia.org/wiki/Register>
6. Manual Lab of Birzeit University Experiment No.5