Faculty of Engineering & Technology

Electrical & Computer Engineering Department

Digital Electronics and Computer Organization Lab ENCS 2110

Report #5:

Sequential Logic Circuits

Prepared by:

Aya Dahbour 1201738

Partners:

Nour Makhalfa 1201505

Wael Awawda

Instructor: Dr. Bilal Karaki
Assistant: Eng. Ali Hamodeh

Section: 2

Date: 28-4-2022

Abstract:

The aim of the experiment is to understand the differences between combinational and sequential Logic circuits, and the applications of various memory units. Also, to study the operating principles and applications of various flip-flops. Moreover, to understand the operating principles of counters and how to construct counters with JK flip-flops. Finally, to study the synchronous and asynchronous counters.

Contents:

Table of figures:

List of tabels:

Pre-Lab:

1. SR latch using NOR gates

Design the Logic Diagram, function table of the SR latch using NOR gates, and explain how it works?

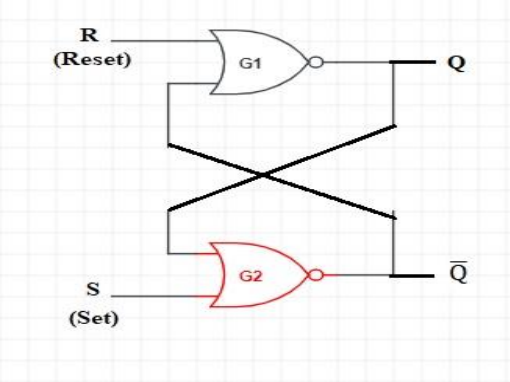


Figure 1.SR-Latch using not gate

Table 1.SR-Latch using nor gate truth table

S	R	Q	Q^{-}
0	0	Previous state	
0	1	0	1
1	0	1	0
1	1	Not used	

For SR-Latch using two nor gate there is 4 cases:

Case 1:

S=0, r=0 then Q will be the same as the previous state(memory).

Case 2:

S=0, r=1 then Q will equal 0(Reset).

Case 3:

S=1, r=0 then Q will equal 1(set).

Case 4:

S=1, r=1 this case is not used hence Q and Q' will be equaled and that not make since

Theory:

A Sequential circuit is a combinational logic circuit that consists of inputs variable (X), logic gates (Computational circuit), and output variable (Z).

A combinational circuit produces an output based on input variable only, but a Sequential circuit produces an output based on current input and previous input variables. That means sequential circuits include memory elements that are capable of storing binary information. That binary information defines the state of the sequential circuit at that time. A latch capable of storing one bit of information.

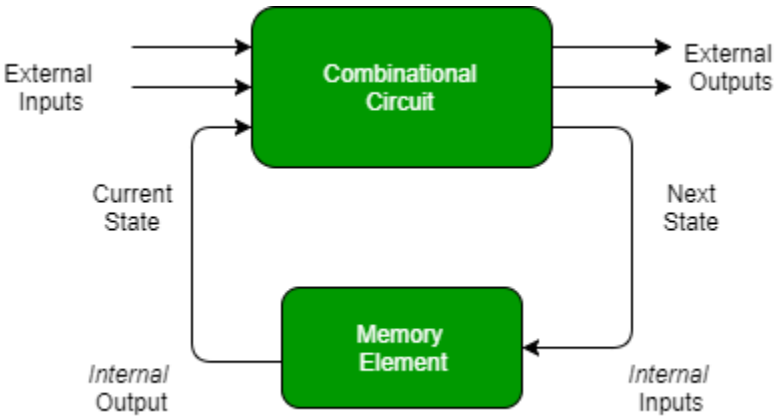


Figure 2. Sequential circuit

Latches:

Latches are basic storage elements that operate with signal levels (rather than signal transitions). Latches are level-sensitive. Latches are useful for the design of the asynchronous sequential circuit.[2]

1.SR (Set-Reset) Latch:

SR Latch is a circuit with:
(i) 2 cross-coupled NOR gate or 2 cross-coupled NAND gate.
(ii) 2 input S for SET and R for RESET.
(iii) 2 output Q, Q'.

Table 2. SR latch truth table

Q	Q'	STATE
1	0	Set
0	1	Reset

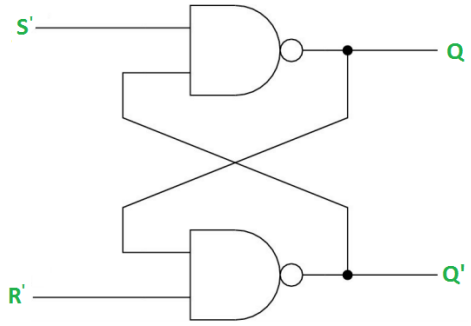


Figure 3. SR latch with NAND gate

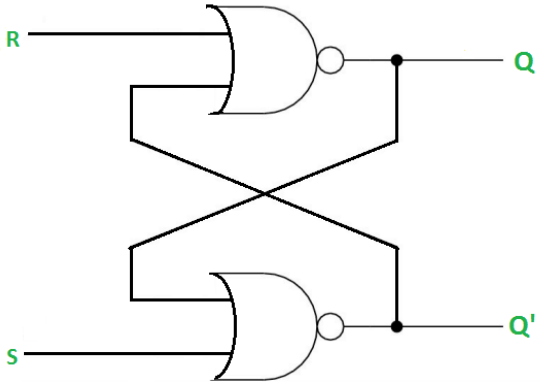


Figure 4. SR latch with NOR gate

S	R	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Table 3. SR NOR latch truth table

$\bar{S}$	$\bar{R}$	Q	$\bar{Q}$
1	1	Q	$\bar{Q}$
0	1	1	0
1	0	0	1
0	0	1	1

Table 4. SR NAND latch truth table

A control input (C) can be added to the SR Latch. If C=0 the output Q will not be changed for any values of R and S but if C=1 the circuit will work normally.

Table 5. SR latch with control input truth table

E	S	R	Q	Q_{next}	Q_{next}'
0	×	×	0	0	1
0	×	×	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	×	0	1
1	1	0	×	1	0
1	1	1	×	1	1

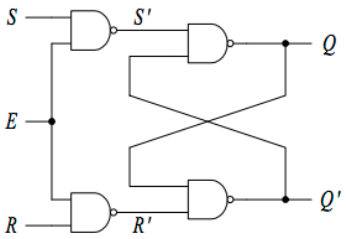


Figure 5. SR latch with control input

2.D-latch:

D latch, which has just two inputs: DATA and ENABLE. When a HIGH is received at the ENABLE input, the DATA input is copied to the output. Even if the ENABLE input then goes low, the output remains unchanged.

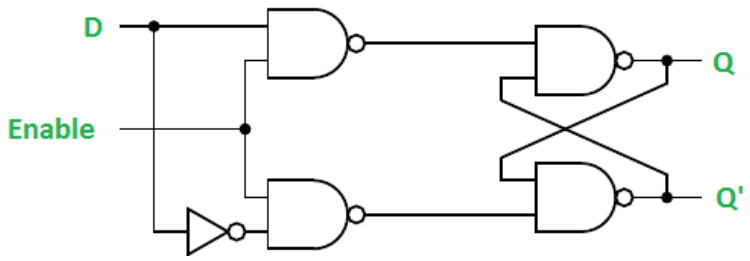


Figure 6. D latch

Table 6. D latch truth table

Enable	D	Q(n)	Q(n+1)	STATE
1	0	x	0	RESET
1	1	x	1	SET
0	x	x	Q(n)	No Change

Flip-Flops:

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. It is the basic storage element in sequential logic. The basic difference between a latch and a flip-flop is a gating or clocking mechanism. Flip Flop is edge-triggered and a latch is level triggered.

Block diagrams and truth tables of the different types of flip-flops are as follows:

1. D-Flip-Flop:

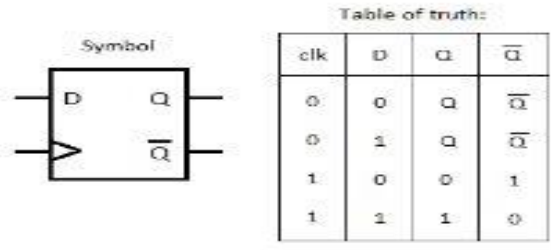
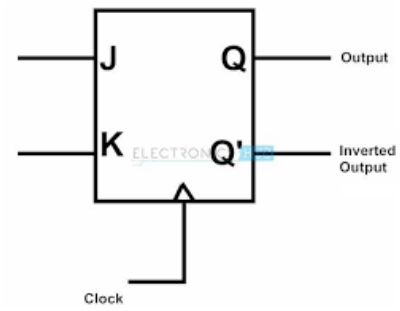


Figure 7. D-Flip-Flop

2. JK-Flip-Flop:

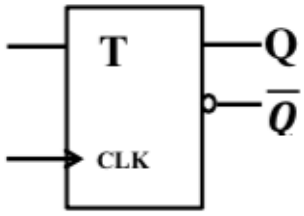


J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

Figure 8.JK-flip-flop

Table 7.JK-flip-flop truth table

3. T-Flip-Flop:



The truth table of a T-flip-flop

T	Q_{t+1}
0	Q_t
1	$\overline{Qt}$

Figure 9. T-Flip-Flop

Registers:

Registers are used to hold binary data. The register is a collection of cascaded Flip-Flops connected together. In registers all Flip-Flops share a common clock and they all reset together. N-bit register require N Flip-Flops.[1]

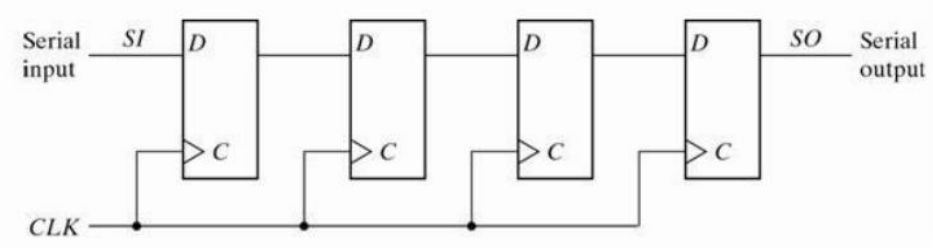


Figure 10. 4-Bit register

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.

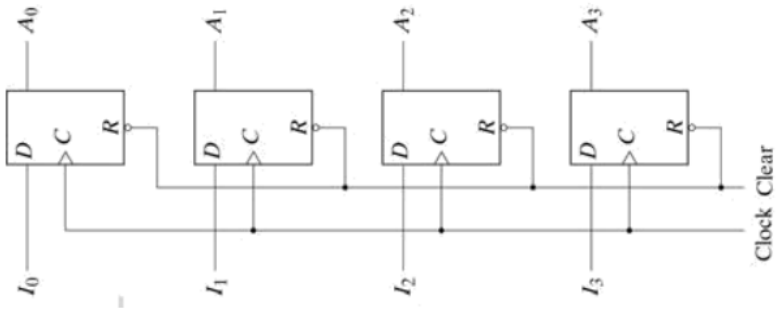


Figure 11. 4-bit shift- right register

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. [1]

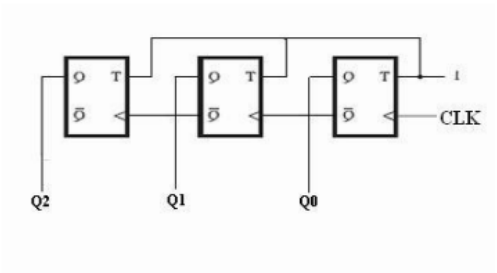


Figure 12. 3-bit synchronous counter

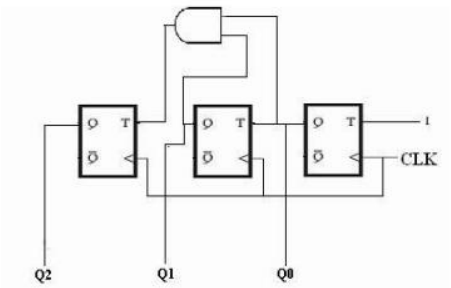


Figure 13. 3-bit ripple counters

Procedure & Discussion:

1. Latches & Flip-Flops:

A) Constructing RS Latch with Basic Logic Gates:

We used IT-3008 module, and connect the inputs A3, A4, to Pulser Switches SWA A(TTL), SWB B(TTL), and connect the output F6, F7 to logic indicators L1, L2.

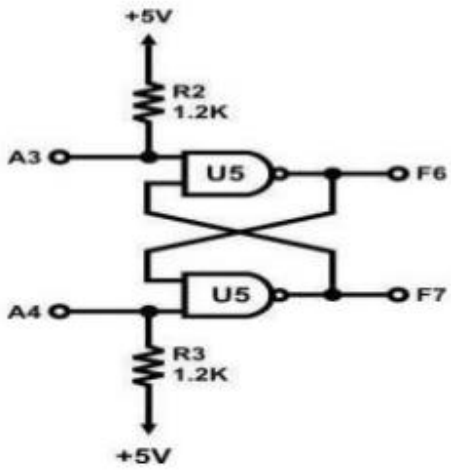


Figure 14. RS latch

Table 8. RS latch truth table

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

We noticed that when A3 and A4 were set to equal zero, both Q and Q' became 1 which is un-valid state, so this case is prohibited, if A3 = 0 and A4 = 1(SET state), then Q equal 1. And when reversing both inputs (A3 = 1, A4 = 0) Q will be 0 (RESET state), if both inputs were set to equal 1, no change will happen to the output it will remain the same as before.

B) Constructing RS latch with control input:

We use IT-3008 module. Connect inputs A1, A5 to Pulser Switches SWA A, SWB B.

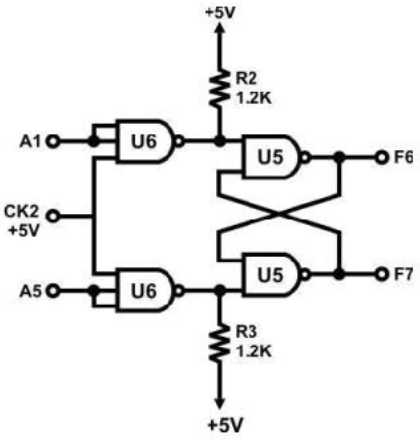


Figure 15. RS Latch with control input

Table 9. . RS Latch with control input truth table

S(A1)	R(A5)	Q(F6)	Q`(F7)
0	0	Q(t)	Q(t)
0	1	0	1
1	0	1	0
1	1	1	1

This circuit is active high. When the two input states are zeros, Q and Q` Will keep as the previous output this condition is called “no change”. And, when the input A1 (S) is 1, the output is 1 (set condition), when A5 (R) is 1, the output is 1 (reset condition). When both A1&A5 are ones, both outputs will equal one, so it’s un-valid state.

C) Constructing D latch with RS latch:

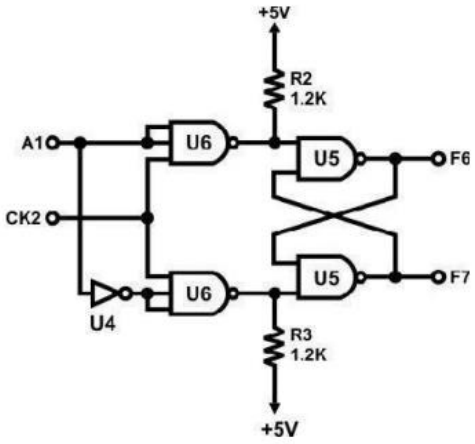


Figure 16. D Latch

Table 10. D latch truth table

CK2	A1	F6
0	0	memory
0	1	memory
⎍	0	0
⎍	1	1

When we set the clock to zero, the output will be equal to the previous output Q(t) no matter A1 was, but after changing the clock into 1 and setting the input to 0 the output will equal zero, and after setting the input A1 to 1 the output will equal one, after any case of the last two cases, if we changed the clock back to zero the output will keep holding the result it already got from the last time.

D) Constructing JK latch with RS latch:

We use IT-3008 module to construct the circuit. And connect CK2 to SWB B output; A1 and A5 to Switches 0, 1, and connect F6 to Led 1.

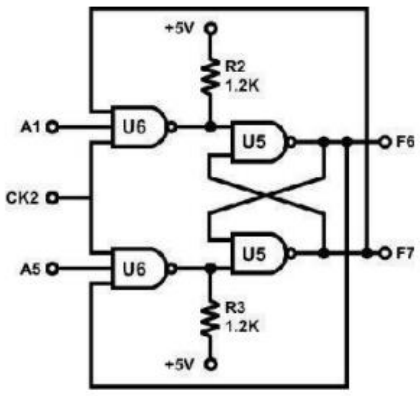


Figure 17. : JK Latch

Table 11. JK latch truth table

CK	A1	A5	F6
⎍	0	0	memory
⎍	0	1	0 Reset
⎍	1	0	1 Set
⎍	1	1	Complement

AS we noticed 0,0 is the memory state, the output will equal the previous output always.

And if inputs equal 0,1 this is the reset state so Q will equal 0, also if inputs equal 1,0 this is the set state the output will equal 1, finally, input 1,1 will give the complement state.

E) Constructing JK Flip-flop with master- slave RS latches:

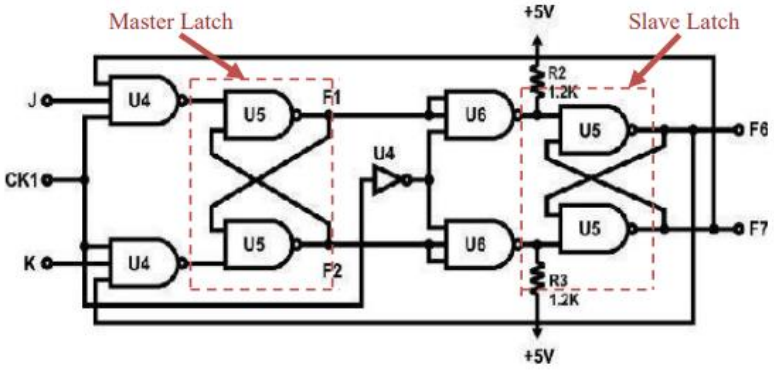


Figure 18. JK Flip-Flop

Table 12. JK Flip-Flop truth table

CK	K	J	F1	F2	F6	F7
⎍	0	0	1	0	1	0
⎍	0	1	0	1	1	0
⎍	1	0	1	0	0	1
⎍	1	1	0	1	1	0
⎍	1	1	1	0	0	1

Registers:

A) Constructing Shift Register with D Flip-Flops:

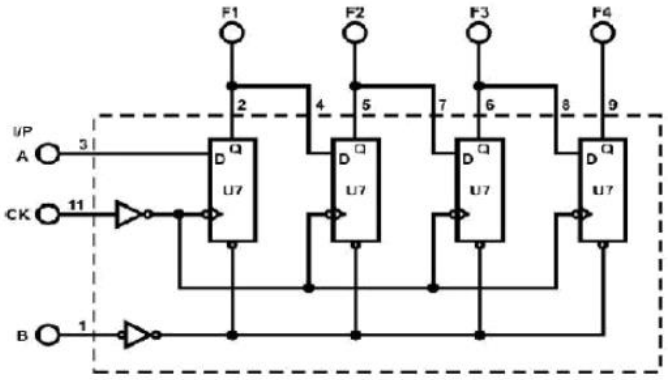


Figure 19. Shift Right Register

When (A) equal one. And sending a clock signal to the CLK input of the Flip-Flops all outputs (F1- F4) will become one sequentially.

When (A) equal zero. And sending a clock signal to the CLK input of the Flip-Flops all outputs (F1- F4) will become 0 sequentially. And then but(A) another time equal to zero, and sending a clock signal to the CLK input again all outputs will stay zero and nothing will change. Finally, At A= 1 again and after sending a clock signal all outputs will become 1 sequentially.

B) 4-Bit Shift Register with serial and parallel load:

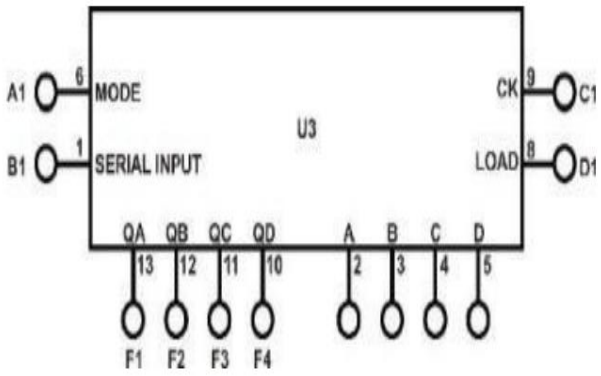


Figure 20. shift register with serial and parallel load

We connected the input A, B, C, and D to SW0, SW1, SW2, SW3, and outputs F1,F2,F3,F4 to L0,L1,L2L,L3. B1 to DIP2.0 and A1 to DIP2.1.

-When We connected CK(C1) to the clock generator TTL level output and change data at B1 with DIP2.0:

Table 13. Shift register with serial load table

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

-We connected LOAD(D1) to the clk generator TTL level outputs at 1 HZ and we set A1 to 1.

Table 14. : Shift register with parallel load table

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

Counters:

A) 2-bit Synchronous Counter:

This counter is implemented using two JK flip-flops. It counts up from 0 to 3.

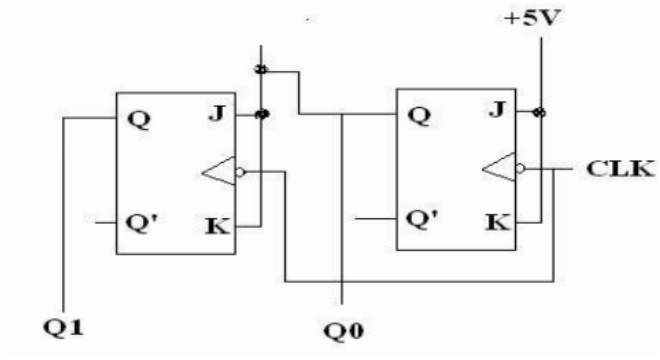


Figure 21. 2-bit Synchronous Counter

Table 15. 2-bit Synchronous Counter truth table

CLK	Q1	Q0
⎍	0	0
⎍	0	1
⎍	1	0
⎍	1	1
⎍	0	0
⎍	0	1
⎍	1	0
⎍	1	1

-We noticed that in every clock, the counter increased by one until it reached the number three and then returned to zero again.

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

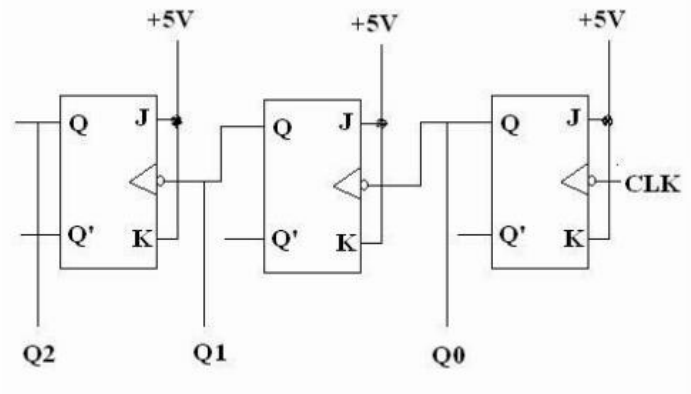


Figure 22. : 3-bit Ripple Counter

Table 16. 3-bit Ripple Counter truth table

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

-This counter counts from 0 to 7 and then goes back to zero again.

C) BCD Counter:

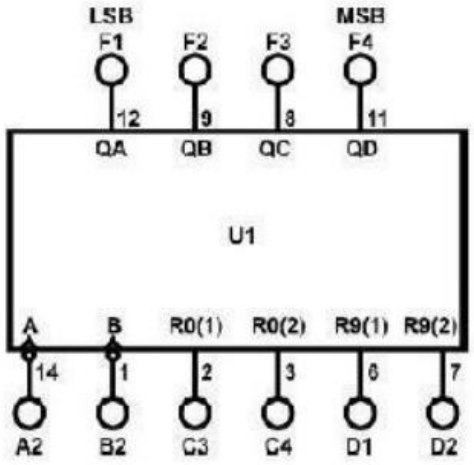


Figure 23. IC 7490 BCD Counter

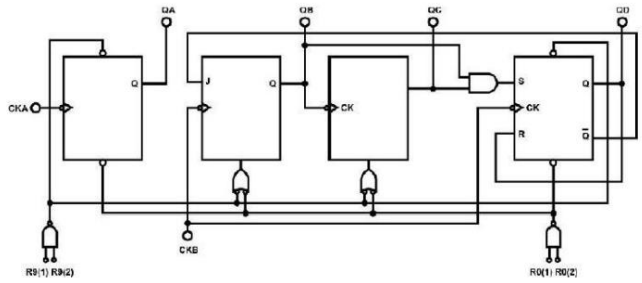


Figure 24. IC 7490 BCD Counter (2)

-This counter counts the BCD numbers from 0-9 and then return to 0 again.

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

- using the same BCD chip counter as before we Connect R0(2) (pin3) to +5V, and connect R0(1) (pin2) to QD (pin11) output. This will make counter reset after 111 (or 7).

Tasks:

Task2: Modifying 3-bit ripple counter to be 3-bit synchronous counter:

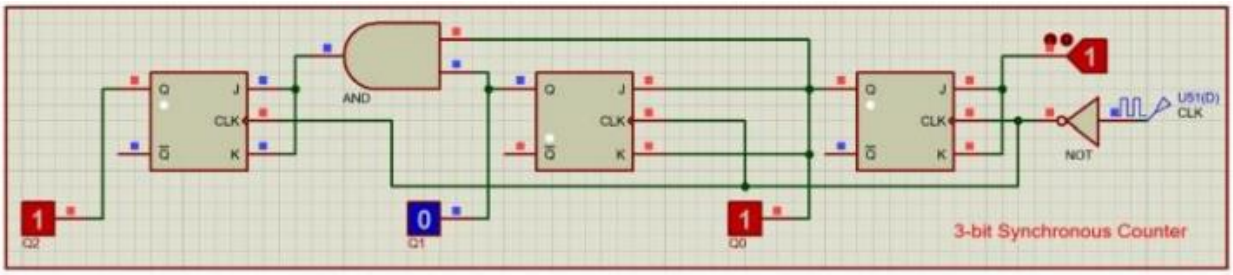


Figure 25. 3-bit synchronous counter

Task3: Change the connection of the BCD counter to make two counters:

- 0-to-5 Counter:

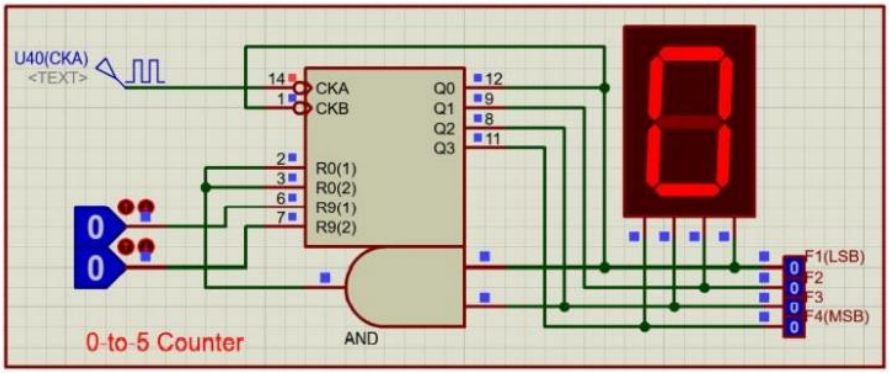
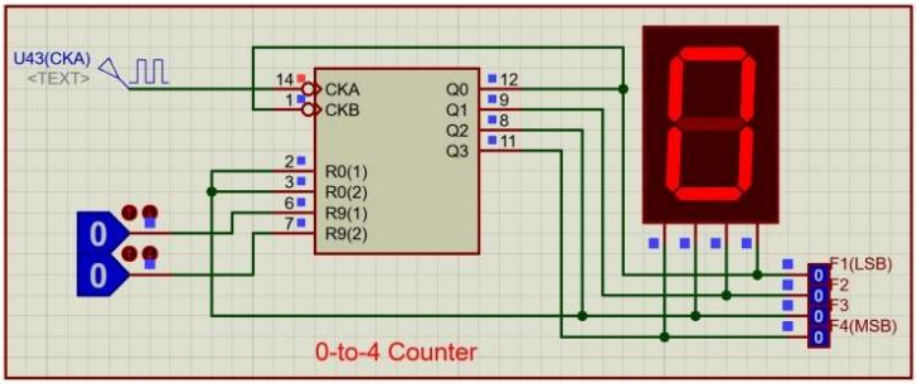


Figure 26. 0-to-5 Counter

-0-to-4 Counter:



Discussion:

1) Because latch is level-triggered (outputs can change as soon as the inputs changes) a while Flip-Flop is edge-triggered (only changes state when a control signal goes from high to low or low to high). And Latches will cause race condition which leads to error in the results.
2) The main disadvantage is when the R and S equal 1 in this case the value of Q and Q' will be the same, and this is neither logical nor true.
3) The main difference between them, is that in the Ripple Counter each flip-flops are used with a different clock pulse. While in the synchronous counter all flip-flops have the same clock, so it's faster than ripple counters.

Conclusion:

After completing this experiment, we got acquainted with sequential logic circuits and differentiated between the different types of latches (SR, D) and the different types of flip flops (JK, D, T). and we realized the main difference between latches and flip flops that is the latch is level sensitive and is used in asynchronous circuits, while flip flop is edge triggering and used in synchronous circuits. Also counters, which are an application of flip fops, were implemented using flip flops and ICs. And the difference between synchronous and asynchronous counters. In the final analysis, the theoretical outputs were seen as outcomes to the practical parts.

References:

- [2] DIGITAL ELECTRONICS – Atul P. Godse, Mrs. Deepali A. Godse
- [1] (Digital Electronics and Computer Organization Lab manual)
- Hamacher, C., Vranesic, Z., Zaky, S., & Manjikian, N. (2012).