Digital Systems Laboratory

Rev 1.2
Agust 2014

LIST OF EXPERIMENTS

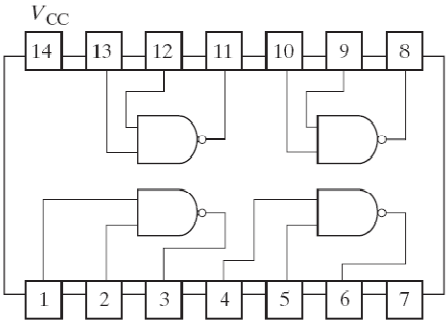
1. INTRODUCTION TO LAB, USING MATERIALS
2. DIGITAL LOGIC GATES
3. INTRODUCTION TO PROTEUS
4. BINARY AND DECIMAL NUMBERS
5. CODE CONVERSION I
6. CODE CONVERSION II
7. ADDERS/SUBTRACTORS
8. MULTIPLEXERS
9. FLIP-FLOPS
10. COUNTERS AND SEQUENTIAL LOGIC

Experiment	**Date**
1	
2	
3	
4	
5	
6	
7	
8	
9	

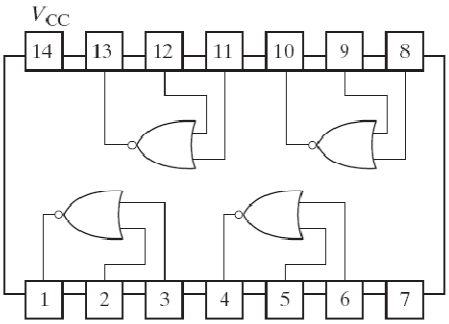
LIST OF THE ITEMS WILL USE FOR THE LAB.

IC Number	Description	Experiment #
7400	Quadruple 2-input NAND gates	
7402	Quadruple 2-input NOR gates	
7404	Hex Inverter	
7408	Quadruple 2-input AND gates	
7410	Triple 3-input NAND gate 4	
7420	Dual 4-input NAND gates 4	
7432	Quadruple 2-input OR gates	
7447	BCD-to-seven segment decoder	
7476	Dual JK master-slave flip-flop	
7483	4-bit binary adder	
7485	4-bit magnitude comparator	
7486	Quadruple 2-input XOR gates	
7493	4-bit ripple counter	
74151	8x1 multiplexer	
74153	Dual 4x1 multiplexer 2	
Display	Seven-segment LED display, common anode	

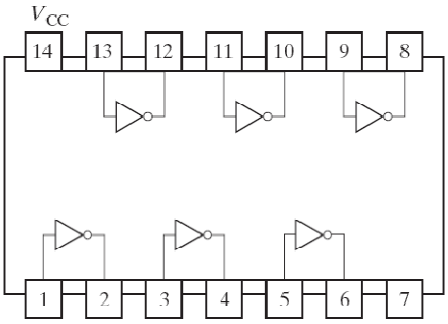
Digital gates in IC packages with identification numbers and pin assignments



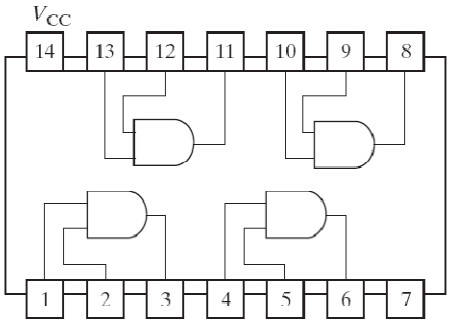
2-input NAND
7400



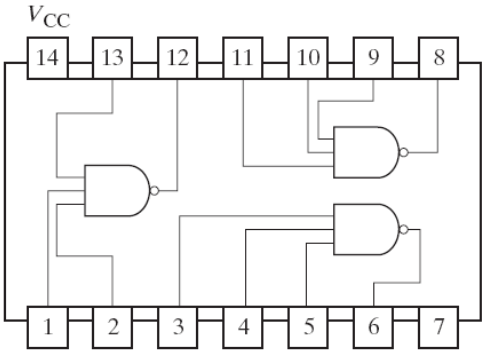
2-input NOR
7402



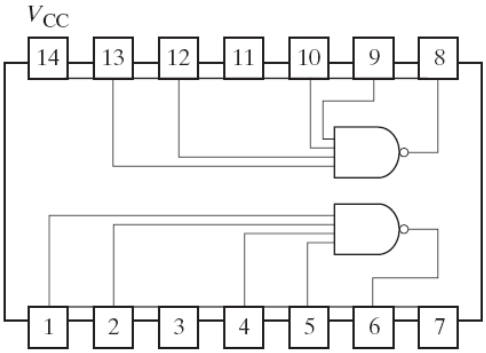
Inverters
7404



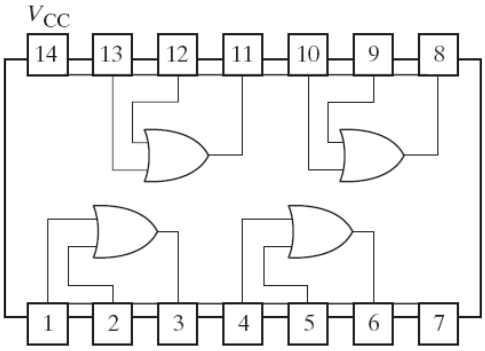
2-input AND
7408



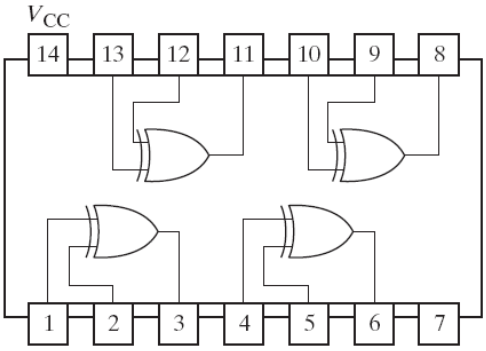
3-input NAND
7410



4-input NAND
7420

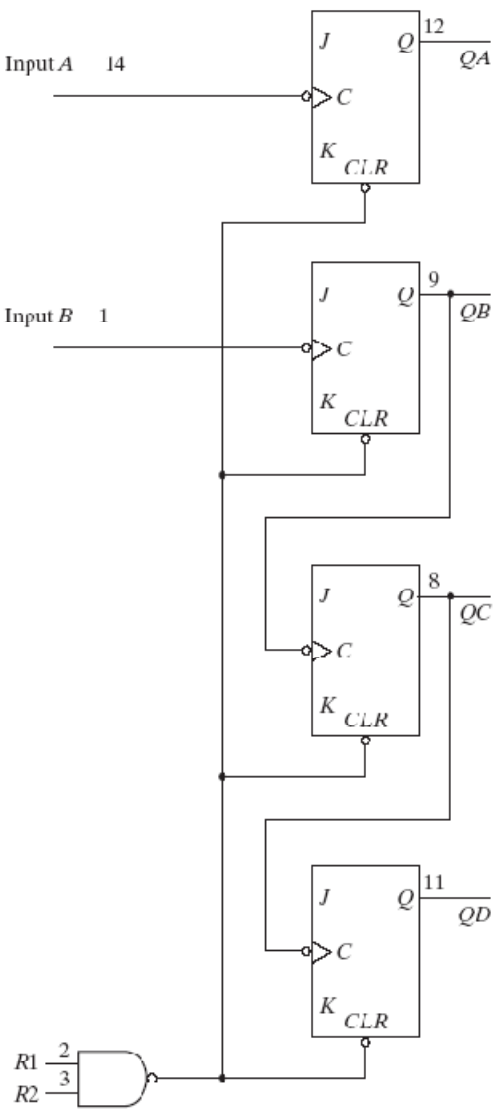


2-input OR
7432

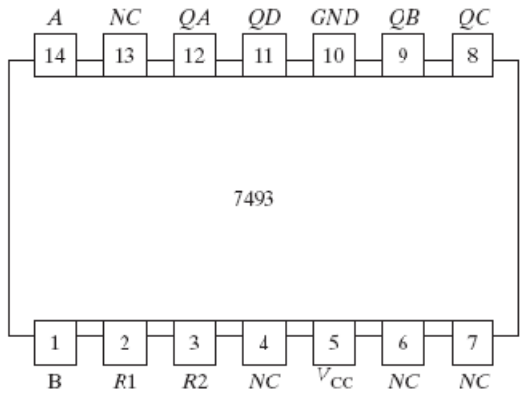


2-input XOR
7486

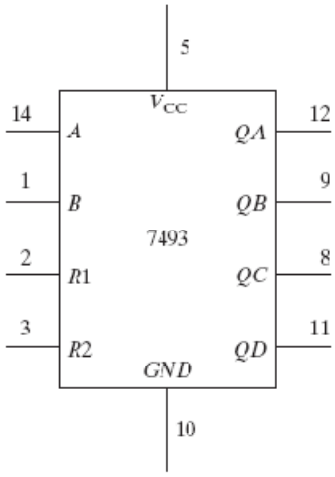
IC Type 7493 Ripple Counter



(a) Internal circuit diagram



(b) Physical layout (*NC*: no connection)



(c) Schematic diagram

Lab 1 : Introduction to Lab, Using materials

Objective

The objective of this exercise is to become familiar with the operation and usage of basic circuit materials, namely resistors, breadboard and Transistor-Transistor Logic (TTL or chip).

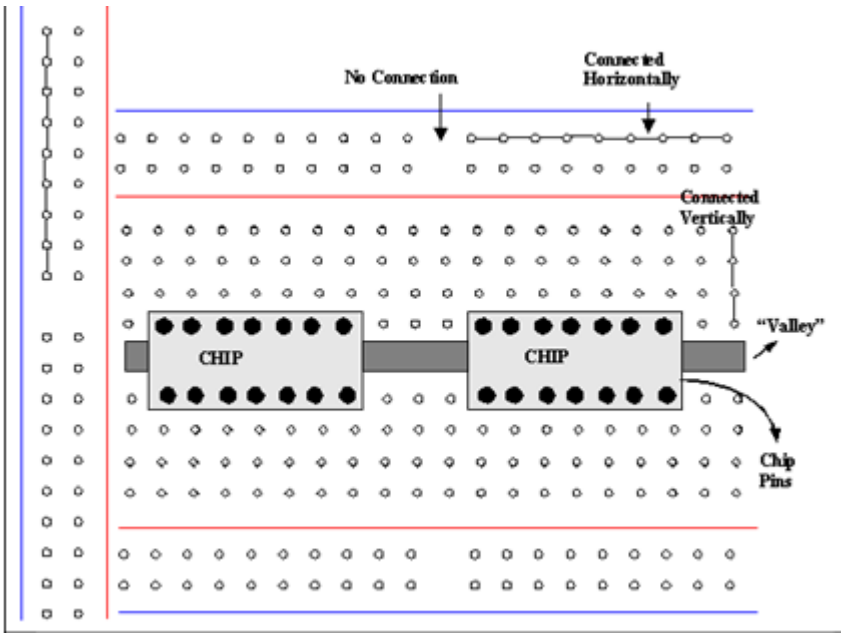
Equipment

(1) Adjustable DC Power Supply
(1) Digital Multimeter
(1) Resistors
(1) Breadboard

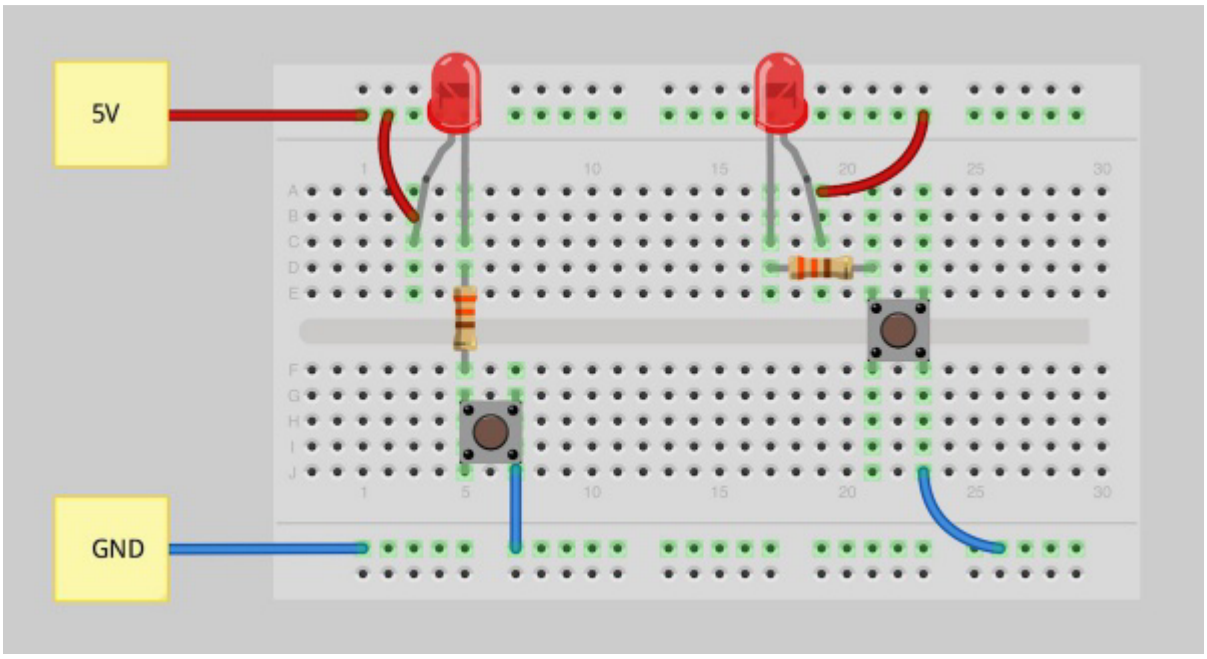
Breadboard:

An electronics breadboard is actually referring to a solder less breadboard. These are great units for making temporary circuits and prototyping, and they require absolutely no soldering.

Prototyping is the process of testing out an idea by creating a preliminary model from which other forms are developed or copied, and it is one of the most common uses for breadboards. If you aren't sure how a circuit will react under a given set of parameters, it's best to build a prototype and test it out. A simple breadboard is shown in the figure. Each color lines are in series each other. The lab assistant teach you how to use the breadboard in lab.



Example:

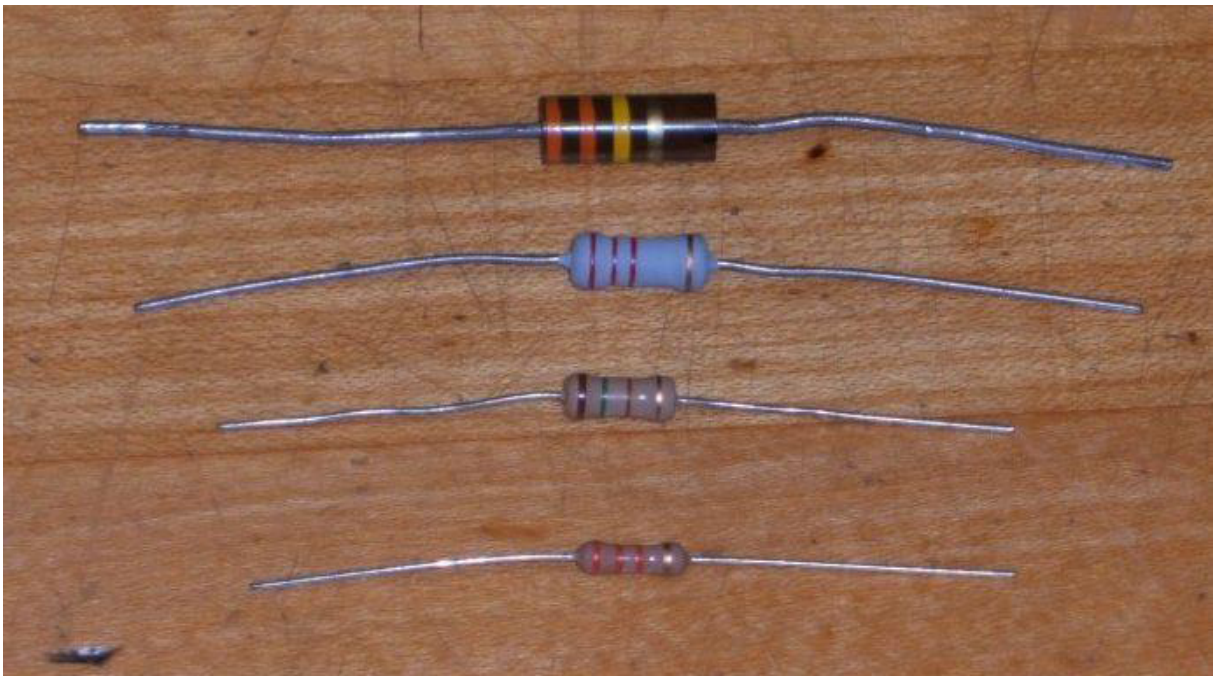


In this figure, simple LED example is given. LED is one of the most important component in electronics gives illumination or light. LEDs have two legs, one of them is anode and the other one is cathode. The small leg is cathode and must be connected to the ground. You can realize by figure, each green color line is series each other.

Resistors:

The resistor is perhaps the most fundamental of all electrical devices. Its fundamental attribute is the restriction of electrical current flow: The greater the resistance, the greater the restriction of current. Resistance is measured in Ohms. The measurement of resistance in unpowered circuits may be performed with a digital multimeter. Like all components, resistors cannot be manufactured to perfection. That is, there will always be some variance of the true value of the component when compared to its nameplate or nominal value. For precision resistors, typically 1% tolerance or better, the nominal value is usually printed directly on the component. Normally, general purpose components, i.e. those worse than 1%, usually use a color code to indicate their value.

The resistor color code typically uses 4 color bands. The first two bands indicate the precision values (i.e. the mantissa) while the third band indicates the power of ten applied (i.e. the number of zeroes to add). The fourth band indicates the tolerance. It is possible to find resistors with five or six bands but they will not be examined in this exercise. Examples are shown below:



It is important to note that the physical size of the resistor indicates its power dissipation rating, not its ohmic value.

Each color in the code represents a numeral. It starts with black and finishes with white, going through the rainbow in between:

0 Black	1 Brown	2 Red	3 Orange	4 Yellow
5 Green	6 Blue	7 Violet	8 Gray	9 White

For the fourth, or tolerance, band:
5% Gold 10% Silver 20% None

For example, a resistor with the color code brown-red-orange-silver would correspond to 1 2 followed by 3 zeroes, or 12,000 Ohms (more conveniently, 12 k Ohms). It would have a tolerance of 10% of 12 k Ohms or 1200 Ohms. This means that the actual value of any particular resistor with this code could be anywhere between 12,000-1200=10,800, to 12,000+1200=13,200. That is, 10.8 k to 13.2 k Ohms. Note, the IEC standard replaces the decimal point with the engineering prefix, thus 1.2 k is alternately written 1k2.

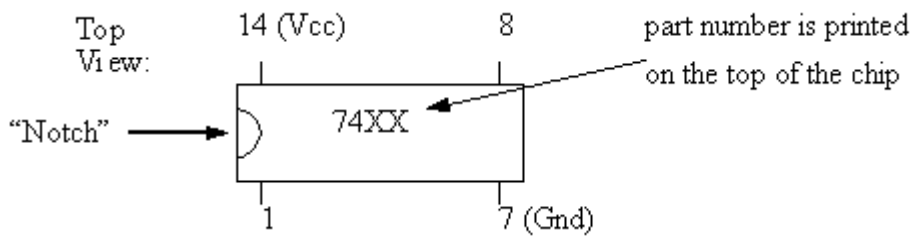
Similarly, a 470 k 5% resistor would have the color code yellow-violet-yellow-gold. To help remember the color code many mnemonics have been created using the first letter of the colors to create a sentence. One example is the *picnic mnemonic* **B**lack **B**ears **R**obbed **O**ur **Y**ummy **G**oodies **B**eating **V**arious **G**ray **W**olves.

Measurement of resistors with a DMM is a very straight forward process. Simply set the DMM to the resistance function and choose the first scale that is higher than the expected value. Clip the leads to the resistor and record the resulting value.

Transistor-Transistor Logic (TTL) Small Scale Integration (SSI) Packages

The chips that you will use in this lab are Small Scale Integration (SSI - meaning not much logic on a single chip) Transistor-Transistor Logic (TTL) family. TTL refers to the nature of the circuit of the digital logic gate; since we will only be using TTL this one time in the lab. It is important to know, however, that the logic probe has a switch that must set appropriately for the different type of logic on it.

All of the TTL chips you will use are “Dual In-line Packages” or DIPs. Most of the packages are 14 pins, and the pins are number from looking at the chip from the top: Below the “notch” is pin 1 to pin 7, and above the notch is pin 14 down to 8.

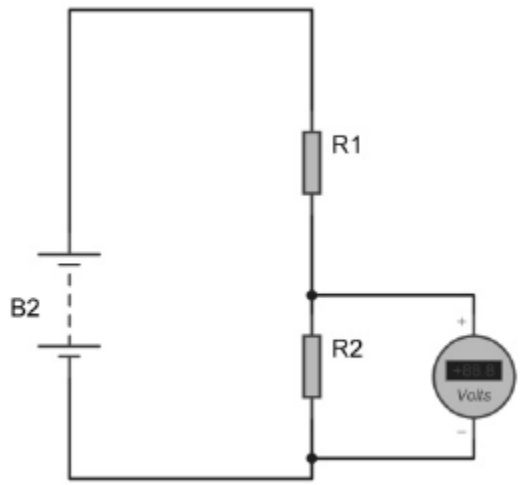


NOTE that Pin 14 must always be connected to VCC (+5V) and pin 7 to ground (0V).

Experiment:

Set up the given following circuit in the breadboard. Measure current and voltage on R2.

R1:1 k ohm
R2: 2 k ohm
B2: 6 V



Lab 2 : Digital Logic Gates

Objectives

• To study the basic logic gates: AND, OR, INVERT, NAND, and NOR.
• To study the representation of these functions by truth tables, logic diagrams and Boolean algebra.
• To observe the pulse response of logic gates.
• To measure the propagation delay of logic gates.

Apparatus

7400 Quadruple 2-input NAND gates
7402 Quadruple 2-input NOR gates
7404 Hex Inverters (x2)
7408 Quadruple 2-input AND gates
7432 Quadruple 2-input OR gates
7486 Quadruple 2-input XOR gate

Theory

AND : A multi-input circuit in which the output is 1 only if all inputs are 1.The symbolic representation of the AND gate is shown in Fig. 2. 1. a.
OR : A multi-input circuit in which the output is 1 when any input is 1. The symbolic representation of the OR gate is shown in Fig. 2. 1. b.
INVERT : The output is 0 when the input is 1, and the output is 1 when the input is 0. The symbolic representation of an inverter is shown in Fig. 2. 1.c.
NAND : AND followed by INVERT. The symbolic representation of the NAND gate is shown in Fig 2. 1.d.
NOR : OR followed by INVERT as shown in Fig. 2.1.e.
EX-OR : The output of the Exclusive –OR gate, is 0 when it's two inputs are the same and it's output is 1 when its two inputs are different, Fig. 2.1.f.
Truth Table: Representation of the output logic levels of a logic circuit for every possible combination of levels of the inputs. This is best done by means of a systematic tabulation.

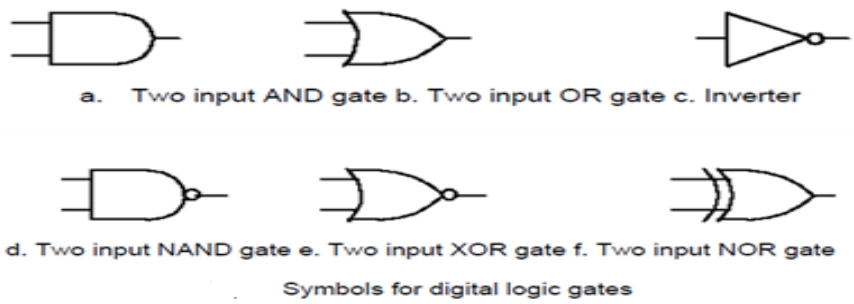


Figure 2.1

Part 1: Logic Functions
I. AND, OR, NAND, and NOR gates.

1. Use one gate for each IC 7400 (NAND), 7402 (NOR), 7408 (AND), 7432 (OR), 7486 (XOR). Each has input pins, 1 and 2, and output pin 3.

2. Connect pin 1 to switch S1-1, pin 2 to switch S1-2, and pin 3 to LED-1 for every gate as shown in Fig. 2.2 as an example for the NAND gate.

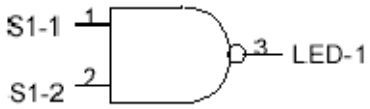


Fig. 2.2 Two input NAND gate

3. Using logic switches S1-1 and S-2, apply the logic levels 0 and 1 to gate inputs (pin 1, pin 2), in the sequence shown in table 1. Record the output logic levels (see LED-1) in Table 2.1. Repeat the recordings for each gate. Remember: LED ON = Logic 1, (High) LED OFF = Logic 0 (Low)

Table 2.1

Pin 1	Pin 2	Pin 3

4. Use an inverter gate from IC 7404 whose input pin is pin 1 and whose output pin is pin2.

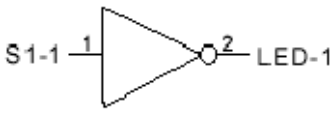


Fig. 2.3 Inverter gate

5. Using logic switches S1-1, apply the logic levels 0 and 1 in the sequence shown in Table 2. Record the output logic levels in Table 2.2.

Table 2.2

Pin1	Pin2
0	
1	
1	
0	

Part-2: Response of Logic Gates:

Connect the circuits of Fig. 6 and 7 and write the corresponding truth tables 3 and 4, respectively.

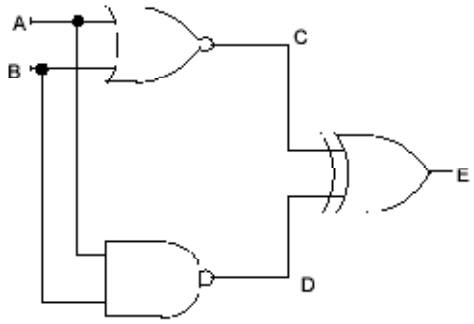


Fig. 2.4

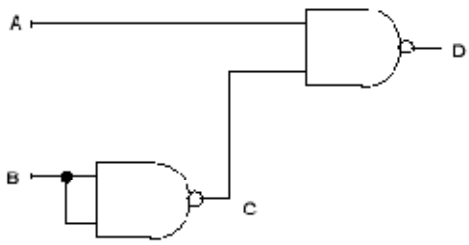


Fig.2.5

Table 3

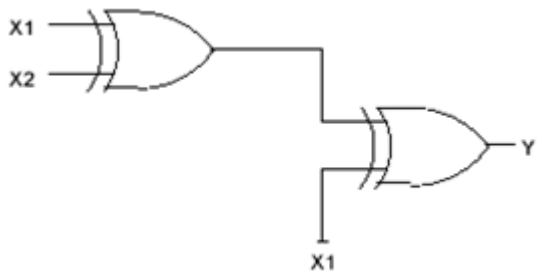
A	B	C	D	E
0	0			
0	1			
1	0			

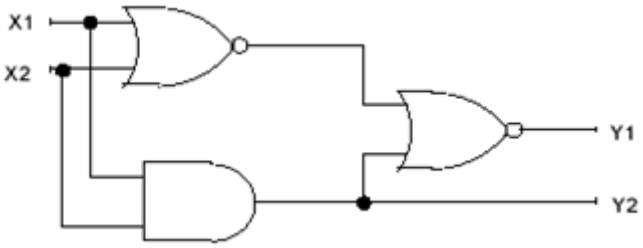
Table 4

A	B	C	D
0	0		
0	1		
1	0		

Part 3: Review Questions:

1. Write a truth table for each circuit. Derive Boolean expressions for all outputs.





Lab 3: Binary and Decimal Numbers

Objective

To demonstrate the count sequence of binary number and the binary-coded decimal (BCD) representation.

Apparatus

7490 Decade Counter
CADET trainer
Dual-trace Oscilloscope

Procedure

Binary Count

Binary counting will be shown on simulator.

BCD Count

1. Turn off the power switch.
2. Connect the IC type 7490 as shown in Fig.3.1 use push button for generate the pulse.
3. Turn the power on and observe the four logic indicator lamps/LEDs. The 4-bit number in the indicators is incremented by one for every pulse generated by pushing the pushbutton following the sequence 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2,3, ….
4. Disconnect the input of the counter and connect it to TTL output of the function generator. Set frequency selector to "time 1" (1 Hz). This will provide an automatic binary count.

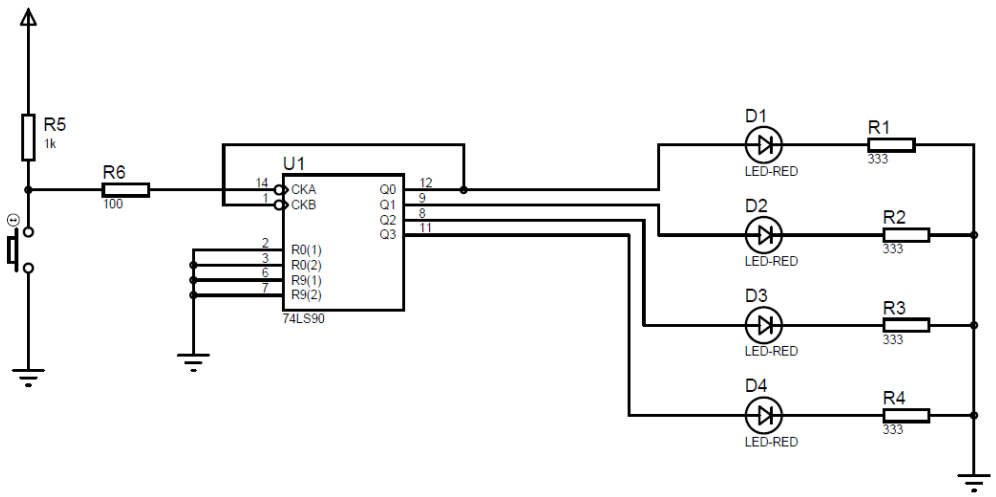


Fig. 3.1 BCD counter

8 decade counter

1. Turn off the power switch.
2. Connect the IC type 7490 as shown in Fig 3. 2.
3. Turn the power on and observe the four logic indicators/LED. The 4-bit number in the out is incremented by one for every pulse generated by pushing the pushbutton.Observe the following the sequence 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2,3, …..

4. Disconnect the input of the counter at pin 14 and connect it to the Function Generator (lead TTL).
5. Set frequency selector to “time 1” (1 Hz). This will provide an automatic binary count.
6. Increase the frequency of the clock to 10 kHz or higher and connect its output to an oscilloscope.
Observe the clock output on the oscilloscope and sketch its waveform.

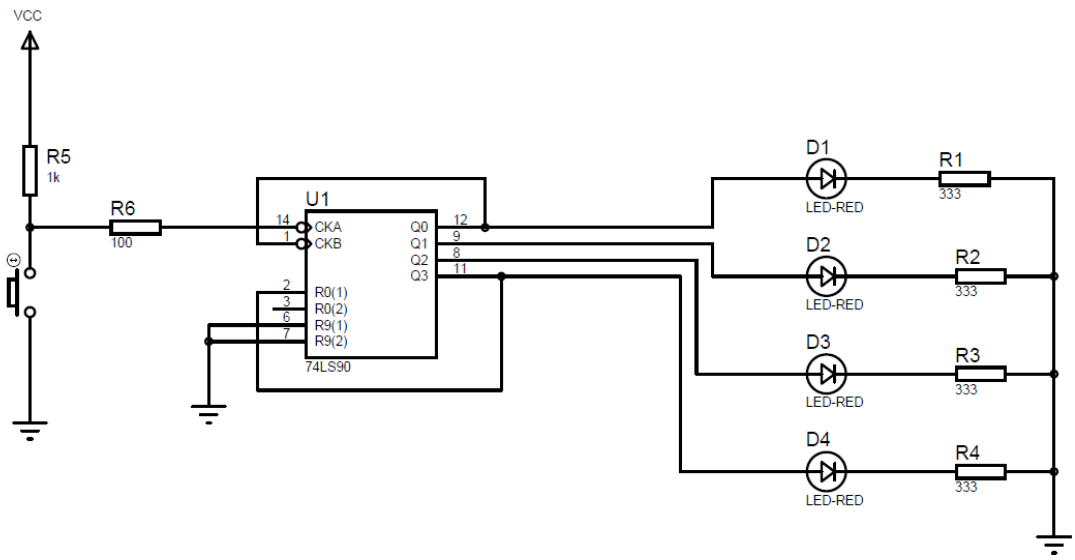


Fig3.2. Eight Decade Counter

Other Counts

R0(1) and R0(2) are the reset inputs of the IC type 7490. By connecting one or two outputs to the reset inputs R0(1) and R0(2), the counter can count from 0 to a variety of final count like in figure 3.2.

Count	Binary	
0	0000	
1	0001	
2	0010	
	…	
	…	
7	0111	
8-> 0	1000 -> 0000	Resets the counter (R0(1)=Q3)
1	0001	
2	0010	

Utilizing your knowledge of how R0(1) and R0(2) affect the final count, find out which outputs should be connected to the resets inputs to count 0000 to the following counts:

a. 0010
b. 0101
1. Turn of the power switch.
2. Connect the 7490 IC to count from 0 to one of the final counts given above.
3. Verify the count by applying pulses from the pushbuttons and observing the output count in the logic indicators/LEDs.

Lab 4 : Introduction to Proteus Isis

- Note that this lab will be held on computer labrotary.
- The propose of this lab, the learning and using proteus to simulate designs.
- Once the teaching assistant show you how to use this simulation, you will be supposed to do realize lab2 and lab 3 respectively.

Lab 5 : Code Conversion I- Gray Code to Binary converter

Objectives

• Design and build gray code to binary converter.

Apparatus

Seven segment display, common anode (CA).
7410 triple 3-input NAND gates
7420 dual 4-input NAND gates
7404 HEX inverter
7486 EX-OR gate

LogicWorks or Proteus can be used in the digital circuit simulations.

Theory

The conversion from one code to another is common in digital systems. Sometimes the output of a system is used as the input to the other system. A conversion circuit is necessary between two systems if each system uses different codes for the same information. In this experiment you will design and construct 3-combinational circuit converters.

PreLab Questions

Gray code to Binary converter

Design a combinational circuit with 4 inputs and 4 outputs that converts a four-bit gray code number (Table 2) into an equivalent four-bit Binary number. Use Karnaugh map technique for simplification.

Procedure:

Gray code to Binary converter:

Gray code is one of the codes used in digital systems. It has the advantage over binary numbers that only one bit in the code word changes when going from one number to the next. (See Table 5.1). Construct the circuit (PreLab Q.1) and verify its operations.

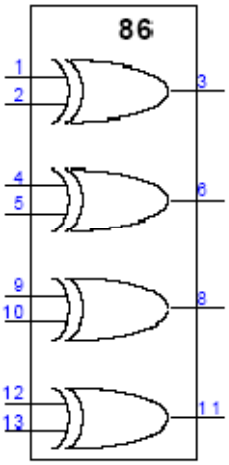


Fig.5.1 XOR chip 7486

Table 5.1

Decimal	Gray	Binary
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

Lab 6 : Code Conversion II- BCD-to-seven Segment converter

Objectives

• Design and build BCD-to-7 segment converter.

Apparatus

Seven segment display, common anode (CA).
7400 quad 2-input NAND gates
7410 triple 3-input NAND gates
7420 dual 4-input NAND gates
7404 HEX inverter
7486 EX-OR gate
7447 BCD-to-seven segment decoder.

LogicWorks or Proteus can be used in the digital circuit simulations.

Theory

The conversion from one code to another is common in digital systems. Sometimes the output of a system is used as the input to the other system. A conversion circuit is necessary between two systems if each system uses different codes for the same information. In this experiment you will design and construct 3-combinational circuit converters.

BCD-to-seven Segment converter

Design a combinational circuit which would simulate the BCD-to-seven Segment decoder function for only the segment "a", of the display. This can be done in the following steps:

a. Write down the truth table with 4 inputs and 7 outputs (Table 6.1)

Table 6.1

Dec	BCD				Outputs						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0							
1	0	0	0	1							
2	0	0	1	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0							
9	1	0	0	1							

b. For only the output “a”, obtain a minimum logic function. Realize this function using NAND gates and inverters only. For example if decimal 9 is to be displayed a, b, c, d, f, g must be 0 and the others must be 1 (For common anode type display units), if decimal 5 is to be displayed then a, f, g, c, d must be 0 and the others must be 1.

Procedure

BCD-to-seven Segment converter:

A light emitting Diode (LED) is a PN junction diode. When the diode is forward biased, a current flows through the junction and the light is emitted. See Fig.6.1.

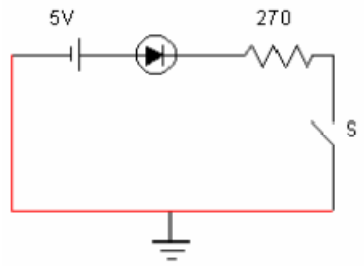


Fig 6.1

A seven segment LED display contains 7 LEDs. Each LED is called a segment and they are identified as (a, b, c, d, e, f, g) segments Fig.6.2.

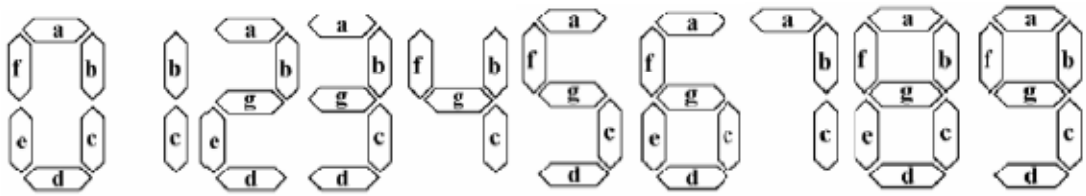


Fig. 6.2 Decimal digits represented by the 7 segments

The display has 7 inputs each connected to an LED segment. All anodes of LEDs are tied together and joined to 5 volts (this type is called common anode type). A current limiting resistance network must be used at the inputs to protect the 7-segment from overloading. BCD inputs are converted into 7 segment inputs (a, b, c, d, e, f, g) by using a decoder, as shown in Fig.6.3.

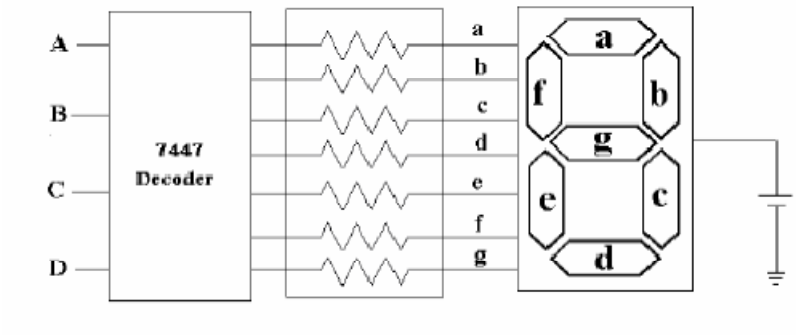


Fig 6.3

A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2 n output lines. The input to the decoder is a BCD code and the outputs of the systems are the seven segments a, b, c, d, e, f, and g. For further information and pin connections, consult the specification sheet for decoder and 7-segment units.

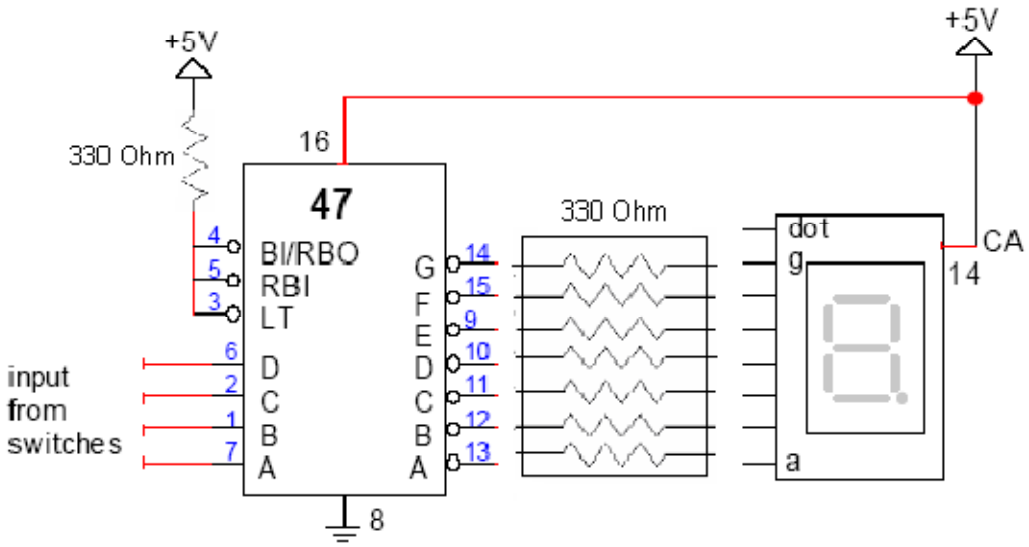
The 7447 BCD-to-seven segment decoder is used to drive a seven-segment LED display. The outputs, ag, drive the corresponding segments on the seven-segment display according to the binary number present at the inputs A-D, D being the most significant bit of the number. Three additional inputs, LAMP TEST, BI/RBO, and RBI are provided. The blanking input/ripple-blanking output (BI/RBO) blanks (turns off) the display when set LOW. Otherwise, when BI/RBO is set high, the outputs drive the display according to the inputs A-D. The ripple blanking input (RBI) must be HIGH if blanking of a decimal zero is not desired. This input is useful in blanking higher order zeroes when using several displays for a multidigit decimal number. Finally, LAMP TEST selects (turns on) all the segments when set LOW. It is used to test the segments on the display.

a. Construct the combinational circuit which will simulate the decoder function for only the segment "a", of the display.

b. Connect the output "a" of your circuit to appropriate input of 7-segment display unit through a current limiting resistor, 330/470 Ω. By applying BCD codes verify the displayed decimal digits for that segment for "a" of the display.

c. Replace your circuit by a decoder IC 7447 for all of the seven segments. Use current limiting resistors at the inputs of the seven-segment display. Observe the display and record the segments that will light up for invalid inputs sequence.

d. Comment on the design if you don't want to see any digit for invalid input sequence.



BCD-to-Seven Segment Decoder and 7-segment display

Lab 7 : Adders/Subtractors

Objectives

• To construct and test various adders and subtractor circuits.
• To construct and test a magnitude comparator circuit.

Apparatus

7486 quad 2-input XOR gates
7400 quad 2-input NAND gates
7404 Hex inverter
7408 Quadruple 2 input AND gates
7420 dual 4-input NAND gates
7483 4-bit binary adder
7485 4-bit magnitude comparator.

LogicWorks or Proteus can be used in the digital circuit simulations.

Addition

IC type 7483 is a 4-bit binary adder with fast carry. The pin assignment is shown in Fig 7.1. The two 4-bit input binary numbers are A1 through A4 and B1 through B4. The 4-bit sum is obtained from S1 through S4. Ci is the input carry and Co is the out carry. This IC can be used as an adder-subtractor as a magnitude comparator.

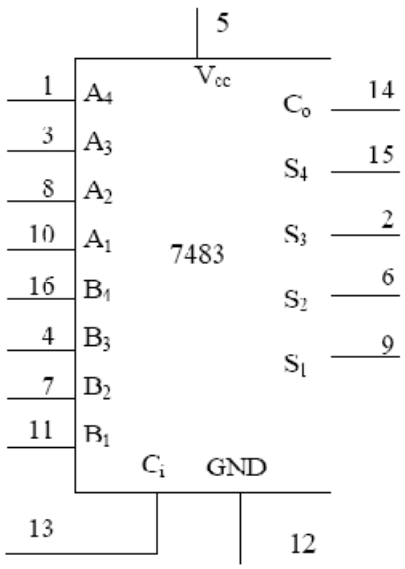


Fig. 7.1 IC type 7483 4-bit adder.

Subtraction

The subtraction of two binary numbers can be done by taking the 2's complement of the subtrahend and adding it to the minuend. The 2's complement can be obtained by taking the 1's complement and adding 1. To perform A - B, we complement the four bits of B, add them to the four bits of A, and add 1 to the input carry. This is done as shown in Fig 7.2. Four XOR gates complement the bits of B when the mode select M = 1 (because x .1=x') and leave the bits of B unchanged when M = 0 (because x .0=x) thus, when the mode select M is equal to 1, the input carry Ci is equal to 1 and the sum output is A plus the 2's complement of B. When M is equal to 0, the input carry is equal to 0 and the sum generates A + B.

Magnitude comparison

The comparison of two numbers is an operation that determines whether one number is greater than, equal to, or less than the other number.

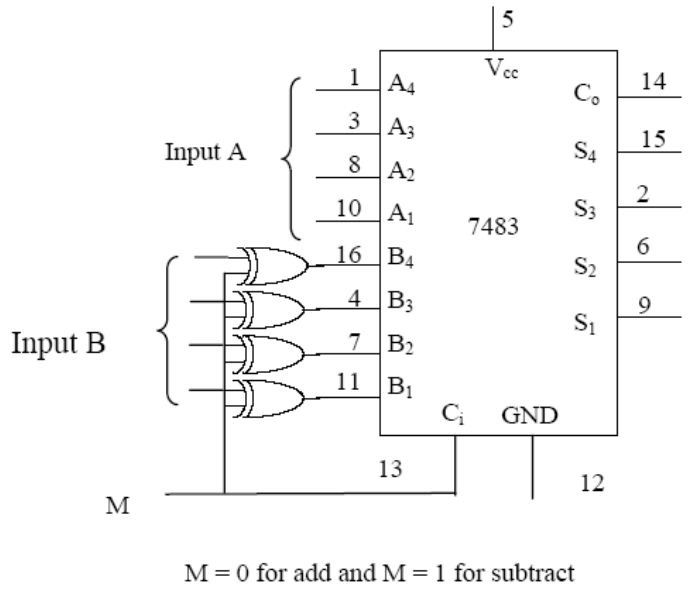


Fig. 7.2 4-bit adder/subtractor

The IC 7485 is a 4 bit magnitude comparator. It compares two 4-Bit binary numbers (labeled as A and B) generates an output of 1 at one of three outputs labeled A > B, A < B, A = B. Three inputs are available for cascading comparators. See Fig.7.3.

PreLab Questions:

1. a. Design using LogicWorks a half adder circuit using only XOR gates and NAND gates. Then during the Lab construct the circuit and verify its operation.

b. Design using LogicWorks a full adder circuit using only XOR gates and NAND gates. Then during the Lab construct the circuit and verify its operation.

2. A magnitude comparator can be constructed by using a subtractor as in Fig 20. and an additional combinational circuit. This is done with a combinational circuit which has 5 inputs S1, S2, S3, S4, and Co, and three outputs X, Y, Z see Fig.7.4.

X = 1 if A = B where S = 0000
Y = 1 if A < B where Co = 0
Z = 1 if A > B where Co = 1 S ≠ 0000.

Design this logic circuit with minimum number of gates.

Procedure:

a. Construct the circuit of PreLab 1(a) and verify its operation.
b. Construct the circuit of PreLab 1(b) and verify its operation.
c. Use IC 7483 to add the two 4-bit numbers A and B shown in Table 7.1

Table 7.1

A3	**A2**	**A1**	**A0**	**B3**	**B2**	**B1**	**B0**	**SUM**				**Carry Out**
1	0	0	1	0	0	1	0					
0	1	1	0	1	0	1	1					
1	1	0	0	1	0	1	0					

Input carry Ci is taken as logic 0. Show that if the input carry is 1, it adds 1 to the output sum. In the Lab use the logic switches S1-1 to S1-8 for the two numbers and use the SPDT (Single-Pole Double Throw) switch S1 for the input carry Ci. For sum and carry out, use LED-1 to LED-5.

d. Connect the adder-subtractor circuit as shown in Fig 7.2. Perform the following erations and record the values of the output sum and the output carry Co.

Table 7.2

Decimal A B	Sum				Carry Out C0
9 + 5					
9 - 5					
9 + 13					
10 + 6					
6 - 10					
9 - 9					

• Show that Co =1 when sum exceeds 15.
• Comment on sum and Co for the subtraction operations when A > B and A < B.

e. Use IC7485 to compare the following two 4 bit numbers A and B. Record the outputs in Table 7.3.

Table 7.3

A	B	Outputs
1001	0110	
1100	1110	
0011	0101	
0101	0101	

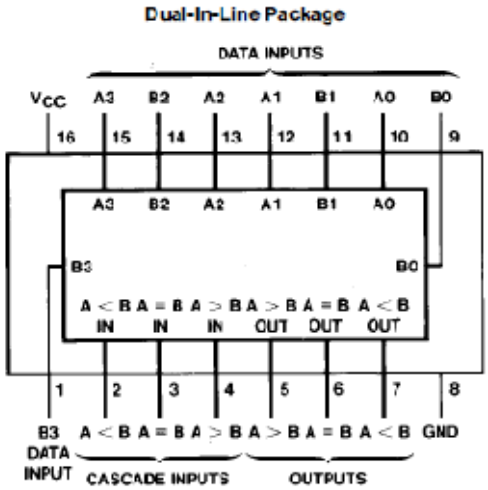


Fig. 7.37485 4-bit magnitude comparator

f. Construct the magnitude comparator of Fig. 7.4. The design of the logic circuit is described in PreLab Q2. Check the comparator action using part (e).

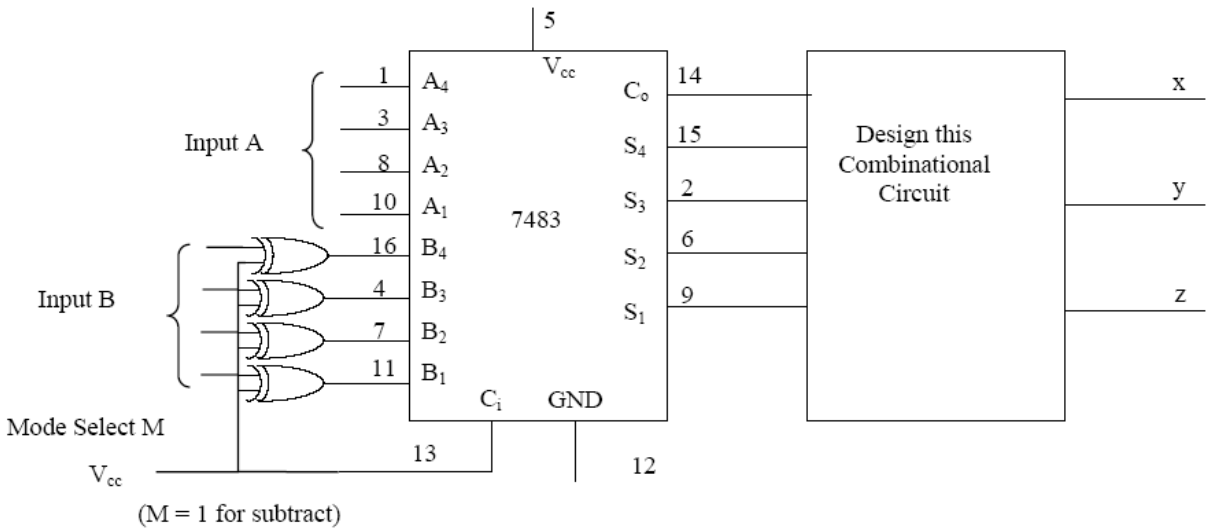


Fig. 7.4 A magnitude comparator using a subtractor

Lab 8 : Design with Multiplexers

Objectives

• To design a combinational circuit and implement it with multiplexers.
• To use a demultiplexer to implement a multiple output combinational circuit from the same input variables.

Apparatus

7404 HEX inverter
74151 8x1 multiplexer
74153 dual 4x1 multiplexer (x2)
7447 BCD-to-Seven-Segment decoder
Seven-Segment Display

LogicWorks or Proteus can be used in the digital circuit simulations.

PreLab Questions:

Design the circuits described in Part 1 and Part 2.

IC Description:

74151 is a 8 line-to-1 line multiplexer. It has the schematic representation shown in Fig 8.1. Selection lines S2, S1 and S0 select the particular input to be multiplexed and applied to the output. Strobe S acts as an enable signal. If strobe =1, the chip 74151 is disabled and output y = 0. If strobe = 0 then the chip 74151 is enabled and functions as a multiplexer. Table 8.1 shows the multiplex function of 74151 in terms of select lines.

Table 8.1

Strobe Select	Lines		Output	
S	S2	S1	S0	Y
1	X	X	X	0
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7

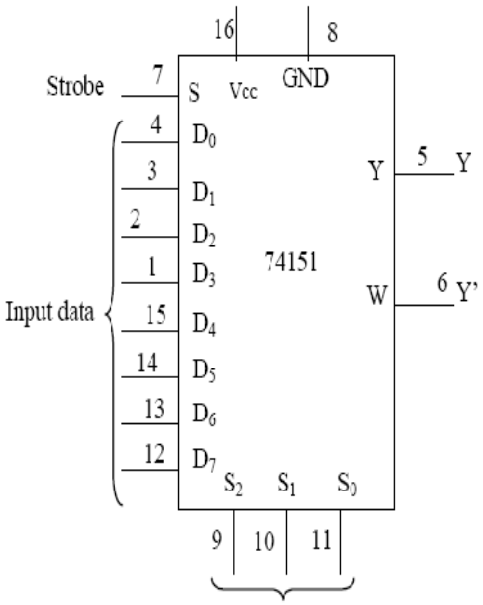


Fig. 8.1 IC type 74151 Multiplexer8×1

74153 is a dual 4 line-to-1 line multiplexer. It has the schematic representation shown in Fig 8.2. Selection lines S1 and S0 select the particular input to be multiplexed and applied to the output IY{1 = 1, 2}.

Each of the strobe signals IG {I = 1, 2} acts as an enable signal for the corresponding multiplexer.

Table 8.2 shows the multiplex function of 74153 in terms of select lines. Note that each of the on-chip multiplexers act independently from the other, while sharing the same select lines S1 and S0.

Table 8.2

Multiplexer 1				***Multiplexer 2***			
Strobe	**Select lines**		**Output**	**Strobe**	**Select lines**		**Output**
1G	**S1**	**S0**	**1Y**	**2G**	**S1**	**S0**	**2Y**
1	x	x	0	1	x	x	0
0	0	0	1D0	0	0	0	2D0
0	0	1	1D1	0	0	1	2D1
0	1	0	1D2	0	1	0	2D2
0	1	1	1D3	0	1	1	2D3

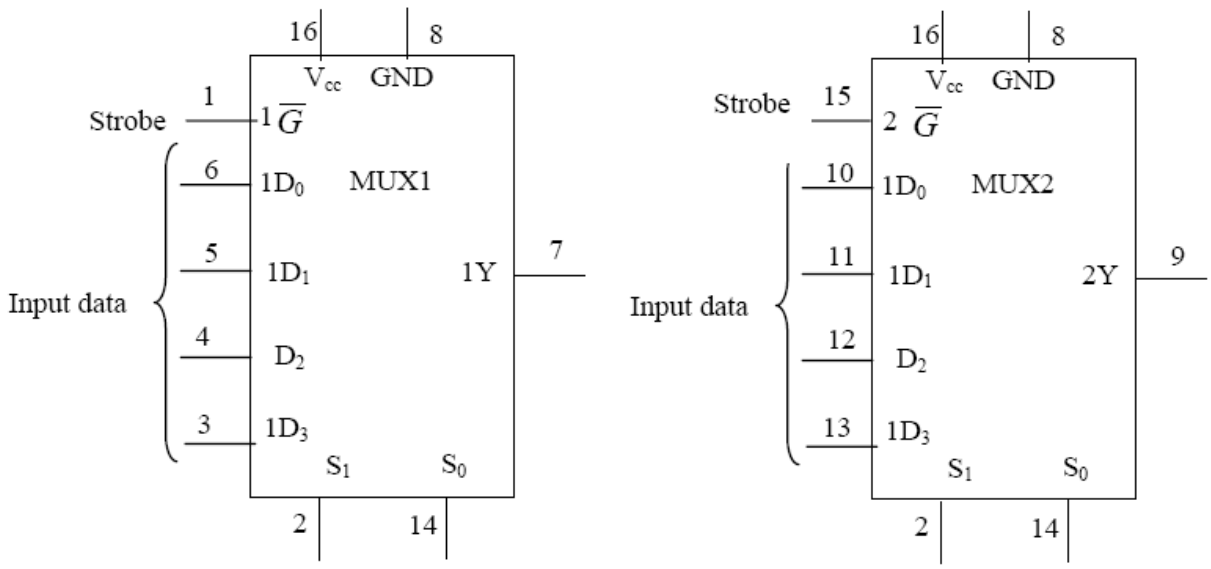


Fig. 8.2 Chip 74153

IC 7447 is a BCD to seven segment decoder driver. It is used to convert the combinational circuit outputs in BCD forms into 7 segment digits for the 7 segment LED display units.

Procedure:

Part I: Parity Generator:

a. Design a parity generator by using a 74151 multiplexer. Parity is an extra bit attached to a code to check that the code has been received correctly. Odd parity bit means that the number of 1's in the code including the parity bit is an odd number. Fill the output column of the truth table in Table 12 for a 5-bit code in which four of the bits (A, B, C, D) represents the information to be sent and fifth bit (x), represents the parity bit. The required parity is an odd

parity. The inputs B, C and D correspond to the select inputs of 74151. Complete the truth table in Table 12 by filling in the last column with 0, 1, A or A'.

b. Simulate the circuit using LogicWorks, use 74-151 multiplexer and Binary switches for inputs and binary Probes for outputs. The 74151 has one output for Y and another inverted output W. Use A and A' for providing values for inputs 0-7. The internal values "A, B, C" are used for selection inputs B, C, and D.

Simulate the circuit and test each input combination filling in the table shown below.

Table 12

Inputs				Outputs	Connect data to
A	B	C	D	X	
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

In the Lab connect the circuit and verify the operations. Connect an LED to the multiplexer output so that it represents the parity bit which lights any time when the four bits input have even parity.

Part 2: Vote Counter: (Performance grade)

A committee is composed of a chairman (C), a senior member (S), and a member (M). The rules of the committee state that:

- The vote of the member (M) will be counted as 2 votes
- The vote of the senior member will be counted as 3 votes.
- The vote of the chairman will be counted as 5 votes.

Each of these persons has a switch to close ("l") when voting yes and to open ("0") when voting no. It is necessary to design a circuit that displays the total number of votes for each issue. Use a seven segment display and a decoder to display the required number.

If all members vote no for an issue the display should be blank. (Recall from Experiment #4, that a binary input 15 into the 7447 blanks all seven segments). If all members vote yes for an issue, the display should be 0. Otherwise the display shows a decimal number equal to the number of 'yes' votes. Use two 74153 units, which include four multiplexers
to design the combinational circuit that converts the inputs from the members' switch to the BCD digit for the 7447.

In LogicWorks use +5V for Logic 1 and ground for Logic 0 and use switches for C, S, and M. Use two chips 74153 and one decoder 7446 verify your design and get a copy of your circuit with the pin numbers to Lab so that you could connect the hardware in exactly the same way.

Lab 9 : Flip-Flops

Objectives:

• To become familiar with flip-flops.
• To implement and observe the operation of different flip-flops.

Apparatus

7400 quad 2-input NAND gates (2)
7476 dual JK master-slave flip-flops.

LogicWorks or Proteus can be used in the digital circuit simulations.

Procedure

1. In the pre-lab using LogicWorks construct the circuit shown in Fig.9.1.

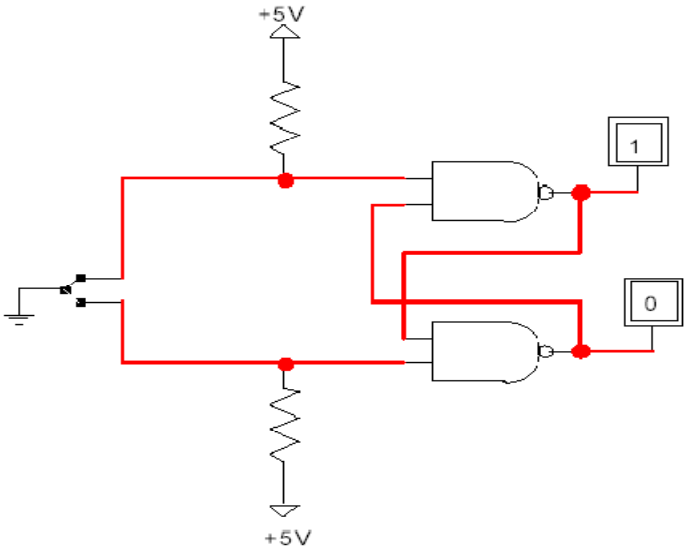


Fig. 9.1

You can use generic NAND gates or 74-00 and Binary Probes to simulate LEDs. Finally, we use SPDT for the bouncing switch. Using the simulated circuit fill in the truth table.

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

In the lab, build the RS latch shown in Fig.9.2. Use SPDT switch S as a bouncing switch. Q and Q' outputs are connected to LED's of the CADET. Verify the truth table of SR latch experimentally.

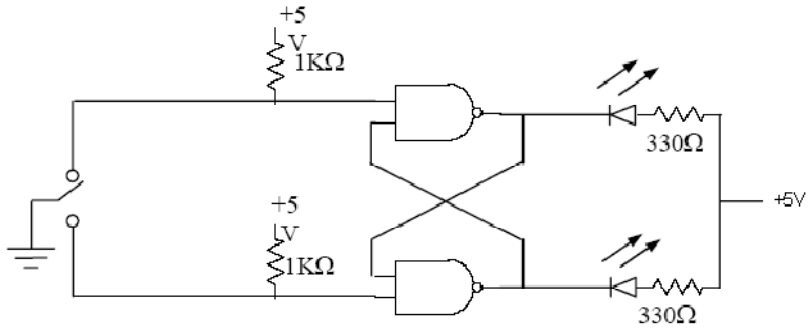


Fig. 9.2

2. Modify the basic RS latch into a D latch by adding the steering gates and the inverter shown in Fig.9.3 Connect the D input to the pulse generator of the CADET and set it at 1 Hz. Connect the enable input to a high through 1k resistor. Observe the output; obtain the truth table experimentally then change the enable to a low.

Is the enable an active high or an active? Leave the enable low and place a momentary short to ground first on one output and then on the other. What happens?

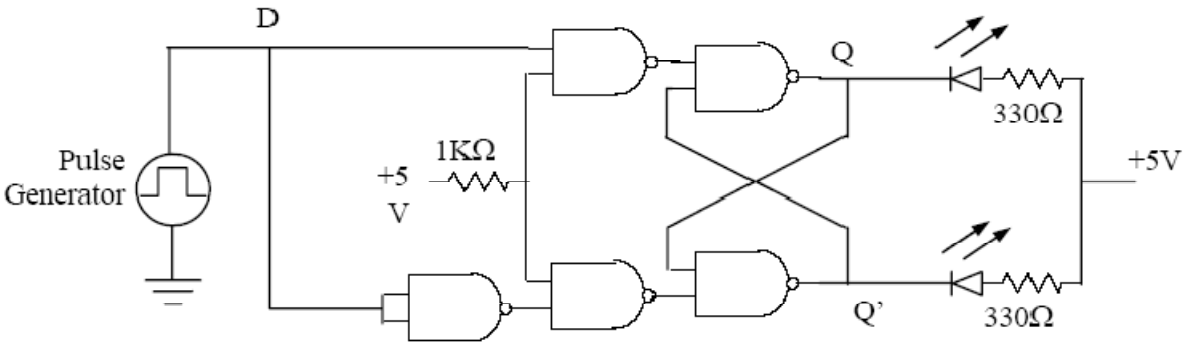


Fig. 9.3

3.The 7476 is a dual JK master-slave flip-flops with preset and clear inputs. The function table given in Table 9.2 defines the operation of the flip-flop. The positive transition of the CLOCK (CP) pulse changes the master flip-flop, and the negative transition changes the slave flip-flop as well as the output of the circuit.

Table 9.2

Input					Output	
Preset	Clear	Clock	J	K	Q	Q'
0	1	X	X	X	1	0
1	0	X	X	X	0	1
0	0	X	X	X	1	1
1	1	↓	0	0	No change	
1	1	↓	0	1	0	1
1	1	↓	1	0	1	0
1	1	↓	1	1	Toggle	

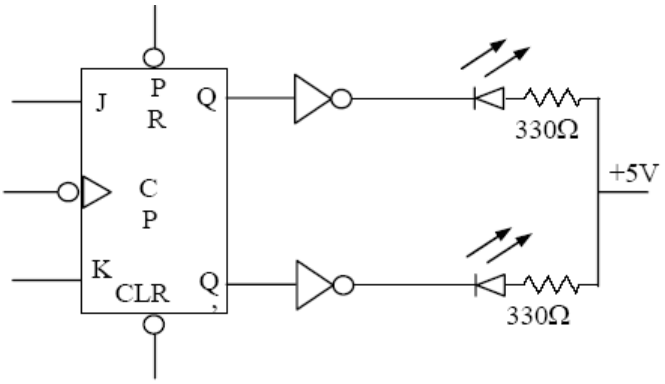


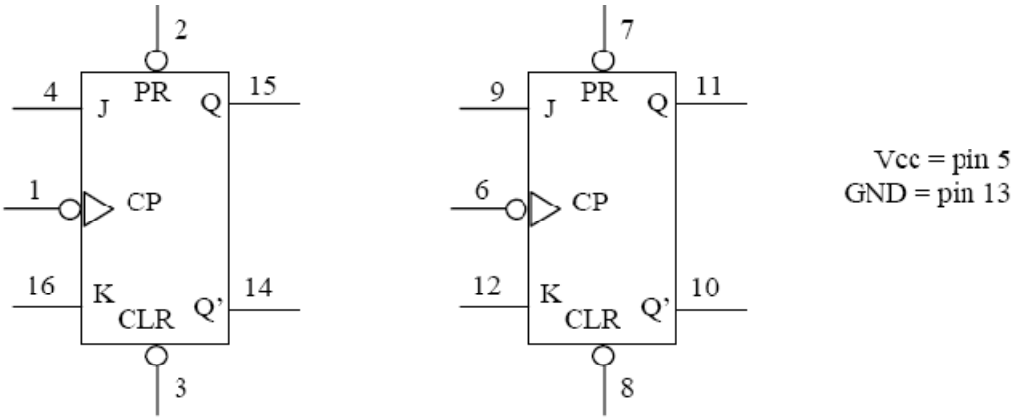
Fig. 9.4

In the lab, construct the circuit of Fig 9.4. Look at the data sheet for the 7476 and determine the inactive logic required at the PRE and CLR inputs.

Connect the 7476 for the SET mode by connecting J = 1, K = 0. With CLOCK (CP) = 0; test the effect of PRE, CLR by putting a 0 on each, one at a time.

Put CLR = 0, then pulse the clock (CP) by putting a HIGH then a LOW, on the clock. Does the CLR input override J input?

Verify the operation of the JK flip flop by experimentally obtaining the characteristic.



Lab 10 : Clocked Sequential Circuits and Counters

Objectives

• To design, build and test synchronous sequential circuits
• To design, build, and test synchronous counters
• To design, build and test asynchronous counters

Apparatus

7476 dual JK master-slave flip-flops (x2)
7400 quad 2-input NAND gates
7404 HEX inverter
7408 quad 2-input AND gate

LogicWorks or Proteus can be used in the digital circuit simulations.

Pre-Lab Questions

1. **a.** Design, construct and test a sequential circuit whose state is shown in Fig.10.1. Use JK flip-flops in the design.

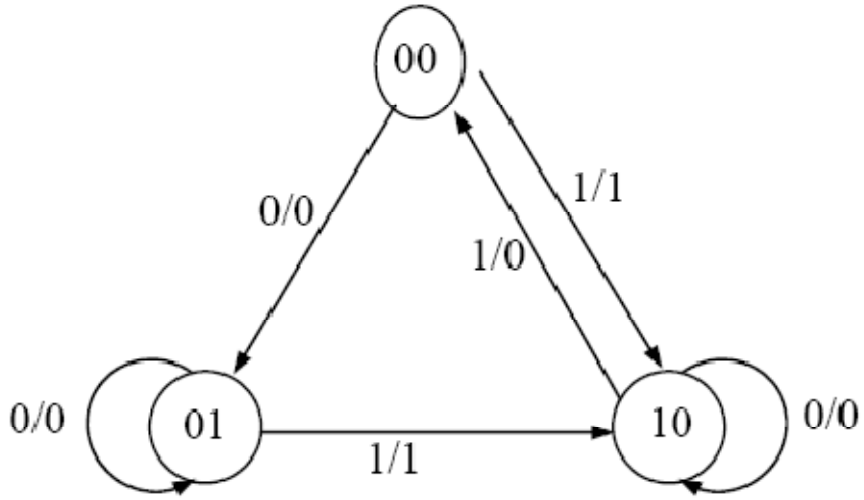


Fig. 10.1.

The circuit has two flip-flops A, B, one input x and one output y. The circuit is to be designed by treating the unused states as don't care conditions. The final circuit must be analyzed to ensure that it is selfcorrecting.If not suggest a solution.

b. Complete the excitation table shown in Table 10.1.

Table 10.1

Present state		Input	Next state		Output	Flip-flop input functions			
A	B	X	A	B	Y	JA	KA	JB	KB
0	0	0	0	1	0	0	X	1	X
0	0	1	1	0	1	1	X	0	X
0	1	0							
0	1	1							
1	0	0							
1	0	1							
1	1	0							
1	1	1							

c. Using Karnaugh maps obtain minimal expressions for the flip-flop input functions JA, KA, JB and KB.

d. Simulate the circuit using LogicWorks. LogicWorks does not have the JK master-slave flip-flop IC 7476.Use instead the generic JK flip-flop as you did in Lab 8. In the Lab, build the circuit and check the output to verify the state table values.

2. a. In the pre-lab using LogicWorks and then in the lab using hardware ICs, design a 2-bit gray code counter using JK flip-flops. The required sequence is the binary equivalent of (0-1-3-2-0). A state diagram for this counter is given in Fig. 10.2.

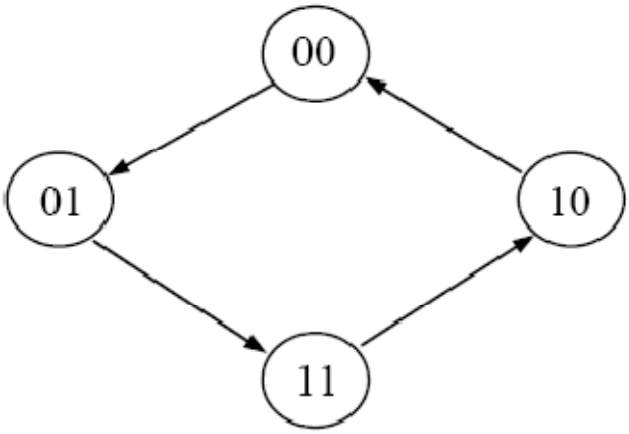


Fig. 10.2

b. Complete the excitation table in Table 10.2 for the counter and obtain logic expression for the JK flip-flop input functions.

Table 10.2

Present state		Next state		Flip-flop input functions			
A	B	A	B	JA	KA	JB	KB
0	0						
0	1						
1	1						
1	0						

Procedure:

1. Synchronous Sequential Circuits

Build the sequential circuit that is designed in Pre-Lab Q1, and test the circuit by applying pulse from the CADET. Check that the output is the designed sequence.

2. Synchronous Counters

Synchronous counters have all clock lines tied to a common clock causing all flip-flops to change at the same time. The count sequence of a counter can be analyzed by placing the counter into every possible number in the sequence and determining the next number in the sequence state diagram is developed as the analysis proceeds. (A state diagram is an illustration of the transitions that occur after each clock pulse).

In the lab, build the 2-bit gray code counter circuit and test it by pulsing it from the CADET. Check that the output is the designed sequence.

3. A Synchronous Counters

Asynchronous counters are a series of flip-flops each clocked by the previous state, one after the other. Since all the stages of the counter are not clocked together, a ripple effect propagates as various flip-flops are clocked. For this reason they are called ripple counters. The modulus of a counter is the number of different output states the counter may take (i.e. Mod 4 means the counter has four output states).

In the pre-lab using LogicWorks construct a 4-bit asynchronous counter shown in Fig.10.3. (It is also called binary ripple counter). Use four generic JK flip-flops. Connect four Binary Probes to Q outputs. Connect all R and S inputs to Logic 1 and connect a switch to the CP input.

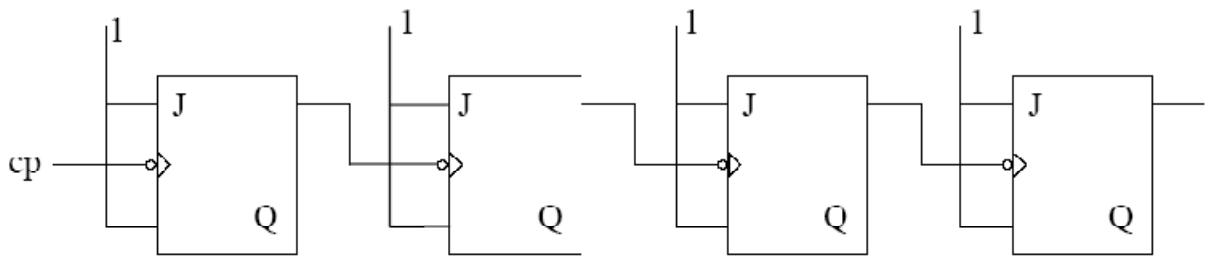


Fig. 10.3. 4-bit ripple counter.

In the lab use two 7476 ICs to implement 4-bit asynchronous counter as shown in Figure 3. Connect Q outputs of flip-flops to logic indicator lamps/LEDs of the CADET. Connect all clear (CLR) and preset (PRE) inputs to logic 1. Connect the CP input to the pulse output of the CADET and check the counter for proper operation.

Write down the count sequence in Table 10.3. Identify this count sequence (up or down). Comment on what happens after the application of 15 pulses to CP input. Count sequence for the 4-bit ripple counter.

Table 10.3

A	B	C	D