Birzeit University

Faculty of Information Technology
Computer Systems Engineering Department

Digital Lab ENCS 211 EXP. No. 5

Sequential Logic Circuits

5.1 OBJECTIVES

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

5.2 EQUIPMENT REQUIRED

1. KL-22001 Basic Electricity Circuit Lab
2. KL-26006 sequence Logic Circuit Experiment Module
3. KL-26007 sequence Logic Circuit Experiment Module

5.3 PRE LAB

Prepare all sections and be ready for a quiz!

5.4 INTRODUCTION

5.4.1 Sequential Circuits

Any digital circuit could be classified as combinational or sequential circuit. Combinational logic circuits implement Boolean functions. Boolean functions are mappings of inputs to outputs. These circuits are functions of input only.
Sequential circuits are *two-valued* networks in which the outputs at any instant are dependent not only upon the inputs present at that instant but also upon the past history (sequence) of inputs. Sequential circuits are classified into: The block diagram of a sequential circuit is shown in Figure.1. The basic logic element that provides memory in many sequential circuits is the *flip-flop*.

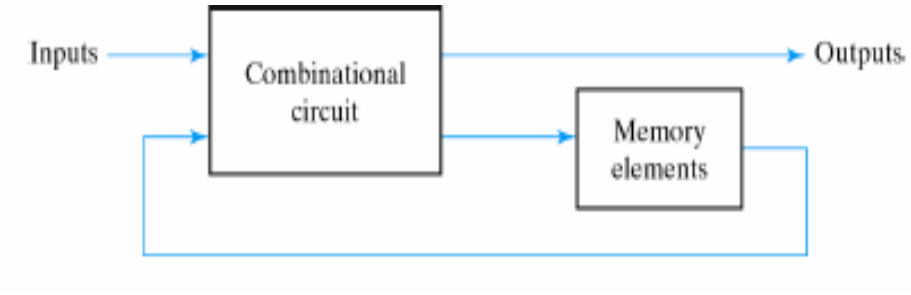


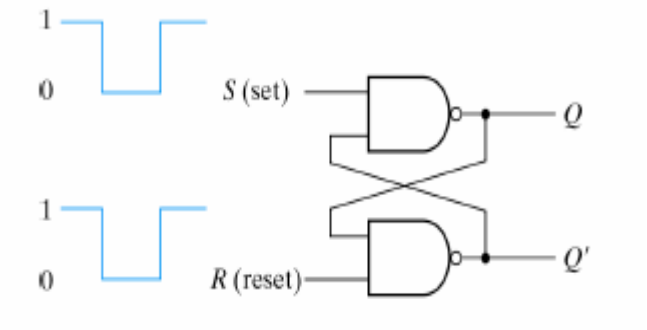
Figure.1: Sequential Circuit Block Diagram

5.4.2 Latches

Latches form one class of flip-flops. This class is characterized by the fact that the timing of the output changes is not controlled. Although latches are useful for storing binary information and for the design of asynchronous sequential circuits, they are not practical for use in synchronous sequential circuits.

5.4.2.1 The SR (set-Reset) Latch

It is a circuit with two cross-coupled NOR or NAND gates. The one with *NAND* gates is shown in Figure.2. 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. The condition that is undefined is when both inputs are equal to 0 at the same time.



S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

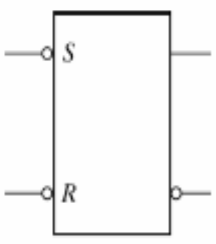
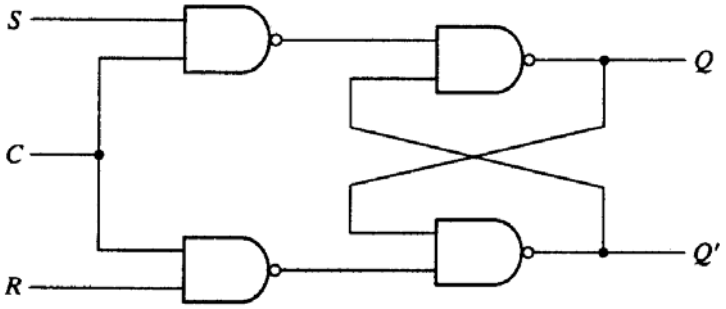


Figure.2: SR latch with NAND gate

RS latch with control input C is shown in figure.3. If C=0 the output Q does not change regarding less the R and S values. If C= 1the circuit will work normally.

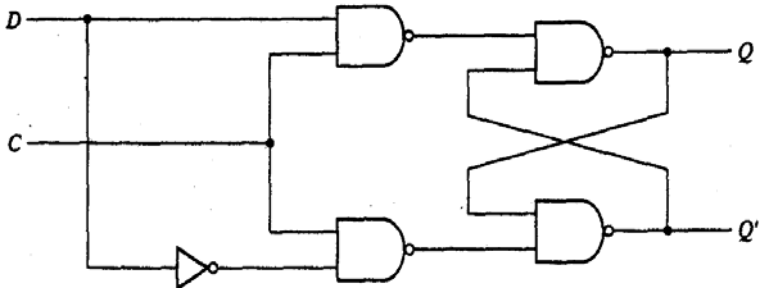


C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; set state
1	1	1	Indeterminate

Figure.3: RS latch with control input

5.4.2.2 The D Latch

To eliminate the undefined condition of the indeterminate state in the RS latch, the D latch was developed. The D latch and its state table is shown in figure.4



C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

Figure.4: D-Latch

5.4.3 Flip-Flops

Also a flip used for storage binary information as the latches, but the different is:
The output change in the flip-flop occurs only at the clock edge while in the latch it occurs at the clock level.
A flip-flop can be implemented using two separated latches. Figure.5 shows the D flip flop implemented with two D latches.

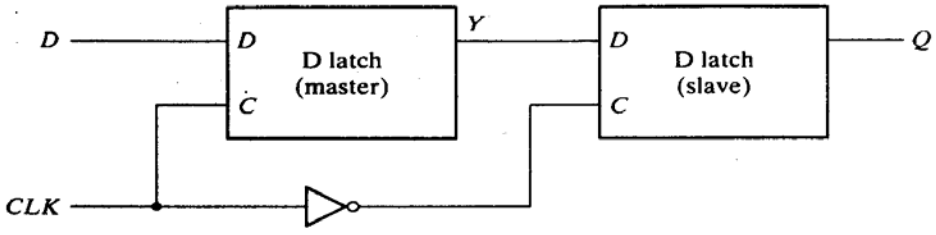


Figure.5: D flip flop implemented with two D latches

There are several types of flip-flops, the common ones are D, T, and JK flip flops. Figure.6 shows these flip flops and their function tables.

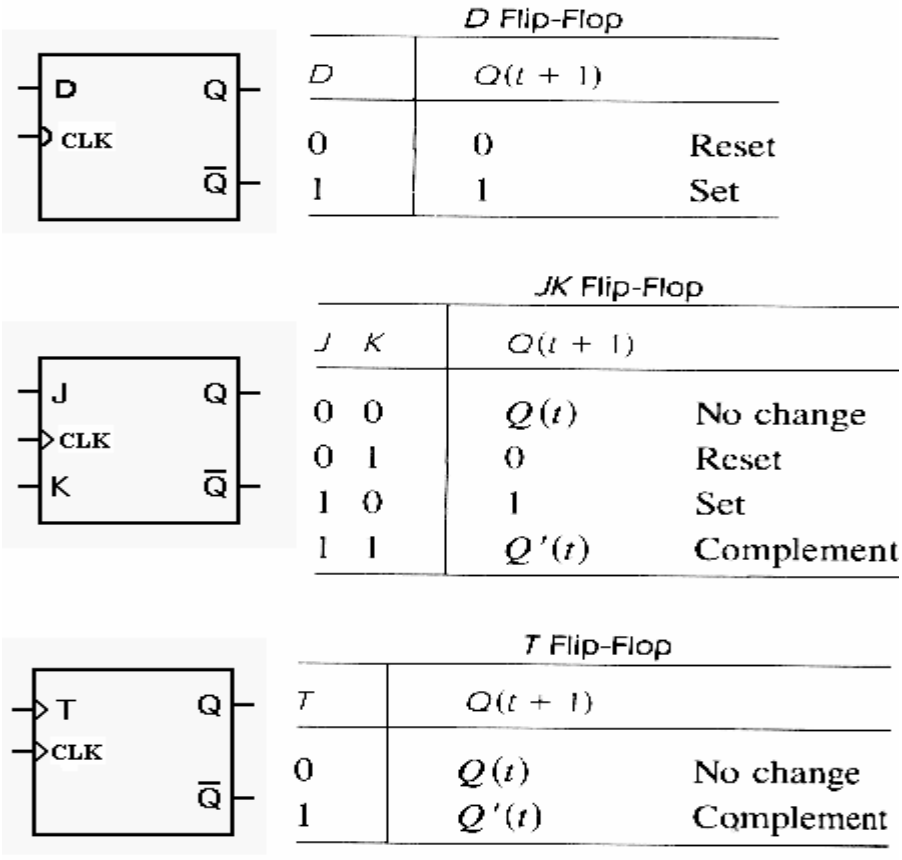


Figure.6: D, JK, and T flip flops

5.4.4 Registers

Digital systems use registers to hold binary entities. The register is a collection of flip flops; N- bit register consists of N flip flops. Figure.7 shows simple 4-bit register implemented with D- flip flops. All the flip-flops are driven by a common clock, and all are reset simultaneously

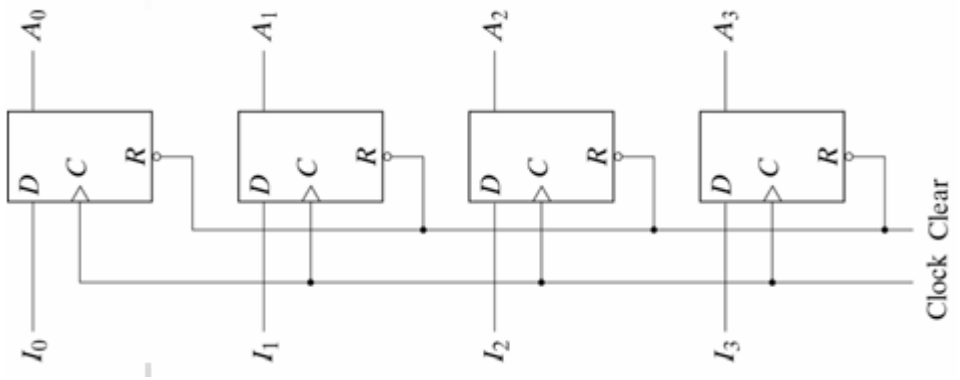


Figure.7: 4-bit Register

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.8 shows 4-bit shift- right register.

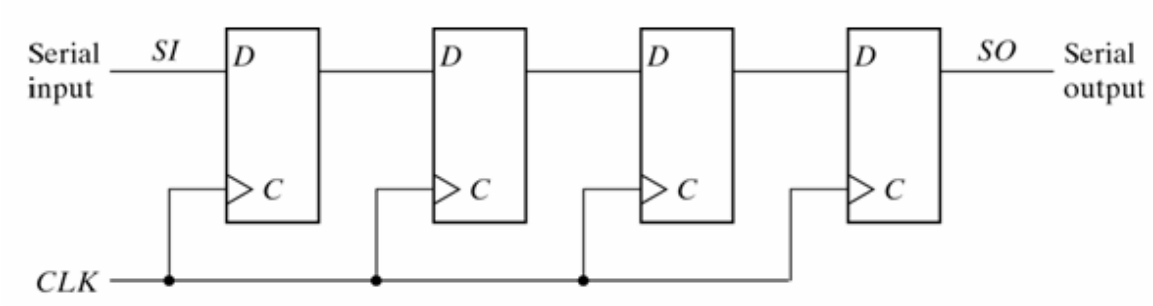


Figure.8: 4-bit shift- right register

5.4.5 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. 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. Figure.9 shows 3-bit ripple and synchronous counters

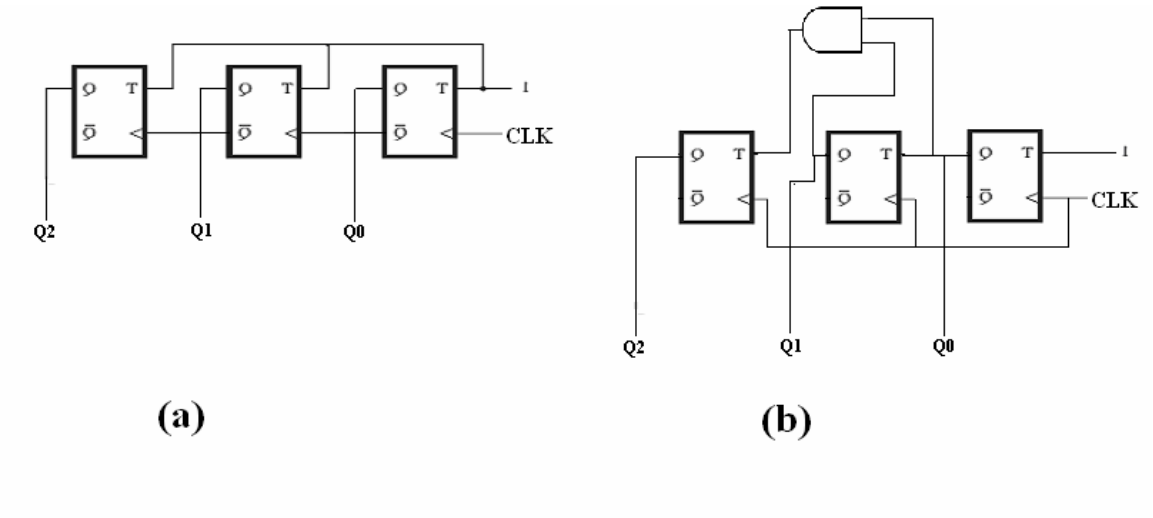


Figure.9: (a) 3-bit ripple counters, (b) 3-bit synchronous counter

5.5 PROCEDURE

5.5.1 Latches and Flip flops:

A) Constructing RS latch with Basic Logic Gates

Use KL-26006 module to construct the circuit shown in Figure.10. Connect switches to S and R inputs, and connect Q and Q' output to indication lamps. Follow the sequences in Table 5.1

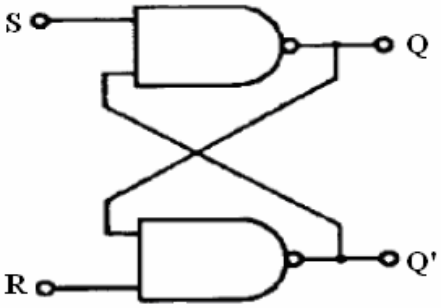


Figure.10: RS latch

S	R	Q	Q'
0	0		
0	1		
1	0		
1	1		

Table 5.1

B) Constructing RS latch with control input

Use KL-26006 to connect the circuit shown in Figure.11. Connect CK2 (control input) to logic 1 (+5V), and follow the input sequence in Table 5.2.

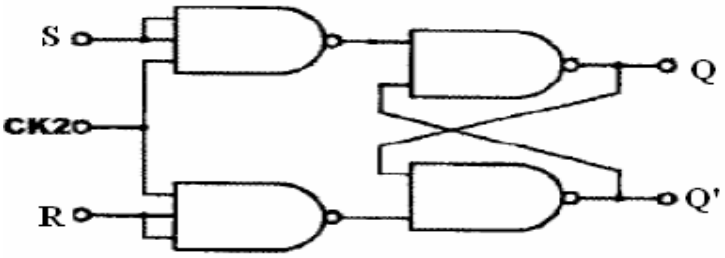


Figure.11: RS Latch with control input

S	R	Q	Q'
0	0		
0	1		
1	0		
1	1		

Table 5.2

C) Constructing D latch with RS latch

Use KL-26006 module to construct the circuit shown in Figure.12. Connect CK2 to **Pulser switch**. Follow the sequences in Table 5.3

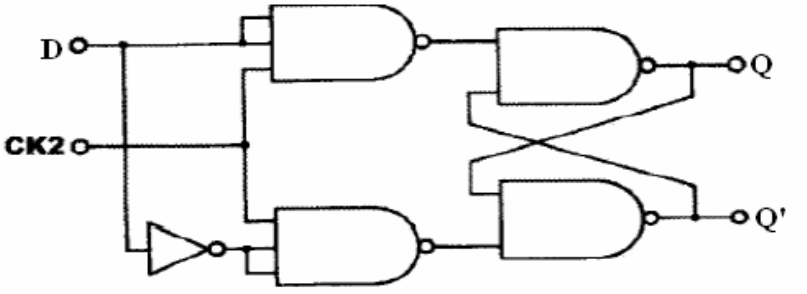


Figure.12

CK2	D	Q	Q'
0	0		
0	1		
⎍	0		
⎍	1		

Table 5.3

D) Constructing JK latch with RS latch

Use KL-26006 module to construct the circuit shown in Figure.13. Connect CK2 to **Pulser switch**. Follow the sequences in Table 5.4

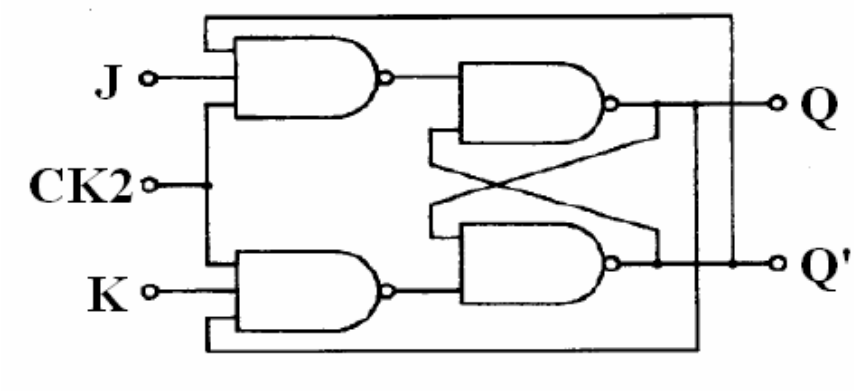


Figure.13: JK Latch

CK2	J	K	Q	Q'
⎍	0	0		
⎍	0	1		
⎍	1	0		
⎍	1	1		

Table 5.4

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

Use KL-26006 module to construct the circuit shown in Figure.14. Connect CK2 to **Pulser switch**. Follow the sequences in Table 5.5

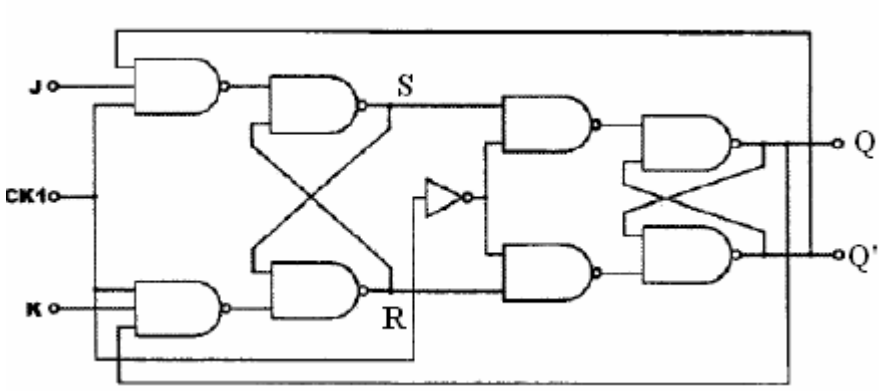


Figure.14: JK Flip-Flop

CK1	K	J	S	R	Q	Q'->	S	R	Q	Q'
⎍	0	0				->				
⎍	0	1				->				
⎍	1	0				->				
⎍	1	1				->				
⎍	1	1				->				

Table 5.5

5.5.2 Registers

A) Constructing Shift Register with D Flip-Flops

1. Use KL-26006 module (block a) to construct the circuit shown in Figure.15

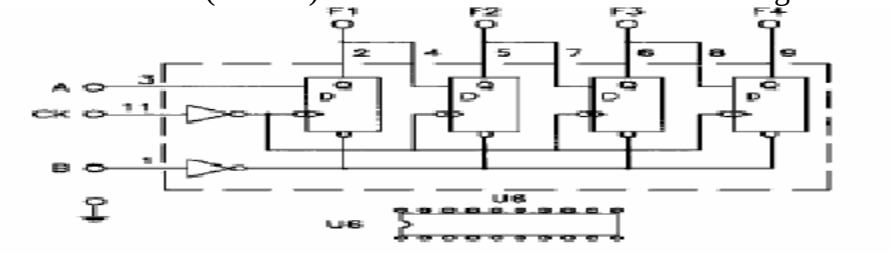


Figure.15: Shift Right Register

2. Connect A (serial input) to switch.
3. Connect B (clear) to other switch.
4. Connect the outputs F1, F2, F3 and F4 to indication lamps.
4. Connect CK to Pulser switch.
5. Set B switch to 1 to clear the register, and then keep it on 0.
6. Set A to 1, then, apply three clock pulses to CK using the pulser. Observe the output.
7. Load **1011** value into the register; **show this to your instructor or lab assistant**.

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

Block b in KL-26006 module is IC 7495 which is 4-Bit Shift Register with serial and parallel synchronous operating modes. It has a Serial input and four Parallel (A–D) Data inputs and four Parallel Data outputs (QA–QD). Figure.16 (a) shows the top view of IC7495 package and Figure.16 (b) shows block b in KL-26006 module.

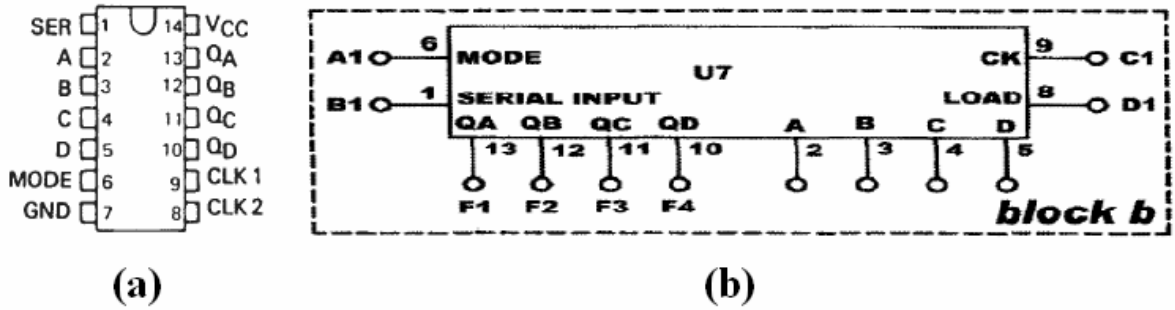


Figure.16: IC7495 shift register with serial and parallel load

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

When the Mode Control input (pin 6) is LOW, CLK1 (pin 9) is enabled. Note that CLK1 is used with serial input mode only

a. Connect the serial input (pin1) to one of the input switches.
b. Connect the outputs QA, QB, QC and QD to indication lambs
c. Connect CLK1 (pin9) to the pulser switch.
d. Apply clock pulses to CLK1 and observe the serial data transfer.
e. Load the register with 1101 value. Show this to your instructor or to your lab assistant.

2. **Parallel- load Register**

When the Mode Control input (pin 6) is HIGH, CLK2 (pin 8) is enabled. Note that CLK2 is used with parallel- input mode only.

To change the mode, the two clocks CLK1 and CLK2 must be at LOW STATE!

a. Connect the inputs A, B, C, and D to switches.
b. Connect the outputs QA, QB, QC and QD to indication lambs
c. Connect CLK1 (pin8) to the pulser switch.

d. Apply clock pulses to CLK1 and observe the parallel data transfer.
e. Load the register with 1101 value. Show this to your instructor or to your lab assistant.

TASK1: How we get **Serial shift-left** using IC7495? Which mode must be used? And what the connections we have to make? Connect the circuit, show it to your instructor or to your lab- assistant, and attach the design with this experiment report.

5.5.3 Counters

A) 2-bit Synchronous Counter

1. Use KL-26007 module, block c, which contains three separated JK flip flops to implement the 2-bit synchronous counter shown in Figure.17.
2. Connect CLK input to pulser switch.
3. Connect counter outputs Q1 and Q0 to indication lamps.
4. Apply clock pulses to CLK input. Observe and record the outputs in Table 5.6 (a)
5. Apply counter outputs Q1 and Q0 to **D1 display** inputs. Observe and record the outputs in Table 5.6 (b).

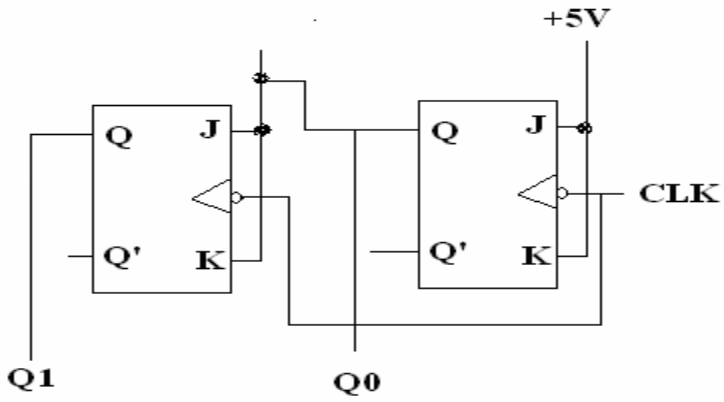


Figure.17: 2-bit Synchronous Counter

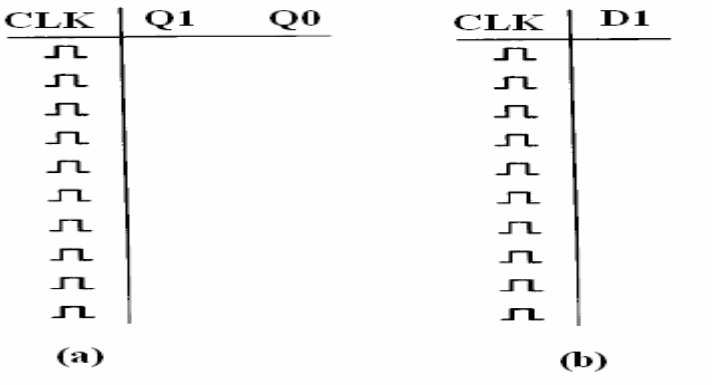


Table 5.6

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

Divide-by-8 counter is 3-bit counter that counts from 0-to-7:
1. Use KL-26007 module, block c, which contains three separated JK flip flops to implement the 3-bit (divide by eight) Ripple counter shown in Figure.18.
2. Connect CLK input to pulser switch.
3. Connect counter outputs Q2, Q1 and Q0 to indication lamps.
4. Apply clock pulses to CLK input. Observe and record the outputs in Table 5.7 (a).
5. Apply counter outputs Q2, Q1 and Q0 to **D1 display** inputs. Observe and record the outputs in Table 5.7 (b).

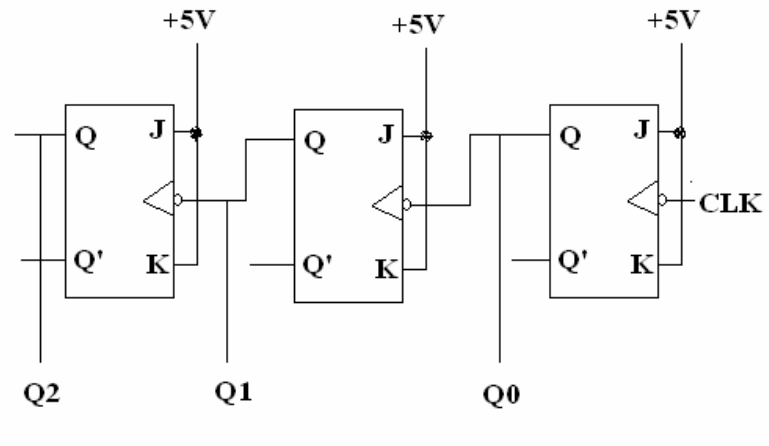


Figure.18: 3-bit Ripple Counter

CLK	Q2	Q1	Q0
⎍			
⎍			
⎍			
⎍			
⎍			
⎍			
⎍			
⎍			
⎍			
⎍			

(a)

CLK	D1
⎍	
⎍	
⎍	
⎍	
⎍	
⎍	
⎍	
⎍	
⎍	
⎍	

(b)

Table 5.7

Task2: Modify the circuit in Figure.16 to be 3-bit **Synchronous Counter**. Attach the design with this experiment report.

C) BCD Counter

Locate BCD counter (IC 7490) on KL-26007 module, block b. Connect the circuit shown in Figure.19.

1. Connect RO(1) and RO(2) (pins 2& 3) to 0v, note that RO(1) and RO(2) are two inputs to internal NAND gate which used to reset the counter.
2. Connect clock INPUT 1 (pin 14) to pulser switch.
3. Connect the outputs A, B, C, and D to indication lamps.
4. Apply clock pulses to INPUT 1, and observe the count sequence (0000-1001).
5. Apply counter outputs A, B, C, and D to **D1 display** inputs. Apply clock pulses to INPUT 1. Observe the count sequence on the display.
6. Connect the INPUT 1 to the output of the clock generator. Observe the count sequence on the display.

Show this to your instructor or to your lab-assistant

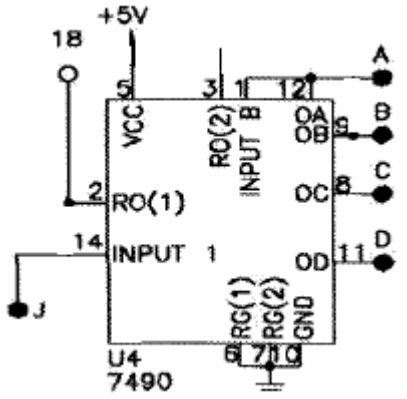


Figure.19: IC 7490 BCD Counter

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

1. Connect RO(2) (pin3) to +5V, and connect RO(1) (pin2) to QD (pin11) output. This will make counter reset after 111 (or 7). WHY?
2. Connect clock INPUT 1 (pin 14) to pulser switch.
3. Connect the outputs A, B, C, and D to indication lamps.
4. Apply clock pulses to INPUT 1, and observe the count sequence (0000-0111).
5. Apply counter outputs A, B, C, and D to **D1 display** inputs. Apply clock pulses to INPUT 1. Observe the count sequence on the display.
6. Connect the INPUT 1 to the output of the clock generator. Observe the count sequence on the display.

Task3: change the connection of counter in Figure.19 to count from:

- **0-to-5**
- **0-to 4**

5.6 DISCUSSION

Answer the following questions:

1. Although latches are useful for storing binary information, they are rarely used in the sequential circuits design, why?
2. What is the disadvantage of the RS flip flop?
3. What is the different between “synchronous” and “ripple” counters?