DRONACHARYA GROUP OF INSTITUTIONS, GREATER NOIDA
Affiliated to Mahamaya Technical University, Noida
Approved by AICTE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Lab Manual for
Computer Organization Lab ECS-453

List of Experiments

1. Bread Board Implementation of various logic gates using NAND gate.
2. Bread Board implementation of Binary Adder (Half and Full)
3. Bread Board implementation of Adder/Subtractor.
4. Bread Board Implementation of Flip-Flops.
5. Experiments with clocked Flip-Flop.
6. Design of Counters.
7. Bread Board implementation of counters & shift registers.
8. Implementation of Arithmetic algorithms.
9. Bread Board implementation of Seven Segment Display.

Experiment 1

Object: Bread Board Implementation of various logic gates using NAND gate.

Apparatus Required: IC 7400, Bread Board, Connecting wires.

Theory:

The NAND Gate:

The NAND, which is composed of two or more inputs and a single output, is a very popular logic element because it may be used as a universal function. That is, it may be employed to construct an inverter, an AND gate, an OR gate, or any combination of theses functions. The term NAND is formed by the concatenation NOT-AND and implies an AND function with an inverted output. The standard symbol for the NAND gate is shown in Figure 1-7 and its truth table listed in Table 1-4. The logical operation of the NAND gate is such that the output is LOW (0) only when all the inputs are HIGH (1).

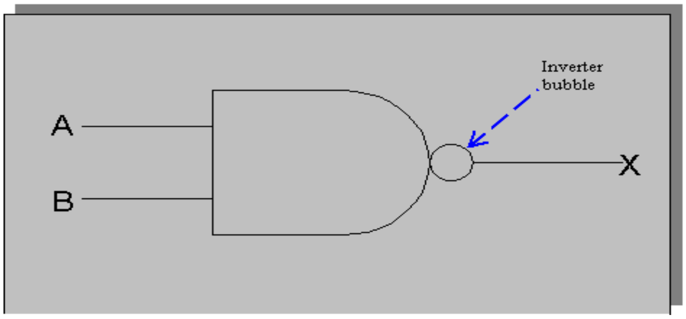
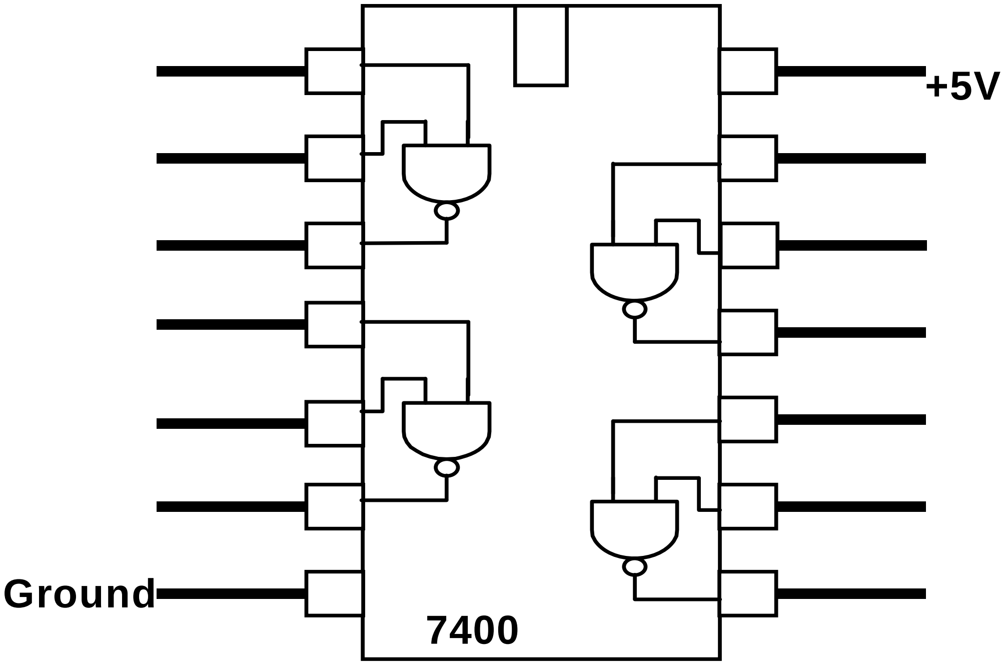


Figure 1-7 Standard logic symbol for NAND gate

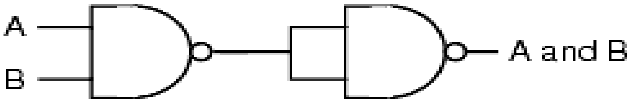
INPUT		OUTPUT
A	B	$X = \overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

Fig: Truth Table for NAND gate.

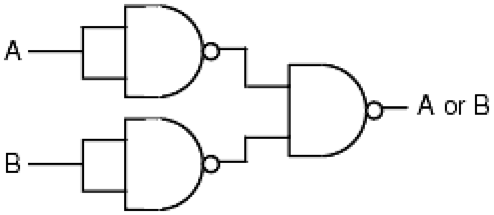
Pin diagram for 7400 Quad NAND gate IC:-



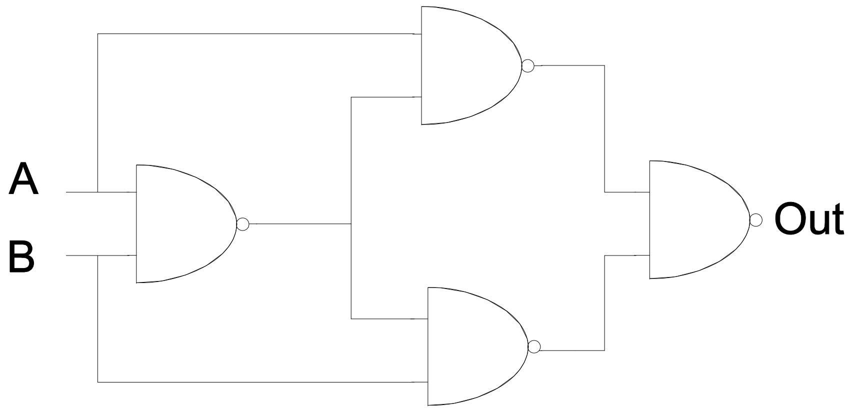
1. **Implementing AND gate using NAND gate:**



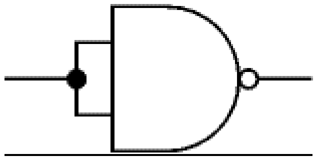
2. **Implementing OR gate using NAND gate:**



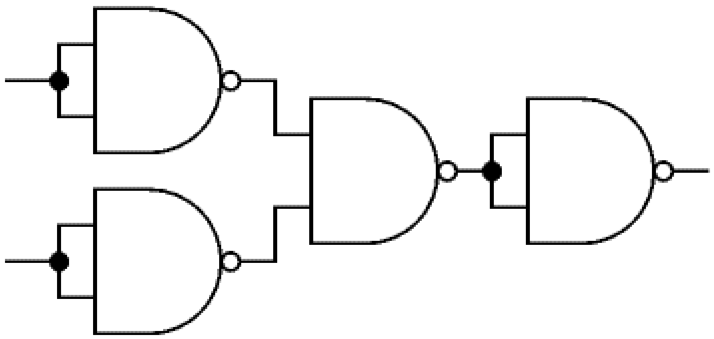
3. **Implementing X- OR gate using NAND gate:**



4. **Implementing NOT gate using NAND gate:**



5. **Implementing OR gate using NAND gate:**



Experiment 2

Object: Bread Board Implementation of binary adders and binary subtractors (half and full).

Apparatus Required: IC 7408, 7432, 7486, Bread Board, Connecting wires.

Theory:

A Half-Adder

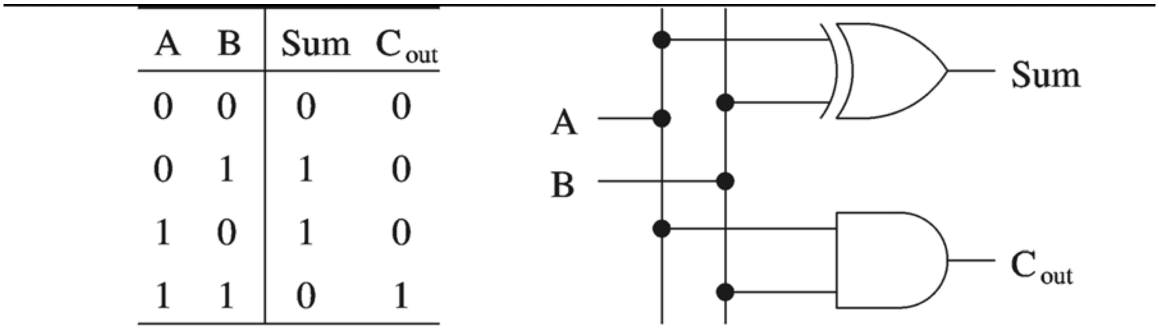
As a first example of useful combinational logic, let's build a device that can add two binary digits together. We can quickly calculate what the answers should be:

$0 + 0 = 0$ $0 + 1 = 1$ $1 + 0 = 1$ $1 + 1 = 10_2$

So we well need two inputs (a and b) and two outputs. The low order output will be called Σ because it represents the sum, and the high order output will be called C_{out} because it represents the carry out.

The truth table is

Simplifying boolean equations or making some Karnaugh map will produce the same circuit shown below, but start by looking at the results. The Sum column is our familiar XOR gate, while the C_{out} column is the AND gate. This device is called a half-adder for reasons that will make sense in the next section.



A	B	Sum	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(a) Half-adder truth table and implementation

A Full-Adder:

The half-adder is extremely useful until you want to add more that one binary digit quantities. The slow way to develop a two- binary digit adder would be to make a truth table and reduce it. Then when you decide to make a three binary digit adder, do it again. Then when you decide to make a four-digit adder, do it again. Then when ... The circuits would be fast, but development time would be slow.

Looking at a two binary digit sum shows what we need to extend addition to multiple binary digits.

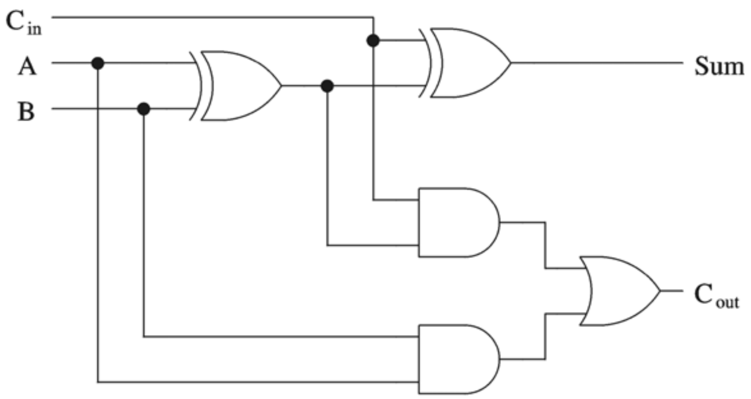
11

110

Look at how many inputs the middle column uses. Our adder needs three inputs; a, b, and the carry from the previous sum, and we can use our two-input adder to build a three input adder.
Σ is the easy part. Normal arithmetic tells us that if $\Sigma = a + b + C_{in}$ and $\Sigma_1 = a + b$, then $\Sigma = \Sigma_1 + C_{in}$.

In order to calculate the high order bit, notice that it is 1 in both cases when a + b produces a C_1. Also, the high order bit is 1 when a + b produces a Sum and C_{in} is a 1. So we will have a carry when C_1 OR (Sum AND C_{in}). Our complete three input adder is:

A	B	C_{in}	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



(b) Full-adder truth table and implementation

For some designs, being able to eliminate one or more types of gates can be important, and you can replace the final OR gate with an XOR gate without changing the results.

A Half-Subtractor:

A half subtractor is a combinational circuit that subtracts two bits and produces their difference. It also has an o/p to specify if a1 has been borrowed. Designate the minuend bit by X and the subtrahend bit by Y. to perform X-Y we have three possibilities 0-0=0,1-0=1,0-1=1,1-1=0.the half subtractor needs two o/p's. One o/p generates the difference and will be designed by the symbol D. The second o/p designated by B for borrow, generates the binary signal that informs the next stage that 1 has been borrowed.

EXPRESSION FOR HALF SUBTRACTOR:
Difference= X'Y+XY'
Borrow= X'Y

TRUTH TABLE: -

INPUT		OUTPUT	
X	**Y**	**Difference**	**Borrow**
0	0	0	0
0	1	1	1
1	0	1	0

1	1	0	0

A Full- Subtractor:

A Full Subtractor is a combinational circuit that performs a subtraction between two bits; taking into account that a1 may have been borrowed by a lower significant stage. This circuit has two inputs and two outputs. The three inputs, Y and Z, denotes the minuend, subtrahend and previous borrow respectively. The two outputs and B represents the difference and output borrow respectively.

EXPRESSION FOR FULL SUBTRACTOR:

Difference= X’Y’Z+X’YZ’+XY’Z’+XYZ
Borrow = X’Y+YZ+ZX’

TRUTH TABLE: -

INPUT			**OUTPUT**	
X	**Y**	**Z**	**Difference**	**Borrow**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

RESULT:- The operation of half adder, half subtractor, full adder, full subtractor has been verified.

PRECAUTIONS: -

1. Connection should be tight.
2. O/P should be finding sequentially.
3. IC’s should be handled carefully.

Experiment 3

Object: Bread Board Implementation of binary adder- subtractor for 4- bits.

Apparatus Required: IC 7483, 7486, Bread Board, Connecting wires.

Theory:

The addition and subtraction operation can be combined into one common circuit by including an X- OR gate with each Full adder. The mode input M controls the operation. When M=0 the circuit is an Adder and when M=1 the circuit becomes subtractor. Each X- OR gate receives input M and one of the inputs of B. When M=0, we have B ⊕ 0= B. The full adders receive the value of B, the input carry is 0, and the circuit performs A plus B. When M=1, we have B ⊕ 1= B' and C_1=1. The B inputs are all complemented and a 1 is added through the input carry. The circuit performs the operation A plus the 2's complement of B. For unsigned numbers, this gives A-B if A>= B or the 2's complement of (B-A) if A<B. For signed numbers, the result is A- B provided that there is no overflow.

A 4-bit Adder-Subtractor is shown in fig.1. Addition operation is done with SUB input kept LOW. When SUB input is LOW, the output is the same as that of the addend. Hence

Augend	A_3	A_2	A_1	A_0
Addend	B_3	B_2	B_1	B_0
Sum	S_3	S_2	S_1	S_0

and carry output Cout.

Subtraction operation is done by using 2'scomplement. For subtraction, SUB input is kept HIGH. When SUB input is HIGH, the output is the complement of the input (i.e. $B_{3,}$ $B_{2,}$ $B_{1,}$ B_0). If a 1 is added to the 1's complement of this input, we get 2's complement of the data input. Now subtraction can be done by 2's complement method by adding the addend and subtrahend. Hence

Addend	B_3	B_2	B_1	B_0
1's complement	B_3	B_2	B_1	B_0
SUB				-1
2's complement	B_3	B_2	B_1	B_0

The data $A_3A_2A_1A_0$ will be added to the 2's complement of $B_3B_2B_1B_0$ to produce the SUM (i.e. the difference between the addend and the augend).

7486 is a quad Ex-OR gate, which can be used as a controlled inverter. 7483 is a TTL circuit with 4-full adders i.e. it can add a nibble. The carry out pin 14 may be connected to the carry in pin 13 of the next 7483 to add 8-bit numbers.

PROCEDURE:

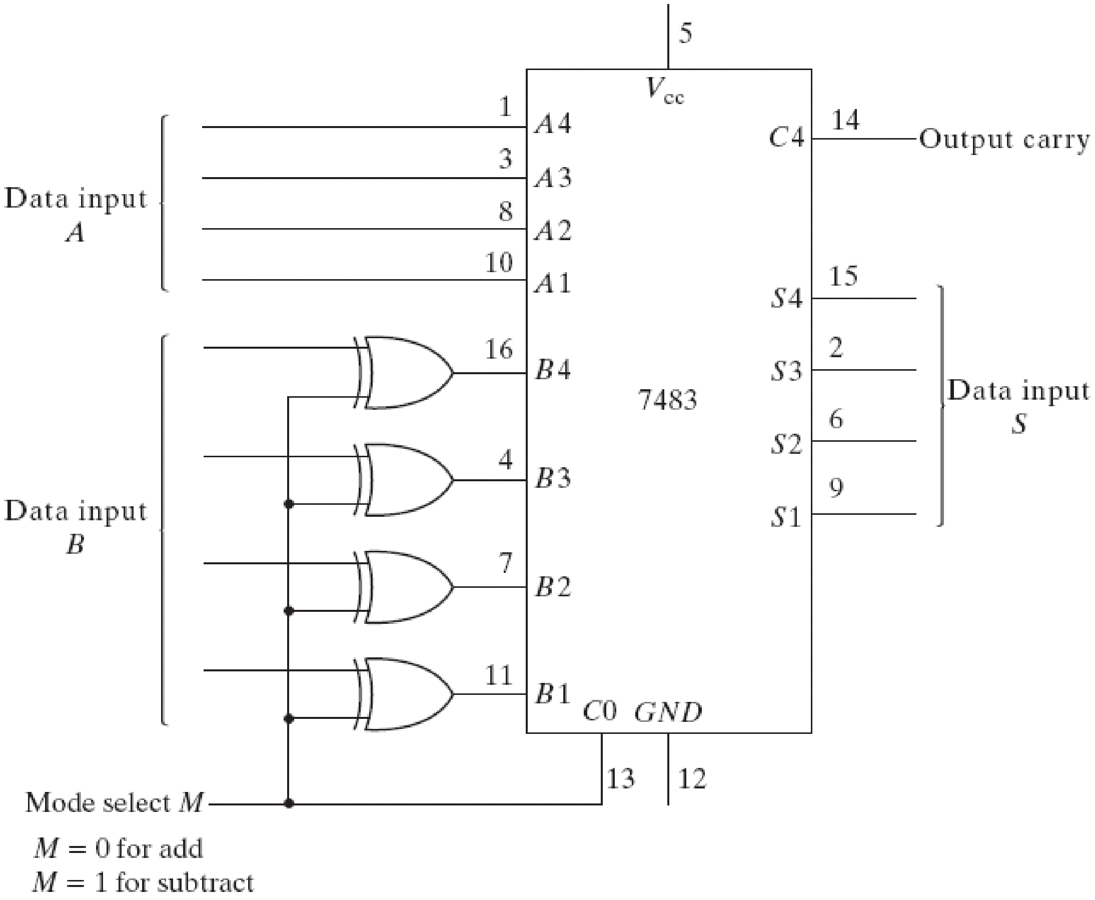
1) Put IC on the breadboard.
2) Apply 5V supply at pin 5.
3) Apply ground at pin 12.
4) Make connections as shown in the circuit diagram.
5) Keep SUB input LOW for Addition.
6) Keep SUB input HIGH for Subtraction.

7) Observe the different outputs.

OBSERVATION TABLE:

S. No.	Carry in	Input	Input	Carry out	Output
	C_0	$A_4A_3A_2A_1$	$B_4B_3B_2$ B_1	C_4	$S_4S_3S_2S_1$
1.					
2.					
3.					
4.					
5.					

CIRCUIT DIAGRAM:



RESULT: The sum and difference of 4 bit numbers is exactly same as calculated theoretically.

PRECAUTIONS: -

1. Connection should be tight.
2. IC's should be handled carefully

Experiment 4

OBJECT: Breadboard implementation of SR and D- flip flop and verify their characteristic table.

APPARATUS REQUIRED: - IC -7400, IC- 7404, and logic trainer board, connecting wires.

THEORY: S-R FLIP FLOP: -
A S R flip flop can be built using NOR gate or NAND gate .It has two inputs R and S and two O/P are Q and Q .In a flip flop the two O/Ps are complementary, If Q=1 then Q=0.A low R and low S result in inactive state (there is no change). A low R and high S results in set state while high R and low S results in reset state. If R and S are high sate, the O/P is in determined and this is called race condition.

D-FLIP FLOP: -
The storage elements employed in clocked sequential circuits are called flip-flop. A flip-flop is a binary storage bit of information .A flip-flop maintains a binary state until directed by clock pulse to switch states. Theory of T flip-flop is presented below.

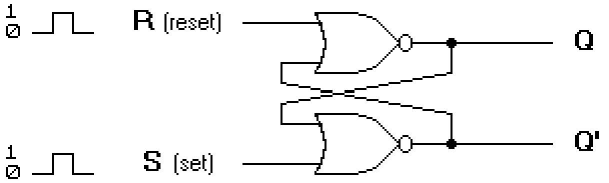
Flip-Flops

The memory elements in a sequential circuit are called *flip-flops*. A flip-flop circuit has two outputs, one for the normal value and one for the complement value of the stored bit. Binary information can enter a flip-flop in a variety of ways and gives rise to different types of flip-flops.

Introduction - Basic Flip-Flop Circuit

A flip-flop circuit can be constructed from two NAND gates or two NOR gates. These flip-flops are shown in Figure 2 and Figure 3. Each flip-flop has two outputs, *Q* and *Q'*, and two inputs, *set* and *reset*. This type of flip-flop is referred to as an *SR flip-flop* or *SR latch*. The flip-flop in Figure 2 has two useful states. When Q=1 and Q'=0, it is in the *set state* (or 1-state). When Q=0 and Q'=1, it is in the *clear state* (or 0-state). The outputs Q and Q' are complements of each other and are referred to as the normal and complement outputs, respectively. The binary state of the flip-flop is taken to be the value of the normal output.

When a 1 is applied to both the set and reset inputs of the flip-flop in Figure 2, both Q and Q' outputs go to 0. This condition violates the fact that both outputs are complements of each other. In normal operation this condition must be avoided by making sure that 1's are not applied to both inputs simultaneously.

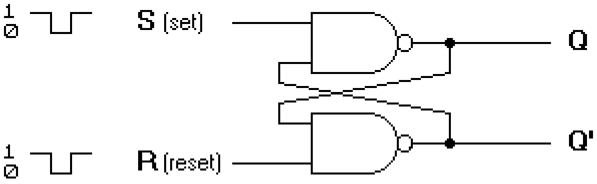


(a) Logic diagram

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

(b) Truth table

Figure 2. Basic flip-flop circuit with NOR gates



(a) Logic diagram

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	

(b) Truth table

Figure: Basic flip-flop circuit with NAND gates

The NAND basic flip-flop circuit in Figure 3(a) operates with inputs normally at 1 unless the state of the flip-flop has to be changed. A 0 applied momentarily to the set input causes Q to go to 1 and Q' to go to 0, putting the flip-flop in the set state. When both inputs go to 0, both outputs go to 1. This condition should be avoided in normal operation.

TRUTH TABLE FOR S-R FLIP-FLOP: -

INPUT	**INPUT**	**OUTPUT**		**COMMENT**
S	**R**	**Q (t)**	**Q (t+1)**	
0	0	0	0	Previous stage
0	1	0	0	Reset
1	0	1	1	Set
1	1	1	1	In determined

TRUTH TABLE FOR S-R FLIP-FLOP: -

INPUT D	OUTPUT Q (t+1)	COMMENT
0	0	No change
1	1	Set

PROCEDURE:

1. Insert ICs according to the circuit diagram on the trainer board.
2. Give +5V supply to the pin 14 and ground to the pin 7 to all ICs.
3. Give inputs S, R and CLK to the respective pins of the ICs and observe the output at output logic.
4. Observe LEDs output.
5. Try with different combination of input S and R.
6. Prepare truth table, observe and verify it.

RESULT: Truth table of S-R and D flip flop are verified.

PRECAUTION:

1. All connection should be tight.
2. After all connection of the circuit, the main supply should be ON.

CIRCUIT DIAGRAM:

Experiment 5

OBJECT: Design of counters.

APPARATUS REQUIRED: -

S. No.	Equipment	Qty.
1	Module-n-counter trainer	1
2	Connecting Leads	6

THEORY: A counter is one of the most useful and versatile sub systems in a digital system. A counter driven by a clock can be used to count the number of the clock cycles. Since the clock pulses occur at known intervals, the counter can be used as an instrument for measuring time and therefore period or frequency. There are basically two different types of counters: Synchronous and Asynchronous.

Module-N-counter is a one type of counter, in which, instead of counting from the beginning to the ending, we can restrict the counter to count up to some set value and then to the beginning value. This can be achieved by feeding the particular output to the reset input through digital gates.

The 74190 and 191 are synchronous, reversible up/down counters. Having all flip-flops clocked simultaneously, so that the outputs change coincide with each other when so instructed by the steering logic provides synchronous counting operation. The mode of operation eliminates the output counting spikes normally associated with asynchronous (ripple clock) counters. The outputs of the four flip-flops are triggered on a low-to-high level transition of the clock input if the enable (CTEN) is LOW. A high at CTEN inhibits counting the direction of the count is determined by the level of the down/up (D/U) input. When D/U is LOW, the counter counts up and when D/U is high, it counts down.

PROCEDURE:

1. Switch ON the experimental kit.
2. Make sure that the counter output is zero and put the U/D switch in 'U' position.
3. Connect the pulser output to the clock input of the counter. At the time of connecting pulser, counter output may change. In that case, switch OFF the trainer and again switch it ON.
4. Now for each pulse, counter output changes from 0000 to 1001.
5. Put the U/D switch in down position and the counter counts in reverse direction.
6. Again put the U/D switch in 'U' mode and connect the 3-input NAND gate output to the load input of the counter.
7. Now depending upon the requirement, particular outputs are connected to the NAND gate inputs.

For example: To construct a module-5counterwhose binary output is 0101, connect second MSB bit and LSB bit (which are '1' (high outputs) to the NAND gate inputs. Now the counter counts from 0000 to 0100 resulting as a module-5-counter.

NOTE: In down counting mode, we cannot use this counter as module-N-counter because MSB bit initially activates.

PRECAUTIONS:

1.Make the connection according to the circuit diagram.

2.Check the connections before on the supply.

3.Care should be taken in case of designing mode-n-counter for counting less than n-bit in taken connection to the NAND gate.

EXPERIMENT No- 6

OBJECT: Breadboard implementation of 4 bit Ripple Counter and verify the characteristic table using IC's 7493

APPARATUS: 4-bit Ripple counter kit using IC 7493, Patch chord.

THEORY: The Counter driven by a clock is used to count number of clock pulses occurs at known intervals. A binary ripple counter can be constructed by use of clocked JK flip-flops. The system clock is square wave, drives flip flop A, the output of A drives B & the O/P of B drives flip flop C. All the JK I/P is tied to +Vcc. This means that each flip-flop will change stage (toggle) with negative transition at its clock input.
When the O/P of the clock is used as the clock I/O for the next flip flop, We call the Counter a Ripple counter & asynchronous counter .The A-flip flop must change state before it can trigger the B flip flop, and the B-flip flop has to change states before it can trigger the C flip flop. The triggers move through the flip flop like a Ripple in water, Because of this the over all propagation delay time is the sum of the individual delays.

OBSERVATION:

MR1	MR2	Output
0	0	Count
0	1	Count
1	0	Count
1	1	No Count

OBSERVATION WITH CLOCK PULSE: -

C/k	D	C	B	A	Count
Π	L	L	L	L	0
Π	L	L	L	H	1
п	L	L	H	L	2
п	L	L	H	H	3
п	L	H	L	L	4
п	L	H	L	H	5
п	L	H	H	L	6
п	L	H	H	H	7
п	H	L	L	L	8
п	H	L	L	H	9
п	H	L	H	L	10
п	H	L	H	H	11
п	H	H	L	L	12
п	H	H	L	H	13
п	H	H	H	L	14
п	H	H	H	H	15

PROCEDURE:

1. Switch on the experimental kit.
2. Connect the monopulse output to the clock input of 7493.

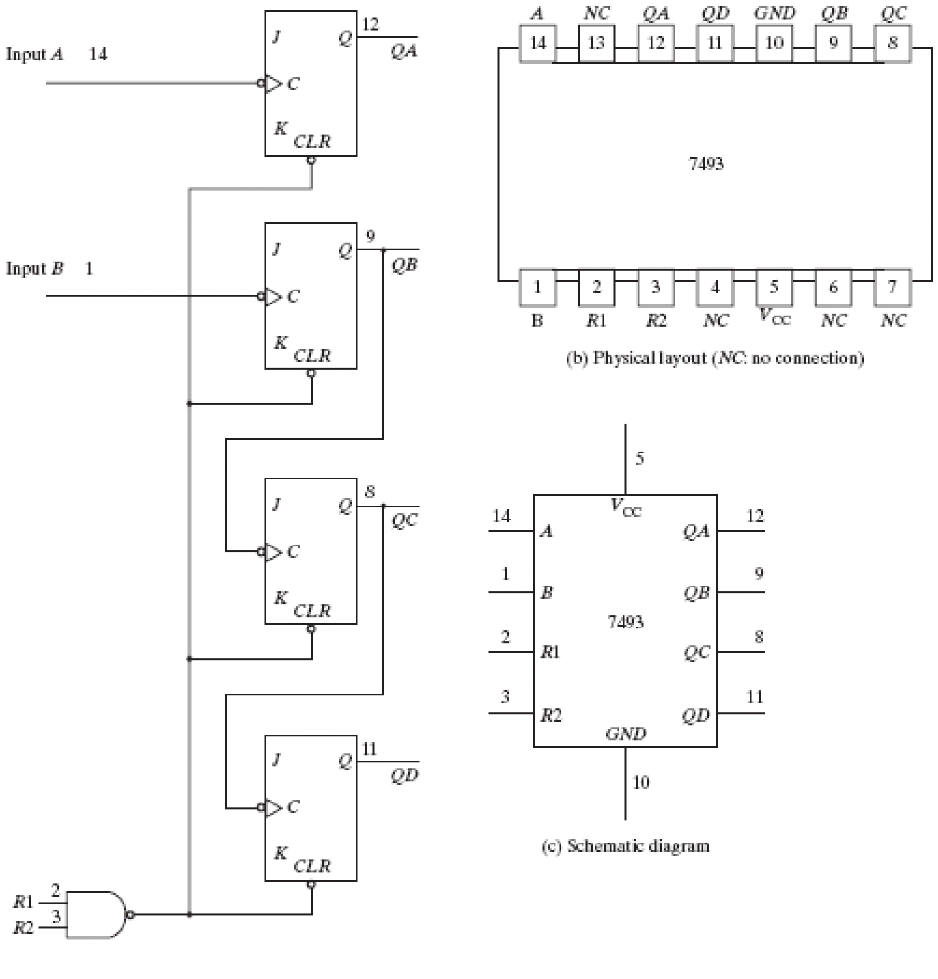
3. Reset the counter with MR1 And MR2 switches and set the counter.
4. Observe the counter and verify the truth table.

RESULT: - 4 bit ripple counter counts the no. Sequentially.

PRECAUTIONS: -

1. Connection should be tight.
2. O/P should be finding sequentially.
3. IC's should be handled carefully.

CIRCUIT DIAGRAM:



(a) Internal circuit diagram

(b) Physical layout (NC: no connection)

(c) Schematic diagram

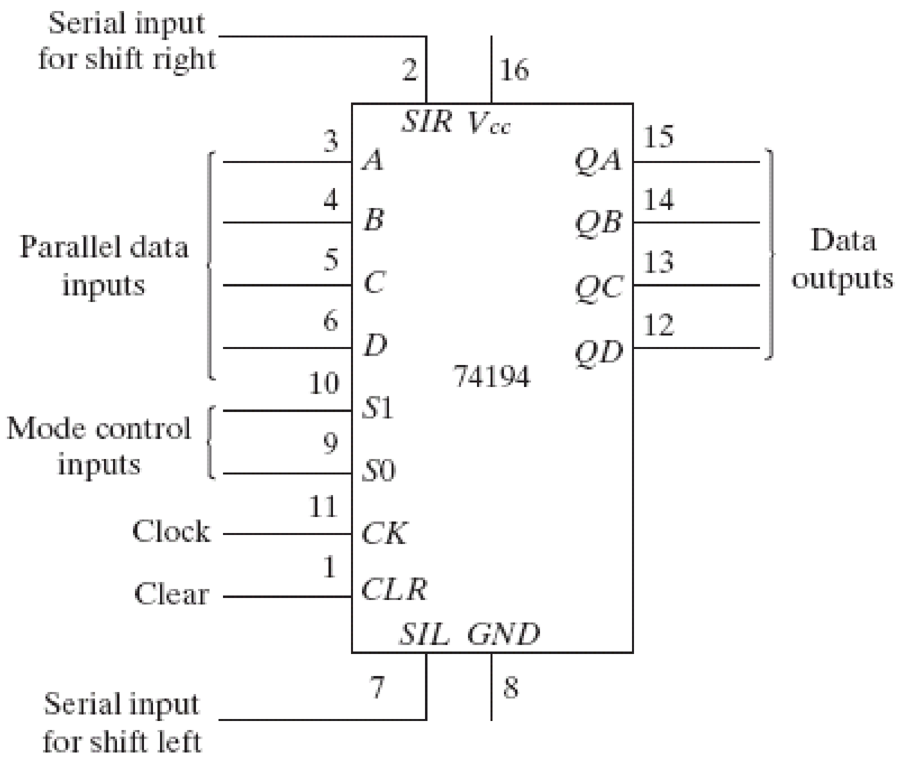
EXPERIMENT No - 7

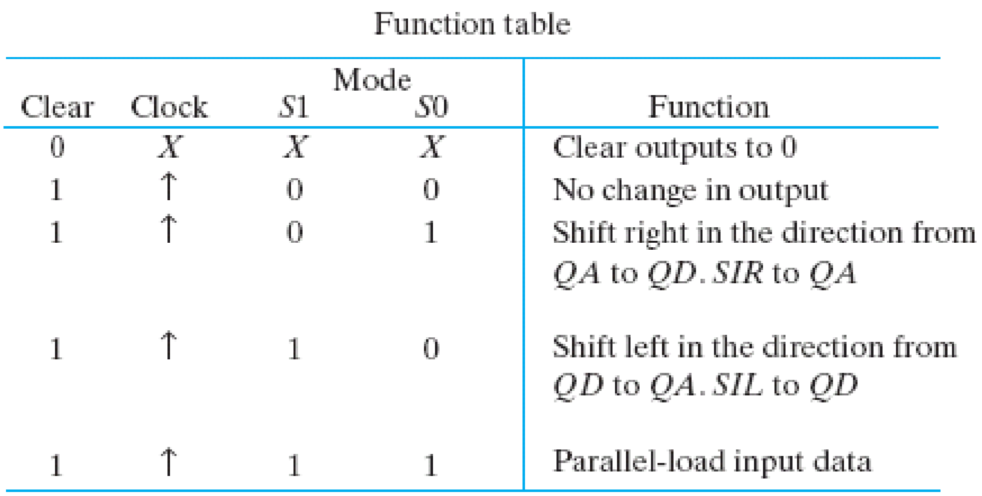
OBJECT: -Breadboard implementation of Shift Register.

APPARATUS USED: -

1. General-purpose trainer board.
2. Ic7491- Serial In Parallel Out
3. Ic74164- Serial In Parallel Out
4. Ic74165 – Parallel In Serial Out
5. Ic74194- Parallel In Parallel Out
6. Patch Cords

THEORY: - A register is simply a group of flip-flop that can be used to store a binary no. of a group of Flip- flop connected to provide either or both of these function is called shift register.
To allow the data in the word to read in to the register serially. The o/p of the flip-flop is connected to the i/p of the following binary such a configuration called a Shift register.
There are two ways to shift the data into a register (serial and parallel) and similarly two ways to shift the data out of the register. This leads to construction of four types of registers.





Function table

Clear	Clock	Mode S1	 S0	Function
0	X	X	X	Clear outputs to 0
1	↑	0	0	No change in output
1	↑	0	1	Shift right in the direction from *QA* to *QD*. *SIR* to *QA*
1	↑	1	0	Shift left in the direction from *QD* to *QA*. *SIL* to *QD*
1	↑	1	1	Parallel-load input data

Fig. 11-19 IC Type 74194 Bidirectional Shift Register with Parallel Load

Fig. IC Type 74194 Bidirectional Shift Register with Parallel Load

EXPERIMENT No. - 8

EXPERIMENT: Implementation of arithmetic algorithms.

APPARATUS REQUIRED:

S.No.	Equipment	Qty.
1	General purpose digital trainer	1
2	IC-74181	1

THEORY: - Arithmetic logic unit is a multipurpose device capable of providing several different arithmetic and logic operations. The specific operation to be performed is selected by the user by placing a specific binary code on the mode select i/p. ALU s are available in large scale integrated circuit packages.

Functional block diag. For 74181 ALU is shown in fig. It is a 4-bit ALU, which provides 16 arithmetic plus 16 logic operations. The unit accepts two 4-bit words (A_3 A_2 A_1 A_0 and B_3 B_2 B_1 B_0) and a carry i/p Cn as i/p's.

The operation to be performed on these i/p are determined by logic levels on i/ps

PROCEDURE:

1. Put IC on the breadboard.
2. Apply Vcc supply at pin 24.
3. Apply ground at pin 12.
4. Make connections as shown in the circuit diagram.
5. Observe the different outputs.

OBSERVATION TABLE:

SELECTION				M=1 LOGIC FUNCTION	M=0 ARITHMRTIC OPERATION
S_3	S_2	S_1	S_0		
0	0	0	0	F=A	F=A
0	0	0	1	F=A+B	F=A+B
0	0	1	0	F=AB	F=A+B
0	0	1	1	F=0	F= -1
0	1	0	0	F=AB	F=A+AB
0	1	0	1	F=B	F=(A+B)+ AB
0	1	1	0	F=A O B	F=A – B -1
0	1	1	1	F=AB	F=AB -1
1	0	0	0	F=A+B	F=A + AB
1	0	0	1	F=A O B	F=A+B
1	0	1	0	F=B	F= (A+B) + AB
1	0	1	1	F=AB	F=AB -1
1	1	0	0	F=1	A + A*
1	1	0	1	F=A+B	F=(A+B) +A
1	1	1	0	F=A+B	F=(A+B) +A
1	1	1	1	F=A	F=A - 1

RESULT: The functional table is verified for 74181 IC.

PRECAUTIONS: -

1. Connection should be tight.
2. O/P should be finding sequentially.
3. IC's should be handled carefully.

2.Truth Table (SISO)

Input		**Output**
tn		**tn+8**
Ds1	Ds2	Q7
L	L	L
L	H	L
H	L	L
H	H	H

3.Truth Table (SIPO)

Operating mode	**Input**			**Output**	
	MR	**DS1**	**DS2**	**Q0**	**Q1 - Q7**
Reset	L	X	X	L	L - L
Shift	H	L	L	L	q0 - q6
	H	H	H	L	q0 - q6
	H	H	H	L	q0 - q6
	H	H	H	H	q0 - q6

4.Truth Table (PIPO)

Operating mode	**Input**						**Output**			
	MR	**S1**	**S0**	**DSR**	**DSL**	**PN**	**Q0**	**Q1**	**Q2**	**Q3**
Reset	L	X	X	X	X	X	L	L	L	L
Hold	H	L	L	X	X	X	q0	q1	q2	q3
Shift left	H	H	L	X	L	X	q1	q2	q3	L
	H	L	H	X	L	X	q1	q2	q3	H
Shift right	H	L	H	L	X	X	L	q0	q1	q2
	H	L	H	H	X	X	H	q0	q1	q2
Parallel load	H	H	H	X	X	Pn	P0	P1	P2	P3

RESULT:- All Shift Register Verify With the given truth table.

PRECAUTIONS: -

1. Connection should be tight.
2. O/P should be finding sequentially.
3. IC's should be handled carefully.

EXPERIMENT No.-9

EXPERIMENT: Breadboard implementation of seven-segment display.

APPARATUS REQUIRED:

S.No.	Equipment	Qty.
1	General purpose digital trainer	1
2	LEDs	1

THEORY: A BCD to seven-segment decoder is a combinational circuit that accepts a decimal digit in BCD and generates the appropriate outputs for selection of segments in display indicator used for displaying the decimal digits. The seven outputs of decoder (a, b, c, d, e, f, g) selects the corresponding segments in the display as shown in the fig.1.the numeric designation chosen to represent the decimal digits

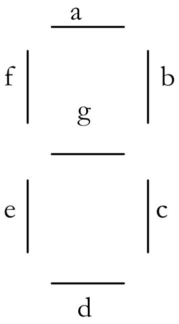


fig.1

PROCEDURE:

1. Put IC on the breadboard.
2. Make connections as shown in the circuit diagram.
3. Observe the different outputs.

OBSERVATION TABLE:

S.No.	BCD Input	seven segment Output
1.		
2.		

RESULT: the operation of BCD to seven-segment decoder has been performed

PRECAUTIONS: -

1. Connection should be tight.
2. IC's should be handled carefully.

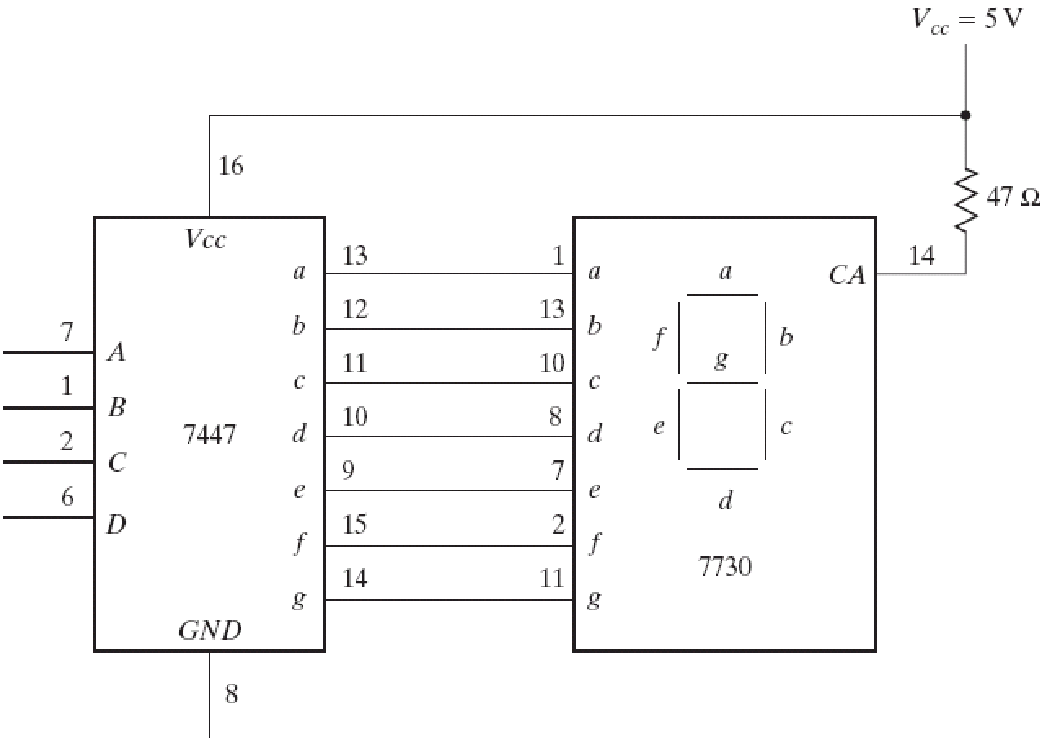


Fig. 11-8 BCD-to-Seven-Segment Decoder (7447) and Seven-Segment Display (7730)