Practical Workbook
CS-205/CS-251
Logic Design & Switching Theory
(EE / SCIT / SE)

Name : _____

Year : _____

Batch : _____

Roll No : _____

Department: _____

Dept. of Computer & Information Systems Engineering

NED University of Engineering & Technology

INTRODUCTION

The logic designing area covers the digital building blocks, tools, and techniques in the design of computers and other digital systems. Logic Design and Switching Theory covers a variety of basic topics, including switching theory, combinational and sequential logic circuits, and memory elements.

In this Practical Workbook, laboratory sessions based on both combinational and sequential logic are covered. The lab sessions fall into two categories:

1. Hardware implementation and IC testing. It covers combinational and sequential circuit building on a bread board or logic trainer board, and testing of various MSI ICs including registers and different types of counters.
2. Logic circuit simulation on CAD software – Electronics Workbench (EWB). EWB is excellent simulation software, where circuits can be designed and tested before physical implementation. Various laboratory sessions of this workbook provide activities and exercises on EWB.

All laboratory sessions of this workbook incorporate brief theoretical backgrounds, as details may be covered in the respective theory classes. Exercises / activities are included with almost all the sessions for the students to practice.

Three appendices are also included in this workbook. The first one provides pin diagrams for all the ICs required for the laboratory work provided in this workbook. It will help the students in preparing the pin diagrams for the circuits. Second appendix covers hardware equipment /components other than ICs that are commonly required in building circuits / mini projects. Third appendix discusses generation of square wave via 555 timer IC and a hardware debouncing circuit for mechanical switches as such switches are extensively used for input purpose in logic circuits.

PREREQUISITES

The students coming for the lab sessions of Logic Design & Switching Theory, should have a personal hardware kit containing the following items:

1 Power supply (preferably student made).

2 An Input-Output panel as shown in figure A (preferably student made).

3 2-3 ICs of each of the following type:

7400, 7404, 7408, 7432, 7447, 7473, 7474, 7476, 7486

4 1 IC of each of the following type:

7490, 74138, 74148, 74150, 74194, any EEPROM

5 Breadboard

6 Connecting wires

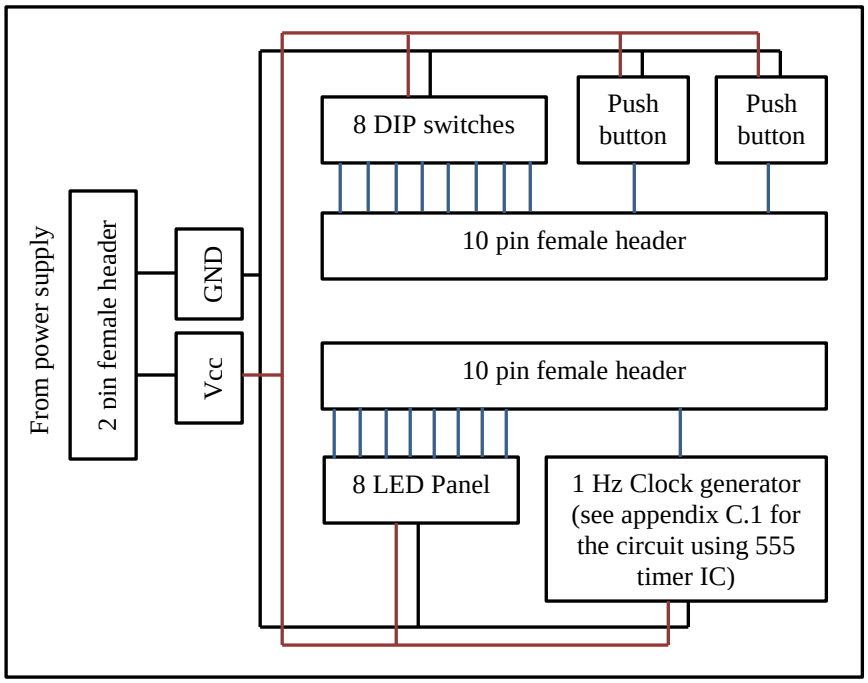


Figure A: *Input-Output panel*

CONTENTS

Lab Session 01

OBJECTIVE

Experimenting with logic gates, implementing the circuit on bread board and observing output for various combinations of inputs.

COMPONENTS REQUIRED

- Bread board, 5 V - power supply, multimeter, logic probe (or LEDs with resistors), wires and the following ICs:
- 7408 quad 2 input AND gate
- 7432 quad 2 input OR gate
- 7404 hex inverter
- 7400 quad 2 input NAND gate
- 7402 quad 2 input NOR gate

THEORY

Logic Gates

Logic gates are the fundamental building blocks of digital systems. These devices are able to make decisions, in the sense that they produce one output level when some combinations of input levels are present and a different output when other combinations are applied; hence given the name Logic Gates.

The two levels produced by digital circuitry are referred to variously as HIGH and LOW, TRUE and FALSE, ON and OFF, or simply 1 and 0.

There are only three basic gates: AND, OR and NOT. The other gates are merely combinations of these basic gates.

1. **The AND Gate** – An AND gate's output is 1 if and only if all its inputs are 1. If A and B are two inputs to an AND gate then output, F of the gate is given as: $F = A.B$

2. **The OR Gate -** An OR gate's output is 1 if at least one of its input is 1. If A and B are two inputs to an OR gate then output, F of the gate is given as: $F = A+B$

3. **The NOT Gate (Inverter)** – Its output is 1 when its input is 0 and its output is 0 when the input is 1; i.e. it complements a digital variable. If A is the input to a NOT gate then output, F of the gate is given as: $F = \overline{A}$

4. **The NAND Gate** – Its output is 1 if at least one of its inputs is 0. This gate performs the same logic as an AND gate followed by an inverter. If A and B are two inputs to a NAND gate then output, F of the gate is given as: $F = A.B$

5. **The NOR Gate** - The output of a NOR gate is 1 if and only if all its inputs are 0. This gate performs the same logic function as an OR gate followed by an inverter. If A and B are two inputs to a NAND gate then output, F of the gate is given as: $F = \overline{A+B}$

All the above gates have one output and two or more inputs except the NOT gate, which has only one input.

INTERNAL IC DIAGRAMS

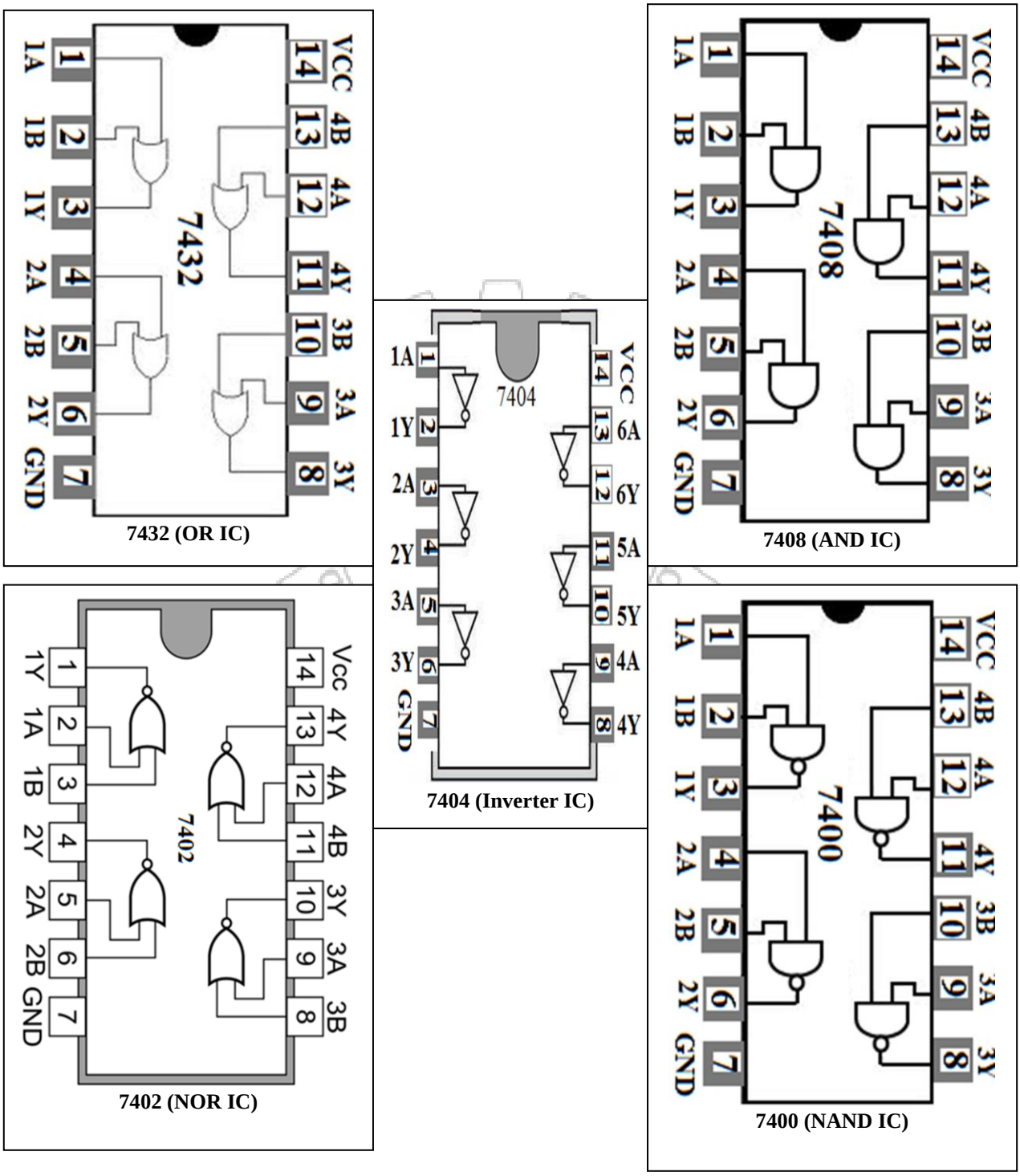


Figure 1: ICs Internal Diagrams

PROCEDURE

1. Set the power supply to 5V. With the help of a multimeter check the voltage at the output knobs of the power supply.
2. Connect wires, long enough to reach the bread board, with the two knobs of the power supply. Again using multimeter, check the voltage at the non-connected end of the wires.
3. Insert the 7408 quad 2 input AND gate IC on to the bread board and make supply and ground connections by joining 5V wire to pin # 14 and 0V wire to pin # 7.
4. Consult IC's internal connection diagram for input and output pins of the first AND gate. Connect input pins to logic 0 (0V) and observe the output using LED or logic probe.
5. Try different combinations of logic levels at the two inputs. Again observe the output.
6. Repeat the last two steps for all other gates of the same IC. Record the observations.
7. Repeat this procedure for all other ICs.

OBSERVATIONS

AND Gate

A	B	Expected Output	Observed Output
0	0		
0	1		
1	0		
1	1		

OR Gate

A	B	Expected Output	Observed Output
0	0		
0	1		
1	0		
1	1		

NOT Gate

A	Expected Output	Observed Output
0		
1		

NAND Gate

A	B	Expected Output	Observed Output
0	0		
0	1		
1	0		
1	1		

NOR Gate

A	B	Expected Output	Observed Output
0	0		
0	1		
1	0		
1	1		

FINDING EXPRESSION FOR GIVEN LOGIC DIAGRAM

LOGIC DIAGRAM

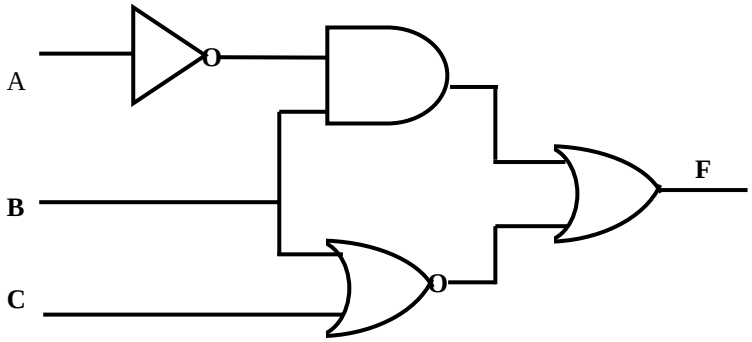


Figure 2: Logic Diagram

PROCEDURE

1. Set the power supply to 5V. With the help of a multimeter check the voltage at the output knobs of the power supply.
2. Connect wires, long enough to reach the bread board, with the two knobs of the power supply. Again using millimeter, check the voltage at the non-connected end of the wires.
3. Insert ICs on the bread board and make their supply and ground connections.
4. As given in the logic diagram, make connections using wires and gates in the ICs.
5. Apply different combinations at the three inputs and observe the output.

OBSERVATIONS

Logic expression for the given logic diagram: _____

A	B	C	Expected Output	Observed Output
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Lab Session 02

OBJECTIVE

Working with Electronics Workbench

ELECTRONICS WORKBENCH - EWB

Electronics Workbench is a computer aided design tool that provides you with all the components and instruments necessary to create board-level designs. It has complete mixed analog and digital simulation and graphical waveform analysis, allowing you to design your circuit and then analyze it using different simulated instruments and analysis options. It is fully integrated and interactive, thus you can change your circuits quickly, allowing fast and repeated what-if analysis.

Electronics Workbench provides the following kind of components:

- Sources parts bin *(AC voltage source, Vcc source, ground, battery, etc)*
- Basic parts bin *(resistors, capacitors, transformers, switches, etc)*
- Diodes parts bin
- Transistors parts bin
- Analog ICs parts bin *(op-amps, etc)*
- Mixed ICs parts bin *(ADCs, DACs, 555 timers, etc)*
- Digital ICs parts bin *(AND, OR, adders, multiplexers, etc)*
- Indicators parts bin *(voltmeter, ammeter, probe, displays, etc)*
- Controls parts bin *(voltage differentiator, multiplier, etc)*
- Instruments parts bin *(multimeter, oscilloscope, function generator, etc)*
- Miscellaneous parts bin *(write data, textbox, etc)*
- Write data: This component allows you to save simulation results as an ASCII file.
- Text Box: Use this to add descriptive text anywhere in a circuit.

DESIGNING LOGIC CIRCUIT FOR A GIVEN LOGIC EXPRESSION

Given Logic Expression: $F = \overline{A}.B + A.\overline{B}.C$

Procedure

1. From *Logic Gates Parts Bin*, drag and drop the required logic gates on the design area. Use *Component Properties* dialog box to customize these gates.
2. Connect the terminal of these gates according to the given expression. Use additional connectors form the *Basic Parts Bin* if more than one wire needs to be connected at a single node.
3. Drag and drop a *probe* from *Indicators Parts Bin.* Use *Component Properties* dialog box to customize the color and other properties of the probe. Connect this probe at the output terminal of the circuit to indicate results.
4. Select four *switches* form *Basic Parts Bin*. Specify the key that controls the switch by typing its name in the *Value* tab of the *Component Properties* dialog box. For example, if you want the switch to close or open when digit '1' is pressed, type 1 in the *Value* tab, then click *OK*. Assign different keys to all the switches.
5. Connect the output terminals of the switches to each of the circuit inputs A, B, and C.

6. Drag and drop *Vcc* and *Ground* form the *Sources Parts Bin*.
7. Connect *Vcc* terminal to one end and *Ground* terminal to the other end of all the switches.
8. Label the circuit properly using *text boxes* found in the *miscellaneous parts bin*.
9. Run the circuit using the *Activate Simulation* switch. Use the keys you have assigned to the switches to toggle them between *Vcc* and Ground connections, thus providing 1 or 0 respectively to the inputs. Record the results as indicated by the probe for all possible combinations of 1s and 0s at the inputs.

EWB Circuit

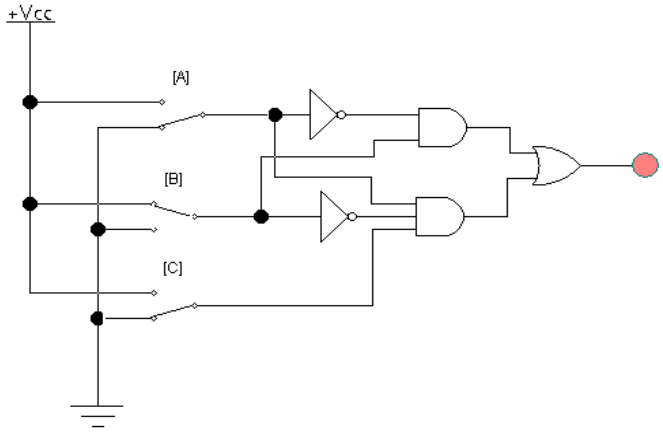


Figure 1: *Designed circuit on EWB*

Observations

A	*B*	*C*	*Expected Output*	*Observed Output*
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

USING LOGIC CONVERTER

Logic converter can be used to derive truth table and logic expression for a given logic circuit or vice versa, i.e. if a logic circuit is expressed in any one of the three ways, other two can be directly obtained using the logic converter.

From the Instruments Parts Bin, drag and drop Logic Converter on the design area. Double click the Logic Converter to reveal Logic Converter dialog box. This dialog box shows various conversion options between truth table, logic expression and logic circuit.

Finding truth table and logic expression for a given logic circuit

1. Create any arbitrary logic circuit on the design area.
2. Attach the input terminals of the logic converter to the input points in the circuit.

3. Connect the single output of the circuit to the output terminal on the logic converter icon.
4. Click the Circuit to Truth Table button. The truth table appears in the logic converter's display.
5. To convert this truth table to a Boolean expression, click the Truth Table to Boolean Expression button. The Boolean expression will be displayed at the bottom of the logic converter.

Creating logic circuit and truth table for a given logic expression

1. Enter the given logic expression in the edit box found at the end of the Logic converter dialog box. Use **'**to represent invert of a variable. For example, $\overline{A}$ is written as **A'**.
2. Click the Boolean Expression to Truth Table button. The truth table appears in the logic converter's display.
3. Now click the Boolean Expression to Circuit button. This creates the logic circuit for the given expression in the design area. Label the diagram if needed.

Creating logic circuit and finding logic expression for a given truth table

1. Click desired number of input channels from A to H, across the top of the logic converter. The display area below the terminals fills up with the necessary combinations of ones and zeros to fulfill the input conditions. The values in the output column on the right are initially set to 0.
2. Edit the output column to specify the desired output for each input condition. To change an output value, select it and type a new value: 1, 0 or x. An x indicates a don't care condition.
3. To convert this truth table to a Boolean expression, click the Truth Table to Boolean Expression button. The Boolean expression will be displayed at the bottom of the logic converter.
4. Simplify the expression by clicking the Simplify button.
5. Now click the Boolean Expression to Circuit button. This creates the logic circuit for the given expression in the design area.

EXERCISE

1. Create a neat pin diagram (using TTL ICs) for the logic expression $F = \overline{A}.B + A.\overline{B}.C$. Test the output and attach hard copy of the diagram here.

Lab Session 03

OBJECTIVE

Simplifying and implementing the given logic expression in hardware and realizing its NAND equivalent.

COMPONENTS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
5. 7400 quad 2 input NAND gate
6. 7404 hex inverter
7. 7408 quad 2 input AND gate
8. 7432 quad 2 input OR gate

GIVEN LOGIC EXPRESSION

$$F(A,B,C,D) = \sum(0,2,3,7,8,10,11,14,15)$$

PROCEDURE

1. Use Karnaugh map to reduce the given function.
2. Draw the circuit diagram for the obtained reduced function.
3. Implement the reduced circuit using digital ICs on a bread board (refer to appendix A for IC pin configurations) and record the observations.
4. Find NAND realization for the simplified circuit.
5. Implement the all NAND circuit using digital ICs on a bread board (refer to appendix A for the pin diagram) and verify the observations taken in step 3.

REDUCTION OF LOGIC EXPRESSION

Reduced logic expression comes out to be: _____

LOGIC DIAGRAM (REDUCED FORM)

NAND REALIZATION OF THE REDUCED LOGIC EXPRESSION

OBSERVATIONS

A	*B*	*C*	*D*	*Expected Output*	*Observed Output*	
					For simplified expression	*For NAND realization*
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			

EXERCISE

1. Simulate the simplified expression implemented in lab using Electronics Workbench or any other simulation tool and generate its NAND equivalent using *Logic Converter* tool. Attach hardcopy of the simulated circuit/output here.

Lab Session 04

OBJECTIVE

Implementing half and full adder circuits.

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
5. 7408 Quad 2-input AND Gate
6. 7432 Quad 2-input OR Gate
7. 7486 Quad 2-input XOR Gate

THEORY

Half Adder

A combination circuit that performs the addition of two bits without accounting for the previous carry is called half adder. It needs two binary inputs and two binary outputs. The input variables designate the augend and addend bits. The output variables produce the sum and carry. The simplified sum of product expressions for a half adder are:

$$S = \bar{x}.y + x.\bar{y} = x \oplus y$$

$$C = x.y$$

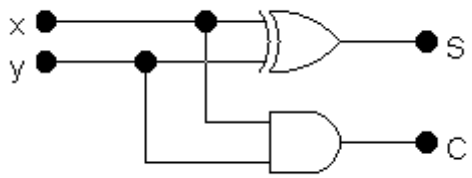


Figure 1: *Logic circuit for Half Adder*

Full Adder

A combinational circuit that performs the addition of three input bits. It consist of three inputs and two outputs. Two of the input variables, represent the two significant bits to be added. The third input, represents the carry from the previous lower significant position. The output variables produce the sum and carry. The simplified sum of product expressions for a half adder are:

$$S = \bar{x}.\bar{y}.z + \bar{x}.y.\bar{z} + x.\bar{y}.\bar{z} + x.y.z = x \oplus y \oplus z$$

$$C = y.x + x.z + y.z = (x \oplus y).z + x.y$$

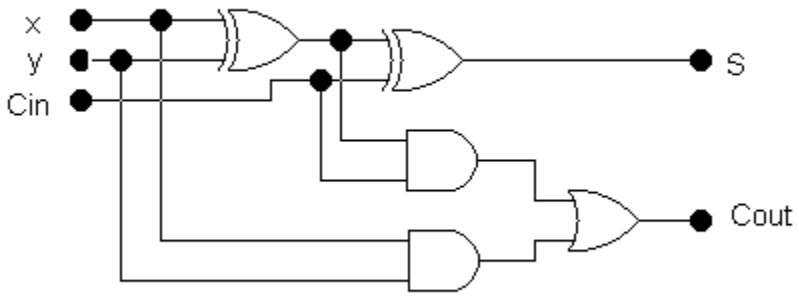


Figure 2: *Circuit diagram for Full Adder*

HARDWARE IMPLEMENTATION AND OBSERVATIONS

Implement the half adder and full adder circuits in hardware (refer to appendix A for IC pin configurations) and record the observations in the following tables:

Half Adder

Inputs		***Outputs***	
X	***Y***	***Carry***	***Sum***
0	0		
0	1		
1	0		
1	1		

Full Adder

Inputs			***Outputs***	
X	***y***	***z***	***Carry***	***Sum***
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

EXERCISES

1. Design a full subtractor circuit and simulate it using Electronic Workbench or any simulation tool. Compare the results of simulation with that of your designed circuit. Attach hardcopy of the simulated circuit/output here.

Lab Session 05

OBJECTIVE

Simulating a 4-bit parallel adder on Electronics Workbench

THEORY

4-Bit Parallel Adder

A 4-bit parallel adder can add two 4-bit values. One way of constructing such an adder is to simply cascade four full adders as shown in figure 1. Here $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ are two 4-bit values to be added. FA3, FA2, FA1 and FA0 are the four full adders that are cascaded in parallel to provide the desired result. $S_3S_2S_1S_0$ is the final 4-bit sum. C_0 is the external carry input (if any) to the circuit and C_4 is the final carry generated by the 4-bit addition.

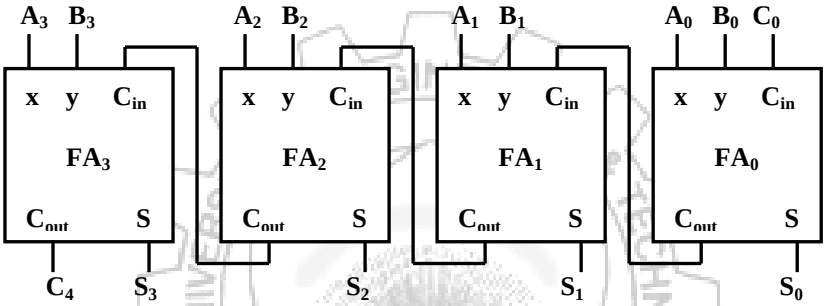


Figure 1: *4-bit Parallel Adder*

Creating Subcircuits in Electronic Workbench

Creating a subcircuit allows you to reuse the circuit multiple times in a design and in future designs. Subcircuits may contain basic circuit elements or other subcircuit definitions. To create a subcircuit in Electronics Workbench:

1. First you have to design the circuit that you want to implement. Let's say you want to create a half-adder, so you simply draw a circuit with a XOR and AND gate (XOR is SUM and AND is Carry Out).
2. Select all components of the design that needs to be included in the subcircuit.
3. From the menu *Circuit*, select *Create Subcircuit.* Type a name and choose appropriate options. This will open a subwindow on the main design area containing the initially selected components.
4. Drag the input and output lines towards the edges to make input and output ports/terminals in the subcircuit.
5. Now click anywhere in the outer design area, this will close the internal structure of the subcircuit. The subcircuit will appear as a block on the main design area with assigned input and output lines. Now this block can be replicated (copy-pasted) where ever required or can be saved for future reuse.

PROCEDURE

1. Create a subcircuit for a full adder unit. Refer to lab session 6 for the gate level circuit of a full adder. This subcircuit should have three inputs namely x, y, C_{in} and two outputs namely C_{out} and S.
2. Replicate the subcircuit to make a total of four full adder subcircuits namely FA3, FA2, FA1 and FA0.
3. Make connections among these full adders as shown in figure 1.
4. Again create a subcircuit containing all these four full adders to make a single unit of a 4-bit parallel adder. This new subcircuit should have nine inputs (C_0, A3, A_2, A_1, A_0, B_3, B_2, B_1, B_0) and five outputs (C_4, S_3, S_2, S_1, S_0).
5. Connect the inputs to switches and outputs to indicators. Apply various combinations of 1s and 0s at inputs and check the binary values at the outputs.

EXERCISES

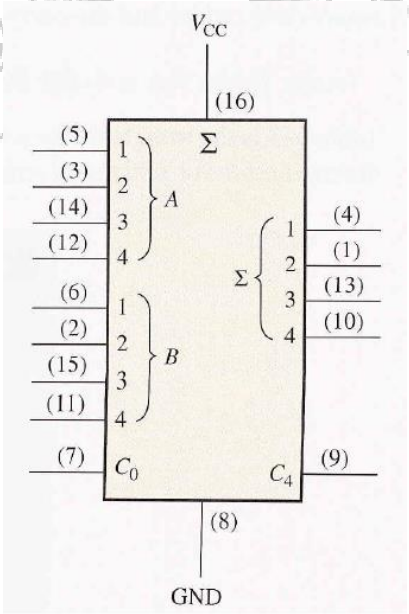
1. Find the sum and output carry for the addition of following two 4-bit numbers if the input carry $C_0 = 0$.
   a. $A_3A_2A_1A_0 = 1100$ $B_3B_2B_1B_0 = 1101$

C_4	S_3	S_2	S_1	S_0

   b. $A_3A_2A_1A_0 = 1011$ $B_3B_2B_1B_0 = 1010$

C_4	S_3	S_2	S_1	S_0

2. Using the logical symbol of parallel adder (given below) realize 2's complement adder/subtractor unit.



3. Test the designed circuit in Q2 for the given values of A and B when $C_0 = 0$.
   a. $A_3A_2A_1A_0 = 1011$ $B_3B_2B_1B_0 = 1101$

C_4	S_3	S_2	S_1	S_0

Lab Session 06

OBJECTIVE

Experimenting with digital decoder IC.

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
5. 74138 - 3 x 8 Line Decoder / Demultiplexer

THEORY

Decoder

A Decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. In practical applications, decoders are often used for selecting one of several devices.

Demultiplexer

A Demultiplexer (DMUX) is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. The selection of a specific output line is controlled by the bit values of n selection lines. A decoder with an enable input can function as a Demultiplexer (DMUX).

74138 - 3 x 8 Line Decoder / Demultiplexer

The 74138 IC has three inputs and eight output lines. It has three enable inputs and for the IC to function all three inputs need to be enabled. Refer to appendix A for IC pin configuration. Function of various pins of this IC is described below:

- ***Y0 through Y7:*** Active low data outputs
- ***A, B, C:*** Active high input / select lines with C being the MSB
- ***G1:*** Active high enable Input
- ***G2A' and G2B':*** Active low enable Inputs
- ***VCC and GND:*** Supply connections lines

TESTING PROCEDURE AND OBSERVATIONS

1 Make Vcc and Gnd supply connections.
2 Connect the three inputs A, B and C to switches.
3 Set appropriate values at enable inputs to activate the IC.
4 Apply different combinations of 1s and 0s at data inputs.
5 Observe the output and record your observations in the following table.

C	*B*	*A*	*Y0*	*Y1*	*Y2*	*Y3*	*Y4*	*Y5*	*Y6*	*Y7*
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

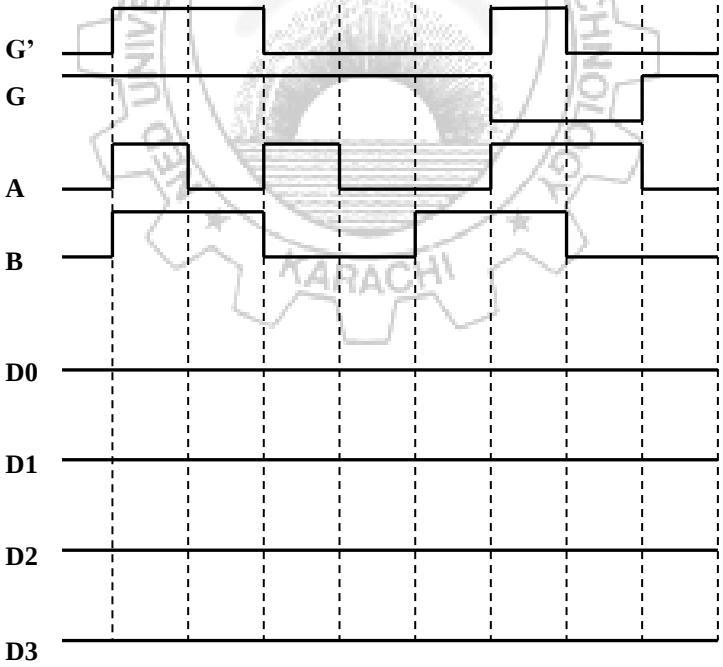
EXERCISES

1. What will be the binary values at the outputs, Y0 through Y7, of 74138 if:
   - All three enable pins are connected to ground?

     _____
     _____
   - All three enable pins are connected to VCC?

     _____
     _____

2. Consider a 2 x 4 Decoder with two enable inputs (one active high - G and one active low – G'). Draw the output wave-forms for D0, D1, D2 and D3, if the two select inputs are A and B (B being the MSB). All outputs are active low.



3. Cascade two 2x4 decoders to form a 3x8 decoder.
   a. Implement the circuit in hardware using two 74138 ICs and observe the output.
   b. Simulate the circuit on Electronics Workbench or any other simulation software using 74139 – dual 2-line-to-4-line decoder/demultiplexer IC. Attach hardcopy of the simulated circuit/output here.

Lab Session 07

OBJECTIVE

Experimenting with digital multiplexer IC.

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
5. 74150 - 16 x 1 Mutiplexer

THEORY

Multiplexers

A digital data Multiplexer (MUX) is a combinational circuit having several data inputs and a single output. A set of *data-select* inputs is used to control which of the data inputs is routed to the single output. A multiplexer is also called a *data selector* because of this ability to select which data input is connected to the output. Normally there are 2^n input lines and n selection lines whose bit combination determine which input is selected.

74150 - 16 x 1 Multiplexer

The 74150 IC has sixteen data inputs and four data-selection lines. It also contains an active low enable and an active low output line. Refer to appendix A for IC pin configuration. Function of various pins of this IC is described below:

- ***E0 through E15:*** Active high data input lines
- ***A, B, C, D:*** Active high data select lines with D being the MSB
- ***W:*** Active low output line
- ***G':*** Active low enable line
- ***VCC and GND:*** Supply connections lines

TESTING PROCEDURE AND OBSERVATIONS

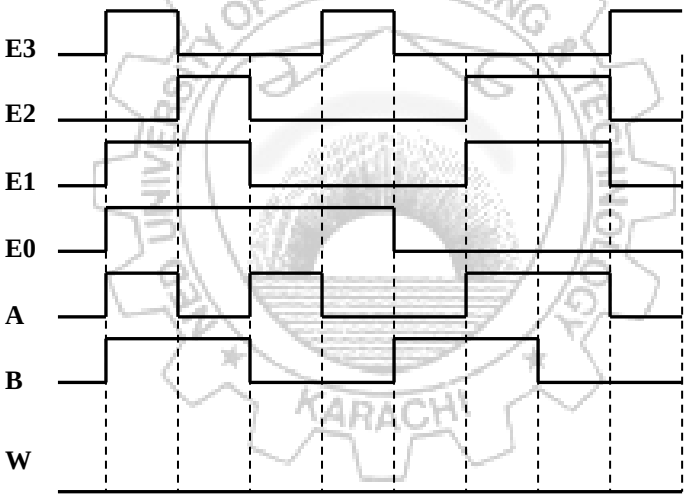
1 Make Vcc and Gnd supply connections.
2 Connect the data select inputs A, B, C and D to switches.
3 Set appropriate values at enable inputs to activate the IC.
4 Connect the data inputs E0 through E15 to switches. For simplicity just two or three data inputs can be connected to switches at a time. Remaining inputs will draw 1 by default (float high characteristic of TTL ICs).
5 Now select any data input with the help of data selectors A, B, C, and D. Apply different data (1 or 0) at this selected data inputs.
6 Observe the output. The invert of the value applied at the selected data input will appear at the output.

7 Record your observations for the input configurations given in the following table. Here only E0, E4 and E15 data inputs are considered.

G'	*D*	*C*	*B*	*A*	*Value at data input*	*W*	*G'*	*D*	*C*	*B*	*A*	*Value at data input*	*W*
0	0	0	0	0			0	1	0	0	1		
0	0	0	0	1			0	1	0	1	0		
0	0	0	1	0			0	1	0	1	1		
0	0	0	1	1			0	1	1	0	0		
0	0	1	0	0			0	1	1	0	1		
0	0	1	0	1			0	1	1	1	0		
0	0	1	1	0			0	1	1	1	1		
0	0	1	1	1			1	0	1	0	1		
0	1	0	0	0			1	1	0	1	0		

EXERCISES

1. Consider 4x1 Multiplexer. Draw the output wave-form for the following data inputs (E_0, E_1, E_2, E_3,) and select lines A, B (B being the MSB). Output is W.



2. Implement the following function using 8x1 multiplexer.

$$F(A,B,C,D) = \sum (1,3,5,6,9,13,15)$$

   a. Implement the circuit in hardware using 74150 IC and observe the output.
   b. Simulate the circuit on Electronics Workbench or any other simulation software using 74151 – 8-line-to-1-line multiplexer IC. Attach hardcopy of the simulated circuit/output here.

Lab Session 08

OBJECTIVE

Experimenting with encoder and seven segment display driver ICs.

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Seven Segment Display (Common Anode Type) with resistor
5. Following ICs and their datasheets or pin configurations:
6. 74148 Octal to Binary Priority Encoder
7. 7447 BCD to Seven Segment Driver
8. 7404 hex inverter

THEORY

Encoder

An Encoder is a digital function that produces a reverse operation from that of a decoder. An Encoder has 2^n (or less) input lines and n output lines. The output lines generate the binary code for the 2^n input variables.

Priority Encoder

A simple encoder may produce an erroneous output if more than one of its inputs is high. A Priority Encoder is one that responds to just one input among those that may be simultaneously high, in accordance with some priority system. The most common priority system is based on the relative magnitudes of the inputs: whichever decimal input is largest is the one that is encoded.

74148 8 x 3 Octal to Binary Priority Encoder

The 74148 is a priority encoder with active-low inputs for decimal digits. There are nine inputs lines (including an enable input) and five output lines, of which three represents the binary code for the octal digit. Refer to appendix A for IC pin configuration. Function of various pins of this IC is described below:

- ***0 through 7:*** Active low data inputs representing the octal digits
- ***A2, A1, A0:*** Active low output lines representing the binary code
- ***E1:*** Active low enable Input
- ***E0:*** Active low output indicating none of the inputs is high
- ***GS:*** Active low output indicating any of the inputs is high
- ***VCC and GND:*** Supply connections lines

7447 BCD to Seven Segment Driver

7447 IC is particularly used to drive common-anode Seven Segment displays. Its input is a BCD number and output drives a seven segment display. Refer to appendix A for IC pin configuration. Function of various pins of these ICs is described below:

- ***A, B, C, D:*** Active high inputs representing BCD digits (D being the MSB).
- ***OA through OG:*** Active low outputs to drive segments *a* through *g* of the display.
- ***RBI:*** Ripple Blanking Input. Turns off all the segments if kept low, provided that LT is kept high and all other inputs (A, B, C, D, BI) are kept low. Should be kept high otherwise.
- ***BI / RBO:*** Wire-AND logic serving as a Blanking Input and / or Ripple Blanking Output.
    - *BI:* Turns off all the segments if low.
    - *RBO:* Goes to a low level (response condition) along with other outputs, when RBI and inputs A, B, C, and D are low with LT input at high level.
- ***LT:*** Lamp Test input. Tests whether all segments are working or not. Illuminates all segments if kept low, provided that BI is kept high. Should be kept high otherwise.
- ***VCC and GND:*** Supply connections lines

Seven Segment Displays

A Seven Segment Display consists of seven light-emitting segments. The segments are designated by letters *a* through *g* (see figure 1). By illuminating various combinations of segments, the numerals 0 through 9 can be displayed. Seven Segment Displays are commonly constructed with light-emitting diodes (LEDs) and with liquid-crystal displays (LCDs). LEDs generally provide greater illumination levels but require much greater power than LCDs.

An LED display can be a common-anode type or common cathode type. In common anode type, a high voltage is applied at the *common* terminal of the display and low voltage is applied at a segment's terminal for illumination. In the common-cathode type, a low voltage is applied at the *common* terminal of the display and high voltage is applied at a segment's terminal for illumination.

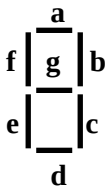


Figure 1: *Seven Segment Display*

IMPLEMENTATION AND OBSERVATIONS

You are required to display the outputs of a 74148 encoder IC on a seven segment display. The circuit is given in figure 2.

1 Make connections as shown in figure 2. (refer to appendix A for IC pin configurations).
2 Select any input from 74148 IC and observe the corresponding decimal code being displayed on the seven segment.

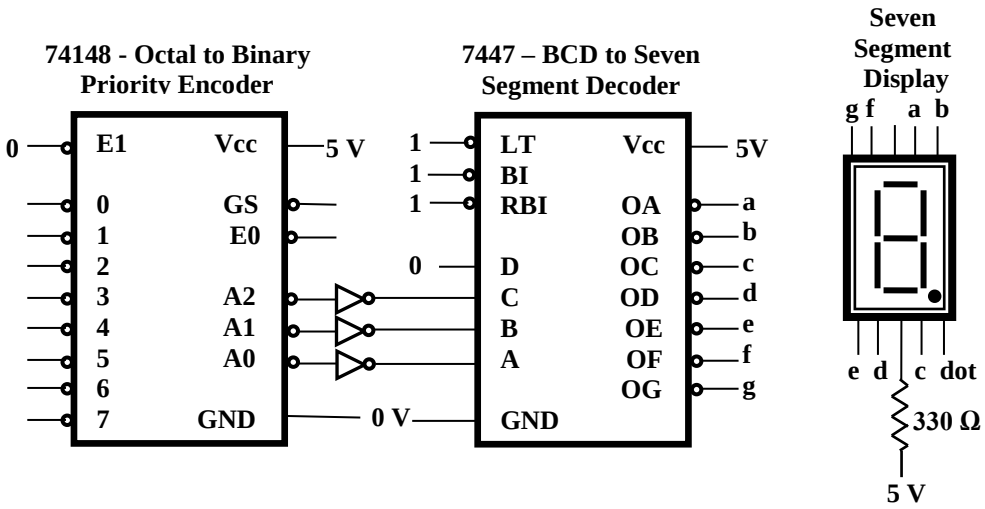


Figure 2: *Circuit for displaying encoders output on 7-segment display*

EXERCISES

1. Perform the Lamp Test for the designed circuit and write your observations.

2. How can you use 7447 IC to drive a common-cathode display?

Lab Session 09

OBJECTIVE

Testing different modes of JK flip flop and implementing a modulo-4 asynchronous up-counter using JK flip flops

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
5. 7473 / 7476 JK Flip-Flop

THEORY

Flip-Flop

A flip-flop circuit can maintain a binary state indefinitely (as long as the power is delivered to the circuit) until directed by an input signal to switch states. The major differences among various types of flip-flops are in the number of inputs they possess and in the manner in which the inputs affect the binary state.

JK Flip-Flop

JK flip flop is an edge triggered device. A typical JK flip flop has three inputs: J, K and a clock input. The flip-flop can be either positive or negative edge triggered. The output Q is available in complemented form as well.

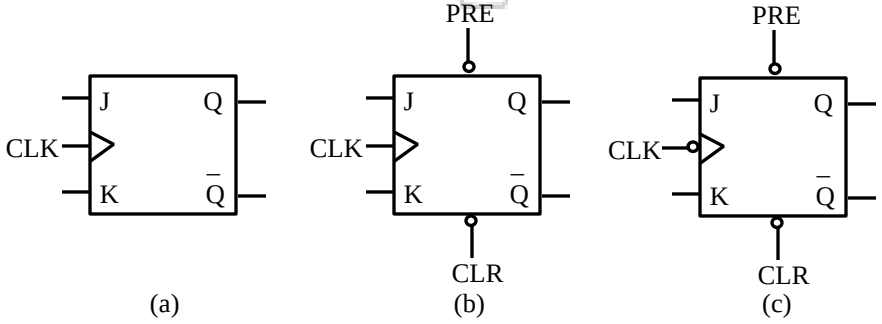


Figure 1: *Symbol for JK flip-flop.*
(a) Positive-edge triggering. (b) Active low Preset (PRE) and Clear (CLR) with positive-edge triggering. (c) Active low Preset (PRE) and Clear (CLR) with negative-edge triggering

Besides the usual inputs and output, most of the flip-flop IC also possess two asynchronous inputs, namely *Preset* and *Clear*. These inputs are usually active low. If used, *Preset* and *Clear* inputs keep the flip-flop in set and reset state respectively, irrespective of the other

inputs. Both of these inputs cannot be used simultaneously, otherwise they will bring the flip-flop in unstable state.

7473 / 7476 Dual JK Flip Flop

Both the ICs 7473 and 7476 are similar in functionality except for one difference. The flip-flops in 7473 have only one type of active low asynchronous input, which is the *Clear* input, whereas the flip-flops in 7476 have both *Preset* and *Clear* inputs. Both these ICs have negative edge triggered flip-flops. Refer to appendix A for IC pin configuration Function of various pins of this IC is described below:

- ***1CLK, 2CLK:*** Negative edge triggered clock inputs for FF1 and FF2 respectively.
- ***1PRE, 2PRE(for 7476 only):*** Active low preset inputs for FF1 and FF2 respectively.
- ***1CLR, 2CLR:*** Active low clear inputs for FF1 and FF2 respectively.
- ***1J, 2J:*** Active high J inputs for FF1 and FF2 respectively.
- ***1K, 2K:*** Active high K inputs for FF1 and FF2 respectively.
- ***1Q, 2Q:*** Active high outputs for FF1 and FF2 respectively.
- ***1Q`, 2Q`:*** Active low outputs for FF1 and FF2 respectively.
- ***VCC and GND:*** Supply connections lines

Digital Counters

A digital counter is a set of flip-flops whose states change in response to pulses applied at the input to the counter. Every counter resets after a certain number of clock pulses. Thus, as it name implies, a counter is used to count pulses. An *n* stage counter can count up to a maximum of 2^n states. *n* is equal to the number of flip-flops required for the construction of counter.

Modulus Counters

The number of input pulses that causes a counter to reset to its initial count is called the *modulus* of the counter. Thus, the modulus equals to total number of distinct *states* (counts), including zero that a counter can store. A binary counter with *n* stages is a *modulo-*2^n (or *MOD-*2^n) counter. The largest count a *mod-N* counter can achieve is *N-1*, i.e. a *mod-N* counter never reaches the binary number equal to its modulus. N is always equal to or less than 2^n.

Counters can be classified as:

- ***Synchronous Counters,*** which are clock driven. All the flip-flops are driven by a single clock.
- ***Asynchronous Counters,*** which are event driven. Clock input is given to the first flip-flop only. Rest of the flip-flops are driven by their preceding flip-flops.

Mod-4 Asynchronous Up Counter

The number of flip-flops required to construct a mod-4 counter is 4. This counter will count from 0 to 3, a total of 4 distinct states.

TESTING PROCEDURE AND OBSERVATIONS

7473 / 7476 Dual JK Flip Flop

1 Make Vcc and Gnd supply connections.
2 Connect the CLK input to some clock source or a switch.
3 Apply different combinations of 1s and 0s at inputs J, K, Preset and Clear.
4 Observe the output and record your observations in the following table.

CLK	*PRE**	*CLR*	*J*	*K*	*Q*
↓	1	1	0	0	
↓	1	1	0	1	
↓	1	1	1	0	
↓	1	1	1	1	
↓	1	0	1	1	
↓	0	1	1	1	
1	1	1	1	1	

*ignore if 7473 IC is being used

MOD-4 Asynchronous Counter

1 Make connections as shown in figure 2. (refer to appendix A for IC pin configurations).
2 Observe the binary values at the outputs of the two flip flops with the incoming clock pulses.

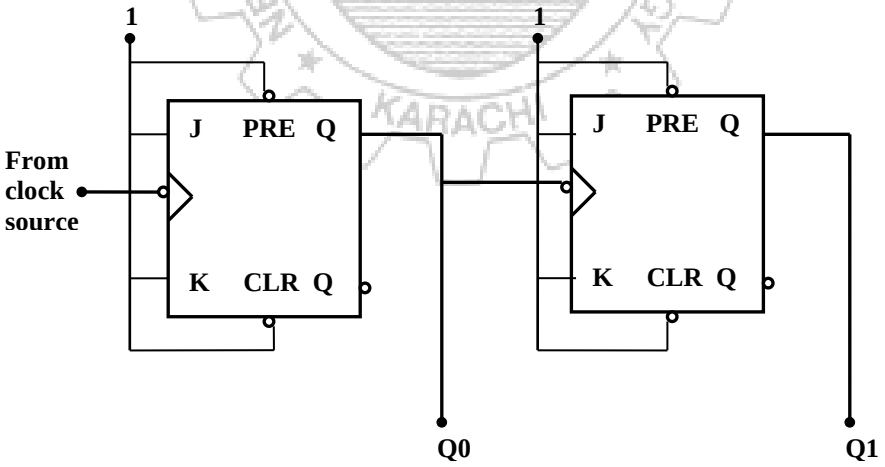


Figure 2: *MOD-4 Asynchronous Counter*

EXERCISES

1. Draw the timing diagram for MOD-4 counter designed in this laboratory session.

2. Simulate MOD-5 asynchronous counter using Electronic Workbench or any other simulation tool. The number of flip-flops required to construct a MOD-5 counter is 3. This counter will count from 0 to 4, a total of 5 distinct states. Since a 3-stage counter can count up to 8 states at maximum, a NAND gate is used to reset it after 5 clock pulses. Attach hardcopy of the simulated circuit/output here.

3. Draw the timing diagram for MOD-5 counter simulated in exercise 2.

Lab Session 10

OBJECTIVE

Experimenting with decade counter IC.

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
5. 7490 decade counter

THEORY

7490 Decade Counter

7490 decade counter can generate the following two sequences:

1. BCD sequence: In this case output QA is connected to input B, external clock is applied to input A
2. Bi-quinary sequence: In this case output QD is connected to input A, external clock is applied to input B

Refer to appendix A for IC pin configuration. Function of various pins of this IC is described below:

- ***Input A, Input B:*** Negative edge triggered clock inputs.
- ***QA, QB, QC and QD:*** Active high outputs.
- ***R91, R92, R01, R02:*** Active high reset inputs. These resets are activated as follows:
- ***VCC and GND:*** Supply connections lines.

Reset Inputs				***Output***			
R01	***R02***	***R91***	***R92***	***QD***	***QC***	***QB***	***QA***
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	COUNT			
L	X	L	X	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			

TESTING PROCEDURE AND OBSERVATIONS

1 Make Vcc and Gnd supply connections.
2 Disable all RESET inputs.

3 Connect input A and input B according to the connections indicated in the previous section to generate BCD count and then bi-quinary count.
4 Observe the output in each case and record your observations given in the following table.

BCD Count				
Clock Pulse	***QD***	***QC***	***QB***	***QA***
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Bi-quinary Count				
Clock Pulse	***QD***	***QC***	***QB***	***QA***
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

EXERCISES

1. Enable the RESET inputs RO1 and RO2 (keeping R91 and R92 disabled) and write your observations below:

   _____
   _____
   _____
   _____

2. Enable the RESET inputs R91 and R92 (keeping R01 and R02 disabled) and write your observations below:

   _____
   _____
   _____
   _____

3. Enable all the RESET inputs R01, R02, R91 and R92 and write your observations below:

   _____
   _____
   _____
   _____

Lab Session 11

OBJECTIVE

Designing and simulating the seconds section of a digital clock

THEORY

Digital Clock

A digital clock is a time keeping circuit that displays seconds, minutes and hours. For seconds' section of a digital clock, a MOD-60 counter needs to be designed. A simple way to do this is to design MOD-10 and MOD-6 counters separately and then cascade them.

Generating Clock (Square Wave) Signals for Sequential Circuits in Electronic Workbench

Most sequential circuits require trigger/pulse for the functioning of their memory elements. This trigger is merely a 0 to1 or 1 to 0 signal transition at the input where it needs to be applied. This can be achieved by simply connecting a switch which can toggle between 0 and 1.

For continuous pulse generation a proper clock source is needed. In EWB, this can be achieved via function generator (in *Instruments Parts Bin*) or more easily as follows:

1. From the *Sources Parts Bin*, drag and drop *Clock* on the design area. Double click the *Clock* to reveal the properties dialog box, where frequency, duty cycle and voltage of the signal can be chosen.
2. Connect the negative end of the component to the ground.
3. Positive end can be connected to the point where the clock signal needs to be applied. You can connect an indicator here to view the generated signal.

PROCEDURE

In this lab session we will use two 7490 ICs to design MOD-10 and MOD-6 counters.

1. For MOD-10 counter, connect a 7490 IC in BCD count mode refer to previous lab session. Disable all RESET inputs and apply clock at input A as illustrated is the previous section. Set the frequency of the clock to 1 Hz.
2. Connect another 7490 IC in BCD mode. To convert it to MOD-6 counter, connect R01 to QB and R02 to QC. This will activate the resets of 7490 when it reaches the count of 6. Disable the other two RESET inputs R91 and R92. Now connect QD of the MOD-10 counter created in the last step to the clock input (input A) of this IC. QD is the MSB of the MOD-10 count and it goes from 1 to 0 only when the counter resets after 9, thus providing the negative edge to increment the MOD-6 counter. Here also a seven segment display can be connected to view the outputs. Now collectively this circuit works as a MOD-60 counter.

3. The output of this circuit can be viewed on seven segment displays. Two kinds of 7 segment displays are available in EWB. One takes four BCD inputs directly and the other one takes seven inputs for the seven segments. The later one is connected via a BCD to seven segment converter. For our application the former one is simpler to use as it provides the BCD to seven segment conversion internally.

EXERCISES

1. Design the hours' section of a digital clock on EWB. The clock should be a 24 hour clock. Attach hardcopy of the simulated design here.

Lab Session 12

OBJECTIVE

Experimenting with bidirectional universal shift register IC

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
5. 74194 4-bit bidirectional universal shift register

THEORY

Shift Registers

A Register is a set of flip-flops used to store binary data. A register, which is capable of shifting its binary information either to the right or left, is called a shift register. The logical configuration of a register consists of a chain of flip-flops connected in cascade, with the output of one flip-flop connected to the input of next flip-flop. All flip-flops receive a common clock pulse, which causes the shift from one stage to the next.

Universal Bidirectional Shift Registers

A bi-directional shift register is one whose bits can be shifted from left to right or from right to left. A universal shift register is a bi-directional register whose input can be in either serial or parallel form and whose output can be in either serial or parallel form.

74194 4-Bit Bidirectional Universal Shift Register

The 74194 register provides parallel as well as serial loading in both directions. A, B, C, and D are inputs for parallel loading, whereas SR and SL are inputs for serial loading with right or left shifting respectively. S1 and S0 are used to select the loading mode. Refer to appendix A for IC pin configuration. Function of various pins of this IC is described below:

- ***A, B, C and D:*** Active high inputs for parallel loading.
- ***QA, QB, QC and QD:*** Active high outputs.
- ***S0 and S1:*** Active high mode control inputs. The following table shows combinations of S1 and S0 to enable various modes.

S1	*S0*	*Clock*	*Action*
0	0	x	No change
0	1	↑	Shift right
1	0	↑	Shift left
1	1	↑	Parallel load

- ***SR:*** Active high serial input for right shifting.
- ***SL:*** Active high serial input for left shifting.
- ***CLR:*** Active low clear input.
- ***CLK:*** Positive edger triggered cock input.
- ***VCC and GND:*** Supply connections lines.

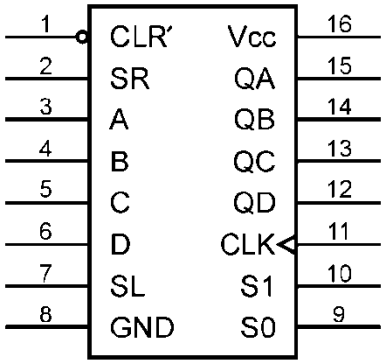
TESTING PROCEDURE AND OBSERVATIONS

1 Make Vcc and Gnd supply connections.
2 Connect the CLK input to some clock source or a switch.
3 Disable CLR input.
4 Apply different combinations of 1s and 0s at inputs J, K, Preset and Clear.
5 Observe the output and record your observations given in the following table.

S1	*S0*	*SL*	*SR*	*A*	*B*	*C*	*D*	*QA*	*QB*	*QC*	*QD*
0	0	0	1	0	0	0	0				
0	1	1	0	0	0	0	1				
1	0	0	0	0	0	1	0				
1	1	0	0	0	0	1	1				
0	0	1	1	0	1	0	0				
0	1	1	1	0	1	0	1				
1	0	1	0	0	1	1	0				
1	1	1	0	0	1	1	1				
1	1	0	0	1	1	1	1				

EXERCISES

1. Show connections of 74194 to convert it into a ring counter with right shifting.



74194

Lab Session 13

OBJECTIVE

Designing and implementing a 3-bit even sequence synchronous up counter using T flip flops

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
   1. 7473 / 7476 JK Flip-Flop

THEORY

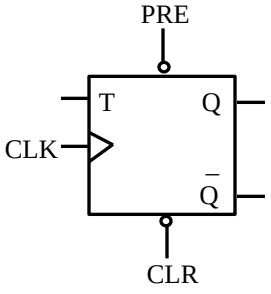
Synchronous counters are clock driven counters. It is a sequential circuit, the design of which involves the following steps:

1. Generate a state diagram and/or a state table from the statement of the problem.
2. Select the type of flip-flop for the circuit and generate the flip flop input equations needed for the required state transitions
3. Derive logic equations for generation of the output from the inputs and current state.
4. Generate a logic diagram of the circuit using ANDs, ORs, inverters, and flip flops.

3-bit Even Sequence Synchronous Up Counter

A 3-bit even sequence synchronous up counter will count through the following sequence: 0, 2, 4, 6 and repeat. The number of flip-flops required to construct such a counter is 3. The state of the flip flops is the final output of the circuit, therefore no explicit output equations need to be generated.

T Flip-Flop



(a)

Q_t	T	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

(b)

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

(c)

Figure 1: *T Flip Flop. (a) Graphical symbol. (b) Characteristic table. (c) Excitation table*

T flip-flop (toggle flip flop) has a single input T. If the T input is high, the T flip-flop changes state (toggles) whenever the clock input strobes. If the T input is low, the flip-flop holds the previous value. This behavior is described by the characteristic table and transition table given in figure 1.

Characteristic equation of T flip-flop is: $Q_{(t+1)} = T \oplus Q_t$

DESIGNING THE COUNTER

State Diagram

State Table

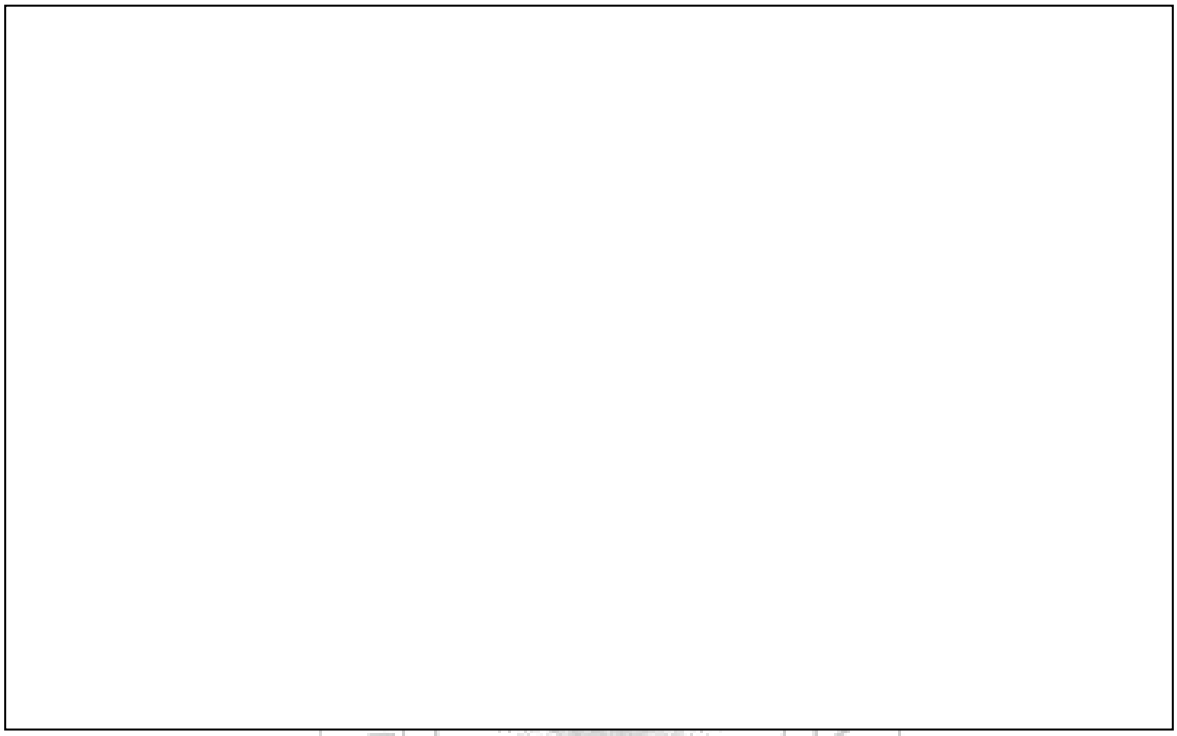
The design requires three flip flops: FF-0, FF-1 and FF-2.

Present State			***Next State***			***Flip flop inputs***		
Q_{2t}	Q_{1t}	Q_{0t}	$Q_{2(t+1)}$	$Q_{1(t+1)}$	$Q_{0(t+1)}$	***FF-2, T_2***	***FF-1, T_1***	***FF-0, T_0***
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

Input Equations

T_2 = _____ , T_1 = _____ , T_0 = _____

Circuit Diagram



1. Implement the circuit using digital ICs on a bread board (refer to appendix A for IC pin configurations). T flip flop can be obtained by connecting together the two inputs J and K, of JK flip flop.
2. Make Vcc and Gnd supply connections.
3. Connect the CLK input to some clock source or a switch.
4. Observe the output even value sequence of the circuit.

Lab Session 14

OBJECTIVE

Designing and implementing synchronous sequential circuit for the given state diagram using D flip-flop.

GIVEN STATE DIAGRAM

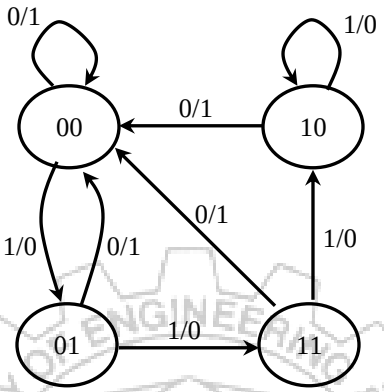


Figure 1: *State Diagram*

COMPONENTS AND APPARATUS REQUIRED

1. Digital logic trainer board or the following components:
   Bread board, 5 V - Power Supply, Multimeter, LEDs with Resistors, Switches
2. Logic probe
3. Connecting wires
4. Following ICs and their datasheets or pin configurations:
5. 7474 D Flip-Flop
6. 7408 Quad 2-input AND Gate
7. 7432 Quad 2-input OR Gate
8. 7404 Hex Inverter

THEORY

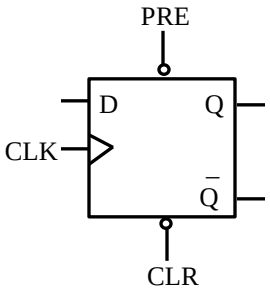
Sequential Circuits

In *Sequential Circuits*, the output not only depends on the present inputs, but also on previous states of the circuit. These circuits use memory elements (latches, flip-flops) and the binary information stored in the memory elements at any given time defines the state of the sequential circuit.

Refer to last lab session for the steps involved in the design of synchronous sequential circuit.

D Flip-Flop

D flip-flop is also called transparent flip-flop as it simply transfers the input data to the output. This behavior is described by the characteristic table and transition table given in figure 2.



(a)

Q_t	D	Q_{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

(b)

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

(c)

Figure 2: *D Flip Flop. (a) Graphical symbol. (b) Characteristic table. (c) Excitation table*

Characteristic equation of D flip-flop is: $Q_{(t+1)} = D$

DESIGNING SYNCHRONOUS SEQUENTIAL CIRCUIT FOR THE GIVEN STATE DIAGRAM

State Table

The given state diagram has four states, so two D flip-flops (FF-0, FF-1) will be required.

Present State		***Input***	***Next State***		***Output***	***Flip flop inputs***	
Q_{1t}	Q_{0t}	x	$Q_{1(t+1)}$	$Q_{0(t+1)}$	z	***FF-1, D_1***	***FF-0, D_0***
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Input and Output Equations

D_1 = _____ , D_0 = _____ , z = _____

Circuit Diagram

IMPLEMENTATION PROCEDURE AND OBSERVATIONS

1. Implement the circuit using digital ICs on a bread board (refer to appendix A for IC pin configurations).
2. Make Vcc and Gnd supply connections.
3. Connect the CLK and the CLR inputs to input switches (preferably push buttons).
4. Apply 0 to the input *x*.
5. Enable CLR input momentarily to clear all the flip flops.
6. Apply CLK through the switch and record your observations in the following table.
7. Apply 1 to the input *x*.
8. Enable CLR input momentarily to clear all the flip flops.
9. Apply CLK through the switch and record your observations in the following table.

x = 0				***x = 1***			
Clock Pulse #	***Q_1***	***Q_0***	***Output z***	***Clock Pulse #***	***Q_1***	***Q_0***	***Output z***
Initial value	0	0		Initial value	0	0	
1				1			
2				2			
3				3			
4				4			
5				5			

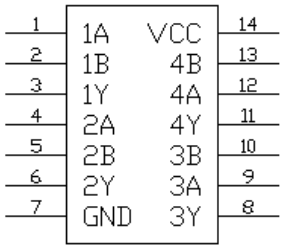
Alternatively, connect the CLK input to a continuous clock source. Now output will change continuously. These outputs can be viewed using an oscilloscope.

Appendix A

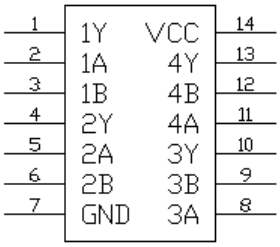
PIN DIAGRAMS OF THE ICS REQUIRED FOR THE LABORATORY SESSIONS

Consult the TTL/IC data book for internal diagrams and electrical characteristics of these ICs.

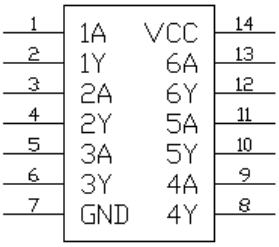
7400 Quad 2-Input NAND



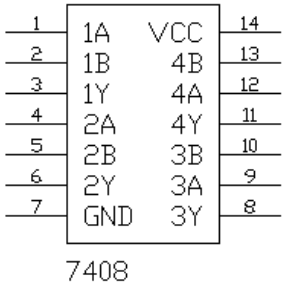
7402 Quad 2-Input NOR



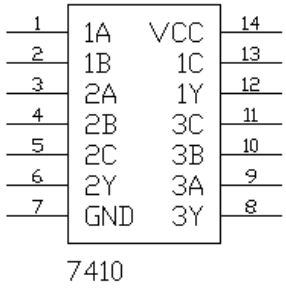
7404 Hex Inverter



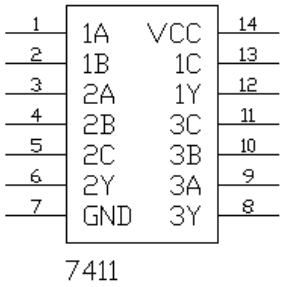
7408 Quad 2-Input AND



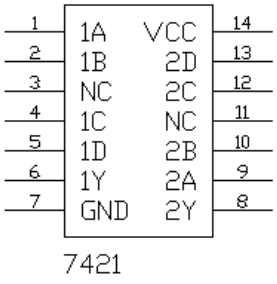
7410 Triple 3-Input NAND



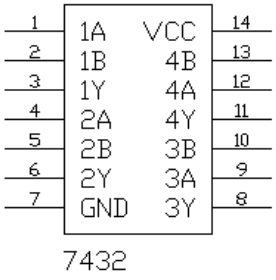
7411 Triple 3-Input AND



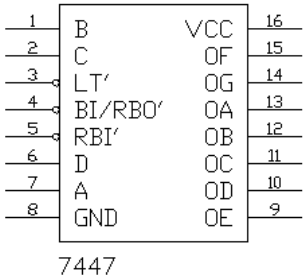
7421 Dual 4-Input AND



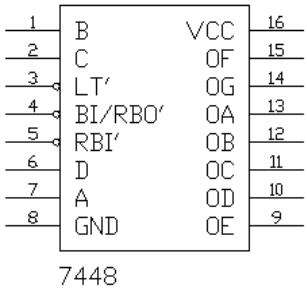
7432 Quad 2-Input OR



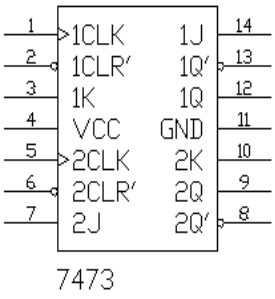
7447 BCD-to-Seven Segment Decoder (15V, O.C.)



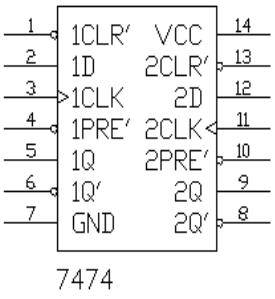
7448 BCD-to-Seven Segment Decoder (2kΩ pull-up output)



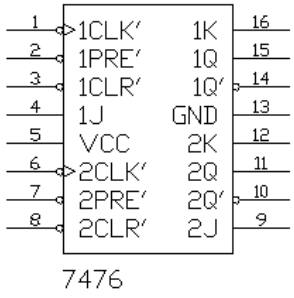
7473 Dual JK Flip-Flop



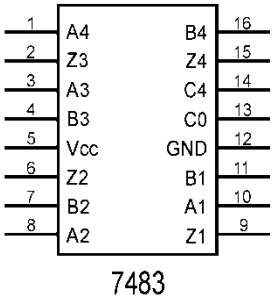
7474 Dual D-Type Flip-Flop



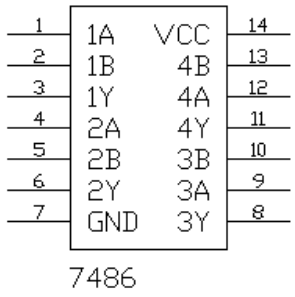
7476 Dual JK Flop-Flop



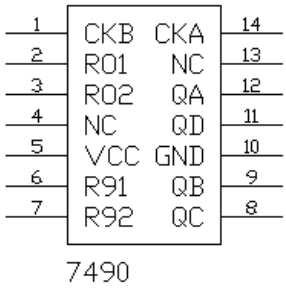
7483 4-Bit Binary Full Adder with Fast Carry



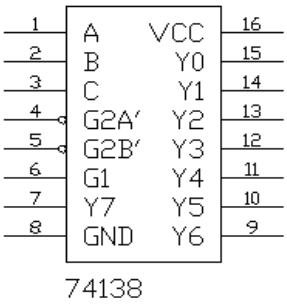
7486 Quad 2-Input Exclusive OR



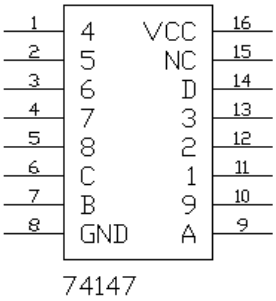
7490 Decade Counter



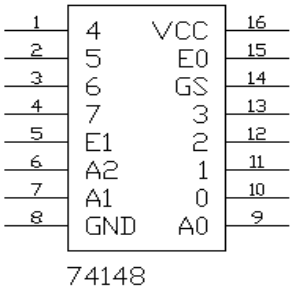
74138 3-to-8 Line Decoder / Demultiplexer



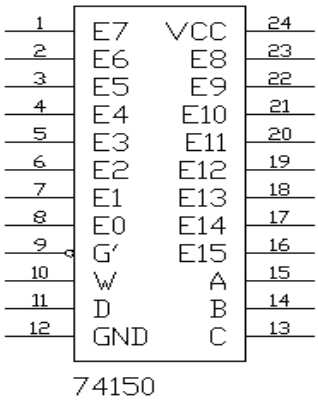
74147 10-to-4 Priority Encoder



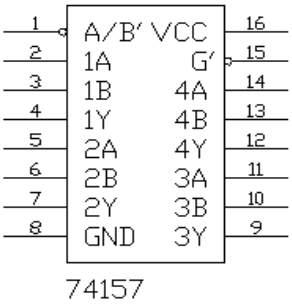
74148 8-to-3 Line Priority Encoder



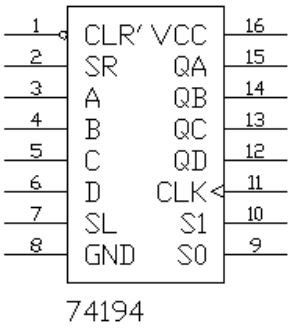
74150 16-to-1 Line Data Selector / Multiplexer



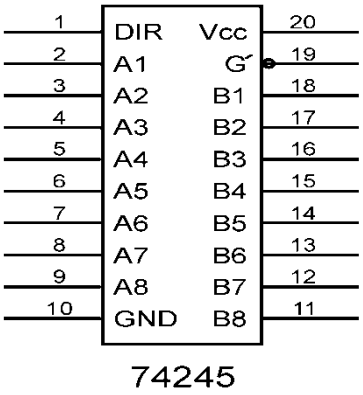
74157 Quad 2x1 Data Selector / Multiplexer



74194 4-Bit Bidirectional Universal Shift Register



74245 Octal Bus Transceivers (3-State)



Appendix B

COMMON COMPONENTS (BESIDE LOGIC ICS) USED IN BUILDING CIRCUITS

Battery

Battery supplies a voltage which drives an electric *current* round the circuit from the positive (+) terminal of the battery to its negative (–) terminal. Voltage is measured in volts (V) and current in *amperes* (A).

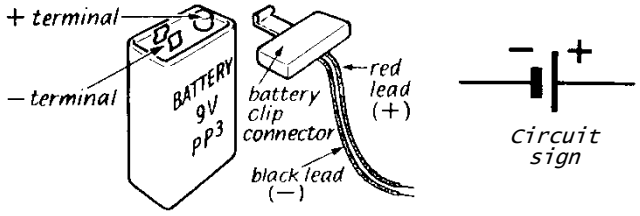


Figure B.1: *Battery (9V, type PP3)*

Connecting Wires

A connecting wire allows current to flow through it easily because it is made of copper which is a good electrical conductor. Insulators like PVC (polyvinyl chloride - a plastic) and enamel are used to cover connecting wires.

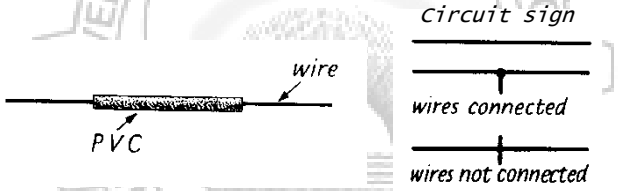


Figure B.2: *Connecting wire (PVC-covered tinned copper wire 1/0.6 mm, i.e. 1 wire of diameter 0.6 mm)*

Resistors

A resistor reduces the current in a circuit because of its resistance.

Color Coded Resistor

The colored bands give the resistance in *ohms*.

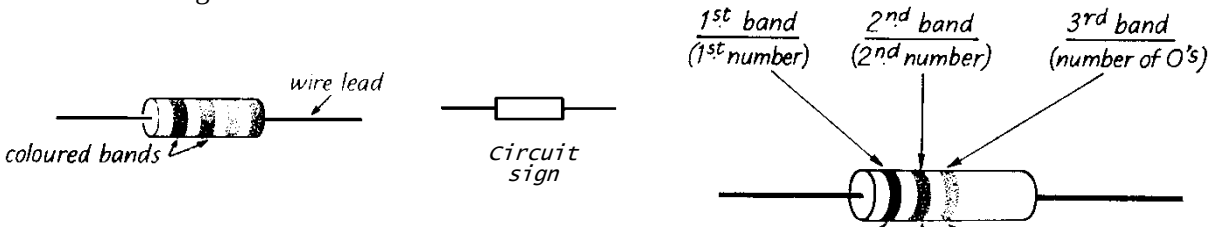


Figure B.3: *Color coded resistor (carbon, ½ watt)*

To distinguish left from right there is a gap between the C and D bands.

- Band A is the first significant figure of component value (left side)
- Band B is the second significant figure

- Band C is the decimal multiplier
- Band D if present, indicates tolerance of value in percent (no color means 20%)

The values are interpreted as given in table B.1.

Color	Significant figures	Multiplier	Tolerance
Black	0	x 10^0	-
Brown	1	x 10^1	±1%
Red	2	x 10^2	±2%
Orange	3	x 10^3	-
Yellow	4	x 10^4	-
Green	5	x 10^5	±0.5%
Blue	6	x 10^6	±0.25%
Violet	7	x 10^7	±0.1%
Gray	8	x 10^8	±0.05%
White	9	x 10^9	-
Gold	-	x 10^{-1}	5%
Silver	-	x 10^{-2}	10%
None	-	-	20%

Table B.1: *Standard color codes*

For example, a resistor with bands of yellow, violet, red and gold will have first digit 4, second digit 7, followed by 2 zeros: 4700 ohms. Gold signifies that the tolerance is ±5%, so the resistance could lie anywhere between 4465 and 4935 ohms.

Variable Resistor

Variable resistors are used when it is necessary to dynamically change the resistance in order to control the current in a circuit, and may also be used when a voltage divider is needed. For example, they are used to control the volume in a radio or the brightness of a lamp.

Variable resistors consist of a resistance track with connections at both ends and a wiper which moves along the track as you turn the spindle (see figure B.4). The track may be made from carbon, cermet (ceramic and metal mixture) or a coil of wire (for low resistances). The track is usually rotary but straight track versions, usually called sliders, are also available.

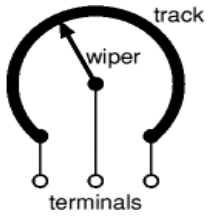


Figure B.4: *Variable resistor*

They are specified by their maximum resistance, linear or logarithmic track, and their physical size. The standard spindle diameter is 6mm. The resistance and type of track are marked on the body (for example: 4K7 LIN means 4.7 kΩ linear track ; 1M LOG means 1 MΩ logarithmic track). Linear (LIN) track means that the resistance changes at a constant rate as you move the wiper. This is the standard arrangement which is assumed if a project does not specify the type of track. Logarithmic (LOG) track means that the resistance changes slowly at one end of the track and rapidly at the other end, so halfway along the track is not half the total resistance. This arrangement is used for volume (loudness) controls because the human ear has a logarithmic response to loudness so fine control (slow change) is required at low volumes and coarser control (rapid change) at high volumes. It is important to connect

the ends of the track the correct way round, if turning the spindle increases the volume rapidly followed by little further change you should swap the connections to the ends of the track.

Variable resistors may be used as a rheostat with two connections (the wiper and just one end of the track) or as a potentiometer with all three connections in use. Miniature versions called presets are made for setting up circuits which will not require further adjustment.

The terminal in the middle is the *wiper*. When a potentiometer is used as a voltage divider, all three terminals are wired separately. But when a potentiometer is used strictly as a rheostat, only need two connections are needed. Either side of the variable resistor may be attached to the circuit board, with the remaining side unattached or grounded, but it is important to always connect the wiper. The wiper must be grounded or affixed to the voltage source. For example, the left terminal of the pot may be attached to the voltage source and the wiper to ground, or the right terminal may be substituted for the left. Changing the connection to a different side alters the direction the knob must be turned in order to achieve maximum resistance. The unused side of the variable resistor may be left unconnected, wired to an unused portion of the breadboard, or wired to ground.

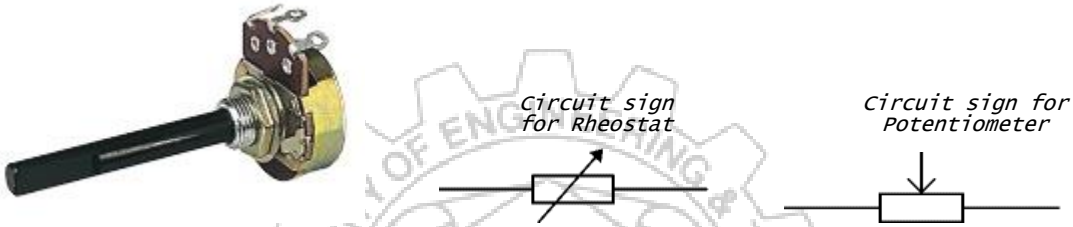


Figure B.5: *Potentiometer and Rheostat*

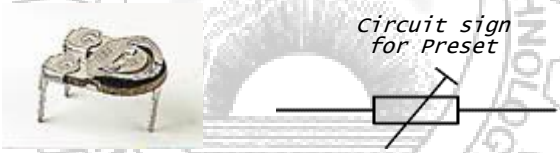


Figure B.6: *Preset*

Light Dependent Resistor (LDR)

When light falls on it, its resistance becomes small; in the dark resistance is high.

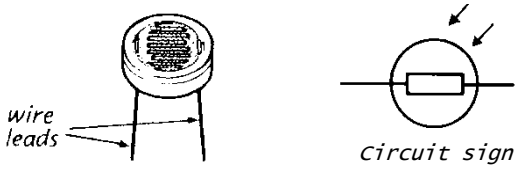


Figure B.7: *Photocell or light dependent resistor (LDR)*

Capacitors

A capacitor stores electricity; the greater the capacitance the more does it store. Capacitance values are measured in microfarads shortened to μF or, less correctly, to mfd. On a capacitor, 0.1 μF may be marked as .l mfd and 0.01μF as 10n. The greatest voltage it can stand is also shown, e.g. 30V.

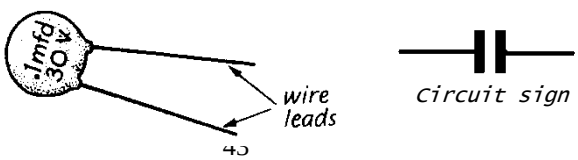


Figure B.8: *Capacitor (disc ceramic type)*

Electrolytic Capacitor

Electrolytic capacitor stores electricity: values usually larger than 1μF. Greatest voltage marked on it. Must be connected the correct way round.

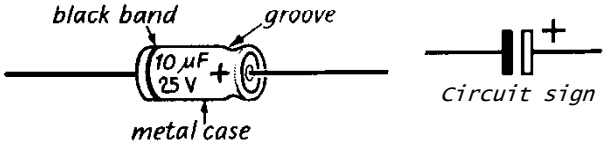


Figure B.9: *Electrolytic Capacitor*

Variable Capacitor

It varies the capacitance in a circuit by moving one set of metal plates in or out of a fixed set when the spindle is rotated. The sets of plates are separated by sheets of an insulator (also called a dielectric).

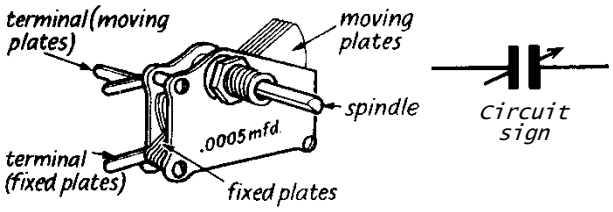


Figure B.10: *Variable Capacitor (0.0005 microfarads)*

Loud Speaker

It changes electric currents into sound.

Figure B.11: *Loud Speaker (2.5 inch, 25 to 80*

Aerial

Changes radio waves into electric currents.

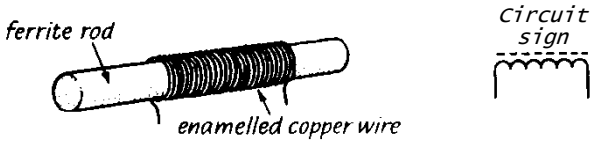


Figure B.12: Ferrite rod aerial

Light Emitting Diode (LED)

An LED lets current flow in one direction but not in the other. When it conducts, light is emitted. Must have a current limiting resistor in series with it. The cathode lead is nearest the 'flat' and may be shorter than the anode lead (but this is not always so). The arrow on the sign shows the conducting direction

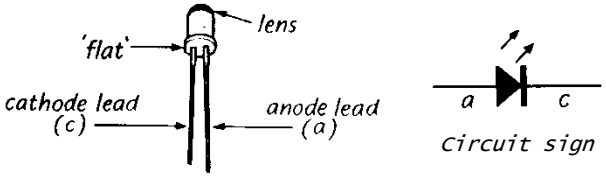


Figure B.13: *LED (light emitting diode)*

Transistor

Transistor amplifies small currents into much larger copies. It acts as a very fast switch. It must be correctly connected with a positive voltage to the collector

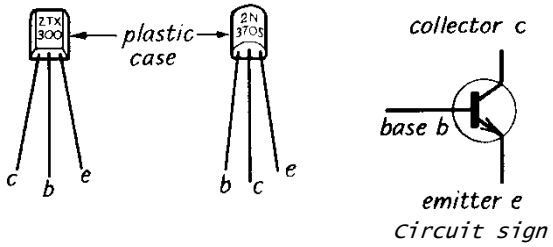


Figure B.14: *Transistor (npn)*
(e.g. ZTX300 or 2N3705)

Switches

A switch is an electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another.

SPDT Switch

Connects terminal A to terminal B or C, i.e. it is a change-over switch.

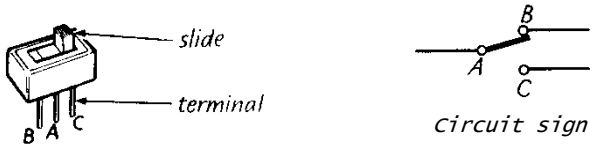


Figure B.15: *Miniature slide switch SPDT (single pole double throw)*

DIP Switch

A DIP switches are manual electric switches that are packaged in a group in a standard dual in-line package (DIP) (the whole package unit may also be referred to as a DIP switch in the singular). It provides an easy way of inputting values to a digital circuit. The Dip Switch requires a Resistor Network Pack for its operation.

The connection configuration is shown in figure B.17. Resistor Network Pack has a pin configuration that must be followed. The side of the resistor pack with the print is the "front." At the far left end of the component there is a black circle. This pin must be connected to the power supply. The other pins do not required to be connected to power. Once the dipswitch and resistor pack is placed on the board, the next thing that needs to be done is to connect wires from all the columns connected to the dip-switch to the ground rail. This way one side of the dip-switch is "tied" to ground.

(a) *(b)*

Figure B.16: *(a) Resistor Network Pack, (b) DIP Switch*

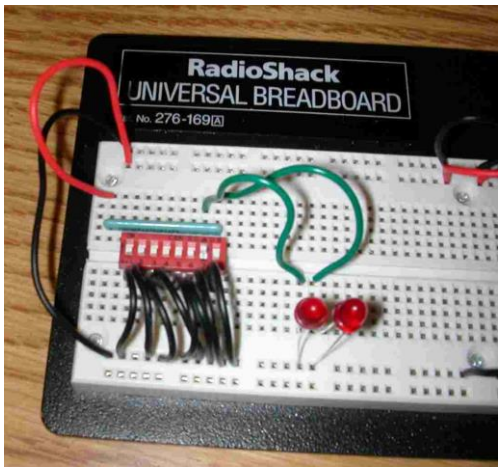


Figure B.17: *Circuit Connections of DIP switch with resistor network pack*

The dip-switch can now be used to "input" values. When the switch is pushed to one side it disconnects the top column from the bottom column on the breadboard. Voltage appears on the column due to the pin of the resistor pack pin connected to that column. When switch is pushed to the other side, the dip-switch connects the top column to the bottom column. Since the top column along with the pin of the resistor pack is "tied" to ground, no voltage appears on the top column. The idea of the dip-switch is that when the dip switch is open, or there is no connection between the top and bottom column, current flows from the resistor pack to the electrical component. However, when the dip-switch is closed, or the top and bottom column are tied together, current flows from the resistor pack to ground rather than to the electrical component. This resistor-pack/dip-switch circuit should be placed as far to the left side of the breadboard as possible to leave space for other components.

Push Button

A Push Switch or Push to make switch, allows electricity to flow between its two contacts when held in. When the button is released, the circuit is broken. Other forms are push to break which, does the opposite.

Relay

A relay is an electrically operated switch. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. These devices use a solenoid to control a heavy-duty switch. The wiring for the solenoid may require only 0.5 amps to activate, while the switch it controls carries 10 to 30 amps.

A Solid State Relay (SSR) is an electronic switch that works without moving parts. Here the low current control and a high current load are isolated optically or with transformers. They are activated by AC control signals or DC control signals from Programmable logic controller (PLCs), PCs, Transistor-transistor logic (TTL) sources, or other microprocessor and microcontroller controls. Since relays are switches, the terminology applied to switches is also applied to relays. A relay will switch one or more *poles*, each of whose contacts can be *thrown* by energizing the coil in one of three ways:

- Normally-open (NO) contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive.
- Normally-closed (NC) contacts disconnect the circuit when the relay is activated; the circuit is connected when the relay is inactive.
- Change-over (CO), or double-throw (DT), contacts control two circuits: one normally-open contact and one normally-closed contact with a common terminal.

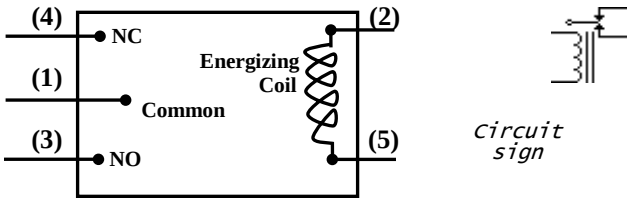


Figure B.18: *Relay*

For circuit connections, consider figure B.18, pins 2 and 5 seem to go to the coil. Pin 1 is the common pin. Pin 4 is the NC (normally closed) pin and pin 3 is the NO (normally open) pin. Connect the control device to the two solenoid pins (pins 2 and 5) of the relay. Supplying a small voltage to these two pins will turn the relay switch on or off. The lower the better until you are sure what voltage the relay can take (can start from 3V or 5V). When the relay is not energized, there should be continuity between pins 1 and 4. When the relay is energized, there should be continuity between pins 1 and 3. Connect the negative terminal of the battery to pin 1. This pin brings the power into the relay for powering the external device, which could be the electric motor of a fan or light. If the external device should only be on when the relay is energized, connect a wire from the negative terminal of the external device to pin 3 of the relay. If the external device should be on at all times except when the relay is energized, connect to pin 4 instead of pin 3. Complete the circuit by connecting a wire from the positive terminal of the external device to the positive terminal of the battery.

Integrated Circuits (ICs)

Transistors, diodes, resistors and capacitors are connected together on a tiny 'chip' of silicon (sand is mostly silicon oxide) to give any desired circuit, e.g. a multistage amplifier; an astable, bistable or monostable multivibrator; a counter; a logic gate for a computer; several stages of a TRF (tuned radio frequency) radio.

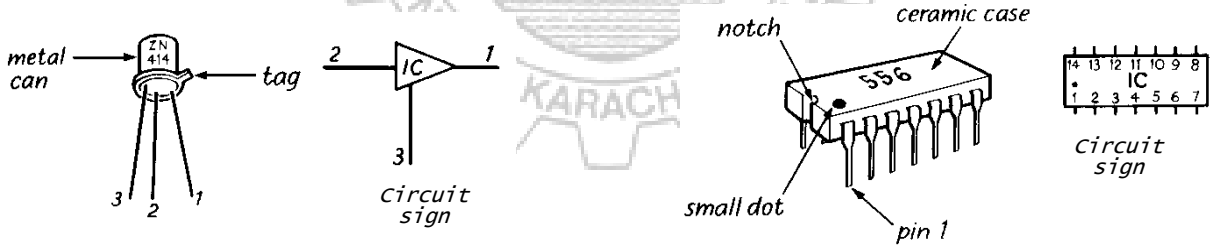


Figure B.19: *Integrated circuits ('chips')* *(b)*
(a) Can type (b) Dual in line (d.i.l) type (14 or 16 pins)

They must be correctly connected. Pin 1 is next to the 'tag' in the can type and on the d.i.l. type it is identified from the 'notch' or 'small dot' on the case. TTL and CMOS are two technologies in which logic gate ICs are available. CMOS 'chips' (standing for *C*omplementary *M*etal *O*xide *S*emiconductors and pronounced 'see-moss') need special care.

CMOS (4000 series) General Characteristics

- Supply: 3 to 15V, small fluctuations are tolerated.
- Inputs have very high impedance (resistance), this is good because it means they will not affect the part of the circuit where they are connected. However, it also means that unconnected inputs can easily pick up electrical noise and rapidly change between high and low states in an unpredictable way. This is likely to make the chip behave erratically and it will significantly increase the supply

current. To prevent problems all unused inputs MUST be connected to the supply (either +Vs or 0V), this applies even if that part of the chip is not being used in the circuit.

- Outputs can sink and source only about 1mA if you wish to maintain the correct output voltage to drive CMOS inputs. If there is no need to drive any inputs the maximum current is about 5mA with a 6V supply, or 10mA with a 9V supply (just enough to light an LED). To switch larger currents you can connect a transistor.
- Fan-out: one output can drive up to 50 inputs.
- Gate propagation time: typically 30ns for a signal to travel through a gate with a 9V supply, it takes a longer time at lower supply voltages.
- Frequency: up to 1MHz, above that the 74 series is a better choice.
- Power consumption (of the chip itself) is very low, a few µW. It is much greater at high frequencies, a few mW at 1MHz for example.
- Damage occurs if static charges build up on input pins when, for example, they touch insulating materials (e.g. clothes, plastic pen) in warm, dry conditions.
    1. Keep the IC in the carrier in which it is supplied until it is inserted in the circuit.
    2. Do not finger the pins or hold them in contact with an insulator.
    3. Connect all unused inputs of the IC to either the positive or the negative of the battery, depending on the circuit.

74LS series TTL Characteristics

- Supply: 5V ±0.25V, it must be very smooth, a regulated supply is best. In addition to the normal supply smoothing, a 0.1µF capacitor should be connected across the supply near the chip to remove the 'spikes' generated as it switches state, one capacitor is needed for every 4 chips.
- Inputs 'float' high to logic 1 if unconnected, but do not rely on this in a permanent (soldered) circuit because the inputs may pick up electrical noise. 1mA must be drawn out to hold inputs at logic 0. In a permanent circuit it is wise to connect any unused inputs to +Vs to ensure good immunity to noise.
- Outputs can sink up to 16mA (enough to light an LED), but they can source only about 2mA. To switch larger currents you can connect a transistor.
- Fan-out: one output can drive up to 10 74LS inputs, but many more 74HCT inputs.
- Gate propagation time: about 10ns for a signal to travel through a gate.
- Frequency: up to about 35MHz (under the right conditions).
- Power consumption (of the chip itself) is a few mW.

Building Circuits

The circuit board shown in figure B.20 accepts ICs as well as separate components. It has 47 rows of 5 interconnected sockets on each side of a central channel across which d.i.l. ICs can be fitted. A wire inserted in a socket in a certain row becomes connected to wires in any of the other 4 sockets in that row by a metal strip under the board. For example, wires in sockets B5, C5, D5, E5 and F5 are all joined. Metal strips under the board connect the sockets.

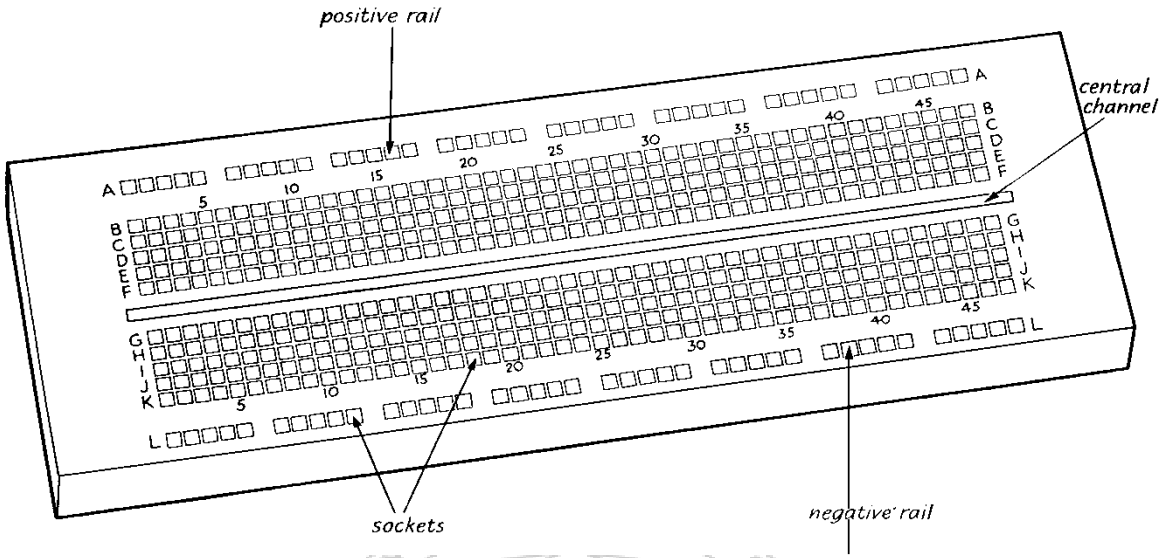


Figure B.20: *Circuit board*

The circuit board shown above accepts ICs as well as separate components. It has 47 rows of 5 interconnected sockets on each side of a central channel across which d.i.l. ICs can be fitted. A wire inserted in a socket in a certain row becomes connected to wires in any of the other 4 sockets in that row by a metal strip under the board. For example, wires in sockets B5, C5, D5, E5 and F5 are all joined. Metal strips under the board connect the sockets. There is a row of 40 interconnected sockets along the top of the board and a similar row along the bottom act as the positive and negative power supply rails (called 'bus bars'). Various makes of circuit board are available, some with vertically mounting removable panels for supporting controls.

To make a connection push about 1 cm of the bare end of a wire (0.25 to 0.85 mm diameter) straight into the socket (not at an angle) so that it is gripped by the metal strip under the board. Do not use wires that are dirty or have kinked ends. Only put one wire in each socket. Bend leads on resistors etc., as shown in figure B.21 before inserting them in the board.

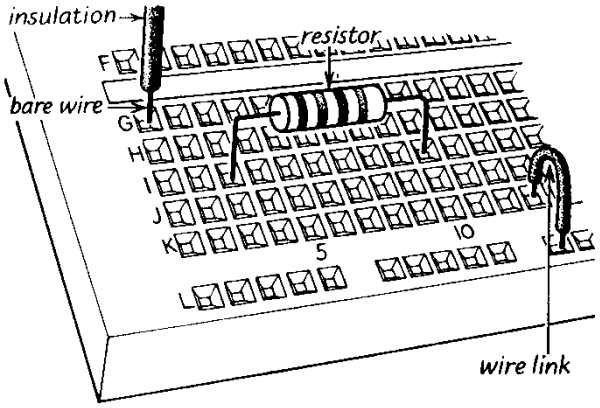


Figure B.21: *Placing various components on the circuit board*

Bare the ends of connecting wire (PVC-covered tinned copper wire 0.6 mm diameter) by removing the insulation (PVC) either with wire strippers or using a pair of blunt-nosed pliers and a pair of side cutters as shown in figure B.22. With practice you should be able to judge just how much the side cutters have to be squeezed and pulled to remove the insulation without cutting the wire.

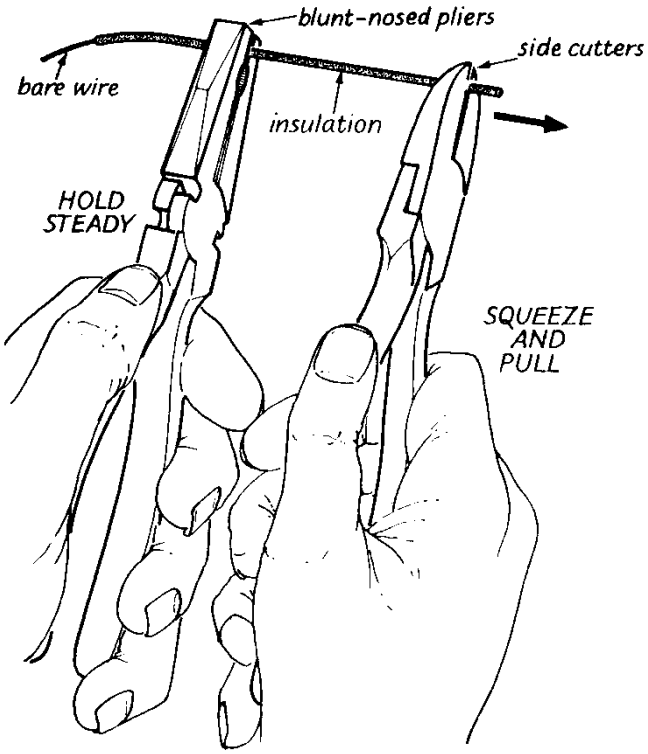


Figure B.22: Stripping wire

Appendix C

C.1 CLOCK GENERATION USING 555 TIMER IC

The 555 timer is a versatile and widely used device which can be used effectively without understanding the function of each pin in detail. It can be configured in two different modes as either a *monostable multivibrator (one-shot)* or as an *astable multivibrator (oscillator).* An astable multivibrator has no stable state and therefore changes back and forth (oscillates) between two unstable states without any external triggering, hence can be used for clock generation of desired frequency and duty cycle. The circuit given in figure C.1 shows the connections of 555 timer in astable mode.

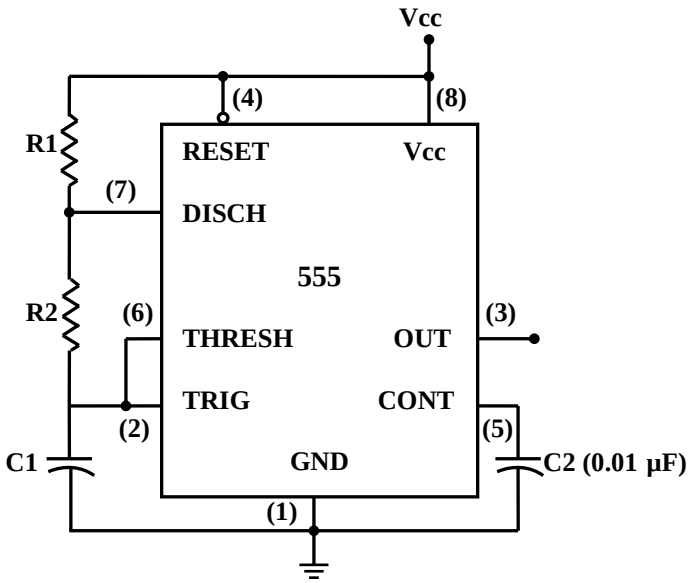


Figure C.1: *Connections of 555 timer IC in astable mode*

The frequency, or repetition rate, of the output pulses is determined by the values of two resistors, R1 and R2 and by the timing capacitor, C1. The design formula for various parameters are as follows:

1. Frequency , $f = \frac{1.44}{(R1 + 2R2) \times C1}$.

2. Period , $t = \frac{1}{f} = 0.69 (R1 + 2R2) \times C1$

3. HIGH time = 0.69 (R1 + R2) x C1

4. LOW time = 0.69 (R2 x C1)

5. Duty Cycle = $\frac{\text{HIGH time}}{\text{period}}$

Before calculating a frequency, it is usual to make R1=1 kΩ because this helps to give the output pulses a duty cycle close to 50%, that is, the HIGH and LOW times of the pulses are approximately equal. By selecting C1 = 10μF and R2 = 140kΩ, output pulses having approximately 1Hz frequency can be generated. Another usual practice is to replace *R*2 with a potentiometer (usually 500kΩ) so that the output frequency can be changed at runtime.

C.2 DEBOUNCING CIRCUITRY FOR MECHANICAL SWITCHES

Mechanical switches form the interface between human beings and computers or other digital systems. For example, a keyboard is a matrix of switches used to supply alphanumeric data to a computer. Whatever the switch design, it is a potential source of problems due to contact bounce. The contact bounce creates a sense of narrow pulses when a switch is opened or closed. An example of device whose operation would be adversely affected by contact bounce is a digital counter used to count the number of times a switch is depressed. Eliminating the effects of contact bounce is called *debouncing.*

When data is entered into a computer via a keyboard, a *software debounce* is often used. This type of debouncing is a program that causes the computer to sample the switch terminal (i.e., to input data from it) many times in succession during the interval of time that contact bounce occurs. If the data is sensed to be 1s (or 0s) for a specific number of consecutive samples, then it is assumed that contact bounce has ended and the last value sensed is valid.

Hardware debouncing is the use of electronic circuitry to eliminate the effects of contact bounce. There are numerous versions of such circuitry, including those that use monostable multivibrators (one-shots), but the most straightforward is simply an RS latch. Figure B.1 shows the circuit. When the switch is in position 1, $R = 0$ and $S = 1$, so the latch is set and the output (Q) is 1. When the switch is in position 0, $R = 1$and $S = 0$, so the latch is reset and the output is 0. When the switch is moved from one position to the other, the latch changes state and bouncing occurs at either the *R* or the *S* input. The bouncing does not affect the latch after it has change state. An RS latch will remain set, for example, when its *R* input is 0 and its *S* input is alternately changed from 1 to 0.

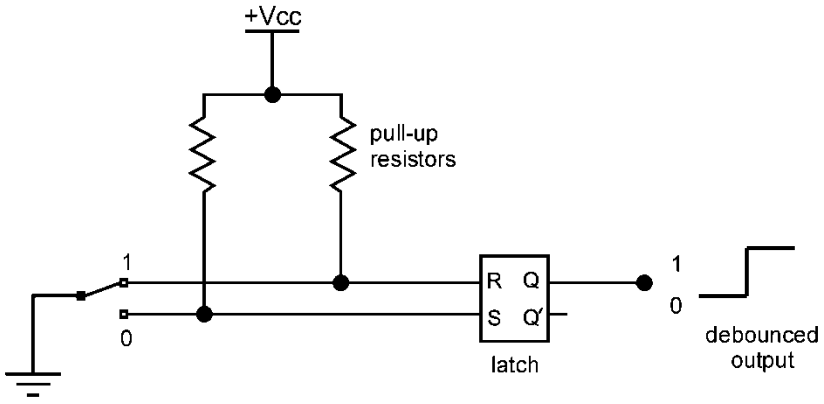


Figure C.2: *Use of RS latch to debounce a mechanical switch*