CPEN 230L

INTRODUCTION TO DIGITAL LOGIC

LAB EXPERIMENTS

C. Tavora
January 2006

CPEN 230L

Contents

Lab Procedures

Attendance

Attendance is mandatory. Each student must complete all labs in order to receive a passing grade. If you miss a lab without prior notification of the instructor, you will be required to make up the lab subject to a 50% penalty in grade. No penalty will be imposed if an arrangement is made with the instructor prior to the lab.

Lab Preparation

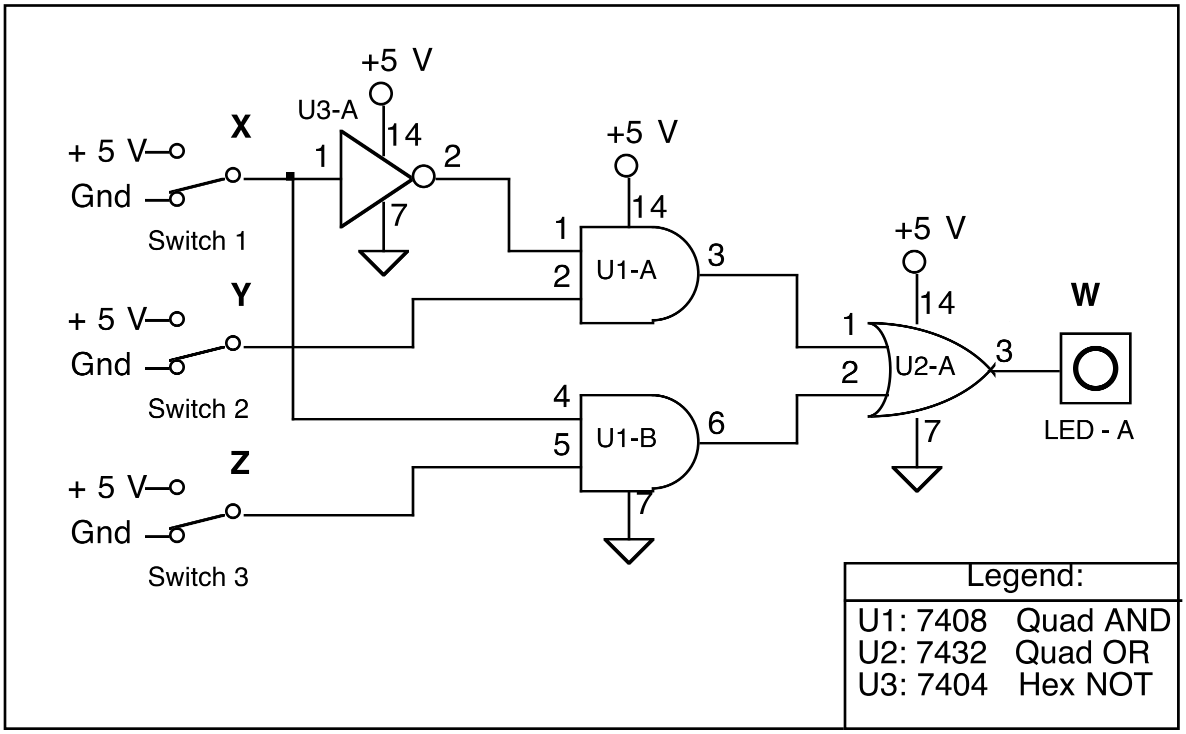
Most labs require the preparation of a prelab which may involve a significant amount of work. The prelab must be completed prior to arrival at the lab. The prelab work will be graded as you enter the lab and is part of your lab grade.

Reports

Lab reports are due at the end of each lab. Each report must be written neatly and must be organized in a way that is easy to read. Each report must include the following information:

Lab #
Lab Title and Objective
Each procedure completely written out
Circuit diagrams
Truth table for combinational logic
Timing diagrams (if applicable)

Circuit diagrams must include IC# and gate# as shown in the diagram below. All pins must be labeled. Power and ground connections must be drawn. A legend listing all ICs (v.g., U1 7400 Quad NAND) must be a part of each circuit diagram.



LAB # 1

Logic Trainer

Purpose: To become familiar with the features and devices on the Logic Trainer. Learn how to document and wire a logic diagram. Learn how to use a logic probe.

1. Lab preparation

   Verify that the logic expressions you plan to use are correct
   Prepare logic diagram that implements the logic expressions
   Check the diagram to have all features shown on the attached diagram
   Develop a test procedure for the logic
   Prepare a table with expected results and space for measured results

2. Logic Trainer - Learn how to use the following features of the Logic Trainer:

   Power Switch
   Power Supply terminals: +5 V (Vcc) & ground (do not use + 15 V and - 15 V)
   Logic indicators
   Data and Debounced switches
   Clock: waveform and frequency selection
   Breadboard connections: buses and tie points

3. IC Testing Station

   How to read a chip part number
   Identification of pin 1 of a chip
   How to place a chip on the tester socket
   How to test and interpret the result

4. How to wire a logic diagram

   Containers with different sizes of wire
   How to place wires on the logic trainer (check that power is off)
   Verify the connections against the logic diagram

5. Test the logic

   Turn power on
   Perform tests and document results
   Analyze results and correct problems by debugging the logic

6. When to use a Logic Probe

   Logic values and corresponding voltage ranges
   What can the LED on the trainer detect
   Levels detected by the Logic Probe

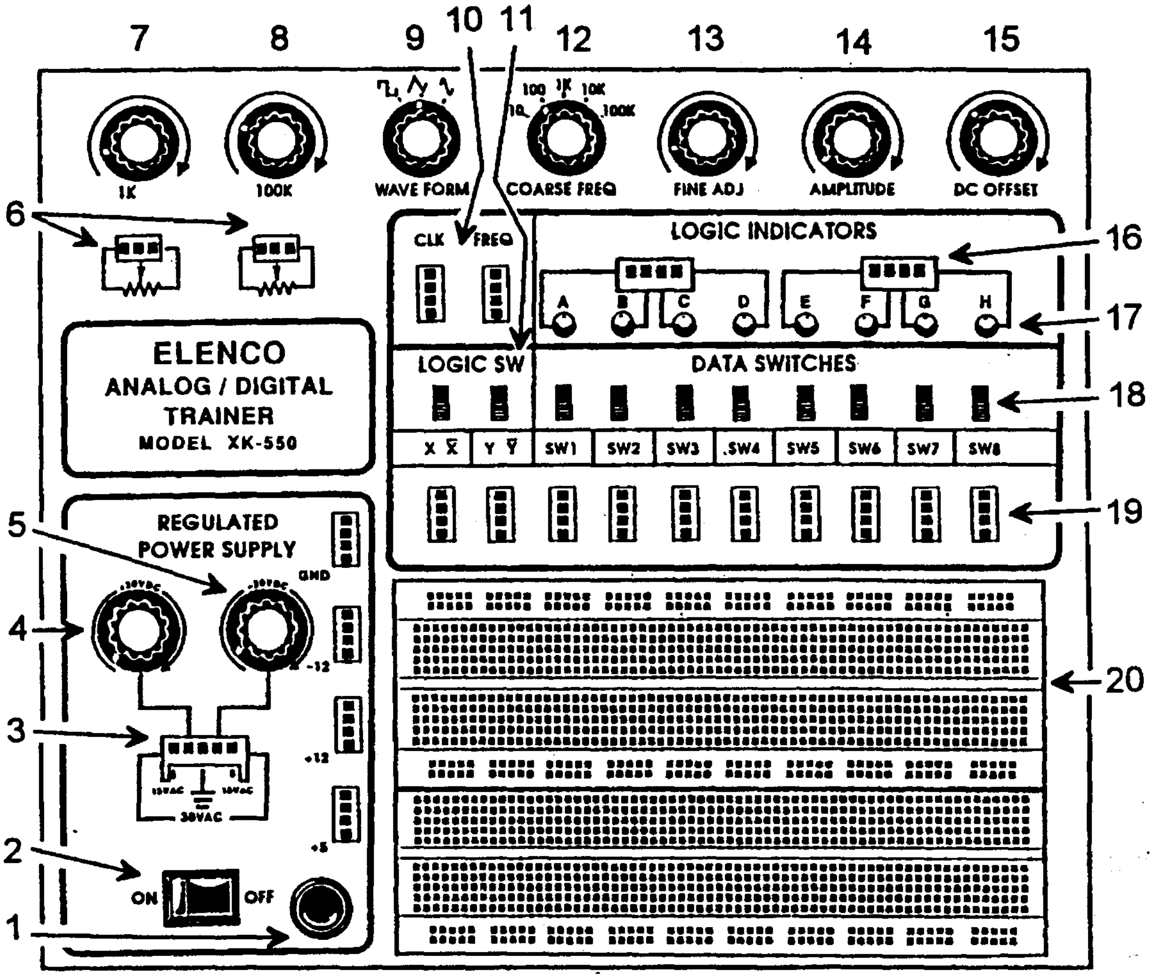
7. Clean up

   Turn off power to all devices
   Remove wires and components from Digital Trainer
   Return components to appropriate storage bins
   Clean up the work bench

8. Logic Trainer Practice

Test the IC chip selected by the instructor for the practice
Make the connections shown by your instructor on the digital trainer
Test the operation of the logic

USERS DESCRIPTION OF FRONT PANEL CONTROL



1) **Fuse holder** - Easy access for replacement of 1.25A 250V fuse.
2) **On-Off switch** - Allows power to be applied to all outputs. Switch will light when on.
3) **Power output terminals** - This provides 30VAC center tapped at 15VAC - also provides output terminal for positive and negative variable voltages.
4) **Variable positive voltage control** - Varies positive voltage from 1.25 to 20V at indicated output connector pin.
5) **Variable negative voltage control** - Varies negative voltage from -1.25 to -20V at indicated output connector pin.
6) **Output terminals** for 1KΩ and 100KΩ undedicated potentiometers.
7) 1KΩ undedicated potentiometer.
8) 100KΩ undedicated potentiometer.
9) **Waveform selection control**, square, triangle or sine generator waveforms.
10) **Output terminals** for all functions as stated, 4 pins per block.
11) **Two logic switches** - These are no bounce logic switches. Give one signal state change per movement of switch.
12) Selects five ranges of frequencies from 10 to 100,000 hertz.
13) **Fine frequency control** - alows easy selection of desired function generator frequency.
14) **Amplitude control** - Controls the function generation output amplitude , 0-15Vpp.
15) **DC offset control** - controls the DC level of the generator output. DC may be varied ±10 volts from zero level.
16) **Input points for logic indicator LEDs.** "A" input corresponds with A lamp, etc.
17) **Logic indicators LEDs**, total eight.
18) **Eight data switches** - Lets output of 5V or 0V depending on position.
19) Output terminal for all functions as stated, 4 pins per block.
20) **Two breadboards** containing a total of 1,660 tie points including 6 independent bus lines.

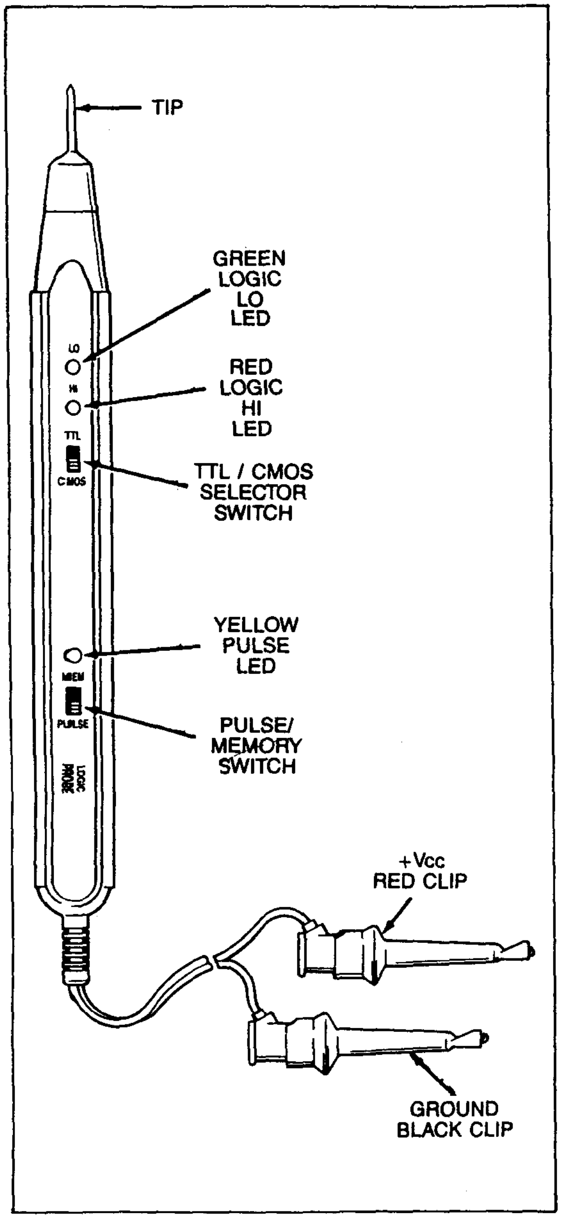


Figure 1.

OPERATION

A. Applying power to the probe.
   1. Connect the black clip to GND (−).
   2. Connect the red clip to the Vcc (+). Be sure Vcc is less than 20V.

B. Application.
   1. TTL/CMOS switch
      The TTL/CMOS switch can be switched to TTL mode for use in TTL circuits. The TTL logic 1 threshold is 2.3V±0.2V, logic 0 threshold is 0.8V±0.2V. When switched to CMOS mode, the CMOS logic 1 threshold is 70% Vcc, logic 0 threshold is 30% Vcc.

   2. Pulse/Memory switch.
      The logic probe can detect and memorize the level transition. Either positive or negative level transition can be detected or memorized, depending on the mode selected.

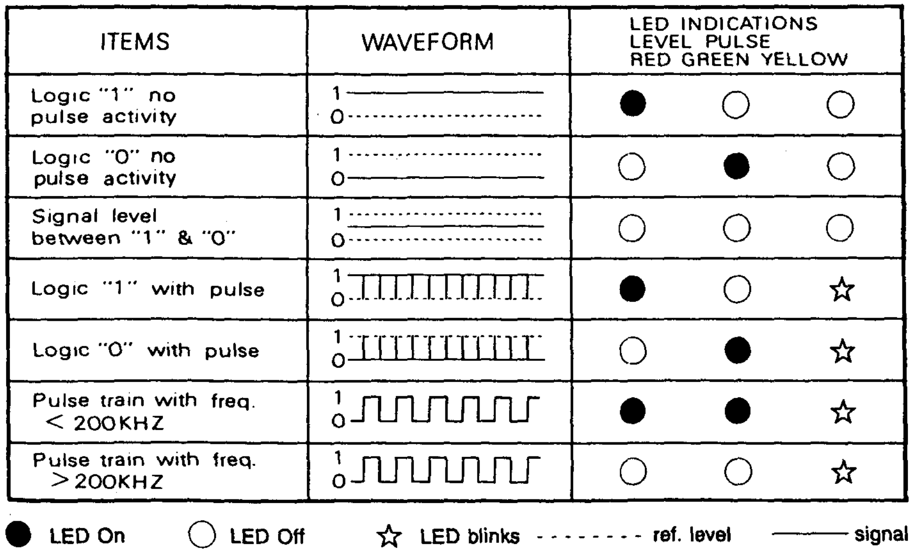
      a) Pulse position:
         The memory function is inoperative in this position. Input state transition is "0" to "1" or "1" to "0" will activate the pulse indicator (flicker for 500ms).

      b) Memory position:
         In this position, memory function is activated. Pulse indicator lights until reset if any pulse or level transition occurs.

If the yellow LED lights when switching from PULSE to MEM modes with the probe tip open circuited, return the switch to the PULSE position and touch the probe tip to the point of measurement before switching to the MEM mode.

   3. See Figure 2 below for some typical combinations of indications from the DP-21 Logic Probe.

TYPICAL SIGNALS AND CORRESPONDING LED INDICATION:



ITEMS	WAVEFORM	LED INDICATIONS LEVEL RED	 PULSE GREEN	 YELLOW
Logic "1" no pulse activity	1 0	●	○	○
Logic "0" no pulse activity	1 0	○	●	○
Signal level between "1" & "0"	1 0	○	○	○
Logic "1" with pulse	1 0	●	○	☆
Logic "0" with pulse	1 0	○	●	☆
Pulse train with freq. < 200KHZ	1 0	●	●	☆
Pulse train with freq. >200KHZ	1 0	○	○	☆

● LED On ○ LED Off ☆ LED blinks - - - - - - - - ref. level ——— signal

Figure 2.

LAB # 2

Combinational Logic with Basic Gates

Purpose: Evaluate the operation of basic logic gates. Use switches, LED lights and power supply on the Logic Trainer to test logic gates. Wire and test the logic implementation of a function.

1. Logic operation of gates

In preparation for the lab, for each of the chips listed below prepare the following:

a. a logic diagram showing how switches and lights are connected to each chip. Show pin numbers, switch and light identifiers, and power supply connections.
b. prepare a truth table for testing one gate on each chip (list the expected result and provide space for each measured result).
c. develop a test procedure that can be used to determine the logic level associated with an open input (i.e., not connected to Vcc or to ground).

Chips to be tested

7400	Quadruple 2-input NAND gates
7402	Quadruple 2-input NOR gates
7404	Hex inverters
7408	Quadruple 2-input AND gates
7432	Quadruple 2-input OR gates
7486	Quadruple 2-input XOR gates

During the lab, do the following:

a. wire each circuit in accordance with the logic diagram developed for the prelab.
b. verify that all circuit connections are correctly implemented.
c. power up the logic and measure the data required for each table.
d. establish the logic level associated with an open input to each gate type.

2. Logic Implementation of a Boolean expression.

In preparation for the lab, the Boolean expression below provides the logic required to turn on or off light L with any one of three switches A, B or C. Changing the status of any one of the switches causes the light L to change state.

$$L = A\, B'\, C' + A'\, B\, C' + A'\, B'\, C + A\, B\, C$$

a. provide a logic diagram showing how switches, lights and logic gates should be connected in order to implement the expression above. Show gates (with pin numbers), switches, light and power supply connections.
b. prepare a truth table that shows the expected value of L for each state of the input switches. Provide space for entering the measured value of L for each input condition.

3. Discussion:

1. In which ways do chip manufacturers identify the location of pin 1 of a chip.
2. Is the logic level associated with an open input the same for all gates?
3. Give a intuitive explanation of why the Boolean expression given for L makes sense.

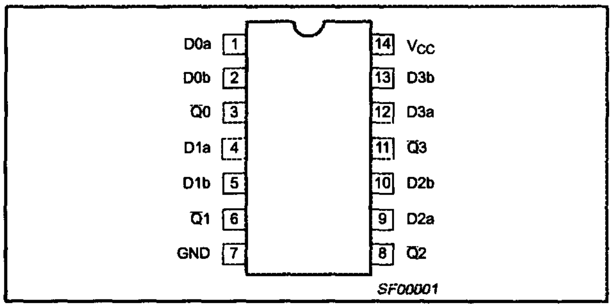
Quad 2-input NAND gate 74F00

FEATURE

- Industrial temperature range available (–40°C to +85°C)

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F00	3.4ns	4.4mA

PIN CONFIGURATION



ORDERING INFORMATION

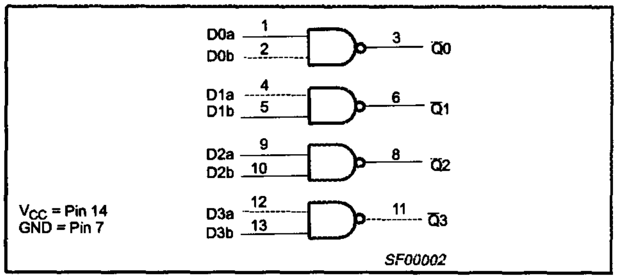
DESCRIPTION	ORDER CODE		PKG DWG #
	COMMERCIAL RANGE V_{CC} = 5V ±10%, T_{amb} = 0°C to +70°C	INDUSTRIAL RANGE V_{CC} = 5V ±10%, T_{amb} = –40°C to +85°C	
14-pin plastic DIP	N74F00N	I74F00N	SOT27-1
14-pin plastic SO	N74F00D	I74F00D	SOT108-1

INPUT AND OUTPUT LOADING AND FAN OUT TABLE

PINS	DESCRIPTION	74F (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
Dna, Dnb	Data inputs	1.0/1.0	20µA/0.6mA
$\overline{Q}n$	Data output	50/33	1.0mA/20mA

NOTE: One (1.0) FAST unit load is defined as: 20µA in the high state and 0.6mA in the low state.

LOGIC DIAGRAM

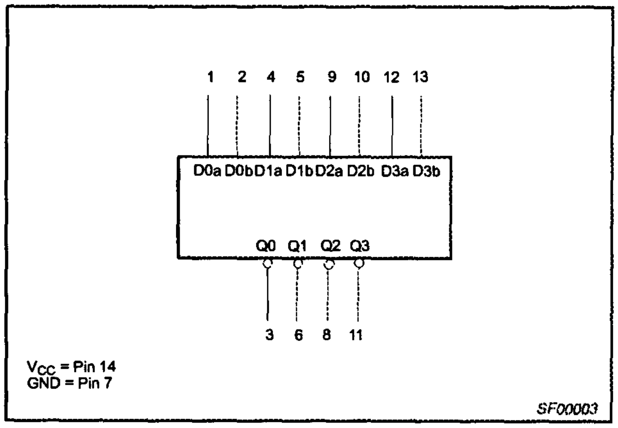


FUNCTION TABLE

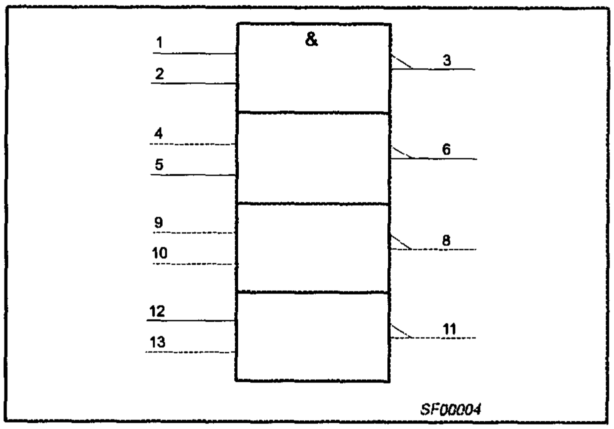
INPUTS		OUTPUT
Dna	Dnb	$\overline{Q}n$
L	L	H
L	H	H
H	L	H
H	H	L

NOTES:
H = High voltage level
L = Low voltage level

LOGIC SYMBOL


1 2 4 5 9 10 12 13
D0a D0b D1a D1b D2a D2b D3a D3b
$\overline{Q}0$ $\overline{Q}1$ $\overline{Q}2$ $\overline{Q}3$
3 6 8 11
V_{CC} = Pin 14
GND = Pin 7
SF00003

IEC/IEEE SYMBOL



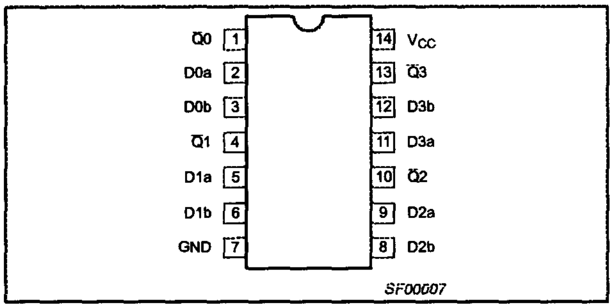
Quad 2-input NOR gate 74F02

FEATURE

- Industrial temperature range available (–40°C to +85°C)

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F02	3.4ns	4.4mA

PIN CONFIGURATION



ORDERING INFORMATION

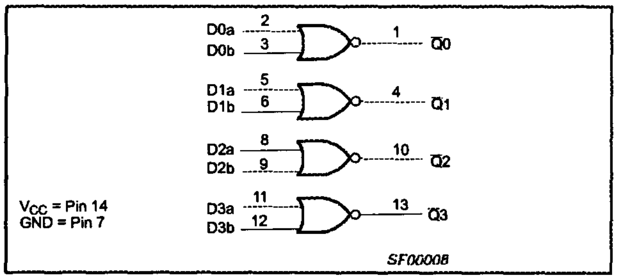
DESCRIPTION	ORDER CODE		PKG DWG #
	COMMERCIAL RANGE V_{CC} = 5V ±10%, T_{amb} = 0°C to +70°C	INDUSTRIAL RANGE V_{CC} = 5V ±10%, T_{amb} = –40°C to +85°C	
14-pin plastic DIP	N74F02N	I74F02N	SOT27-1
14-pin plastic SO	N74F02D	I74F02D	SOT108-1

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
Dna, Dnb	Data inputs	1.0/1.0	20μA/0.6mA
$\overline{Q}n$	Data output	50/33	1.0mA/20mA

NOTE:
One (1.0) FAST unit load is defined as: 20μA in the high state and 0.6mA in the low state.

LOGIC DIAGRAM

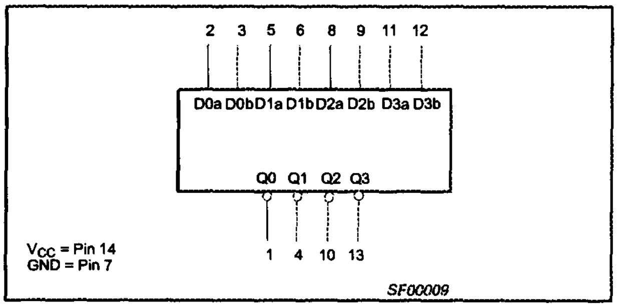


FUNCTION TABLE

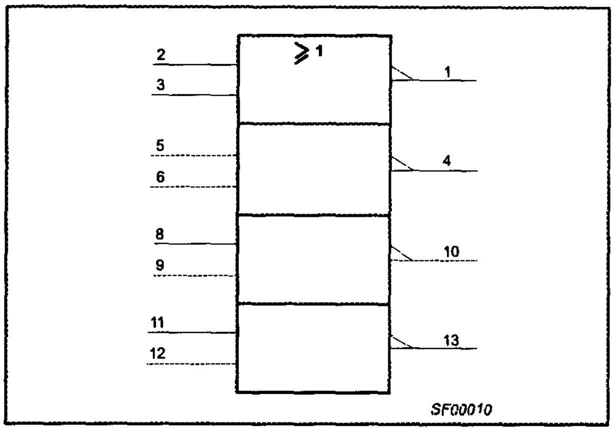
INPUTS		OUTPUT
Dna	Dnb	$\overline{Q}n$
L	L	H
L	H	L
H	L	L
H	H	L

NOTES:
1 H = High voltage level
2 L = Low voltage level

LOGIC SYMBOL



IEC/IEEE SYMBOL



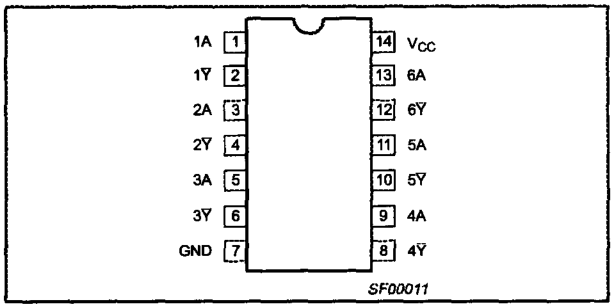
Hex inverter 74F04

FEATURE

- Industrial temperature range available (–40°C to +85°C)

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F04	3.5ns	6.9mA

PIN CONFIGURATION



ORDERING INFORMATION

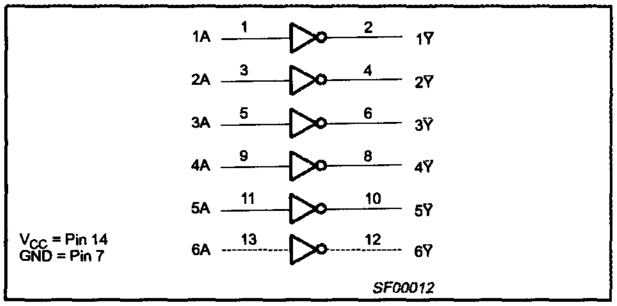
DESCRIPTION	ORDER CODE		PKG DWG #
	COMMERCIAL RANGE V_{CC} = 5V ±10%, T_{amb} = 0°C to +70°C	INDUSTRIAL RANGE V_{CC} = 5V ±10%, T_{amb} = –40°C to +85°C	
14-pin plastic DIP	N74F04N	I74F04N	SOT27-1
14-pin plastic SO	N74F04D	I74F04D	SOT108-1

INPUT AND OUTPUT LOADING AND FAN OUT TABLE

PINS	DESCRIPTION	74F (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
nA	Data inputs	1.0/1.0	20μA/0.6mA
$n\overline{Y}$	Data output	50/33	1.0mA/20mA

NOTE: One (1.0) FAST unit load is defined as: 20μA in the high state and 0.6mA in the low state.

LOGIC DIAGRAM

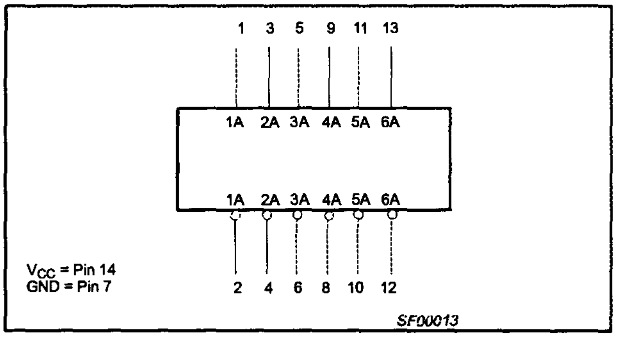


FUNCTION TABLE

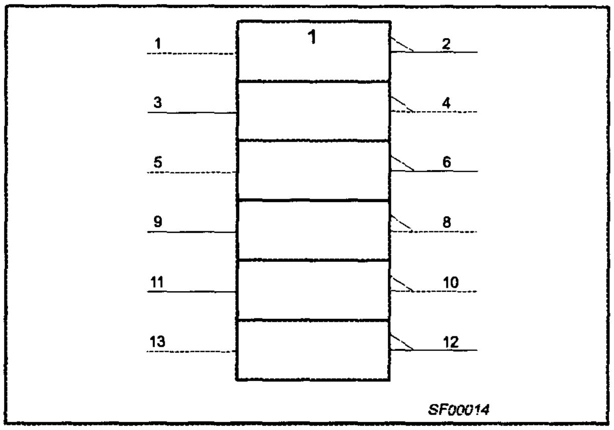
INPUTS	OUTPUT
A	$\overline{Y}$
L	H
H	L

NOTES:
H = High voltage level
L = Low voltage level

LOGIC SYMBOL



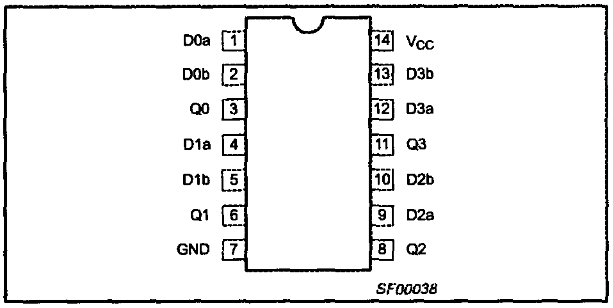
IEC/IEEE SYMBOL



- 74F08 Available for industrial range (–40°C to +85°C)

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F08	4.1ns	7.1mA

PIN CONFIGURATION



ORDERING INFORMATION

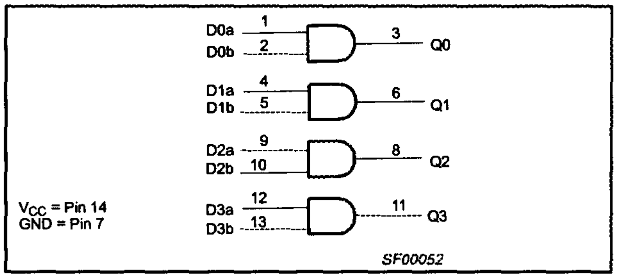
DESCRIPTION	COMMERCIAL RANGE V_{CC} = 5.0V ±10%, T_{amb} = 0°C to +70°C	INDUSTRIAL RANGE V_{CC} = 5.0V ±10%, T_{amb} = –40°C to +85°C	PKG DWG #
14-pin plastic DIP	N74F08N	I74F08N	SOT27-1
14-pin plastic SO	N74F08D	I74F08D	SOT108-1

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
Dna, Dnb	Data inputs	1.0/1.0	20µA/0.6mA
Qn	Data output	50/33	1.0mA/20mA

NOTE: One (1.0) FAST unit load is defined as: 20µA in the High state and 0.6mA in the Low state.

LOGIC DIAGRAM

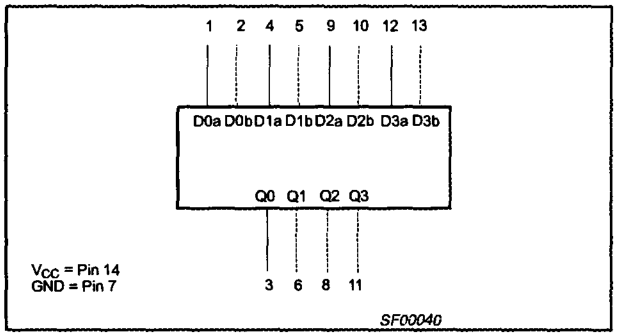


FUNCTION TABLE

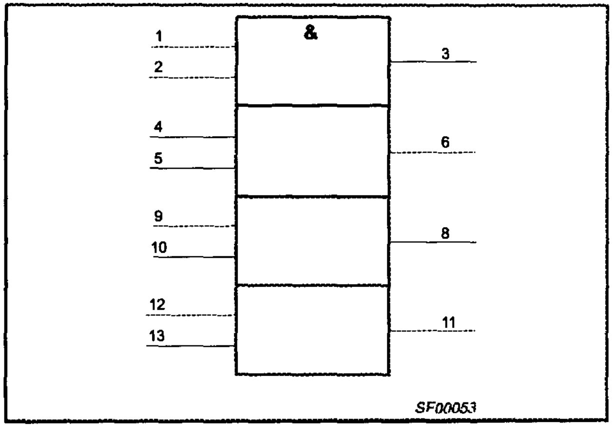
INPUTS		OUTPUT
Dna	Dnb	Qn
L	L	L
L	H	L
H	L	L
H	H	H

NOTES:
H = High voltage level
L = Low voltage level

LOGIC SYMBOL



LOGIC SYMBOL (IEEE/IEC)

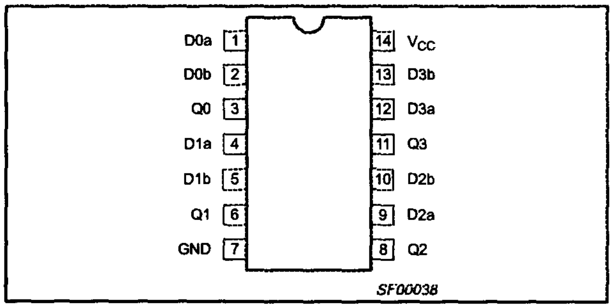


FEATURE

- Industrial temperature range available (–40°C to +85°C)

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F32	4.1ns	8.2mA

PIN CONFIGURATION



ORDERING INFORMATION

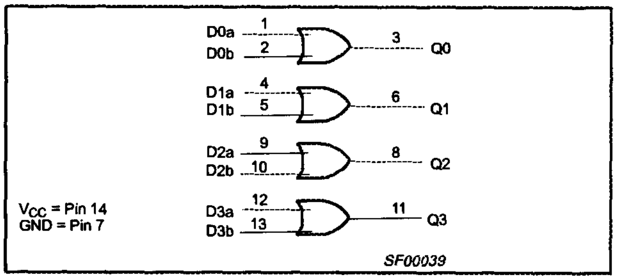
DESCRIPTION	ORDER CODE		PKG DWG #
	COMMERCIAL RANGE V_{CC} = 5V ±10%, T_{amb} = 0°C to +70°C	INDUSTRIAL RANGE V_{CC} = 5V ±10%, T_{amb} = –40°C to +85°C	
14-pin plastic DIP	N74F32N	I74F32N	SOT27-1
14-pin plastic SO	N74F32D	I74F32D	SOT108–1

INPUT AND OUTPUT LOADING AND FAN OUT TABLE

PINS	DESCRIPTION	74F (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
Dna, Dnb	Data inputs	1.0/1.0	20µA/0.6mA
Qn	Data output	50/33	1.0mA/20mA

NOTE: One (1.0) FAST unit load is defined as: 20µA in the high state and 0.6mA in the low state.

LOGIC DIAGRAM

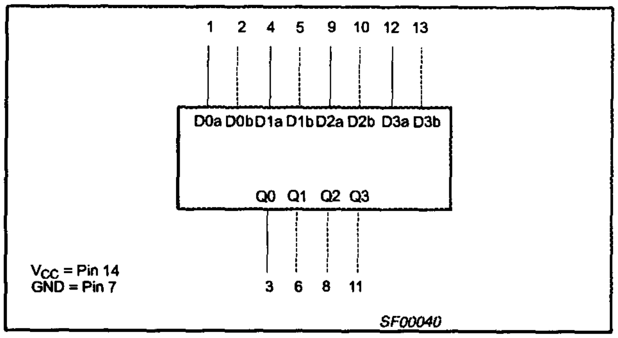


FUNCTION TABLE

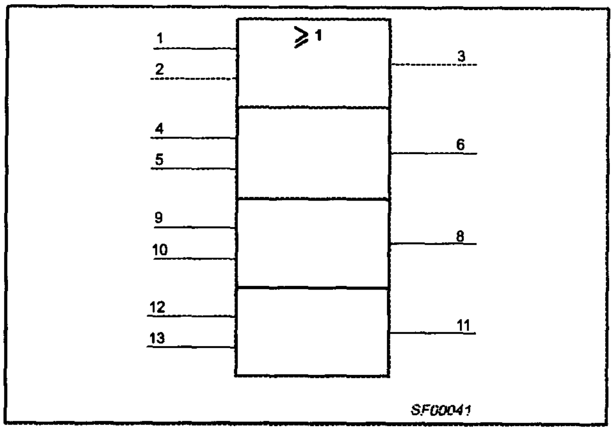
INPUTS		OUTPUT
Dna	Dnb	Qn
L	L	L
L	H	H
H	L	H
H	H	H

NOTES:
1 H = High voltage level
2 L = Low voltage level

LOGIC SYMBOL



IEC/IEEE SYMBOL

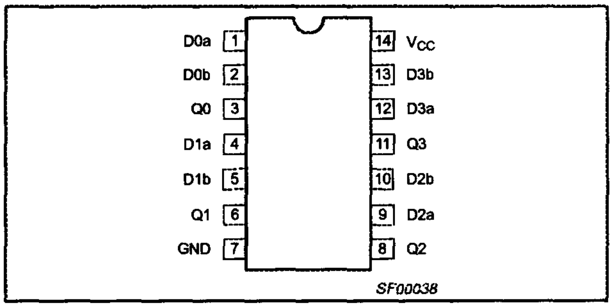


FEATURE

- Industrial temperature range available (–40°C to +85°C)

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F86	4.3ns	16.5mA

PIN CONFIGURATION



ORDERING INFORMATION

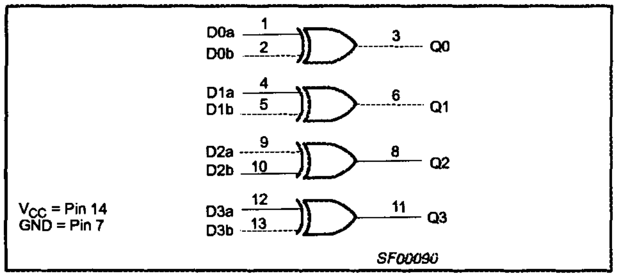
DESCRIPTION	ORDER CODE		PKG DWG #
	COMMERCIAL RANGE V_{CC} = 5V ±10%, T_{amb} = 0°C to +70°C	INDUSTRIAL RANGE V_{CC} = 5V ±10%, T_{amb} = –40°C to +85°C	
14-pin plastic DIP	N74F86N	I74F86N	SOT27-1
14-pin plastic SO	N74F86D	I74F86D	SOT108-1

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
Dna, Dnb	Data inputs	1.0/1.0	20µA/0.6mA
Qn	Data output	50/33	1.0mA/20mA

NOTE:
1. One (1.0) FAST unit load is defined as: 20µA in the High state and 0.6mA in the Low state.

LOGIC DIAGRAM

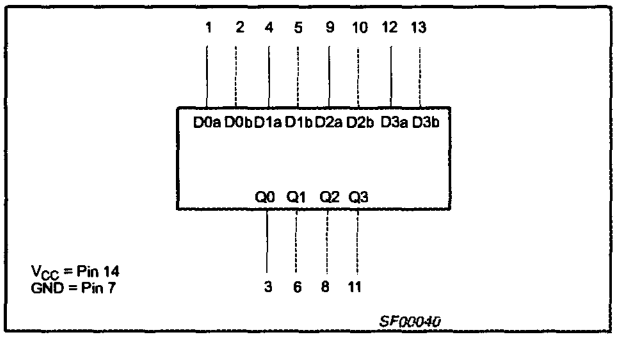


FUNCTION TABLE

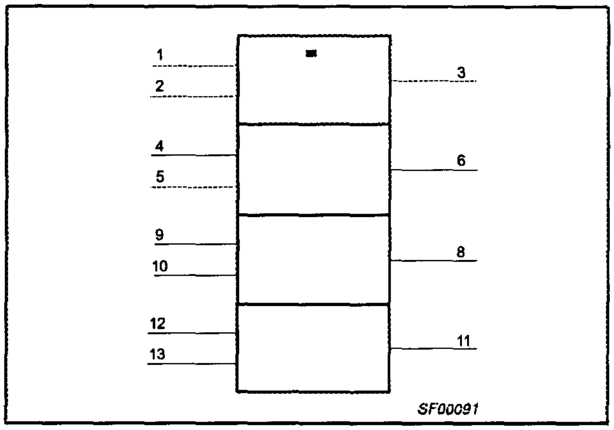
INPUTS		OUTPUT
Dna	Dnb	Qn
L	L	L
L	H	H
H	L	H
H	H	L

NOTES:
H = High voltage level
L = Low voltage level

LOGIC SYMBOL



IEC/IEEE SYMBOL



LAB # 3

Logic Implementation with One Gate Type

Purpose: Use NAND and NOR gates to implement AND, OR and NOT functions.

1. Logic Implementation of a basic gates with NAND or NOR gates.

   In preparation for the lab, develop a logic diagram showing how AND, OR and NOT gates can be implemented using:

   a. NAND gates

   b. NOR gates

>>> You may want to consider the chips shown on the next pages for parts 2, 3 and 4 <<<

2. Logic Implementation of a Boolean expression using NAND gates.

   In preparation for the lab develop a logic diagram for the expression

   $$L = A\,B'\,C' + A'\,B\,C' + A'\,B'\,C + A\,B\,C$$

   So that it can be implemented with NAND gates only.

   During the lab, do the following:

   a. wire the circuit in accordance with the logic diagram developed for the prelab.
   b. test and debug the logic.
   c. when the logic is functioning properly, call the instructor to verify its operation.

3. Logic Implementation of a Boolean expression using NOR gates.

   In preparation for the lab develop a logic diagram for the expression

   $$L = A\,B'\,C' + A'\,B\,C' + A'\,B'\,C + A\,B\,C$$

   So that it can be implemented with NOR gates only.

   During the lab, do the following:

   a. wire the circuit in accordance with the logic diagram developed for the prelab.
   b. test and debug the logic.
   c. when the logic is functioning properly, call the instructor to verify its operation.

4. Logic Implementation of a Majority gate.

   A majority gate produces an output 1 when the majority of the inputs are 1's. The output becomes 0 when the majority of the inputs are 0's.

   In preparation for the lab develop the logic for a 3 input majority gate using NAND gates.
   During the lab, do the following:

   a. wire the circuit in accordance with the logic diagram developed for the prelab.
   b. test and debug the logic.
   c. when the logic is functioning properly, call the instructor to verify its operation.

SN54F10, SN74F10
TRIPLE 3-INPUT POSITIVE-NAND GATES

SDFS039A – MARCH 1987 – REVISED OCTOBER 1993

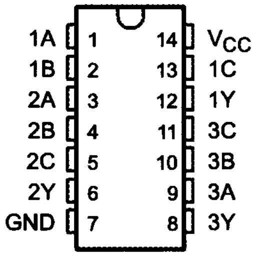
- **Package Options Include Plastic Small-Outline Packages, Ceramic Chip Carriers, and Standard Plastic and Ceramic 300-mil DIPs**

description

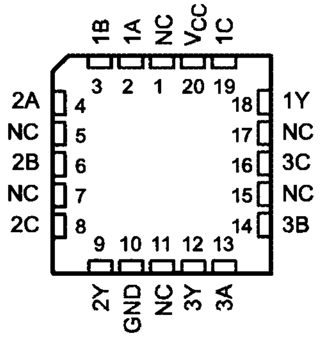
These devices contain three independent 3-input NAND gates. They perform the Boolean functions $Y = \overline{A \bullet B \bullet C}$ or $Y = \overline{A} + \overline{B} + \overline{C}$ in positive logic.

The SN54F10 is characterized for operation over the full military temperature range of –55°C to 125°C. The SN74F10 is characterized for operation from 0°C to 70°C.

SN54F10 . . . J PACKAGE
SN74F10 . . . D OR N PACKAGE
(TOP VIEW)



SN54F10 . . . FK PACKAGE
(TOP VIEW)

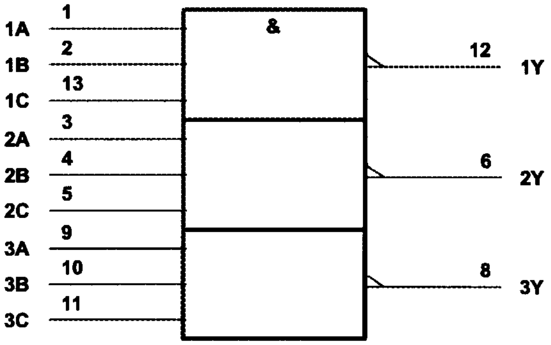


NC – No internal connection

FUNCTION TABLE
(each gate)

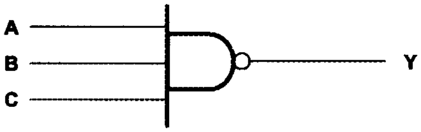
INPUTS			OUTPUT
A	B	C	Y
H	H	H	L
L	X	X	H
X	L	X	H
X	X	L	H

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for the D, J, and N packages.

logic diagram, each gate (positive logic)



SN54F20, SN74F20
DUAL 4-INPUT POSITIVE-NAND GATES

SDFS041A – MARCH 1987 – REVISED OCTOBER 1993

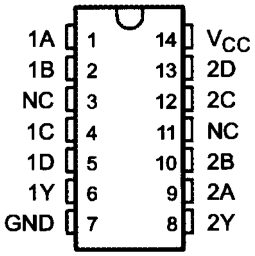
- **Package Options Include Plastic Small-Outline Packages, Ceramic Chip Carriers, and Standard Plastic and Ceramic 300-mil DIPs**

description

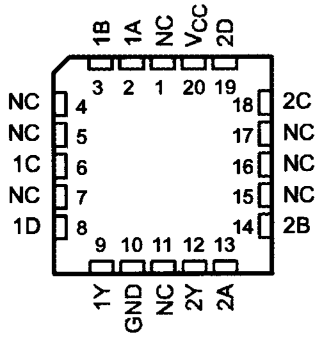
These devices contain two independent 4-input NAND gates. They perform the Boolean functions $Y = \overline{A \bullet B \bullet C \bullet D}$ or $Y = \overline{A} + \overline{B} + \overline{C} + \overline{D}$ in positive logic.

The SN54F20 is characterized for operation over the full military temperature range of −55°C to 125°C. The SN74F20 is characterized for operation from 0°C to 70°C.

SN54F20 . . . J PACKAGE
SN74F20 . . . D OR N PACKAGE
(TOP VIEW)



SN54F20 . . . FK PACKAGE
(TOP VIEW)

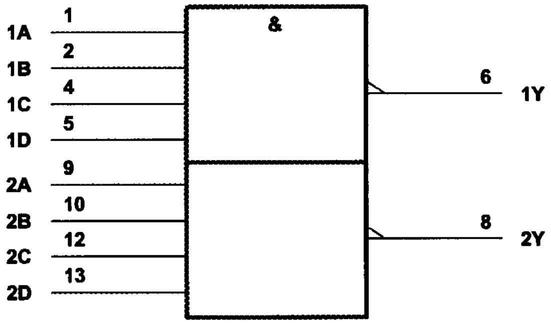


NC – No internal connection

FUNCTION TABLE
(each gate)

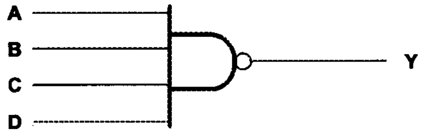
INPUTS				OUTPUT
A	B	C	D	Y
H	H	H	H	L
L	X	X	X	H
X	L	X	X	H
X	X	L	X	H
X	X	X	L	H

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for the D, J, and N packages.

logic diagram, each gate (positive logic)



SN54F27, SN74F27
TRIPLE 3-INPUT POSITIVE-NOR GATES

SDFS042A – MARCH 1987 – REVISED OCTOBER 1993

- **Package Options Include Plastic Small-Outline Packages, Ceramic Chip Carriers, and Standard Plastic and Ceramic 300-mil DIPs**

description

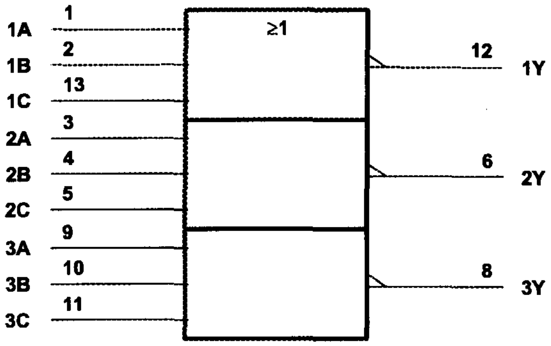
These devices contain three independent 3-input NOR gates. They perform the Boolean functions $Y = \overline{A} \bullet \overline{B} \bullet \overline{C}$ or $Y = \overline{A + B + C}$ in positive logic.

The SN54F27 is characterized for operation over the full military temperature range of –55°C to 125°C. The SN74F27 is characterized for operation from 0°C to 70°C.

FUNCTION TABLE
(each gate)

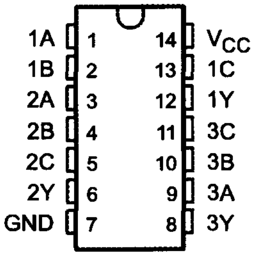
INPUTS			OUTPUT
A	B	C	Y
H	X	X	L
X	H	X	L
X	X	H	L
L	L	L	H

logic symbol†

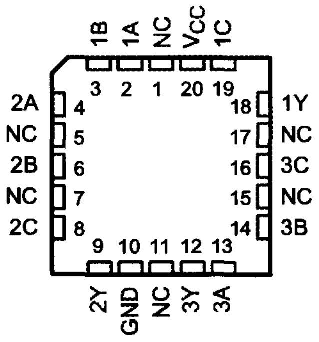


† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for the D, J, and N packages.

SN54F27 . . . J PACKAGE
SN74F27 . . . D OR N PACKAGE
(TOP VIEW)

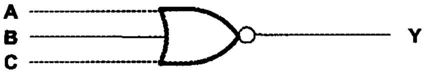


SN54F27 . . . FK PACKAGE
(TOP VIEW)



NC – No internal connection

logic diagram, each gate (positive logic)



LAB # 4

Logic Simplification with K-maps

Purpose: Use of the K-map to simplify logic expressions

1. Division of a single digit BCD number by 5

A BCD digit can be represented by four bits ($B_3B_2B_1B_0$) with values 0 through 9. This number when divided by 5 will yield a one bit quotient (Q_0) and a three bit remainder ($R_2R_1R_0$).

In preparation for the lab design a logic that accepts the BCD number ($B_3B_2B_1B_0$) and compute the quotient (Q_0) and the remainder ($R_2R_1R_0$) resulting from a division by 5. As part of your design provide the following:

a. a truth table with columns corresponding to $B_3B_2B_1B_0$, Q_0 and $R_2R_1R_0$.
b. K-maps to obtain a logic expression for Q_0, R_2, R_1 and R_0 in terms of $B_3B_2B_1B_0$. Be sure to take advantage of don't cares to simplify the logic.
c. a logic diagram that uses NAND gates and inverters to implement the logic expressions obtained in part b. Use switches for entering the BCD number and LED lights to display the quotient and the remainder.

During the lab, do the following:

d. wire the circuit in accordance with the logic diagram developed for the prelab.
e. test and debug the logic.
f. when the logic is functioning properly, call the instructor to verify its operation.

2. A Full-Adder Cell

A full-adder cell allows the addition of two bits (x_i and y_i) with a carry (c_i) from a less significant two bit addition and produces a sum (s_i) and a carry (c_{i+1}) for a more significant two bit addition. The operation and the corresponding truth table are shown aside.

$$\begin{array}{r} c_i \\ x_i \\ +\ y_i \\ \hline c_{i+1}\ s_i \end{array}$$

$c_i\ x_i\ y_i$	c_{i+1}	s_i
0 0 0	0	0
0 0 1	0	1
0 1 0	0	1
0 1 1	1	0
1 0 0	0	1
1 0 1	1	0
1 1 0	1	0
1 1 1	1	1

In preparation for the lab design the logic to accept $c_{i,}$ x_i and y_i as input and to compute the outputs s_i and c_{i+1}. As part of your design provide the following:

a. K-maps to obtain a logic expression for s_i and c_{i+1} in terms of $c_{i,}$ x_i and y_i .
b. based on the chips available on the lab provide a logic diagram that will implement a Full-adder cell with a minimum number of chips.
c. obtain a truth table for the expected value of s_i and s_i in terms of c_i, x_i and y_i

During the lab, do the following:

d. wire the circuit in accordance with the logic diagram developed for the prelab.
e. test and debug the logic.
f. when the logic is functioning properly, call the instructor to verify its operation.

LAB # 5

Programming Logic with Verilog

Purpose: Learn how to to enter, simulate and debug logic expressions using VeriWell.

1. An Example of Combinational Logic

In preparation for the lab study the Verilog program listed below and:

a. extract the logic expression that is being implemented by the “logic1” modules.
b. explain what Verilog models were used to implement logic1 in Model A and Model B
c. obtain a truth table for the expected value of d in terms of a, b and c
d. explain what the instruction {a,b,c} = count does
e. describe the function of $display and $monitor

```
module stimulus; // tested for module logic1reg a,b,c;
  reg [2:0] count;
  wire d;

  // instantiate the module logic1
  logic1 trylogic(d,a,b,c);

  // use count to assign values to a,b and c
  initial
        count = 3'b0; // set count = 0

  always
    begin
        {a,b,c} = count; // assign values to a,b and c
        #5 count = count+1; // compute next set of values
    end

  initial
        #40 $finish; // terminate the simulation

  // display the value of inputs and output
  initial
    begin    $display("              time ","a ","b ","c ","d");
        $monitor($time," ",a," ", b," ", c," ", d);
    end
endmodule

module logic1(W,X,Y,Z);  //   model A
  output W;
  input X,Y,Z;
  wire P,Q;

  not(P,Y);
  or(Q,X,P);
  and(W,Q,Z);
endmodule
```

```
/*
module logic1(W,X,Y,Z);  //   model B
  output W;
  input X,Y,Z;

  assign W = (X || !Y) && Z;
endmodule
*/
```

During the lab, do the following:

a. copy the code of this program from the server into your account
b. run the Verilog program with model A using VeriWell
c. print the Verilog program and the results of the simulation on the VeriWell Console
d. ask the instructor to verify your results and initial your listing.
e. do steps b. through d. using model B.

2. Majority gate Implementation with Verilog

In preparation for the lab develop the logic for a 3 input majority gate using:

a. Verilog behavioral model (use assign, &&, || and ! to enter logic expressions)
b. Verilog gate model (use built in NAND function)
c. a stimulus program to test the models developed in parts a and b.

During the lab, do the following:

d. program the logic developed for the prelab.
e. test and debug the logic.
f. when the logic is functioning properly, call the instructor to verify its operation.
g. document the source code with comments
h. print the source code and the test output

3. Full-Adder Cell Implementation with Verilog

a. use the information provided in lab 4.2 to develop a behavioral model for a Full-Adder (use assign, &&, || and !)
b. design a stimulus program to test the models developed in part a.

During the lab, do the following:

c. program the logic developed for the prelab.
d. test and debug the logic.
e. when the logic is functioning properly, call the instructor to verify its operation.
f. document the source code with comments
g. print the source code and the test output

LAB # 6

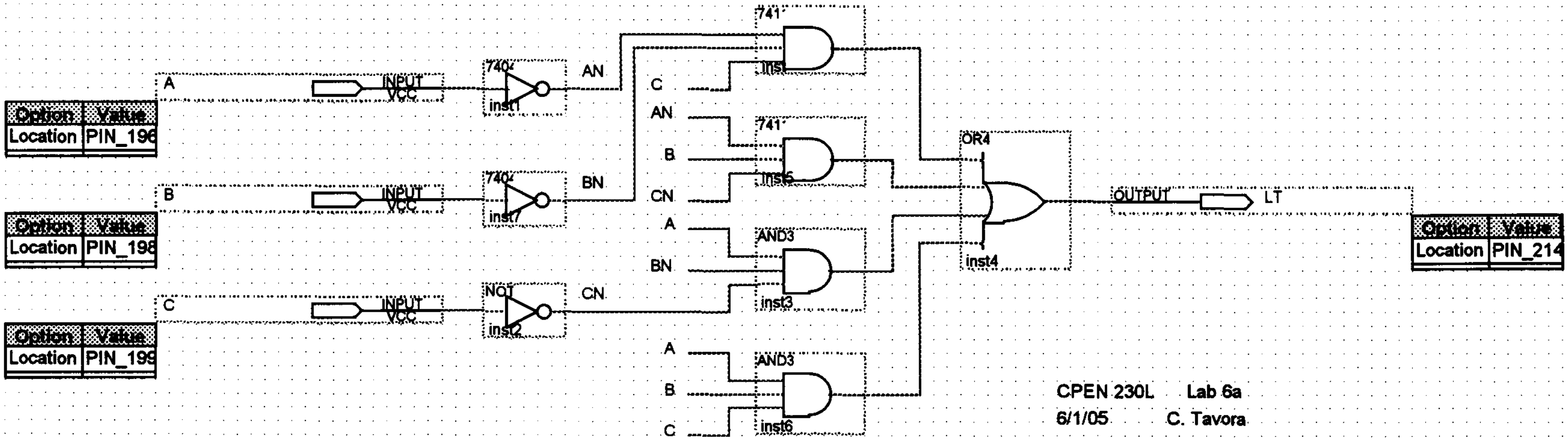
Logic Implementation with a PLD

Purpose: Learn how to use Quartus to enter, simulate, program and test logic expressions on a PLD using the Programmable Logic Unit (PLU).

1. Logic Entry with Quartus

a. Use Windows Explore to create a folder “Lab6a” inside the folder “CPEN230L” in the server under your account. Use Quartus to create a project by selecting
> File > New Project Wizard
enter the path to the folder “Lab6a”
directory: n:\cpen230L\Lab6a
Project Name: Lab6a
Top-level Entity: Lab6a
use the “finish” button to exit the window

b. Enter Schematic
> File > New -> Block Diagram/Schematic file -> OK (*.bdf)
Expand the drawing window to full screen by pressing “Ctrl” “Alt” “space”.
Press the same set of keys to return to the original arrangement of windows.
Select a component
Left double click anywhere on the grided window. Enter under “Name” each of the following parts (one at a time) and “return”:
7404, 7411, not, and3, or4, gnd, vcc, input, output.
Place the components as shown in the figure Lab6a.
Delete the vcc and gnd by selecting the component and using “delete =>”.
Duplicate required gates by using copy and paste ctrl C, ctrl V
Use the tool to connect the gates as shown in figure Lab6a. Observe that the tool can also be selected by left clicking the cursor at an input (or output) of a gate.
Assign a name to a line by right double clicking on the line and selecting “properties” and assign a name to an I/O pin by left double clicking on pin_name. Enter the names A, B, C, AN, BN, CN, LT as shown in figure Lab6a.
Save the logic diagram (observe that the window name is changed to Lab6a.bdf).
Identify the PLD used for implementation
> Assignments > Device
Family: FLEX 10K select under “Available Devices”
EPF10K20RC240-4 OK
Assign a PLD pin number to each input and output connector as shown in figure Lab6a by entering
> Assignments > Assign Pins
scroll the window to a row with the desired number - left click on that row
enter the pin_name on the window “Pin Name:”
click the button “Add”
repeat this procedure until all I/O pins are numbered then click OK.
Save the diagram
> File > Save
Compile the project
> Processing > Start Compilation
observe the progress of the compilation on the window named “Module”.
Print the logic diagram
double click the entry “Lab6a” on the window “entity” to display the logic diagram.
> File > Print



CPEN 230L Lab 6a
6/1/05 C. Tavora

2. Logic Simulation

a. Enter Input Waveforms

> File > New > Other Files > Vector Waveform File OK (*.vwf)

select the “end time” and the “grid size”

> Edit > End Time enter 10.0 us
> Edit > Grid Size enter 1.0 us
> View > Fit in Window

enter input waveform Names

double click 1st line under Name
enter the Name: A OK
double click 2nd line under Name
enter the Name: B OK
....
....
double click next line under Name
enter the Name: LT OK

enter the waveforms

left-click on the input symbol to the left of variable A (this will cause the row associated with A to be highlighted). Left-click on the C-waveform on the tool palet. Select the “timing” tab and enter

start time: 0 ps
end time: 10.0 us
count every: 1.0 us
multiplied by: 1 OK

do the same thing with B and C with the exception that

for B set “multiplied by:” equal to 2
for C set “multiplied by:” equal to 4

save the waveform file

b. Compile and Simulate

> Processing > Start Compilation & Simulation
observe the process of the compilation and simulation on the module window
*** Errors may be detected that require correction ***
display the simulation results
> Processing > Simulation Report
expand the simulation window to fill the whole screen (Ctrl Alt space)
use the cursor the check the status of all variables at a selected time

left-click and hold-down on the little square box located at 0 ps just above the waveforms. Pull the box along the time scale and observe the change in value of variables on the column “value” next to the column “name”.

reduce the size of the simulation window (Ctrl Alt space)

3. Download the logic configuration to the PLU

a. Set up the programmer

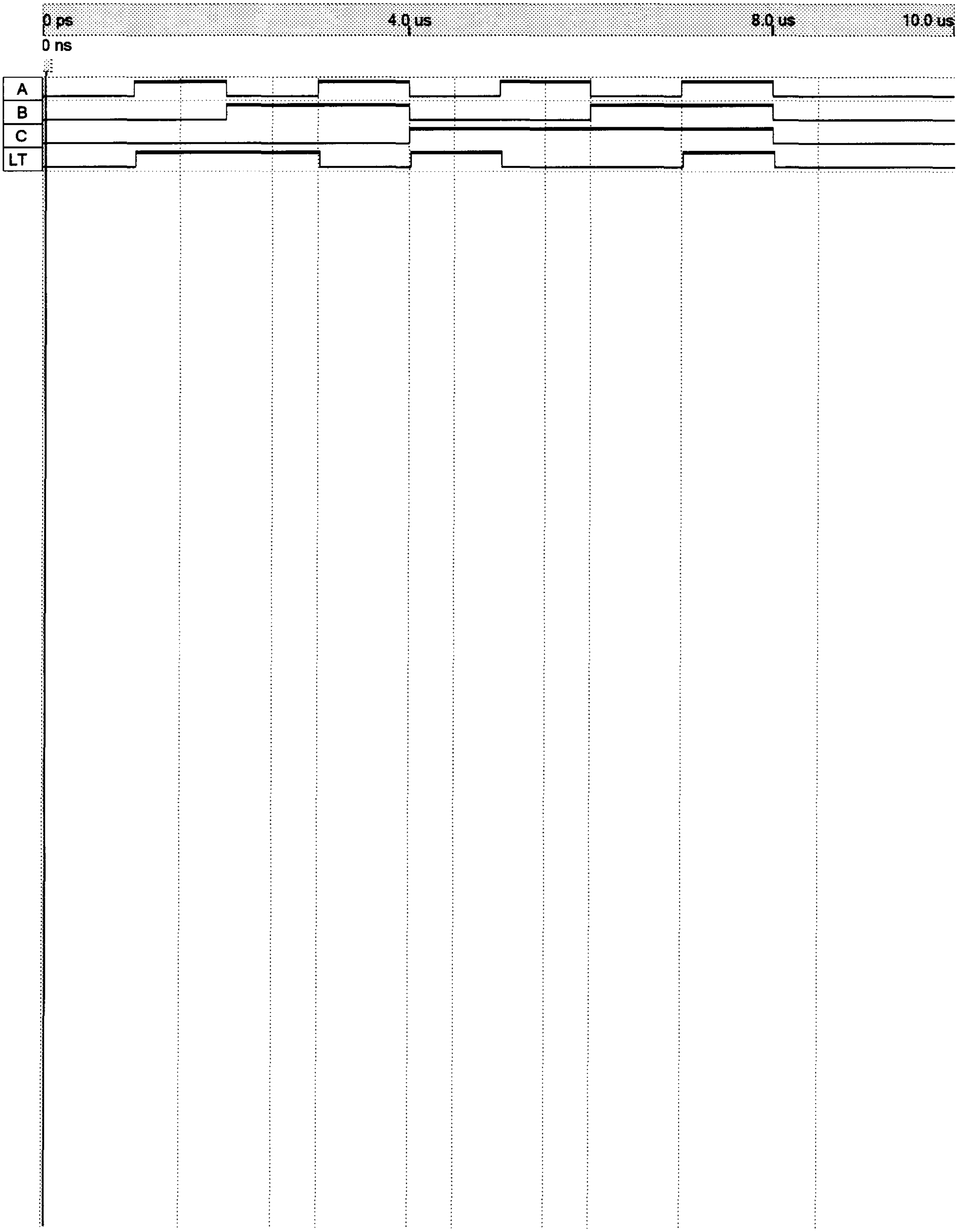
> Tools > Programmer
verify that

the window next to the “hardware setup” shows “ByteBlaster [LPT1]”
the window next to the “Mode” shows “JTAG”.

b. Select programming file

click on the “add file” button and select the file “Lab6a.sof”
click on the square box under “Program/Configure” on the file description line. Observe

Date: June 6, 2005 db/Lab6a-sim.vwf Project: Lab6a



that a check mark is added to the box.
turn on the power to the Programmable Logic Unit box.
observe that the green power light on the board is lit.
left-click the "start" button on the programmer window.
observe that the "configured" light on the board is now lit.

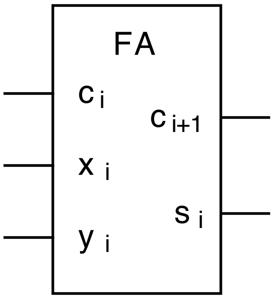
c. Test the logic
flip switches SW0 (196), SW1 (198) and SW2 (199) and observe the output in LT0 (214).

4. A Full-Adder Cell

In preparation for the lab use the logic expressions developed in lab 4 part 2 to:
a. implement a Full-Adder cell using Verilog.
b. check the functionality of the cell using VeriWell.

During the lab, do the following:

c. create the folder Lab6b and project Lab6b.
d. make the logic compatible with Quartus.
e. use the "create default symbol" from the "file" menu to modularize the logic into a virtual device named "FA". This device shall have the inputs and output shown aside.
f. use Quartus simulator with appropriate waveforms to test and debug the logic.
g. when the logic is functioning properly, call the instructor to verify its operation.



5. A 4-bit Adder using with Full-Adder Cells

In preparation for the lab do the following:
a. show how 4 FA modules can be interconnected to implement a 4-bit adder
b. develop a logic diagram for the PLU that will allow testing the 4-bit adder. Use switches 0-3 for x and switches 4-7 for y. Lights 0-5 shall be used for the result.

During the lab, do the following:

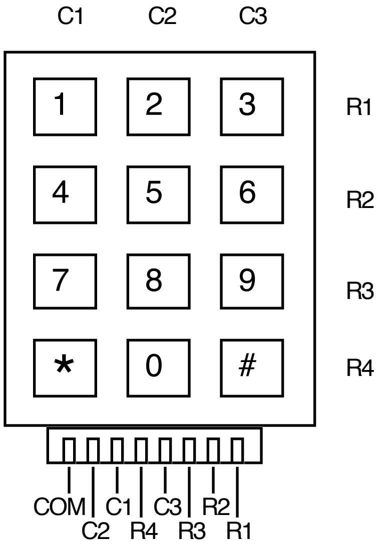
c. create the folder Lab6c and project Lab6c.
d. program the 4-bit adder. Use 4-bit busses for the inputs x and y, and for the output s. Create a symbol for the 4-bit adder. Connect the adder to PLU lights and switches.
e. compile, test and debug the logic
f. when the logic is functioning properly, call the instructor to verify its operation.

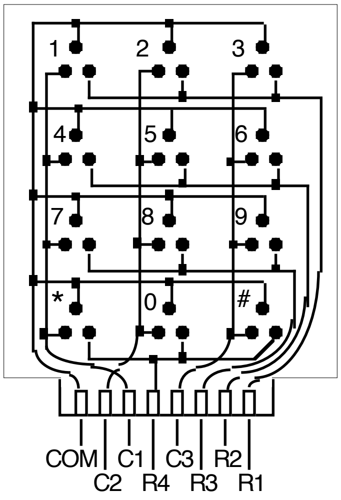
LAB # 7

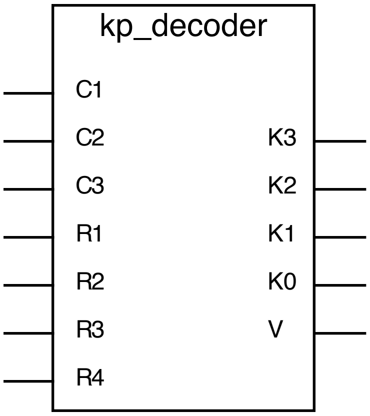
Keypad Decoder

Purpose: Use Verilog and Quartus to design a decoder that translates 2 out of 7 keypad signals into a 4-bit binary value and a valid entry signal.

A diagram of the keypad on the Programmable Logic Unit is shown below. The terminal COM is connected to ground and the terminals linked to rows and columns have pull up resistors.







The decoder shall receive the active-low signals C1, C2, C3, R1, R2, R3 and R4 from the keypad and generate the active-high signals K3, K2, K1, K0 and V. The signals K3, K2, K1, K0 shall provide the binary value of the key being depressed with K3 being the most significant bit. Depressing the keys 0 - 9, *, # shall produce the output 0000 - 1001, 1010 and 1011. When no key is being depressed or when only a row or a column is low, the value 1111 shall be output. The signal V when asserted indicates that a valid key has been depressed and that its value can be read on K3, K2, K1 and K0. The signal V shall not be asserted if no key or more than one key are depressed. Observe that the signal V, if debounced, can be used as a clock for entry of the data on outputs K3, K2, K1 and K0.

1. In preparation for the lab do the following:
   a. obtain logic expressions for K3, K2, K1, K0 and V in terms of C1, C2, C3, R1, R2, R3 and R4.
   b. write the code in Verilog to implement the logic for decoding the keypad.

   During the lab, do the following:

   c. use Quartus to enter the keypad decoder logic. Encapsulate the logic into a module and create a symbol for the device and name it “kp_decoder” (see figure above).
   d. connect the keypad decoder to the 7-segment display decoder. Display the V signal on the decimal point (pin 14) of the display. Use the V signal to blank the display when no key is depressed or if more than one key is depressed.
   e. compile, test and debug the logic
   f. when the logic is functioning properly, call the instructor to verify its operation.

LAB # 8

Addition and Subtraction

Purpose: Design of an arithmetic unit that performs addition and subtraction. Use of multiplexors.

1. Design an 4-bit adder and subtractor
   This logic will allow the PLU to perform the operation C = A ± B. The value A shall be entered on switches SW0 - SW3 with SW0 being the most significant bit. The value B shall be entered on pushbuttons SW4 - SW7 with SW4 being the most significant bit. The result C shall be displayed on lights LT4 - LT7 with LT4 being the most significant bit. The operation shall be specified by switch SW8 (up = +, down = -). A carry or borrow shall be displayed on light LT3.

   <u>In preparation for the lab</u>
   a. review the work done in part 5 of lab 6. Develop a logic diagram for the adder and subtracter (consider Figure 5.13 of the text) name this project Lab 8A .
   b. develop a Verilog module named ADS that has two 4-bit input buses (A and B), an input operation selector line (S) and an 4-bit output bus (C). Use Quartus to interface the module ADS to the switches and lights name this Lab8B.
   c. select test values to validate the operation of the two implementations of an adder and subtracter.

   <u>During the lab</u>, do the following:

   d. use Quartus to enter the logic for Lab8a
   e. compile, test and debug the logic
   f. when the logic is functioning properly, call the instructor to verify its operation.
   g. repeat steps d, e and f for Lab8b

2. Design an 4-bit BCD adder
   Implement a 1 digit BCD adder using the logic shown in Figure 5.43 of the text. Use for inputs and outputs the same switches and lights specified in part 1.

   <u>In preparation for the lab</u>
   a. develop a logic diagram for the BCD adder using Quartus logic parts. Name this diagram Lab 8C.
   b. select test values to validate the operation of the operation of the BCD adder.

   <u>During the lab</u>, do the following:

   c. use Quartus to enter the logic for Lab8C
   d. compile, test and debug the logic
   e when the logic is functioning properly, call the instructor to verify its operation.

LAB # 9

Logic Implementation with MUX, DEMUX and ROM

Purpose: How to use multiplexors, demultiplexors and ROMs to implement logic.

1. MUX Implementation of a Full-Adder cell
   Using the information provided in Lab 4.2 implement the expressions for s_i and c_{i+1} using two 4-to-1 muxes.

   In preparation for the lab
   a. make a diagram for implementing the logic on the PLU with the following I/O assignment:

   c_i = PB2 x_i = PB1 y_i = SW1 s_i = LT1 c_{i+1} = LT0

   b. prepare a truth table with the expected value of s_i and c_{i+1} in terms of c_i, x_i and y_i

   During the lab, do the following:
   c. use Quartus to enter the logic
   d. compile, test and debug the logic
   e. when the logic is functioning properly, call the instructor to verify its operation.

2. DEMUX Implementation of a Full-Adder cell
   Using the information used in part 1, implement the expressions for s_i and c_{i+1} using one demux and OR gates.

   In preparation for the lab
   a. make a diagram for implementing the logic on the PLU with the following I/O assignment:

   c_i = PB7 x_i = PB6 y_i = SW6 s_i = LT6 c_{i+1} = LT5

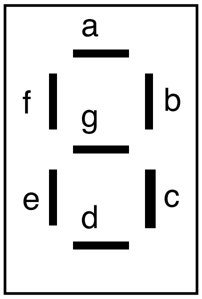
   During the lab, do the following:
   c. use Quartus to enter the logic
   d. compile, test and debug the logic
   e. when the logic is functioning properly, call the instructor to verify its operation.

3. ROM Implementation of a 4-bit to 7-segment HEX decoder Design

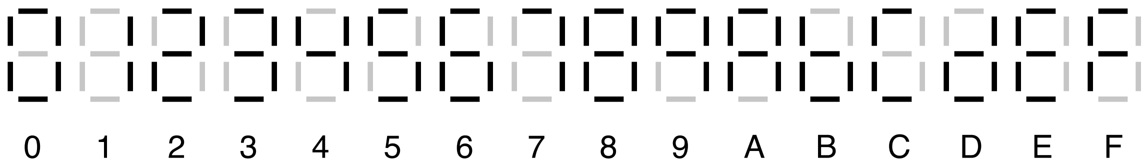
   In preparation for the lab
   a. develop a minterm expression for each LED of a 7-segment display. The letter assignment for the display is shown aside. The HEX display shall be in accordance with the format shown below.

   b. Prepare a programming table for a ROM with 4 address lines and 7 data lines to implement the decoder.

   c. Prepare a logic diagram in which switches SW0, SW1, SW2



and SW3 (weights 8,4,2,1) can be used to enter a binary value to be displayed on LED display.

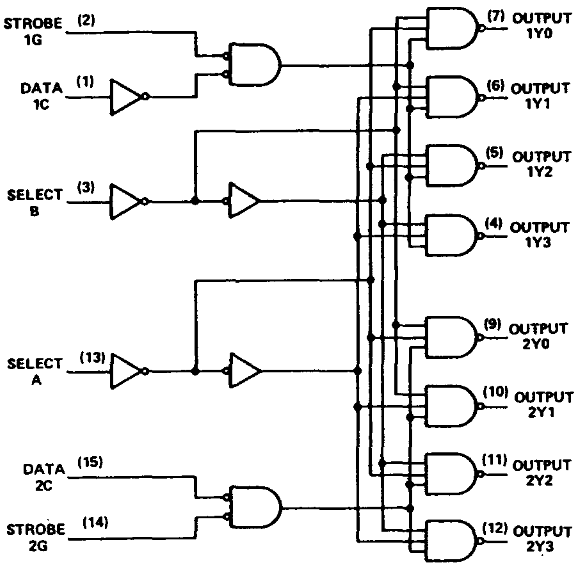


<u>During the lab</u>, do the following:

d. use Quartus to enter the logic and program the ROM.
e. compile, test and debug the logic
f. when the logic is functioning properly, call the instructor to verify its operation.

TYPES SN54155, SN54156, SN54LS155, SN54LS156, SN74155, SN74156, SN74LS155, SN74LS156 DUAL 2-LINE-TO-4-LINE DECODERS/DEMULTIPLEXERS

functional block diagram and logic



FUNCTION TABLES
2-LINE-TO-4-LINE DECODER
OR 1-LINE-TO-4-LINE DEMULTIPLEXER

INPUTS				OUTPUTS			
SELECT		STROBE	DATA	1Y0	1Y1	1Y2	1Y3
B	A	1G	1C				
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

INPUTS				OUTPUTS			
SELECT		STROBE	DATA	2Y0	2Y1	2Y2	2Y3
B	A	2G	2C				
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

FUNCTION TABLE
3-LINE-TO-8-LINE DECODER
OR 1-LINE-TO-8-LINE DEMULTIPLEXER

INPUTS				OUTPUTS							
SELECT			STROBE OR DATA	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
C†	B	A	G‡	2Y0	2Y1	2Y2	2Y3	1Y0	1Y1	1Y2	1Y3
X	X	X	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H
L	H	L	L	H	H	L	H	H	H	H	H
L	H	H	L	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H
H	H	L	L	H	H	H	H	H	H	L	H
H	H	H	L	H	H	H	H	H	H	H	L

†C = inputs 1C and 2C connected together
‡G = inputs 1G and 2G connected together
H = high level, L = low level, X = irrelevant

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)		7 V
Input voltage:	'155, '156	5.5 V
	'LS155, 'LS156	7 V
Off-state output voltage:	'155	5.5 V
	'LS155	7 V
Operating free-air temperature range:	SN54', SN54LS' Circuits	−55°C to 125°C
	SN74', SN74LS' Circuits	0°C to 70°C
Storage temperature range		−65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

Memory Initialization File (.mif)

An ASCII text file (with the extension **.mif**) that specifies the initial content of a memory block (CAM, RAM, or ROM), that is, the initial values for each address. This file is used during project compilation and/or simulation.

A MIF is used as an input file for memory initialization in the Compiler and Simulator. You can also use a Hexadecimal (Intel-Format) File (**.hex**) to provide memory initialization data.

A MIF contains the initial values for each address in the memory. A separate file is required for each memory block. In a MIF, you are also required to specify the memory depth and width values. In addition, you can specify the radixes used to display and interpret addresses and data values.

Following is a sample MIF:

```
DEPTH = 32;      % Memory depth and width are required   %
WIDTH = 14;      % Enter a decimal number        %

ADDRESS_RADIX = HEX;     % Address and value radixes are required
        %
DATA_RADIX = HEX;        % Enter BIN, DEC, HEX, OCT, or UNS; unless
        %
                         % otherwise specified, radixes = HEX    %

-- Specify values for addresses, which can be single address or range

CONTENT
        BEGIN

[0..F]  :       3FFF;    % Range--Every address from 0 to F = 3FFF
        %
        6       :        F;       % Single address--Address 6 = F %
        8       :        F E 5;   % Range starting from specific address--
        %

END ;   % Addr[8] = F, Addr[9] = E, Addr[A] = 5 %
```

If multiple values are specified for the same address, only the last value is used.

You can create a MIF using the Memory Editor.

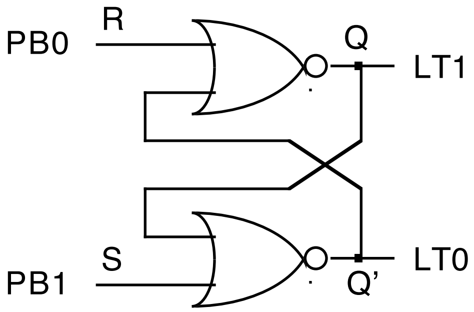
LAB # 10

Latches, Flip-Flops and Counters

Purpose: How to use latches, flip-flops and counters. When to debounce a switch.

1. Set-Reset (S-R) Latch

   The figure below shows a S-R latch implemented with NOR gates.



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

<u>In preparation for the lab</u>

a. fill in the truth table for this latch.

b. what combination of S-R inputs must not be allowed? If this condition is allowed to occur, will the latch be damaged? Could it introduce an error in the logic? Why?

<u>During the lab</u>, do the following:

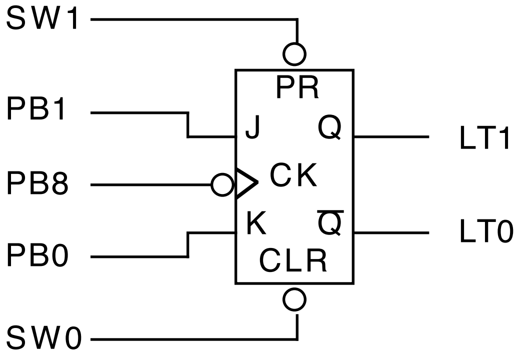
c. use Quartus to enter the logic for the latch. Save the work as project Lab10a.

d. check that the latch performs as specified in the next state table.

e. when the logic is functioning properly, call the instructor to verify its operation.

2. J-K Flip-Flop

   The figure below shows a 7476 J-K flip-flop connected to switches and lights.



J	K	Q	Q'
0	0		
0	1		
1	0		
1	1		

<u>In preparation for the lab</u>

a. fill in the truth table for this flip-flop.

b. which inputs have higher precedence J & K or PR & CLR?

c. in order to use the J & K inputs what level should be applied to PR & CLR?

d. on what edge of the clock do the inputs J & K get serviced? When will this happen, when PB8 is depressed or when it is released?

e. can this flip-flop be operated as a latch? How?

f. what is meant by switch bounce? Sketch the waveform of a signal produced by a closing switch. What flip-flop input is sensitive to a switch bounce?

During the lab, do the following:

g. use Quartus to enter the connections to the flip-flop. Save the work as project Lab10b.

h. check that the flip-flop performs as specified in the next state table and that the inputs PR & CLR function as expected.

i. observe that the flip-flop does not toggle consistently with the clock when J = K = 1. Explain why.

j. review with the instructor the operation of the flip-flop.

3. J-K Flip-Flop with debounced clock

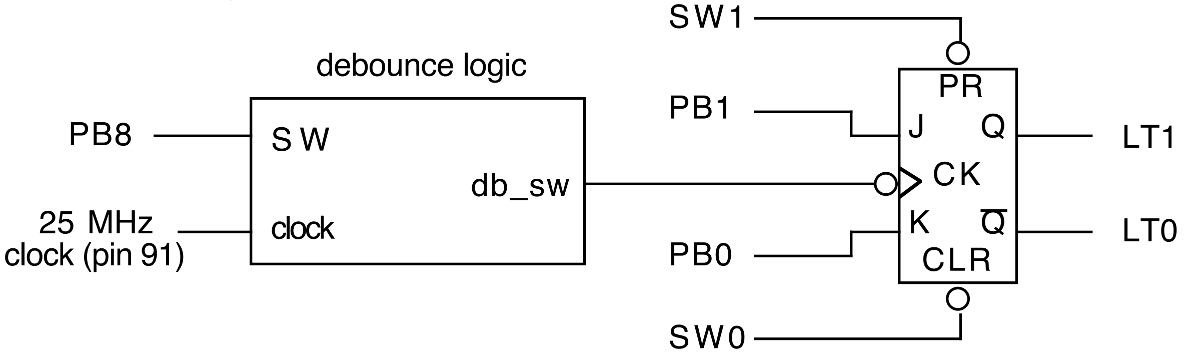
The pushbutton providing the clock to the flip-flop will be debounced as shown in the figure below.

During the lab, do the following:

a. create a new project on a new folder and name it Lab10c.

b. copy the files debounce1.gdf and debounce1.sym from \\davinci\class\cpen230\library into folder Lab10c.

c. place the debounce1 logic between PB8 and the 7476 flip-flop wired for Lab10b as shown in the figure below.



d. observe that the flip-flop toggle consistently with the clock when J = K = 1.

e. when the logic is functioning properly, call the instructor to verify its operation.

4. Ripple Counter

A 4-bit binary counter will be tested using a pushbutton as the clock and a 7-segment display for monitoring the output.

In preparation for the lab

a. prepare a logic diagram showing how switch PB8 can be used to increment the count of a 7493 counter. Use the debounce logic to make PB8 a clean clock. The output of the counter shall be connected to a 7-segment display using the 7-segment HEX decoder developed in part 3 of lab 9. Pushbutton PB9 shall be used to reset the counter.

b. design the logic required to convert this counter into a 0 to 9 counter. Pushbutton PB9 shall still be able to reset the counter.

c. design the logic required to convert this counter into a 0 to 6 counter. Pushbutton PB9 shall still be able to reset the counter.

d. does pushbutton PB9 require debounce logic?

During the lab, do the following:

d. use Quartus to enter the logic.
e. check that the counter performs as expected.
f. when the counter is functioning properly, call the instructor to verify its operation.
g. convert the counter to a 0 to 9 counter using the logic developed in part b.
h. test the operation of the 0 to 9 counter.
i. convert the counter to a 0 to 6 counter using the logic developed in part b.
j. test the operation of the 0 to 6 counter.
k. connect the pushbutton PB8 directly to the clock input of the counter (bypass the debounce logic). Test the counter operation and establish if the debounce logic is needed for proper counting.

5. Verilog Implementation of a D Flip-Flop

The Verilog code for a D Flip-Flop is listed below.

```
module DFF(D, clk, Q);
input D, clk;
output Q;
reg Q;

always @(posedge clk)
    Q = D;

endmodule
```

In preparation for the lab

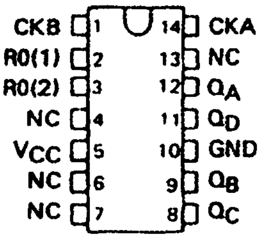
a. Design the logic required to test this Flip-Flop using SW0 for D, PB0 for clock and LT0 to display the output Q.

During the lab, do the following:

b. use Quartus to enter the Verilog code and the test logic.
c. compile, test and debug the logic
d. when the logic is functioning properly, call the instructor to verify its operation.

'93A, 'LS93 . . . 4-Bit Binary Counters

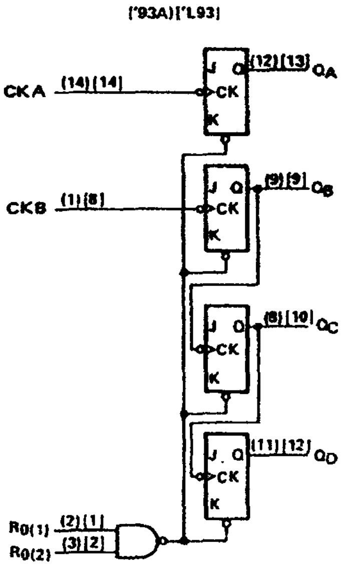
SN5493A, SN54LS93 . . . J OR W PACKAGE
SN7493 . . . N PACKAGE
SN74LS93 . . . D OR N PACKAGE
(TOP VIEW)



'93A, 'LS93
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

'93A, 'LS93



'92A, 'LS92, '93A, 'LS93
RESET/COUNT FUNCTION TABLE

RESET INPUTS		OUTPUT			
$R_{0(1)}$	$R_{0(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

NOTES: A. Output Q_A is connected to input CKB for BCD count.
B. Output Q_D is connected to input CKA for bi-quinary count.
C. Output Q_A is connected to input CKB.
D. H = high level, L = low level, X = irrelevant

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

LAB # 11

Registers

Purpose: Use of registers to transfer and store information

1. Storing a keypad number

   In preparation for the lab

   a. modify the logic diagram developed for lab 7 by adding a parallel register to store the value of a key that is depressed. Consider using the 74175 register shown in the attached data sheet.

   b. the signal valid signal (V) from the keypad decoder should be used as the clock for the register. It would be wise to debounce this signal.

   c. the output of the register shall be displayed on a 7-segment display using a 7-segment display decoder (7448).

   During the lab, do the following:

   d. use Quartus to enter the logic developed in parts a through c.

   e. compile, test and debug the logic

   f. when the logic is functioning properly, call the instructor to verify its operation.

2. Modes of operation of a shift register

   In preparation for the lab

   a. study the operation of a 74195 shift register. Examine the device input/output signals and the function table shown on the data sheet. The device has two modes of operation: parallel loading and serial shifting. Parallel loading is illustrated in the 2nd row of the function table. Right shifting is shown in the 5th and 6th rows of the table.

   b. prepare a logic diagram showing how the shift register can be controlled with:

      SW8 - mode switch 0 = parallel load, 1 = right shift
      SW0 through SW3 - data for parallel inputs A, B, C, D
      PB8 - when pressed will cause either a parallel load or serial right shift as specified by SW8. Why should this signal be debounced?

      The parallel outputs QA, QB, QC and QD of the register should be displayed on the lights LT0 through LT3.

   c. develop a plan for testing parallel loading followed by serial shifting of data.

   During the lab, do the following:

   d. use Quartus to enter the logic developed in parts a through c.

   e. compile, test and debug the logic

   f. when the logic is functioning properly, call the instructor to verify its operation.

3. Serial transfer of data between registers

   The information obtained in part 2 will be used to implement the transfer of data from 4 switches to 4 lights one-bit-at-a-time through serial transmission between registers. Two 74195 shift registers will be required: one to accept parallel data and transmit this data serialy and the other to receive the serial data and present this data through a parallel output.

In preparation for the lab

a. prepare a logic diagram showing how the two shift registers can be controlled with:

SW8 - mode switch 0 = parallel load, 1 = right shift
SW0 through SW3 - data for parallel inputs A, B, C, D
PB8 - when pressed will cause parallel load or serial shift as specified by SW8.

The parallel outputs QA, QB, QC and QD of the transmisison register should be displayed on the lights LT0 through LT3. The parallel outputs QA, QB, QC and QD of the receive register should be displayed on the lights LT4 through LT7.

b. in your design, carefully consider the fact that latching of data on the receiver should occur prior to a data shift on the transmitter. Keep in mind that a pulse has two edges: a positive edge and a negative edge!

c. develop a plan for testing parallel loading followed by serial shifting of data.

During the lab, do the following:

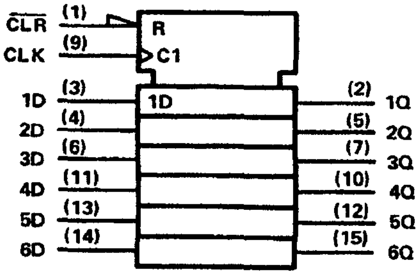
d. use Quartus to enter the logic developed in parts a through c.

e. compile, test and debug the logic

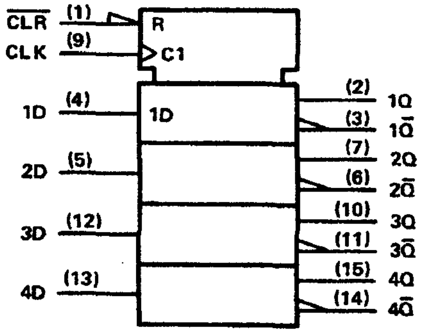
f. when the logic is functioning properly, call the instructor to verify its operation.

logic symbols†

'174



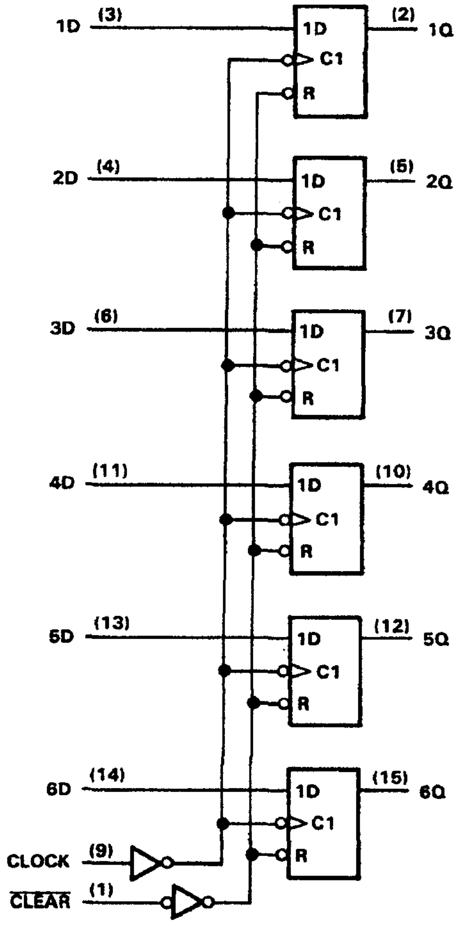
'175



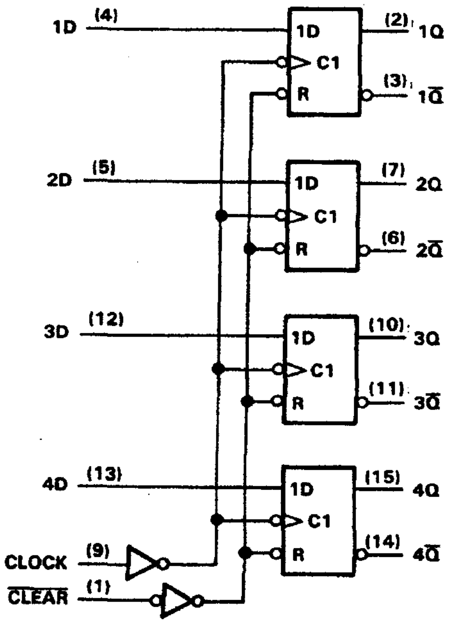
†These symbols are in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.
Pin numbers shown are for D, J, N, and W packages.

logic diagrams (positive logic)

'174, 'LS174, 'S174



'175, 'LS175, 'S175


1D (4) 1D C1 R (2) 1Q (3) 1Q̄
2D (5) 1D C1 R (7) 2Q (6) 2Q̄
3D (12) 1D C1 R (10) 3Q (11) 3Q̄
4D (13) 1D C1 R (15) 4Q (14) 4Q̄
CLOCK (9)
CLEAR (1)

Pin numbers shown are for D, J, N, and W packages.

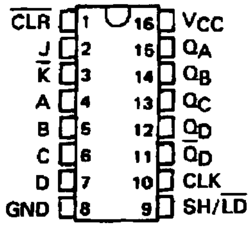
SDLS076

SN54195, SN54LS195A, SN54S195, SN74195, SN74LS195A, SN74S195 4-BIT PARALLEL-ACCESS SHIFT REGISTERS

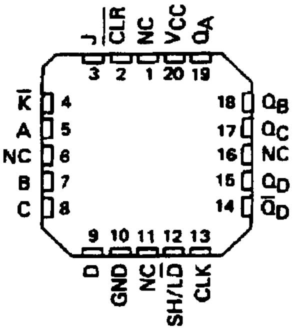
MARCH 1974—REVISED MARCH 1988

- **Synchronous Parallel Load**
- **Positive-Edge-Triggered Clocking**
- **Parallel Inputs and Outputs from Each Flip-Flop**
- **Direct Overriding Clear**
- **J and $\overline{K}$ Inputs to First Stage**
- **Complementary Outputs from Last Stage**
- **For Use in High Performance:**
  Accumulators/Processors
  Serial-to-Parallel, Parallel-to-Serial Converters

SN54195, SN54LS195A, SN54S195 . . . J OR W PACKAGE
SN74195 . . . N PACKAGE
SN74LS195A, SN74S195 . . . D OR N PACKAGE
(TOP VIEW)



SN54LS195, SN54S195 . . . FK PACKAGE
(TOP VIEW)



NC - No internal connection

description

These 4-bit registers feature parallel inputs, parallel outputs, J-$\overline{K}$ serial inputs, shift/load (SH/$\overline{LD}$) control input, and a direct overriding clear. All inputs are buffered to lower the input drive requirements. The register has two modes of operation:

Parallel (broadside) load
Shift (in the direction Q_A toward Q_D)

Parallel loading is accomplished by applying the four bits of data and taking SH/$\overline{LD}$ low. The data is loaded into the associated flip-flop and appears at the outputs after the positive transition of the clock input. During loading, serial data flow is inhibited.

Shifting is accomplished synchronously when SH/$\overline{LD}$ is high. Serial data for this mode is entered at the J-$\overline{K}$ inputs. These inputs permit the first stage to perform as a J-$\overline{K}$, D-, or T-type flip-flop as shown in the function table.

The high-performance 'S195, with a 105-megahertz typical maximum shift-frequency, is particularly attractive for very-high-speed data processing systems. In most cases existing systems can be upgraded merely by using this Schottky-clamped shift register.

TYPE	TYPICAL MAXIMUM CLOCK FREQUENCY	TYPICAL POWER DISSIPATION
'195	39 MHz	195 mW
'LS195A	39 MHz	70 mW
'S195	105 MHz	350 mW

FUNCTION TABLE

INPUTS									OUTPUTS				
CLEAR	SHIFT/ $\overline{LOAD}$	CLOCK	SERIAL		PARALLEL				Q_A	Q_B	Q_C	Q_D	$\overline{Q}_D$
			J	$\overline{K}$	A	B	C	D					
L	X	X	X	X	X	X	X	X	L	L	L	L	H
H	L	↑	X	X	a	b	c	d	a	b	c	d	$\overline{d}$
H	H	L	X	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}	$\overline{Q}_{D0}$
H	H	↑	L	H	X	X	X	X	Q_{A0}	Q_{A0}	Q_{Bn}	Q_{Cn}	$\overline{Q}_{Cn}$
H	H	↑	L	L	X	X	X	X	L	Q_{An}	Q_{Bn}	Q_{Cn}	$\overline{Q}_{Cn}$
H	H	↑	H	H	X	X	X	X	H	Q_{An}	Q_{Bn}	Q_{Cn}	$\overline{Q}_{Cn}$
H	H	↑	H	L	X	X	X	X	$\overline{Q}_{An}$	Q_{An}	Q_{Bn}	Q_{Cn}	$\overline{Q}_{Cn}$

H = high level (steady state)
L = low level (steady state)
X = irrelevant (any input, including transitions)
↑ = transition from low to high level
a, b, c, d = the level of steady-state input at A, B, C, or D, respectively
Q_{A0}, Q_{B0}, Q_{C0}, Q_{D0} = the level of Q_A, Q_B, Q_C, or Q_D, respectively, before the indicated steady-state input conditions were established
Q_{An}, Q_{Bn}, Q_{Cn} = the level of Q_A, Q_B, or Q_C, respectively, before the most-recent transition of the clock

Lab Project

Electronic Combination Lock

The lock shall accept three digits (0 - 9) entered through the keypad and compare them with three internally stored numbers. The logical signal OPEN shall become true if the numbers entered agree in order and value with those internally programmed. The lock shall be disabled if the numbers entered do not agree in value or order with those internally programmed.

Part 1 - Basic Lock

Inputs:

KP (number) - keypad for entering numbers (0 - 9)
PB9 (CLEAR) - for clearing the entry of numbers (only allowed after 1st or 2nd digit)
PB0 (RESEN) - restricted access reset enable

Outputs:

LT 0 (RESET) - reset state - no numbers have been entered
LT 1 (INUSE) - one or two values have been entered
LT 2 (OPEN) - the lock is open. The lock will remain open for only 3 seconds.
LT 3 (DISABL) - indicates that the lock has been disabled. The RESEN pushbutton must be on in order to enable the CLEAR pushbutton.
MS7 - most significant “Flex-digit” 7-segment display on the UP-1 board.
LS7 - least significant “Flex-digit” 7-segment display on the UP-1 board.

Specifications:

1. The lights LT0, LT1, LT2 or LT3 shall be used to indicate the status of the lock.Only one of the lights shall be on at any time.

2. Digits are entered through the keypad KP.After 3 digits are entered the lock must become either OPEN or DISABLED.

3. Display on the MS7 display the number of digits entered (0, 1 or 2). This display shall show 0 when the lock is in the RESET state.

4. When the correct combination (3 5 7) is entered, the light OPEN shall remain ON for 3 seconds after which the lock goes to RESET.

5. If less than three digits are entered, depression of the CLEAR button will cause the lock to go into its RESET state.

6. The lock shall become DISABLED when three digits are entered and at least one of them is incorrect.

7. A DISABLED lock can be reactivated by pressing the restricted access RESEN pushbutton and the CLEAR pushbutton simultaneously.

8. The lock shall timeout and RESET if the second and third digits are not entered within 5 seconds of the first digit.

Part 2 - Enhanced Lock

Design of this lock shall not take place until an implementation of the Basic Lock has been completed and shown to meet all specifications.

Additional Inputs:

PB4 (PRGEN) - enables entry of a new combination for the lock
PB5 (DISPL) - enables the combination of the lock to be displayed on LS7

Additional Outputs:

LT 4 (PROG) - combination being programmed
LT 5 (DISPL) - combination being displayed

Specifications:

1. When the lock is in the RESET state and the restricted access pushbutton PB4 (PRGEN) is depressed, light LT4 shall go on, light LT0 shall go off and three digits entered on the keypad shall be stored as the combination for the lock. After the third digit is entered the lock shall return to its reset state. If three digits are not entered within 5 seconds of PB4 being depressed, the lock shall return to the RESET state and the combination remains unchanged. MS7 shall indicate how many digits have been entered (0, 1 or 2).

2. When the lock is in the RESET state and the restricted access pushbutton PB5 (DISPL) is depressed, light LT5 shall go on, light LT0 shall go off and the combination of the lock shall be displayed on LS7 each digit for 1 second. MS7 shall display the sequence number of each digit (1, 2, 3). The lock shall then return to its reset state and the display shall go blank.

3. Operation of the Basic Lock, programming a new code and displaying the combination shall be designed as mutually exclusive functions that are enabled when the lock is in the RESET state. Possible conflicts shall be resolved in terms of the following priorities:

   DISPL > PROG > Basic Lock

   When the Basic Lock is being controlled and either PB4 or PB5 are pushed, the operation of the Basic Lock will continue. Pushbuttons PB4 and PB5 are only recognized when the lock is in the RESET state. Likewise, when a new combination is being programmed, a depression of PB5 shall be ignored.

CPEN 230

Project Development Cycle

Requirements

- Description of the purpose for the device
- Specifications - how well the device must perform
- Deliverables - what will be provided at the end of the project

Conceptual Design

- Functional Description of Operation
- Block Diagram
- Truth Table
- Algebraic equations
- State Diagram
- Timing Diagram

Detailed Design

- Schematic Diagram
- Wiring Diagram
- Parts List
- Layout Diagram

Test Procedure

- Test Plan
- Test Scripts

Documentation

- User's Manual
- Maintenance Manual

Appendices

Elenco Logic Trainer - Schematic Diagram

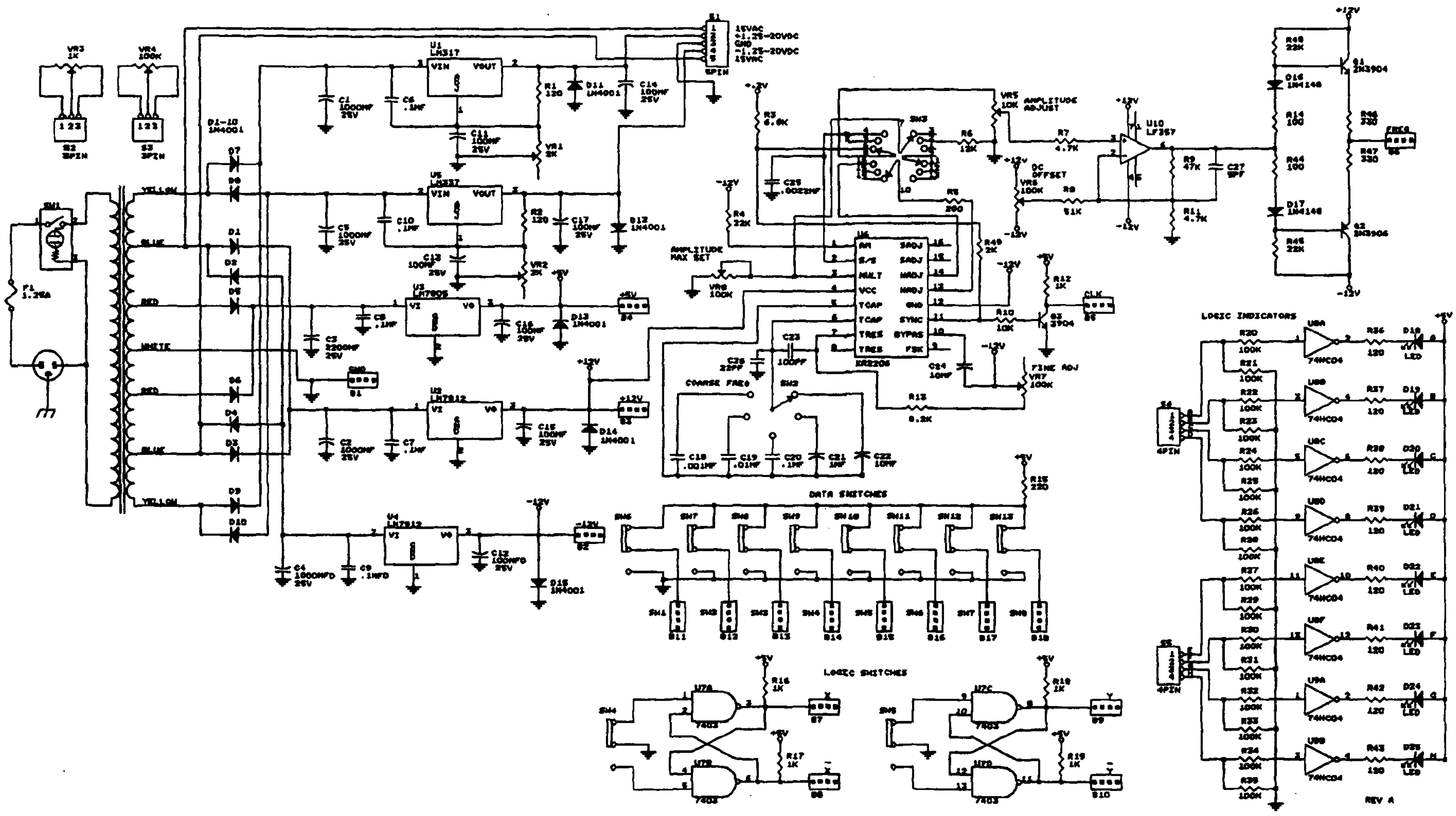
B & K Logic Probe - Schematic Diagram

Logic Gates used in the Lab

PLU Wiring Diagram

Altera’s UP 1 Pin Assignments

TIC DIAGRAM



VR3 1K
VR4 100K
S2 3PIN
S3 3PIN
1 2 3
SW1
F1 1.25A
YELLOW
BLUE
RED
WHITE
RED
BLUE
YELLOW
D1-10 1N4001
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
U1 LM317
VIN VOUT
U5 LM337
C1 1000MF 25V
C6 .1MF
C11 100MF 25V
R1 120
D11 1N4001
C14 100MF 25V
VR1 2K
C5 1000MF 25V
C10 .1MF
C13 100MF 25V
R2 120
C17 100MF 25V
D12 1N4001
VR2 2K
U3 LM7805
VI VO
C3 2200MF 25V
C8 .1MF
C16 100MF 25V
D13 1N4001
+5V
B4
GND
B1
U2 LM7812
C2 1000MF 25V
C7 .1MF
C15 100MF 25V
D14 1N4001
+12V
B3
U4 LM7912
C4 1000MFD 25V
C9 .1MFD
C12 100MFD 25V
D15 1N4001
-12V
B2
S1
15VAC
+1.25-20VDC
GND
-1.25-20VDC
15VAC
5PIN
R3 6.8K
C25 .0022MF
R4 22K
AMPLITUDE MAX SET
VR8 100K
SW3
R6 12K
R5 200
VR5 10K AMPLITUDE ADJUST
DC OFFSET
VR6 100K
R8 51K
R7 4.7K
U10 LF357
R9 47K
C27 5PF
R11 4.7K
U6
AM
S/S
MULT
VCC
TCAP
TCAP
TRES
TRES
SADJ
SADJ
WADJ
WADJ
GND
SYNC
BYPAS
FSK
XR2206
R49 2K
R10 10K
R12 1K
Q3 3904
CLK
B5
C23 100PF
C26 22PF
C24 10MF
FINE ADJ
VR7 100K
R13 8.2K
COARSE FREQ
SW2
C18 .001MF
C19 .01MF
C20 .1MF
C21 1MF
C22 10MF
R48 22K
Q1 2N3904
D16 1N4148
R14 100
R44 100
D17 1N4148
Q2 2N3906
R45 22K
R46 330
R47 330
FREQ
B6
DATA SWITCHES
R15 220
SW6
SW7
SW8
SW9
SW10
SW11
SW12
SW13
SW1
SW2
SW3
SW4
SW5
SW6
SW7
SW8
B11
B12
B13
B14
B15
B16
B17
B18
LOGIC SWITCHES
SW4
U7A
U7B
7403
R16 1K
R17 1K
X
B7
B8
SW5
U7C
U7D
R18 1K
R19 1K
Y
B9
B10
LOGIC INDICATORS
S4
S5
4PIN
R20 100K
R21 100K
R22 100K
R23 100K
R24 100K
R25 100K
R26 100K
R28 100K
R27 100K
R29 100K
R30 100K
R31 100K
R32 100K
R33 100K
R34 100K
R35 100K
U8A
U8B
U8C
U8D
U8E
U8F
U9A
U9B
74HC04
R36 120
R37 120
R38 120
R39 120
R40 120
R41 120
R42 120
R43 120
D18
D19
D20
D21
D22
D23
D24
D25
LED
A
B
C
D
E
F
G
H
REV A

SPECIFICATIONS

Maximum allowable supply voltage	±20V	
Max input voltage	±50V	
Free Air operating temperature	0–50°C	
Operating supply voltage range	TTL 5V	CMOS 5-15V
Input impedance	1MΩ	
Min. detectable pulse width	30ns	
Max input frequency displayable	17MHz min, 20MHz typical	
Logic thresholds	TTL	CMOS
Logic 1 (red LED lights)	2.3V±0.2V	70% Vcc
Logic 0 green LED lights	0.8V±0.2V	30% Vcc
Input overvoltage protection	±220V DC/AC 15 sec.	
Pulse LED (yellow) flashing time	500ms	
Power supply protection	±20V	
Size	205x18x16.5mm	
Weight	60g	

PARTS LIST

SCHEMATIC SYMBOL	DESCRIPTION	B&K-PRECISION PART NO.
Unlisted resistors are ±5%, 1/6W. See schematic diagram for value.		
C1	1000pF ±10%, 500V Ceramic	020-388-9-001
C4	.1µF+80/-20%, 50V, Ceramic	020-388-9-003
C5	120pF ±5%, 50V, Ceramic	020-388-9-004
C6	220pF ±10%, 50V, Ceramic	020-388-9-005
C7	.0047µF, 50V Ceramic	020-388-9-002
C8	.047µF ±20%, 50V, Ceramic	020-388-9-006
C9	.01µF, +80/-20%, 50V Ceramic	020-401-9-001
D1,2,4-6	IN4148 Diode	151-036-9-001
D3	1A, 600 PIV Diode	151-050-9-001
Q1,4	2N3904 NPN Transistor	176-084-9-001

PARTS LIST (Continued)

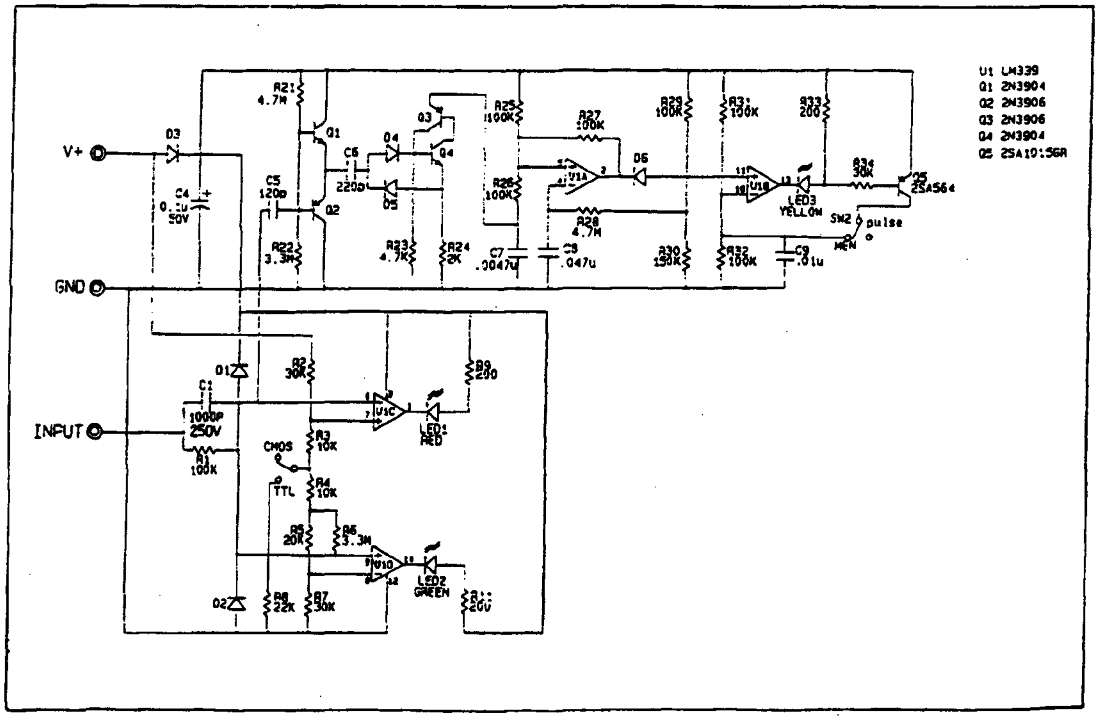
SCHEMATIC SYMBOL	DESCRIPTION	B&K-PRECISION PART NO.
Q2,3	2N3906 PNP Transistor	177-035-9-001
Q5	2SA1015(GR) PNP Transistor	177-057-9-001
R1	100kΩ±5%, 1/2W Carbon	002-102-5-104
SW1,2	Slide Switch	084-158-9-001
U1	LM339 Quad Comparator IC	307-424-9-001
	Red LED	158-077-9-001
	Yellow LED	158-077-9-002
	Green LED	158-077-9-003

U.S.A. PARTS ORDERING INFORMATION

There is a minimum charge for each invoice. Orders will be shipped C.O.D. unless previous open account arrangements have been made or remittance accompanies order. Advance remittance must cover handling, postage or express charges. Specify model and serial number when ordering replacement parts.

ORDER REPLACEMENT PARTS FROM:

B & K-Precision, Maxtec International Corp.
Factory Service Operations
6470 West Cortland Street
Chicago, Illinois 60635
Telephone #: (312) 889-1448
Telex #: 25-3475

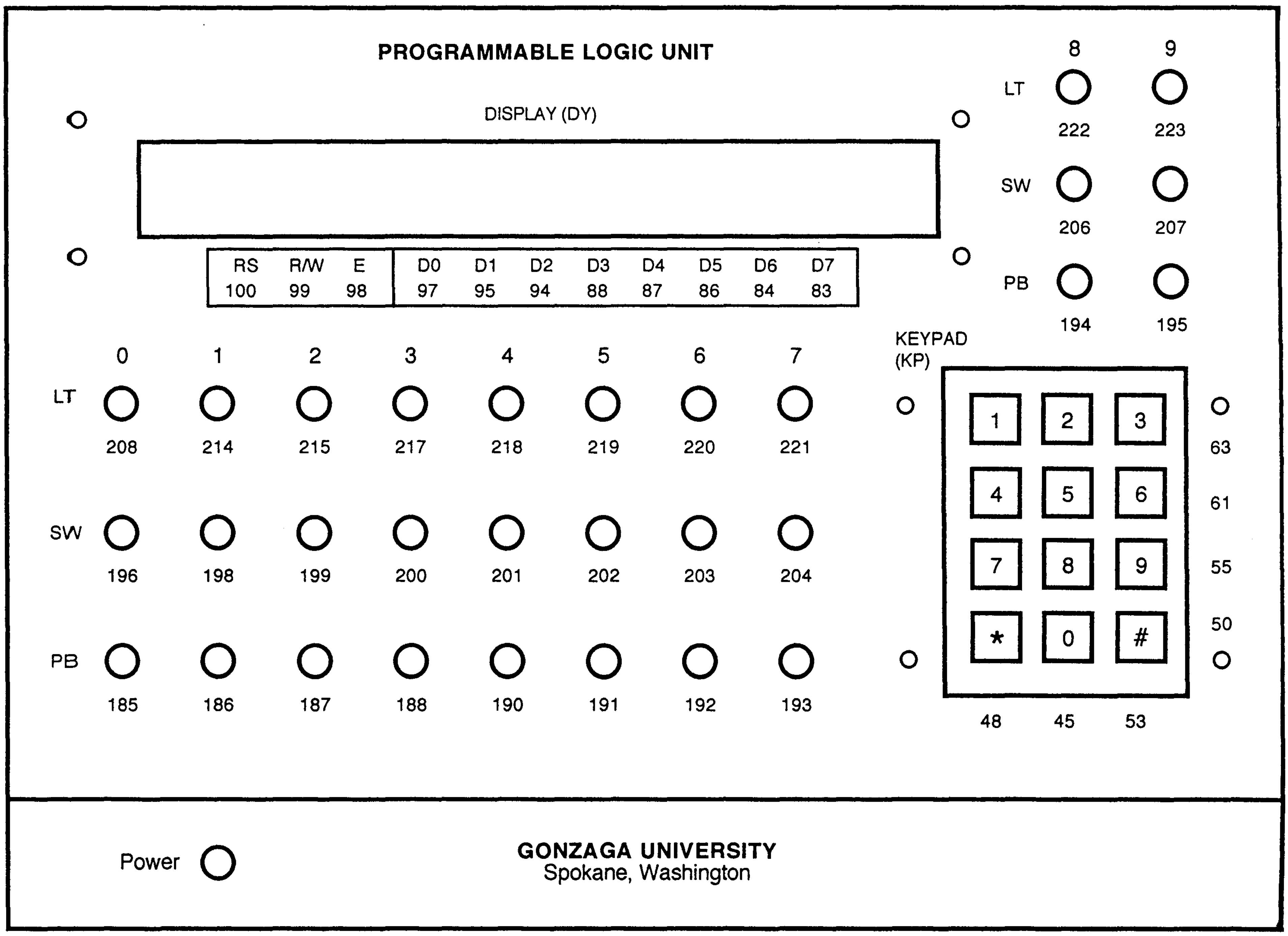


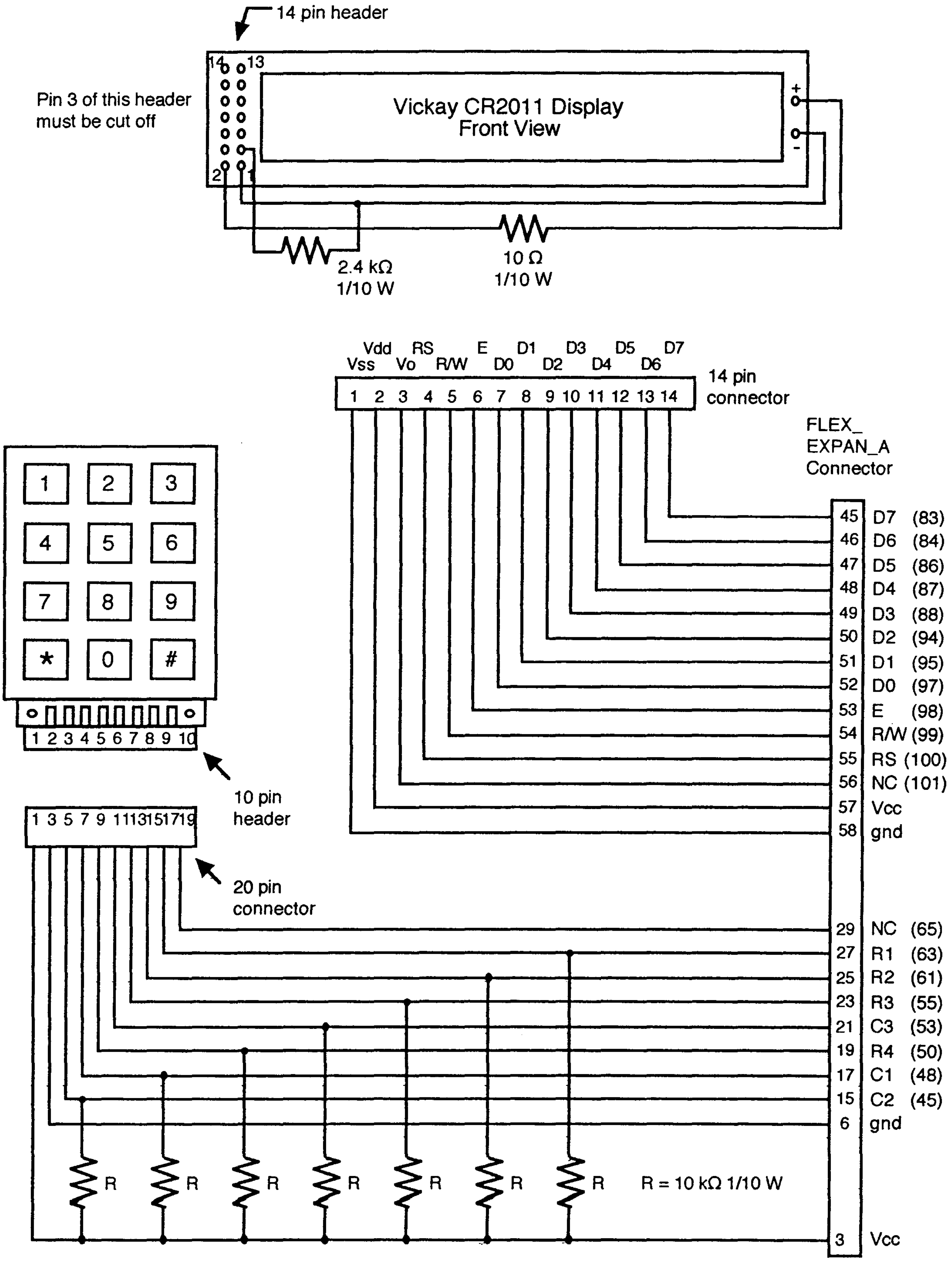
CPEN 230L

ICs Used in the Lab

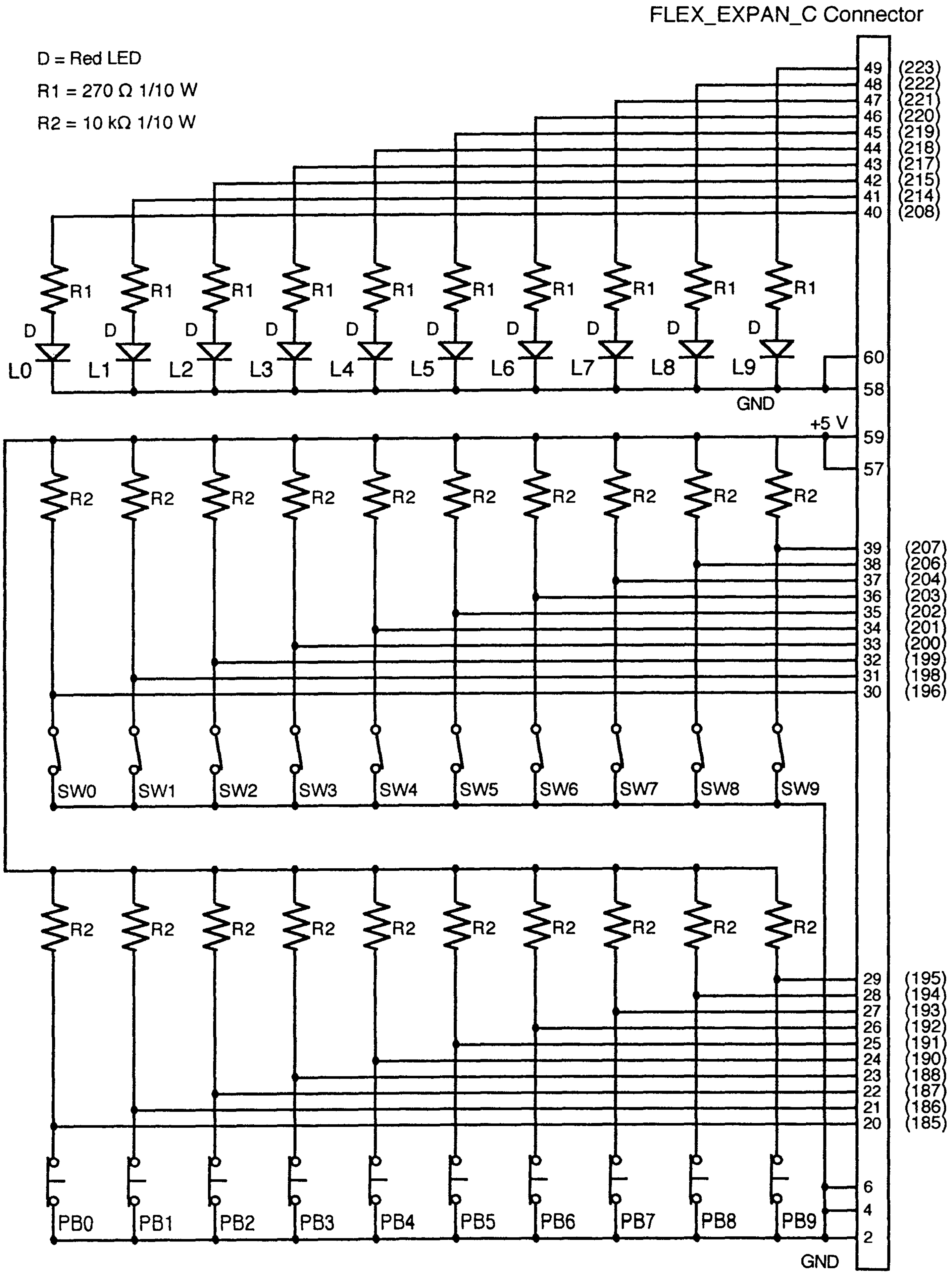
The following chips must be available in the Lab for experiments 1 through 4. All other experiments will be conducted with Programmable Logic Units in room HK 100. The IC are listed with the generic designation 74XX these parts can be replaced by similar parts with designation 74LSXX, 74FXX or 74HCXX.

7400
7402
7404
7408
7410
7420
7427
7432
7486


PROGRAMMABLE LOGIC UNIT
DISPLAY (DY)
RS 100
R/W 99
E 98
D0 97
D1 95
D2 94
D3 88
D4 87
D5 86
D6 84
D7 83
0 1 2 3 4 5 6 7
LT 208 214 215 217 218 219 220 221
SW 196 198 199 200 201 202 203 204
PB 185 186 187 188 190 191 192 193
8 9
LT 222 223
SW 206 207
PB 194 195
KEYPAD
(KP)
1 2 3
4 5 6
7 8 9
* 0 #
63
61
55
50
48 45 53
Power
GONZAGA UNIVERSITY
Spokane, Washington



14 pin header
14
13
2
1
+
-
Pin 3 of this header must be cut off
Vickay CR2011 Display
Front View
2.4 kΩ
1/10 W
10 Ω
1/10 W
Vdd
RS
E
D1
D3
D5
D7
Vss
Vo
R/W
D0
D2
D4
D6
1 2 3 4 5 6 7 8 9 10 11 12 13 14
14 pin connector
FLEX_ EXPAN_A Connector
45 D7 (83)
46 D6 (84)
47 D5 (86)
48 D4 (87)
49 D3 (88)
50 D2 (94)
51 D1 (95)
52 D0 (97)
53 E (98)
54 R/W (99)
55 RS (100)
56 NC (101)
57 Vcc
58 gnd
1 2 3
4 5 6
7 8 9
* 0 #
1 2 3 4 5 6 7 8 9 10
10 pin header
1 3 5 7 9 11 13 15 17 19
20 pin connector
29 NC (65)
27 R1 (63)
25 R2 (61)
23 R3 (55)
21 C3 (53)
19 R4 (50)
17 C1 (48)
15 C2 (45)
6 gnd
R
R
R
R
R
R
R
R = 10 kΩ 1/10 W
3 Vcc



D = Red LED

R1 = 270 Ω 1/10 W

R2 = 10 kΩ 1/10 W

ByteBlaster Parallel Port Download Cable

Designs can be easily and quickly downloaded into the UP 1 Education Board using the ByteBlaster download cable, which is a hardware interface to a standard parallel port. This cable channels programming or configuration data between the MAX+PLUS II software and the UP 1 Education Board. Because design changes are downloaded directly to the devices on the board, prototyping is easy and multiple design iterations can be accomplished in quick succession.

For more information on the ByteBlaster download cable, go to the *ByteBlaster Parallel Port Download Cable Data Sheet.*

UP 1 Education Board Description

The UP 1 Education Board contains the elements described in this section. Figure 1 shows a block diagram of the UP 1 Education Board.

Figure 1. UP 1 Education Board Block Diagram

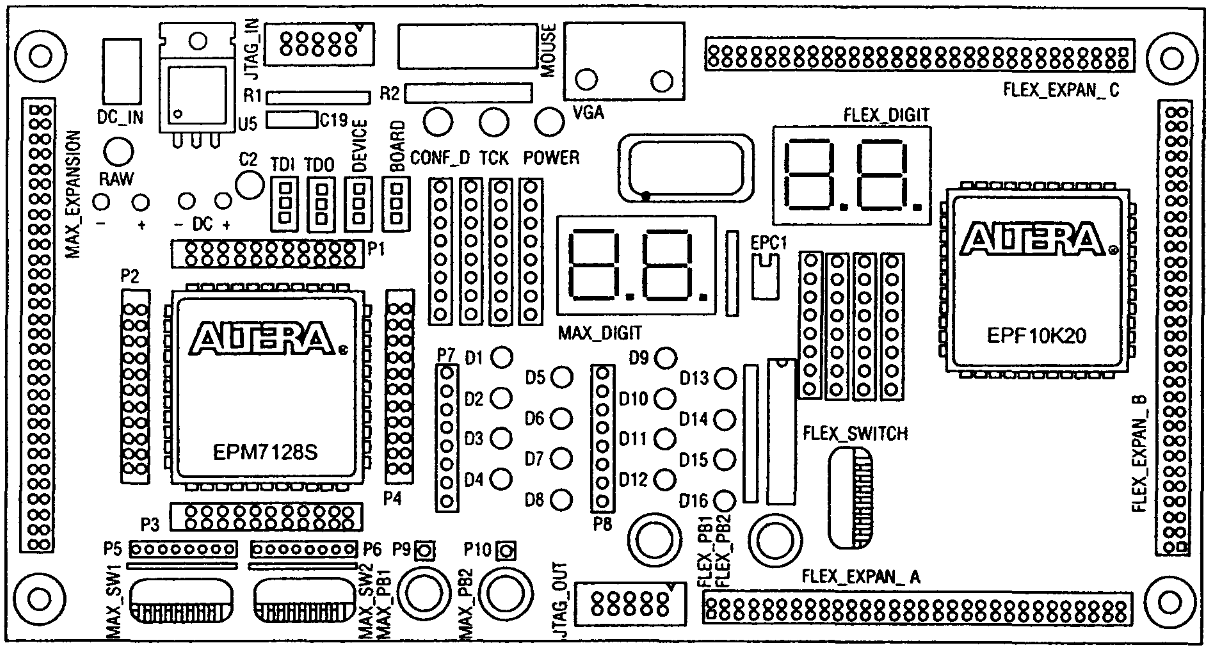


Table C.2 UP 1 Board 10K20RC240 FLEX CHIP I/O pin assignments.

Pin Name	Pin Type	Pin	Function of Pin
MSD_dp	OUTPUT PIN	14	Most Significant Digit of Seven-segment Display - Decimal Point Segment (0 = LED ON, 1 = LED OFF)
MSD_g	OUTPUT PIN	13	MSD Display Segment G (0 = LED ON,1 = LED OFF)
MSD_f	OUTPUT PIN	12	MSD Display Segment F (0 = LED ON, 1 = LED OFF)
MSD_e	OUTPUT PIN	11	MSD Display Segment E (0 = LED ON, 1 = LED OFF)
MSD_d	OUTPUT PIN	9	MSD Display Segment D (0 = LED ON, 1 = LED OFF)
MSD_c	OUTPUT PIN	8	MSD Display Segment C (0 = LED ON, 1 = LED OFF)
MSD_b	OUTPUT PIN	7	MSD Display Segment B (0 = LED ON, 1 = LED OFF)
MSD_a	OUTPUT PIN	6	MSD Display Segment A (0 = LED ON, 1 = LED OFF)
LSD_dp	OUTPUT PIN	25	Least Significant Digit of Seven-segment Display - Decimal Point Segment (0 = LED ON, 1 = LED OFF)
LSD_g	OUTPUT PIN	24	LSD Display Segment G (0 = LED ON, 1 = LED OFF)
LSD_f	OUTPUT PIN	23	LSD Display Segment F (0 = LED ON, 1 = LED OFF)
LSD_e	OUTPUT PIN	21	LSD Display Segment E (0 = LED ON, 1 = LED OFF)
LSD_d	OUTPUT PIN	20	LSD Display Segment D (0 = LED ON, 1 = LED OFF)
LSD_c	OUTPUT PIN	19	LSD Display Segment C (0 = LED ON, 1 = LED OFF)
LSD_b	OUTPUT PIN	18	LSD Display Segment B (0 = LED ON, 1 = LED OFF)
LSD_a	OUTPUT PIN	17	LSD Display Segment A (0 = LED ON, 1 = LED OFF)
FLEX_switch_1	INPUT PIN	41	FLEX DIP Switch Input 1 (1 = Open, 0 = Closed)
FLEX_switch_2	INPUT PIN	40	FLEX DIP Switch Input 2 (1 = Open, 0 = Closed)
FLEX_switch_3	INPUT PIN	39	FLEX DIP Switch Input 3 (1 = Open, 0 = Closed)
FLEX_switch_4	INPUT PIN	38	FLEX DIP Switch Input 4 (1 = Open, 0 = Closed)
FLEX_switch_5	INPUT PIN	36	FLEX DIP Switch Input 5 (1 = Open, 0 = Closed)
FLEX_switch_6	INPUT PIN	35	FLEX DIP Switch Input 6 (1 = Open, 0 = Closed)
FLEX_switch_7	INPUT PIN	34	FLEX DIP Switch Input 7 (1 = Open, 0 = Closed)
FLEX_switch_8	INPUT PIN	33	FLEX DIP Switch Input 8 (1 = Open, 0 = Closed)
PB1	INPUT PIN	28	Pushbutton 1 (non–debounced, 0 = button depressed)
PB2	INPUT PIN	29	Pushbutton 2 (non–debounced, 0 = button depressed)
Horiz_Sync	OUTPUT PIN	240	VGA Video Signal - Horizontal Synchronization
Vert_Sync	OUTPUT PIN	239	VGA Video Signal - Vertical Synchronization
Blue	OUTPUT PIN	238	VGA Video Signal - Blue Video Data
Green	OUTPUT PIN	237	VGA Video Signal - Green Video Data
Red	OUTPUT PIN	236	VGA Video Signal - Red Video Data
PS2_CLK	BIDIRECTIONAL	30	Clock line for PS/2 Mouse and Keyboard
PS2_DATA	BIDIRECTIONAL	31	Data line for PS/2 Mouse and Keyboard
Clock	INPUT PIN	91	25.175 MHz System Clock on low skew Global Clock Line