CDA 3201C
Introduction to Logic Design
Laboratory Manual

Spring
2017

Prepared by
Dr. Bassem A. Alhalabi

Department of Computer & Electrical Engineering & Computer Science
College of Engineering
Florida Atlantic University

FLORIDA ATLANTIC UNIVERSITY
CEECS DEPARTMENT
LOGIC DESIGN AND MICROPROCESSOR LAB

LD/MP Lab Rules and Regulations

1. **Use your student's ID card to admit ONLY yourself and KEEP the door always CLOSED.**
2. **No smoking / drinking / eating is allowed in the lab. Keep the entire lab CLEAN.**
3. **Do not leave your own documents and/or papers in the lab.**
4. **Lab documents may not be checked out.**
5. **Do not INSTALL / COPY / DELETE / MODIFY any software in the lab.**
6. **Keep your lab kit safe and in your possession.**
7. **When your labs are graded, you may keep/recycle your chips and wires (ask TA for recycling).**
8. **Be conservative on wires by using short runs (1", 2", 5", ..). Reuse used wires.**
9. **Teaching Assistants will be available in the lab at scheduled times, which are posted.**
10. **Completed experiments must be checked/graded by the TA's during scheduled lab time.**

Lab Usage Limitation

You must be enrolled in Logic Design or Microprocessor Courses to use this lab. Further, your experiment should be all thought out before taking time in the lab.

Security

You are continuously video taped for your security and against lab vandalism.

Problem Reporting

Any hardware/software failure and discovery of any lost or damaged item in the lab must be reported to the lab attendants or the CEECS technical support staff.

Penalties

Any violation and/or abuse of the Lab regulations and/or facilities may result in disciplinary actions and loss of lab access privileges.

Lab Access

You should aleady have access to the LD/MP lab via your student ID card. In using this access privilege, you indicate you read, understood, and accept all lab regulations.

FLORIDA ATLANTIC UNIVERSITY
CEECS DEPARTMENT
LOGIC DESIGN LAB

Experiments Grading Sheet

1. Students should maintain this sheet along with this whole lab manual for their own records.
2. TAs must sign and date next to every grade, and should indicate if the student cleaned up the workplace after grading each lab.

Lab#	TA Signature	Date	Clean Up	Quartus	Grade
0			(Y) (N)	(Y) (N)	
	Special Remarks:				
1			(Y) (N)	(Y) (N)	
	Special Remarks:				
2			(Y) (N)	(Y) (N)	
	Special Remarks:				
3			(Y) (N)	(Y) (N)	
	Special Remarks:				
4			(Y) (N)	(Y) (N)	
	Special Remarks:				
5			(Y) (N)	(Y) (N)	
	Special Remarks:				
6			(Y) (N)	(Y) (N)	
	Special Remarks:				
Total			(Y) (N)	(Y) (N)	
	Special Remarks:				

Student Name: _____ **Z#:** _____ **Signature:** _____

Basics of Breadboards and Logic Chips

Take the Breadboard out of the case and attach the metal plate onto the bottom of the breadboard; make sure you center it properly. In you have a plastic container, attach the breadboard to the in-side of the cover. Breadboards help build electronic circuits using components and wires without soldering. All connections are made via strips of holes with electric connections underneath. Connection are organized as shown in Figure 1 below. Four long horizontal power strips, two at the top and two at the bottom. All holes along each of the four lines are connected. Some breadboards have the four long horizontal power strips interrupted in the center. All other hole-strips are connected vertically (segments of 5 wholes) with a bridge separating the upper bank form the lower bank.

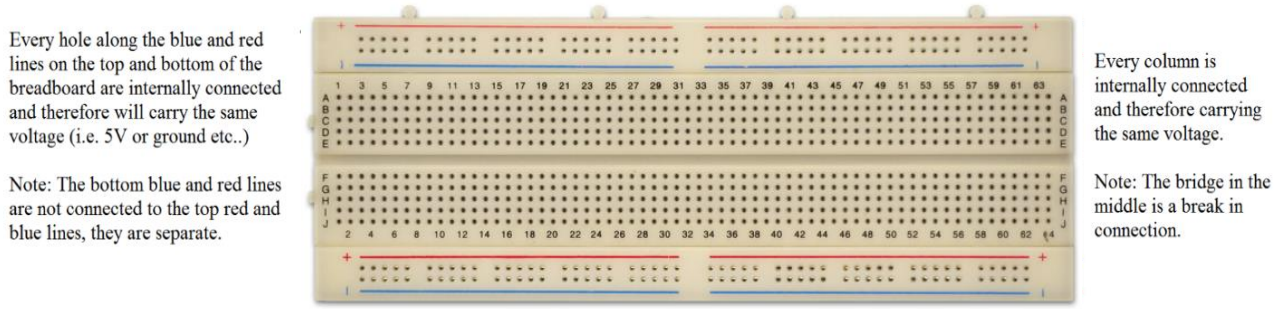


Figure 1: Breadboard

Included in your kit are all components for all lab experiments. The Integrated Circuits (ICs) or simply Chips are actual arrays of logic gates. Use the check list to verify that all parts are in your kit. Each chip is identified by an alphanumeric part number located on the top of it. Figure 2 shows an example of SN74LS00, which is the basic 4x 2-input NAND logic chip within the 7400 logic series. The letters in the part number code indicate the various fabrication technologies. Each chip is marked with a notch or hole on the edge to establish the pin orientation. With the notch on the left, the pins count from the bottom left counterclockwise 1 to 7 and then 8 to 14.

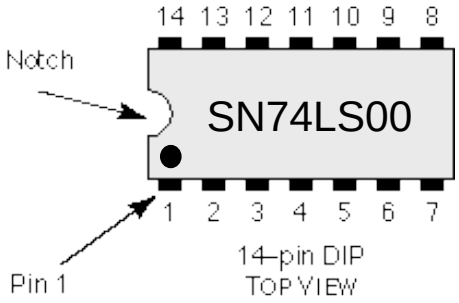


Figure 2: NAND Logic Chip

The schematics or Connection Diagrams of the chips are furnished on the last few pages of this manual. Figure 3 shows the diagram of 7400 (4x 2-input NAND gates). All logic chips must have power connected for the gates to function. Vcc is connected to the +5V of the power source, and GND is connected to Ground of the power source. Note that for all logic chips, Vcc is always the top most left pin and the GND is the bottom most right pin. In a 14-pin chip Vcc is pin 14, and GND is pin 7.

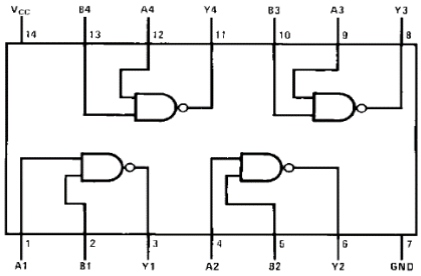
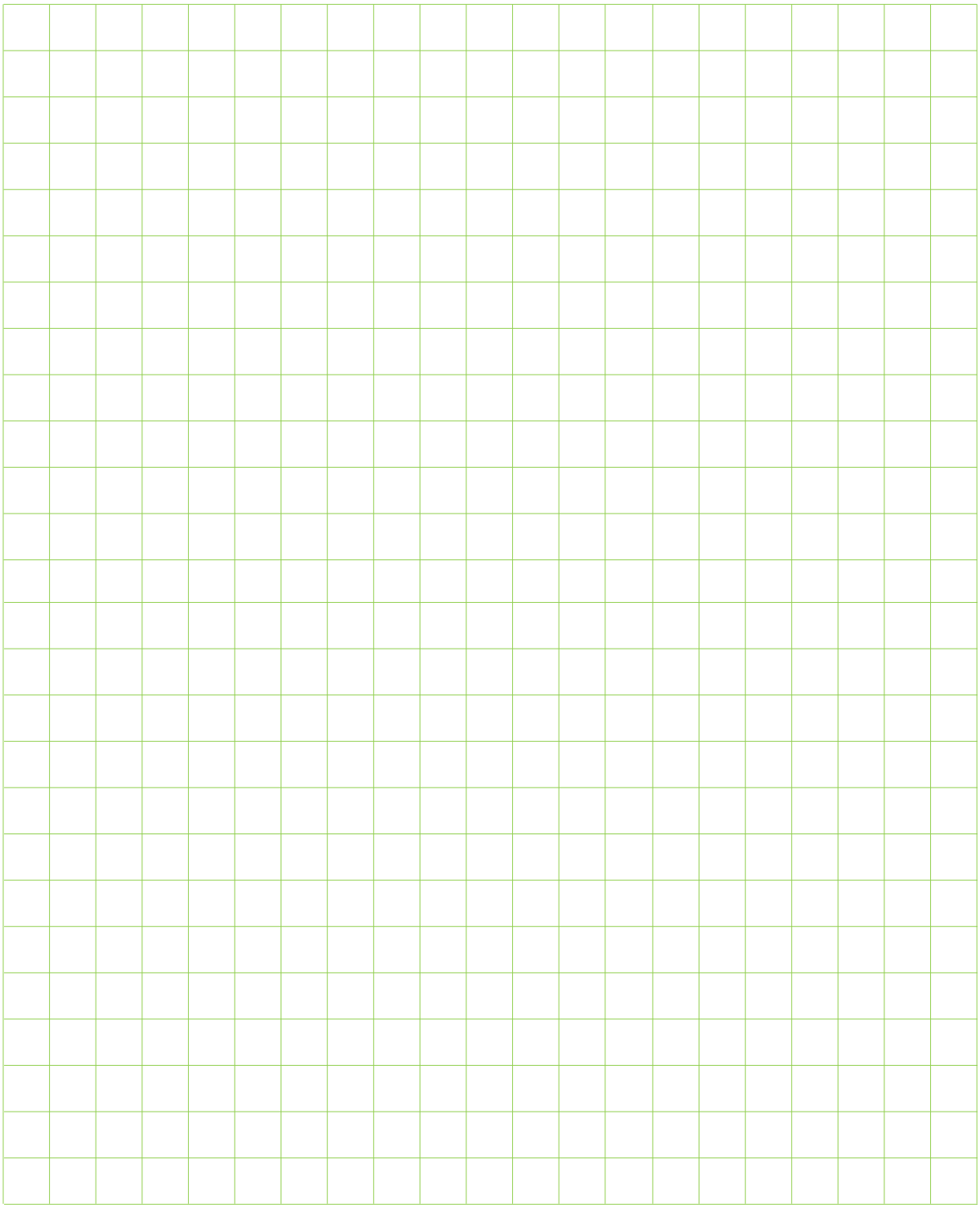


Figure 3: NANA Chip Diagram



Building the Test Platform

In this section, you will build the test platform on one of the breadboards, and keep the second one empty for your logic lab experiments to build one at a time. You will use the same platform for testing all your actual experiments. The test platform includes a number of LEDs as outputs (12 LEDs) and a number of switches as inputs (4 momentary and 12 DIP switches) as depicted in Figure 4. Also as a handy logic-state tester, you need to build the Probe circuit using a LM324 OpAmp chip as shown in Figure 4.

Assemble the entire test platform as shown in Figure 4. Try to imitate the exact layout to make it easier for the lab TAs to assist you. Several video clips are posted on this shared Google Drive which will help you get started with basic instructions.

https://drive.google.com/drive/u/1/folders/0B1D2TdhJGCwmdFc3ZTFESmMwams

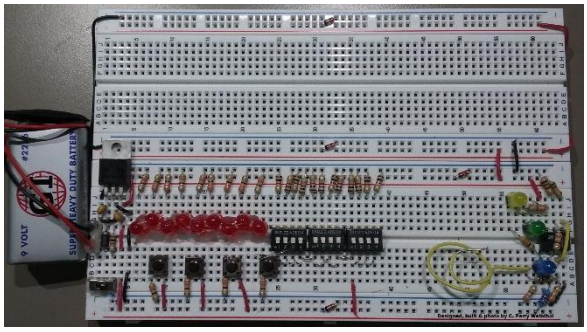
As a registered FAU students you should have access this shared drive. If for some reason you do not have access, then login into bb.fau.edu, select Google Drive, and try again. The schematics of the test platform is furnished next page.

Looking at the lower left side of the breadboard, you find the power supply circuitry. You will use a 9V battery and a 5V regulator. The diode at the regulator input is to protect the regulator against accidentally connecting the battery with reversed polarity. The red LED is a power-on indicator.

For output state indicators, we use 12 red LEDs, each of which is connected to a current limiting resistor, which is connected to GND making the LEDs active high indicators.

For pulse inputs, we use 4 momentary switches with active high connection and pull-down resistor. For continues input states, we use 12-bit DIP switches which are also active-high.

The last circuit on the right side of the bread board is the logic probe circuit. It is built with LM324 OpAmp with LEDs, red indicates false (logic 0) and green indicates true (logic 1), when both are off, if is indication of open circuit, that is neither zero nor 1. The probe input is the long purple wire. To test it, connect it to one of the momentary switches and it should light red, push the switch and probe will light green.



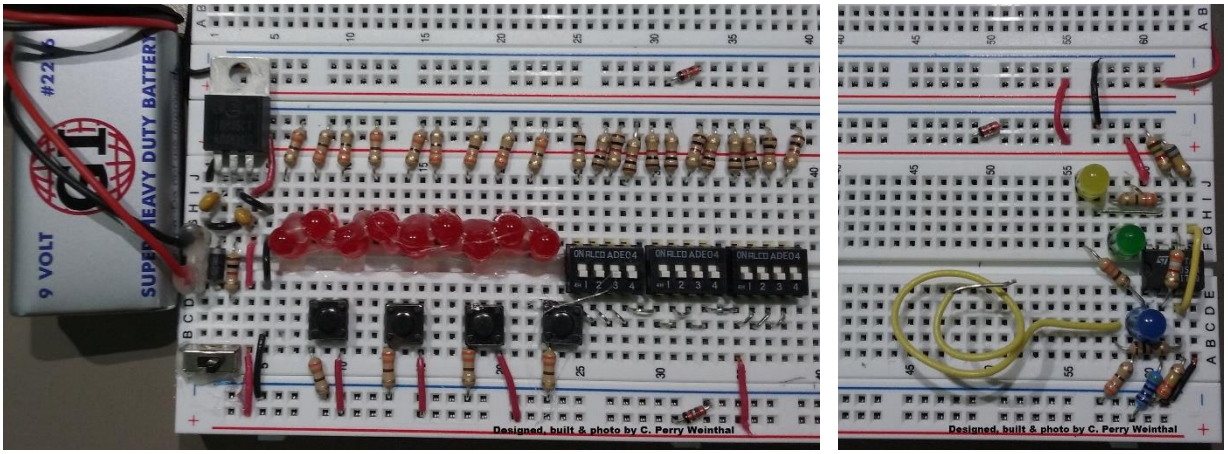
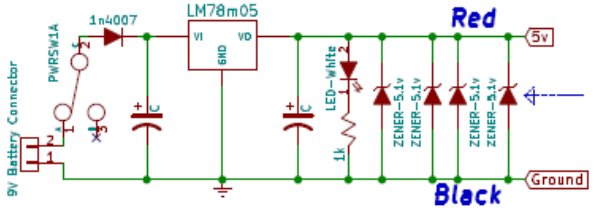


Figure 4: Test Platform with LEDs, Switches, and a Digital Probe

Hint: When unsure please look it up!
Google: "Part_Number" Datasheet ".pdf"

Power Supply

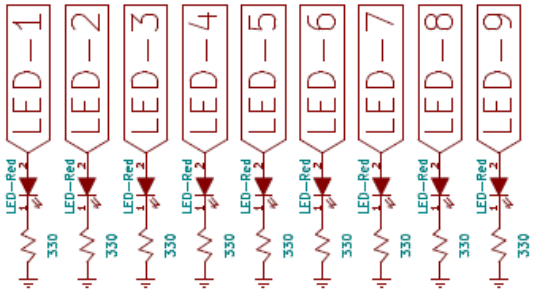


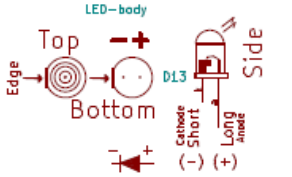
Zener= 1n751a
or 1n5231a
or 1n4733a

OBSERVE:
Zener Diode Black band goes to the 5 Volt Supply.
This is the ONLY Diode that Works like this.

Observe: the Polarity: "PLUS" or "MINUS"
Some Parts have "—" while others have "+"
Polarized Capictors will SMOKE if wired wrong.

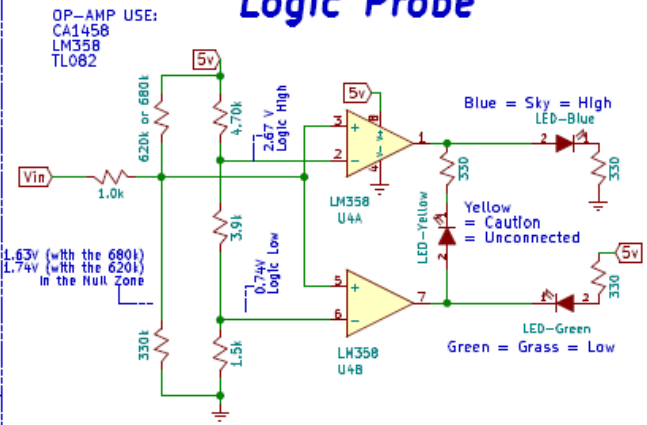
Indcators — Lamps



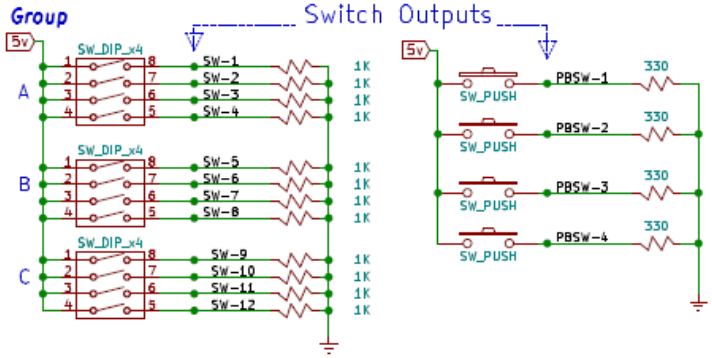


Ohm's Law: Voltage = Current * Resistance
V= I * R
Ohm's Power Law: Power = Voltage * Current
P = V * I == V^2 / R
LEDs Voltage drop at ~20mA, (Typical & ~)
Red = 2.0V, Orange = 2.4V,
Green = 2.7V, Yellow = 2.5V
White = 3.4V, Blue = 4.0V

Logic Probe



Switches — D.I.P. & Push Button



By Perry Weinthal, FAU, Engineering, CEECS Lab Manager

Sheet: /

File: Lab 0 -- Logic Startup -0800.sch

Title: Logic Lab — Bread Board Inital Setup

Size: A	Date:	Rev: 3.0
KiCad E.D.A. eeschema 4.0.2-stable		Id: 1/1

Name: **Grade:** /5

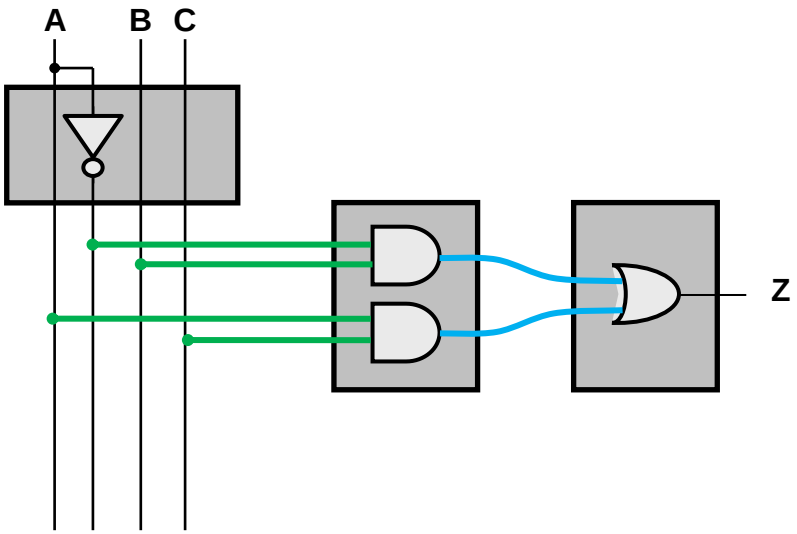
0) [5] This lab experiment is used for the lab orientation where TAs will walk you through practical steps to build the experiment circuit and test it. You will use the following simple function:

$$Z = A^* B + A C$$

0.a) [0] Fill the truth table for the above function

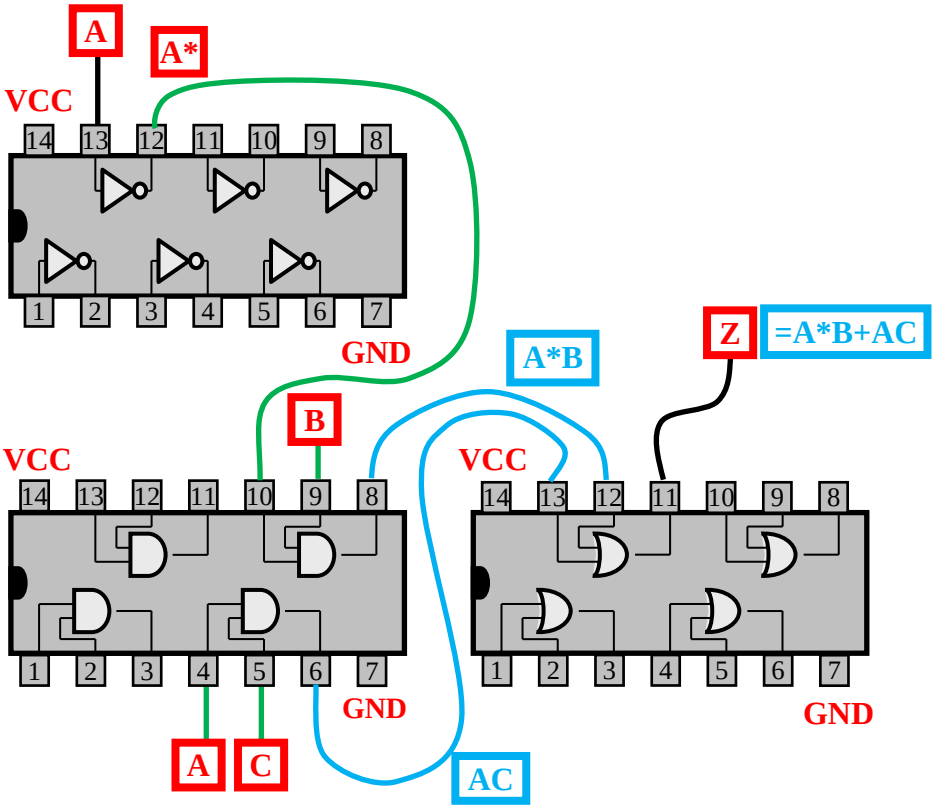
A B C	A*B	A C	Z
000	0	0	0
001	0	0	0
010	1	0	1
011	1	0	1
100	0	0	0
101	0	1	1
110	0	0	0
111	0	1	1

0.b) [0] Draw the switching function (**Z = A* B + A C)** using Inverters, AND gates, and OR gate:



Name: **Grade:** /5

0.c) [0] Using the three logic chips below, draw the wires to make the needed connections between the various gates. Note that not all gates in all chips are used.



0.d) [5] With the wiring diagram above, you are ready to implement the circuit on the breadboard. For inputs ABC, use three DIP switches (most left) with all in off position (ABC= logic 000). For output Z, use one of the LED indicators (most right). If output Z is logic 0, the LED should be off and when Z is logic 1, the LED should be on. Note that all switches and LED indicators are active high.

Next use the 3 DIP switches to try all binary combinations of inputs ABC and observe the output Z if it turns on and off according to the truth table above. If you have a perfect match, it means you wired the circuit correctly, and you completed lab 0.

To check the logic level of any pin on any chip, use the probe wire

Name: **Grade:** /15

1) [15] The following Boolean function is algebraically reduced to two different forms denoted as X1 and X2 as follows. Please verify the simplification process.

$$X = B\,C\,\overline{D} + B\,C\,D + A\,\overline{B}\,\overline{D} + A\,B\,\overline{C}\,\overline{D}$$

$$= BC\,(\overline{D} + D) + (\overline{B} + B\overline{C})\,A\overline{D}$$

$$= BC + (\overline{B} + \overline{C})\,A\overline{D} = \underline{\underline{BC + A\,\overline{B}\,\overline{D} + A\overline{C}\,\overline{D} = X1}}$$

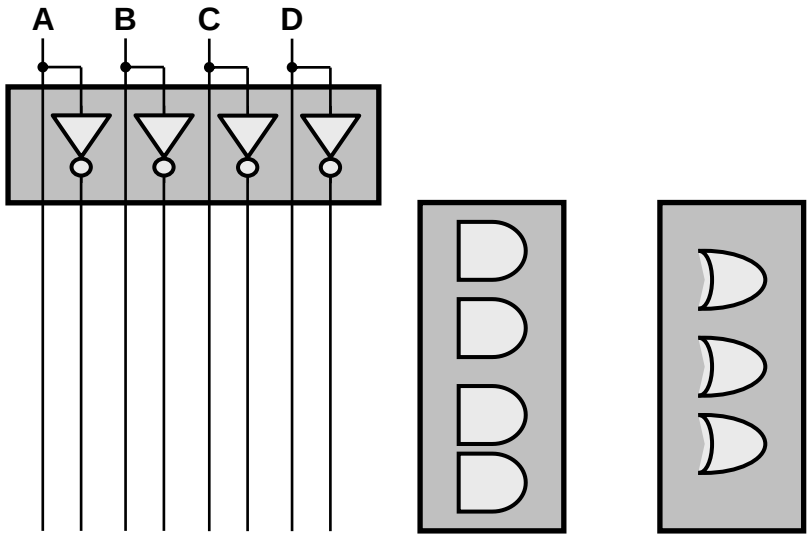
$$\text{or} = BC + \overline{(BC)}\,A\overline{D} = \underline{\underline{BC + A\overline{D} = X2}}$$

1.a) [3] Verify the equivalence of X1 and X2 using the following truth table:

ABCD	BC	$A\,\overline{B}\,\overline{D}$	$A\,\overline{C}\,\overline{D}$	$A\overline{D}$	X1	X2
0000						
0001						
0010						
0011						
0100						
0101						
0110						
0111						
1000						
1001						
1010						
1011						
1100						
1101						
1110						
1111						

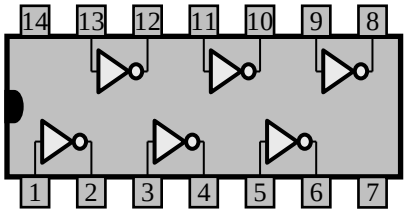
1.b) [4] Implement the above two reduced functions, X1 and X2, using exactly 4 2-input AND gates, 3 2-input OR gates:

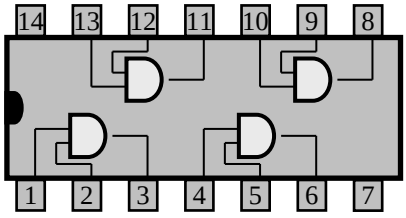
Name: **Grade:** /15

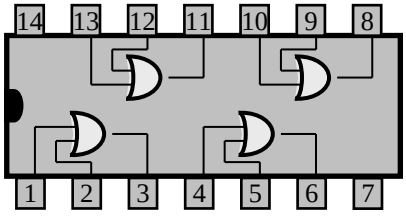


1.c) [6] Build the above 2-output circuit using exactly 1/2 7404, 1 7408 and 3/4 7432 chips on your breadboard and then connect inputs ABCD to 4 DIP switched and the outputs X1 X2 to two LED indicators. Test all the 16 different input combinations and observe the two outputs to be equivalent. Use the following chip pin-out to conveniently plan your wiring.

1.d) [2] You can verify the circuit design/behavior by implementing the circuit using Quartus. You may wish to do this before you actually build the circuit on the breadboard.







Name: **Grade:** /20

2) [20] The following truth table represents a Full Binary Adder. The two binary inputs are A and B and the carry from previous stage is Cin. The sum bit is S and the carry for the next stage is Cout.

A	B	Cin	Sum	Cout
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

2.a) [2] Write the Boolean expression as a sum of Minterms:

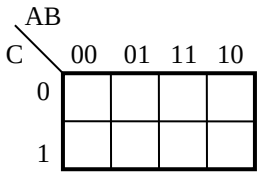
Sum = f (A, B, Cin) = _____

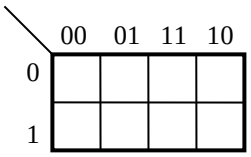
Cout = f (A, B, Cin) = _____

2.b) [2] Optimize the above functions using K-maps:

Sum = f (A, B, Cin) = _____

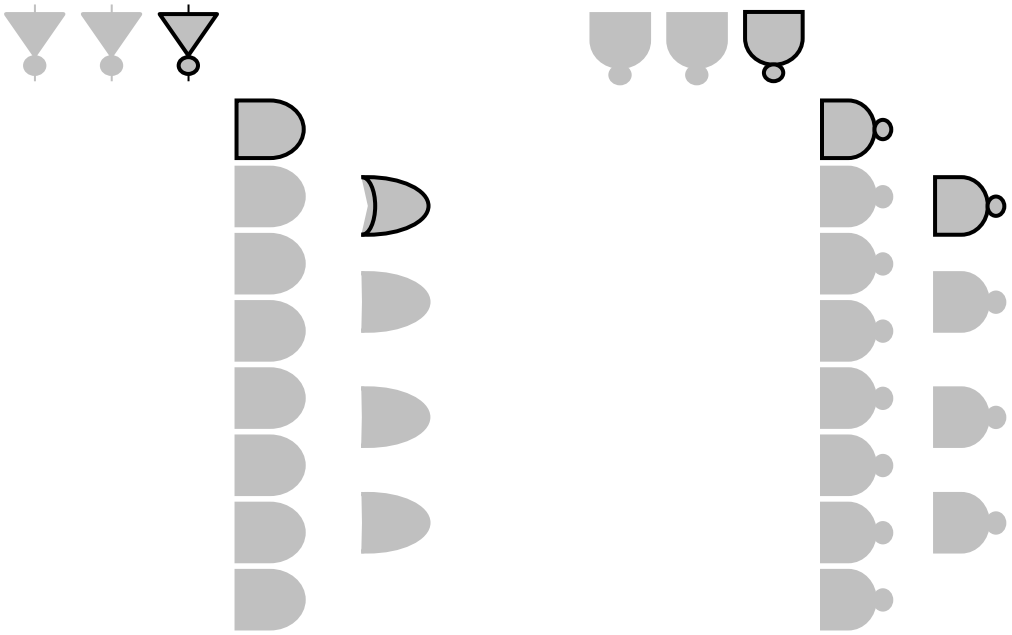
Cout = f (A, B, Cin) = _____





2.c) [2] Implement the above two optimized functions as a 3-input 2-output NOT-AND-OR network:

2.d) [2] Convert the above network to all NAND gates:



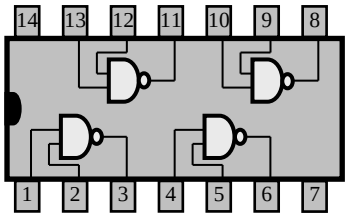
2.e) [4] Build the above all-NAND circuit using the 7400 and 7410 logic chips (maximum 4 chips) on your breadboard and then connect it to your test platform to test it. Use 3 logic switches as inputs and 2 LEDs as outputs. You need to test all 8 different input combinations of the inputs.

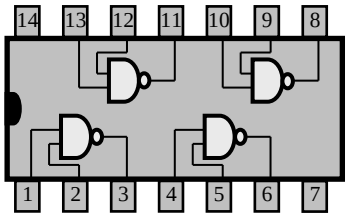
Hint: Leave room on your breadboard for the 16-pin 74283.

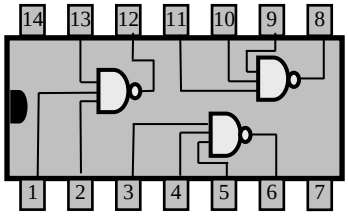
2.f) [6] Connect your all-NAND 1-bit full adder to the standard 4-bit full adder (Chip 74283) making your adder the LSB of the 5-bit resulting adder. Use the following diagram to plan your wiring. To simplify the testing procedure, tie the two MSB bits to low level. Then, use 7 logic switches as inputs and 4 LEDs as outputs. To test your circuit, try as many input combinations as possible.

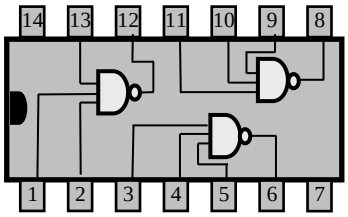
2.g) [2] You can verify the circuit design/behavior (the 5-bit adder) by implementing the circuit using Quartus. You may wish to do this before you actually build the circuit on the breadboard.

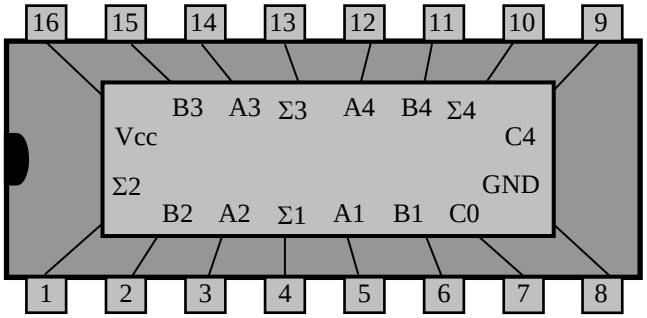
Name: **Grade:** /20

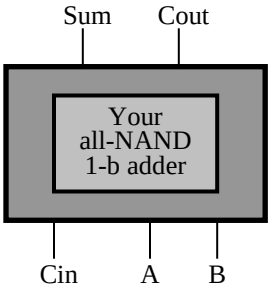












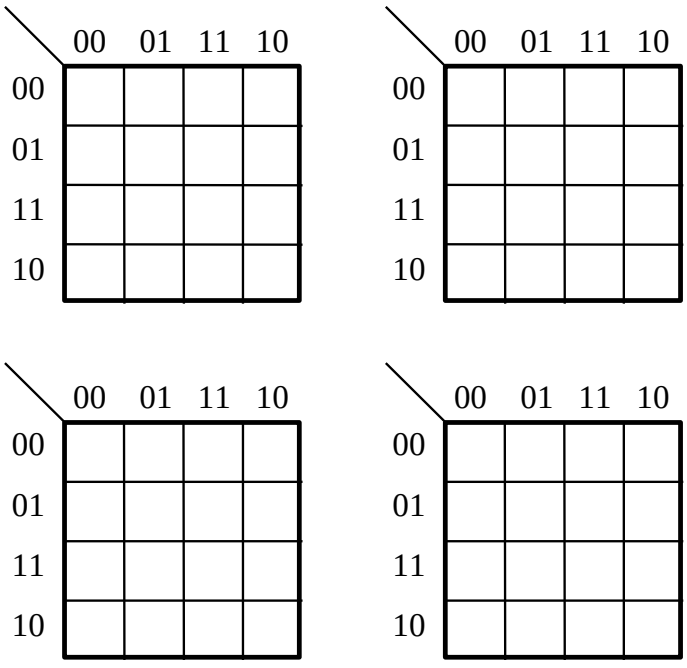
/20

Name: **Grade:**

3) [20] Consider the following two functions:

$$X = \overline{A}BCD + \overline{A}B\overline{C}D + A\overline{B}\overline{C}D + BCD$$
$$Y = A\overline{B}CD + AB\overline{C} + ACD + ABD$$

3.a) [2] Expand the above functions to sum of Minterm using K-maps:



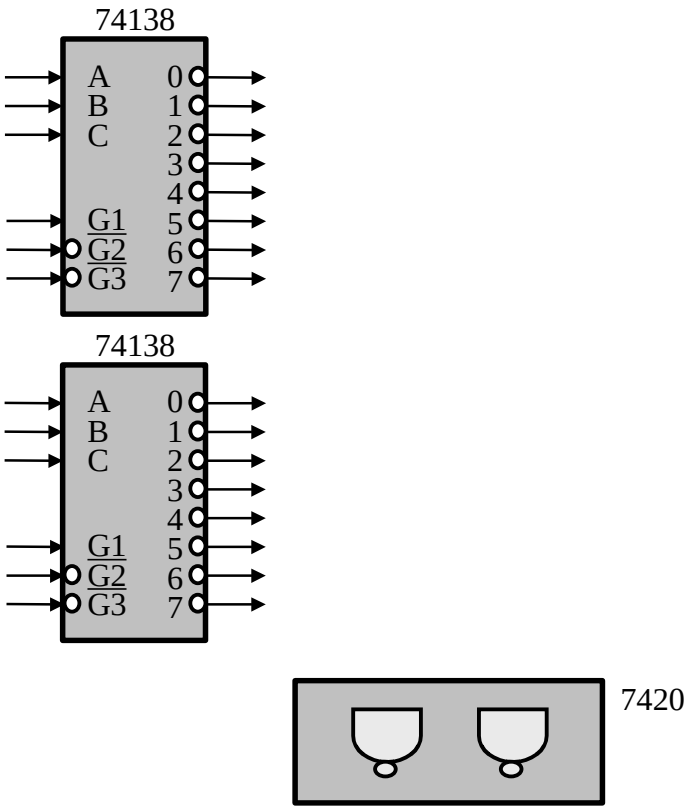
X = f (A, B, C, D) = _____

Y = f (A, B, C, D) = _____

/20

Name: **Grade:**

3.b) [4] Show the input connections necessary to realize the above Boolean expressions using exactly two 3-to-8 Decoders, (2 of 74138 chips) and two 4-input NAND gates (7420 chip). No other gates are allowed.



3.c) [10] Build the above circuit using the specified logic chips on your breadboard and then connect it to the test platform to test it. Use 4 logic switches as inputs and 2 LEDs as outputs. You need to test all the 16 different input combinations of the inputs

3.d) [4] You can verify the circuit design/behavior by implementing the circuit using Quartus. You may wish to do this before you actually build the circuit on the breadboard.

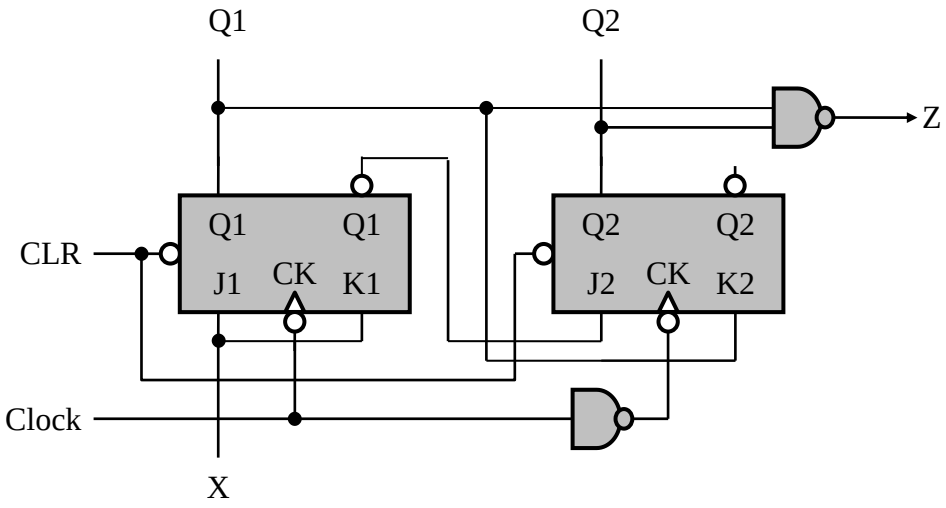
Lab Assignment 3

/20

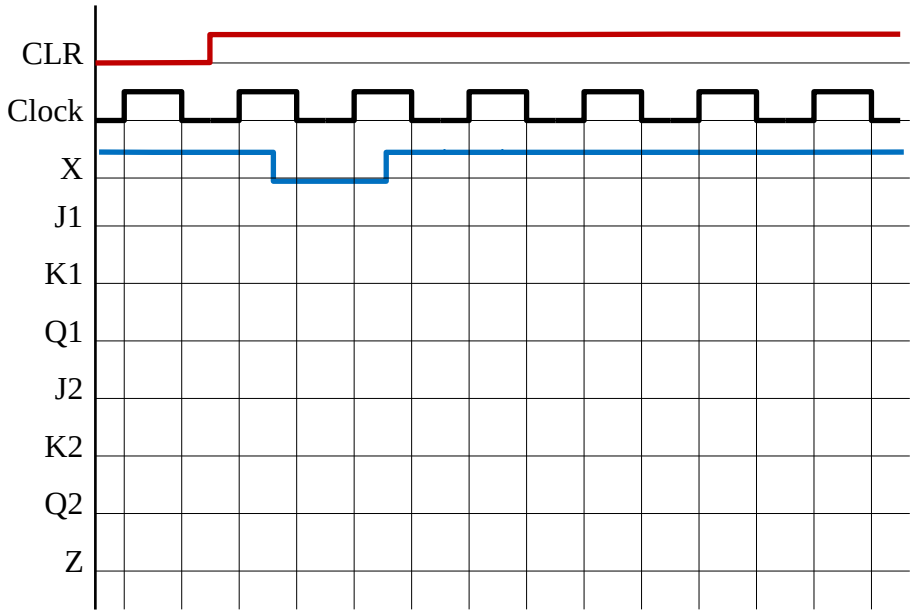
Name: **Grade:**

Name: **Grade:** /20

4) [20] Consider the following sequential circuit with two negative-edge-triggered JK flip-flops.



4.a) [2] Complete the timing diagram for the above circuit.



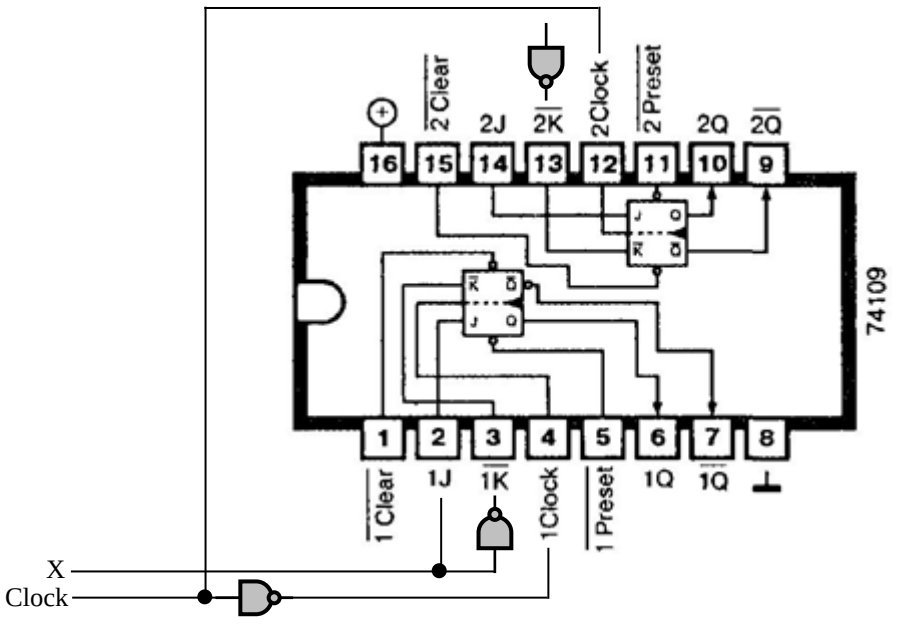
Name: **Grade:** /20

4.b) [8] Build the above circuit using the 74109 dual JK flip-flop logic chip on your breadboard and then connect it to your test platform to test it. Use the chip block diagram below to plan for your wiring. Use 2 logic switches for inputs X and CLR, and one pulse switch for the Clock. Use 4 LEDs to observe X=J1= K1, Q1=J2*=K2, Q2, and Z=Q1*+Q2*. Tie the PRE inputs to high as they are active low.

Observation: If X is kept high, Q1 will toggle at every Negative Clock transition because J1=K1=X=1 (toggle mode).

Warning 1: Note that for 74109 JK package, the K input of the flip-flops is internally inverted so you need to invert it outside the chip by using external inverters. Or simply, in case of 1K, connect it to X*, and in case of 2K, connect it to Q1* (just like 2J).

Warning 2: Note that for 74109 JK package, the Clock input of the flip-flops is actually positive-edge triggered, not negative as we have in the schematics. Therefor for the implementation, use the inverter to invert your Clock input going to 1Clock instead of 2Clock.



Name: **Grade:** **/20**

4.c) [4] You can verify the circuit design/behavior by implementing the circuit using Quartus. You may wish to do this before you actually build the circuit on the breadboard.

4.d) [6] From the analysis of the circuit above, draw the state table and the state graph.

Name: **Grade:** /20

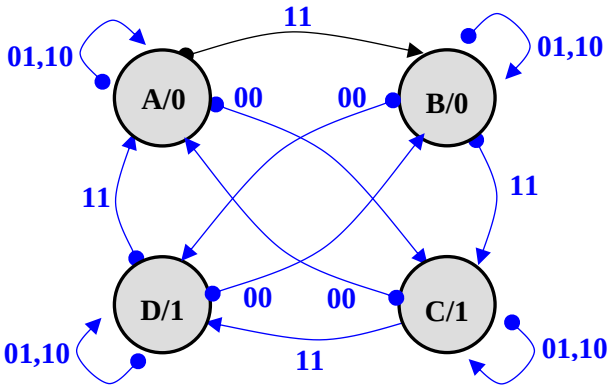
Name: **Grade:** /20

5) [20] Below is a Moore sequential circuit which monitors two inputs X1X2. When the two inputs X1X2 are 00, the output Z toggles at every clock. When the two inputs X1X2 are 11, the output Z toggles at every other clock. When the two inputs X1X2 are different, the output Z holds its state and would not changes until the inputs are equal again. The state diagram of the circuit is given to you.

5.a) [2] Fill in the next state table and the transition table using the indicated binary assignment below.

Present State	Next State X1X2 = 00	01	11	10	Output Z
A					
B					
C					
D					

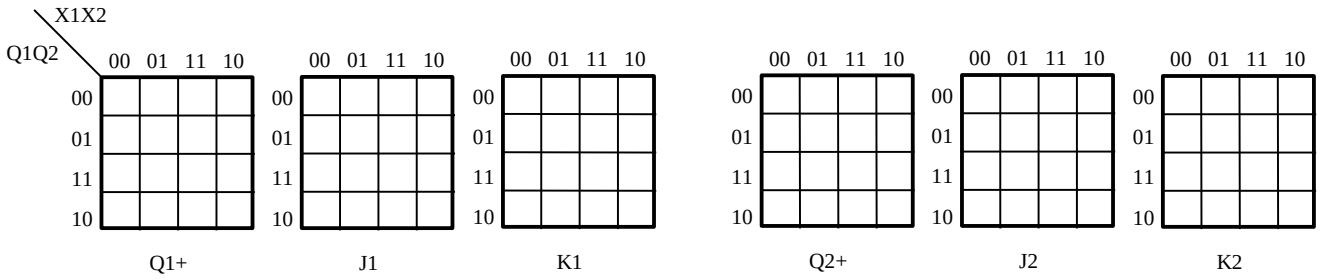
Q1 Q2	Q1+ Q2+ X1X2 = 00	01	11	10	Output Z
00					
01					
11					
10					



5.b) [2] Fill the next-state map for each of the two JK flip-flops below.

5.c) [2] Fill the JK input maps for each of the two flip-flops.

5.d) [2] Write the JK inputs and Z output expressions in a minimized form:



J1= _____ K1= _____

J2= _____ K2= _____

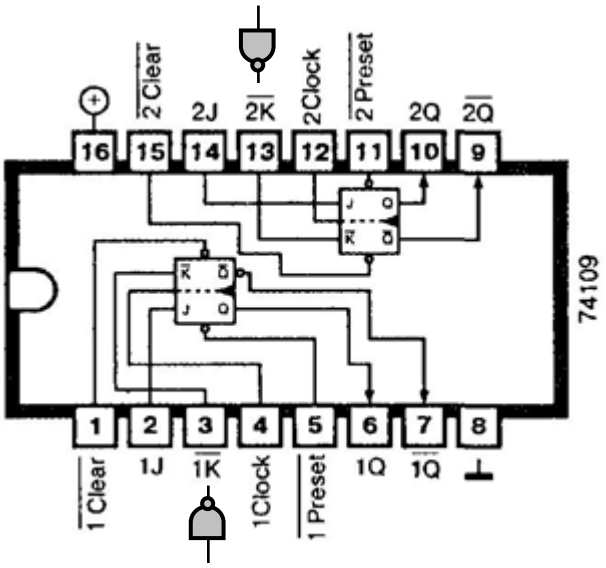
Z= _____

Name: **Grade:** /20

5.e) [8] Build the above circuit using the 74109 dual JK flip-flop and as many 2, 3 or 4 input NAND gates as needed on your breadboard and then connect it to your test platform to test it. Use 2 logic switches for inputs X1 and X2 and 1 pulse switch for the Clock. Use 1 LED to observe the behavior of output Z.

Warning 1: Note that for 74109 JK package, the K input of the flip-flops is internally inverted so you need to invert it outside the chip by using external inverters. Or it might be simpler to implement K1* and K2* and connect them directly to 1K* and 2K* of the chip without external inverters.

5.f) [4] You can verify the circuit design/behavior by implementing the circuit using Quartus. You may wish to do this before you actually build the circuit on the breadboard.

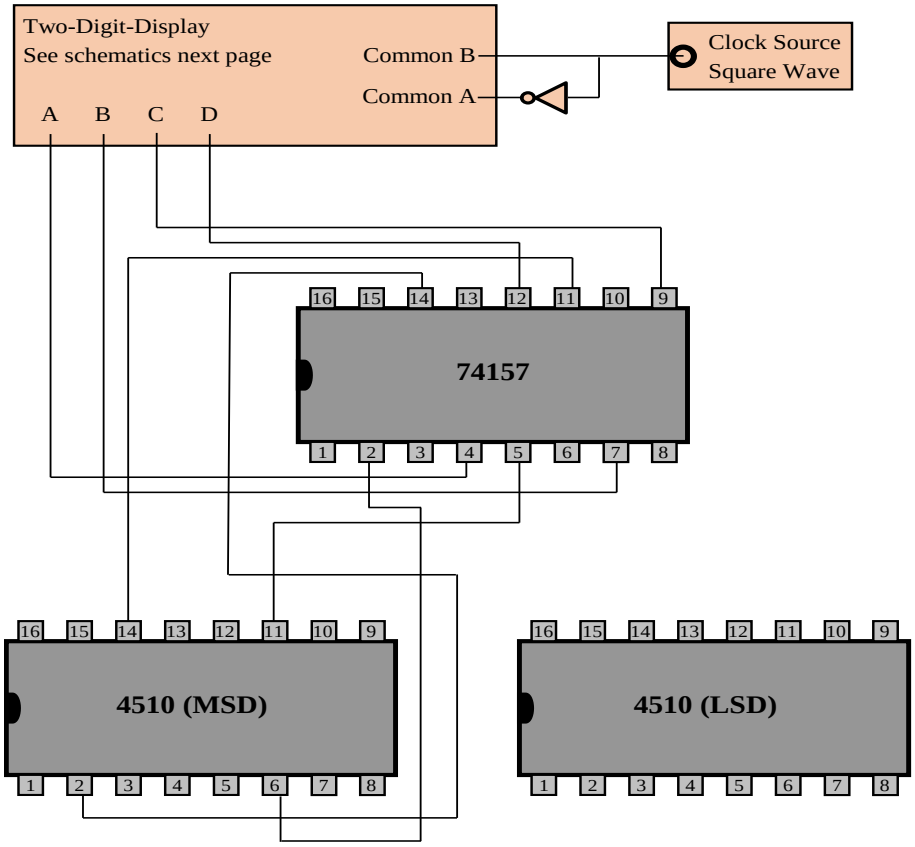


Name: **Grade:** /20

6) [20] Design a 2-digit BCD counter which counts up/down from 00-99 or 99-00 and is capable of parallel loading. The design is partially constructed as shown in the diagram below.

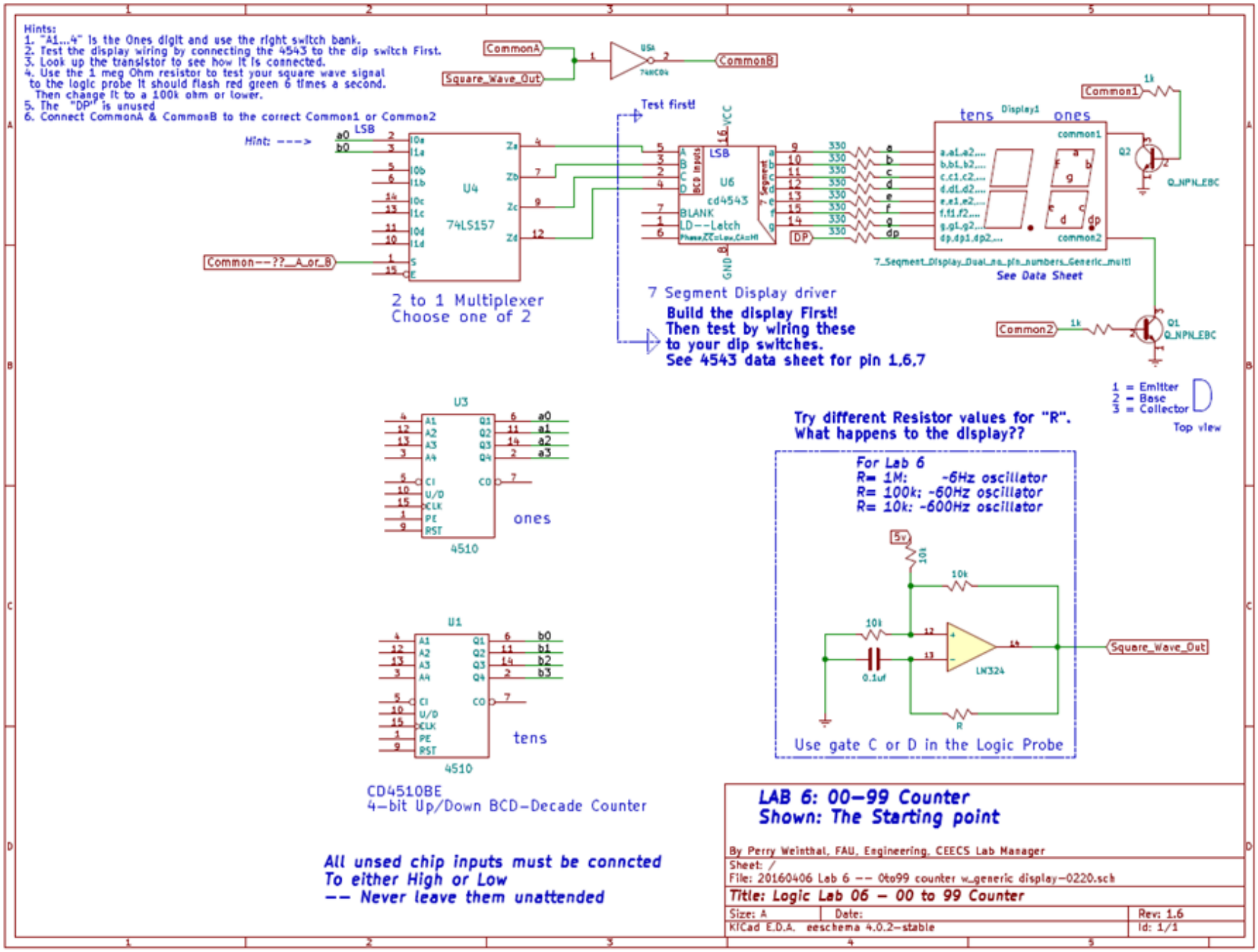
6.1) [8] First, build the Two-Digit-Display block following the schematics next page. Pay attention to use the right NPN/PNP transistors for CA/CC displays. Test this block by connecting the ABCD inputs to 4 DIP switches, and test all combinations form 0000-1001.

6.2) [12] Second, complete the design by adding the rest of the connection on the planning diagram below making the 4510 on the right as the LSD. The select line of the data selector 74157 should be carefully connected to the clock source (with/without inverter) so that the BCD digits (MSD/LSD) correctly show on the 7-segment displays (left/right). Use 8 logic switches (4 for each digit) for parallel loading purposes, 1 more switch for up/down selection and 2 pulse switches: 1 for count pulses and 1 for loading. (Note: Parallel loading is putting the current values of the 8 switches and displaying them simultaneously on the 7-segemnt displays).



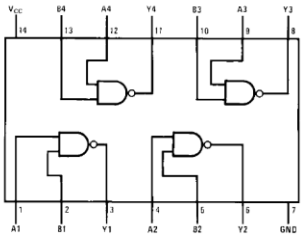
Lab Assignment 6

Name: **Grade: /20**


Hints:
1. "A1...4" is the Ones digit and use the right switch bank.
2. Test the display wiring by connecting the 4543 to the dip switch First.
3. Look up the translstor to see how it is connected.
4. Use the 1 meg Ohm resistor to test your square wave signal
to the logic probe It should flash red green 6 times a second.
Then change it to a 100k ohm or lower.
5. The "DP" is unused
6. Connect CommonA & CommonB to the correct Common1 or Common2
CommonA
Square_Wave_Out
CommonB
Test first!
LSB
Hint: --->
U4
74LS157
Common--??__A_or_B
2 to 1 Multiplexer
Choose one of 2
U6
cd4543
BLANK
LD--Latch
7 Segment Display driver
Build the display First!
Then test by wiring these
to your dip switches.
See 4543 data sheet for pin 1,6,7
DP
tens
ones
Display1
common1
common2
7_Segment_Display_Dual_ns_pin_numbers_Generic_multi
See Data Sheet
Common1
Common2
Q2
Q1
Q_NPN_EBC
1 = Emitter
2 = Base
3 = Collector
Top view
U3
ones
4510
U1
tens
4510
CD4510BE
4-bit Up/Down BCD-Decade Counter
Try different Resistor values for "R".
What happens to the display??
For Lab 6
R= 1M: ~6Hz oscillator
R= 100k: ~60Hz oscillator
R= 10k: ~600Hz oscillator
5v
10k
0.1uf
LM324
R
Square_Wave_Out
Use gate C or D in the Logic Probe
All unsed chip inputs must be conncted
To either High or Low
-- Never leave them unattended
LAB 6: 00-99 Counter
Shown: The Starting point
By Perry Weinthal, FAU, Engineering, CEECS Lab Manager
Sheet: /
File: 20160406 Lab 6 -- 0to99 counter w_generic display-0220.sch
Title: Logic Lab 06 - 00 to 99 Counter
Size: A
Date:
Rev: 1.6
KiCad E.D.A. eeschema 4.0.2-stable
Id: 1/1

74LS00 – Quad 2-input NAND

Connection Diagram



Function Table

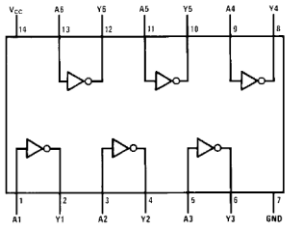
$Y = \overline{AB}$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

74LS04 – Hex Inverting Gates

Connection Diagram



Function Table

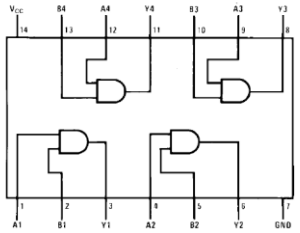
$Y = \overline{A}$

Input	Output
A	Y
L	H
H	L

H = HIGH Logic Level
L = LOW Logic Level

74LS08 – Quad 2-input AND

Connection Diagram



Function Table

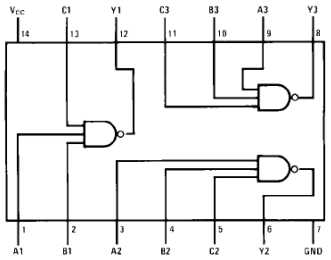
$Y = AB$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = HIGH Logic Level
L = LOW Logic Level

74LS10 – Triple 3-input NAND

Connection Diagram



Function Table

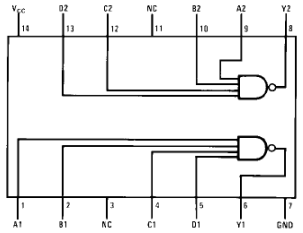
$Y = \overline{ABC}$

Inputs			Output
A	B	C	Y
X	X	L	H
X	L	X	H
L	X	X	H
H	H	H	L

H = HIGH Logic Level
L = LOW Logic Level
X = Either LOW or HIGH Logic Level

74LS20 – Dual 4-input NAND

Connection Diagram



Function Table

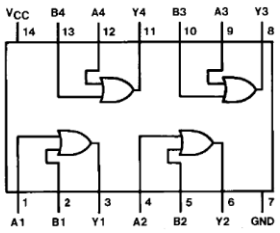
$Y = \overline{ABCD}$

Inputs				Output
A	B	C	D	Y
X	X	X	L	H
X	X	L	X	H
X	L	X	X	H
L	X	X	X	H
H	H	H	H	L

H = HIGH Logic Level
L = LOW Logic Level
X = Either LOW or HIGH Logic Level

74LS32 – Quad 2-input OR

Connection Diagram



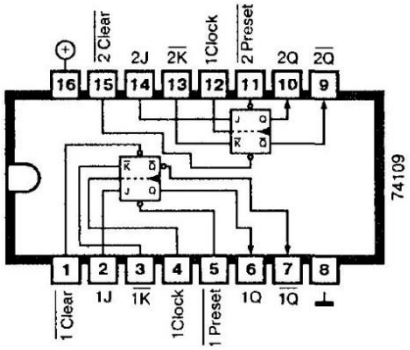
Function Table

$Y = A + B$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

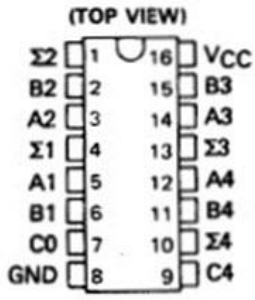
H = HIGH Logic Level
L = LOW Logic Level

74LS109 – Dual JK* Flip-Flop



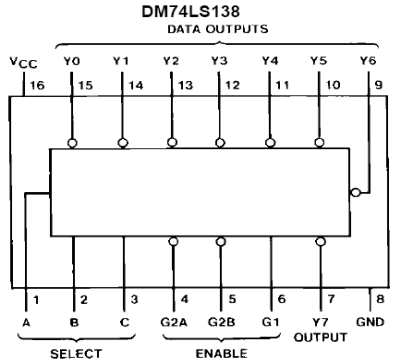
74283 – 4-BIT BINARY FULL ADDER

Connection Diagram

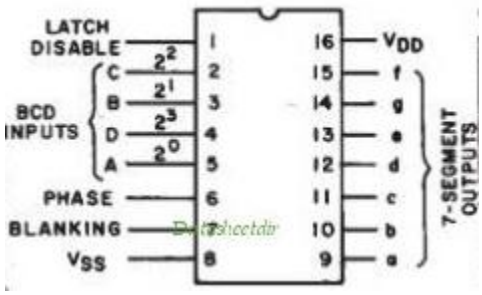


74LS138 – DECODER

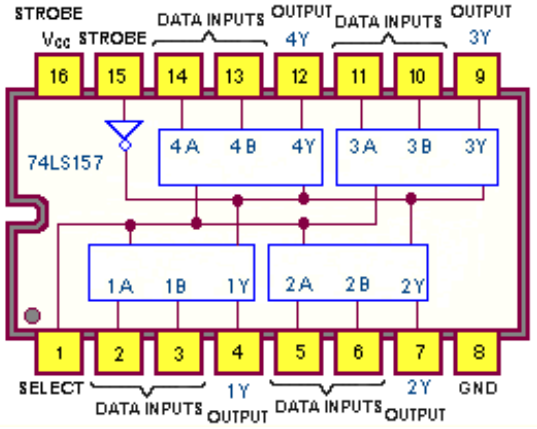
Connection Diagrams



CD4543 – BCD TO 7-SEG DEC

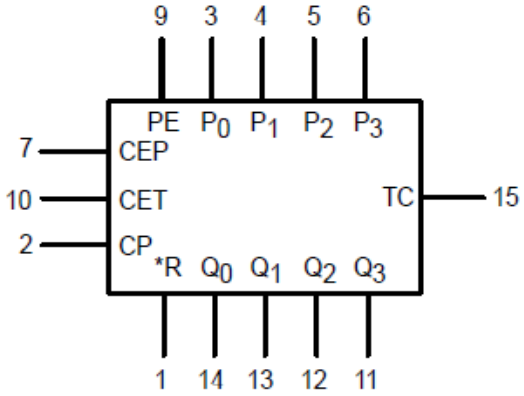


74LS157 – QUAD 2-INPUT MULTIPLEXOR



74LS160 – 4-BIT BINARY COUNTER

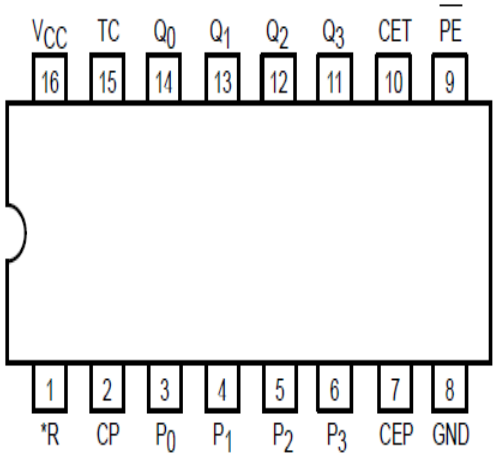
LOGIC SYMBOL



VCC = PIN 16
GND = PIN 8

*MR for LS160A and LS161A
*SR for LS162A and LS163A

CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

*MR for LS160A and LS161A
*SR for LS162A and LS163A