# CA-24 Solderless Prototyping Board Experimenter's Manual

August, 1981

Although great care has been taken in the preparation of this operations manual to insure the technical correctness, no responsibility is assumed by Ohio Scientific for any consequences resulting from the use of its contents. Nor does Ohio Scientific assume any responsibility for any infringements of patents or other rights of third parties which may result from its use.

If you discover misprints or errors please send a letter to the attention of:
Documentation Department, Ohio Scientific Inc., 1333 S. Chillicothe Road, Aurora, Ohio, 44202.

# PREFACE

The CA-24 experimenters board can be a useful learning device for the novice, or it can be a sophistocated tool for the person experienced in computer hardware. If you need an introduction to (or a review of) digital electronics, Section II of this manual will lead you through a series of experiments dealing with gates, flip-flops, and the monostable multivibrator. A computer is not needed for these experiments. If you are already experienced in electronics, Experiments 1, 2, and 3 will introduce you to the operation of the board which you then can use as a stand alone device for breadboarding simple to highly complex analog as well as digital circuits. In either case, of course, you should read Section I which contains a description of the various features of the board.

Section III of this manual contains a series of experiments which leads you through the use of the board as an interactive device with an OSI computer. For these experiments, you must have an OSI C4PMF, C4PDF, or C8PDF computer. (A C1PMF can be used if it has the 63Ø I/O board installed.) Both general use and specific applications are covered. Since the applications in the experiments are for the purpose of illustration rather than as ends in themselves, simplicity within each experiment is stressed.

When the CA-24 board is used with a computer, a certain amount of programming is necessary. Whenever possible, the experiments in this manual which involve interfacing with a computer include programs written in the computer language BASIC. This high level language facilitates convenient and rapid programming. In one experiment, which requires high resolution timing, assembly language routines (with their accompanying machine codes) have been used to provide the necessary speed and flexibility. These routines are explained in detail.

Once you have worked through the experiments, you should be able to use the CA-24 board creatively for a wide range of applications. If you find you need more background in electronics, a bibliography listing several introductory electronics and 65Ø2 assembly language textbooks and manuals is included in Appendix 2.

This manual has been produced by the cooperative effort of many people. The principal organization and writing of the manual in its present form was accomplished by Dr. Lawrence C. Becker, Professor of Physics at Hiram College. Most of the ideas for the experiments, however, were generated by Dr. Louis E. Roemer, Professor of Engineering at the University of Akron. The diagrams and figures (with the exception of Figures 3A and 3B) were put into final form by Craig Bair, a student at Hiram College. Much credit goes to Cookie Beck, the Physics Department secretary at Hiram College, who trained herself on the OSI word processor and put the text of the manual on disk storage. Her enthusiasm and optimism were an encouragement and her attention to accuracy and consistency caught numerous errors during the typing of the text. Julie Maston, a student worker in the secretarial office, helped with the disk storage process.

Careful and extensive proof reading and user testing were performed by Dr. Stewart Becker (retired Professor of Electrical Engineering at the University of Arizona); Dr. Darrell Turnidge, Dr. J. P. Neuzil, and Dr. Victor Nicholson (Associate Professors of Mathematics at Kent State University); Dr. Edward Carlson (Professor of Physics at Michigan State University); Dr. Michael Grajek (Associate Professor of Mathematics at Hiram College); Morgan Paul (student at Hiram College); and Paul Joviak (Systems Programmer at Ohio Scientific). Their corrections and suggestions have been valuable contributions to the final form of this manual.

Finally, the encouragement and patience of David Loynd (Product Support Manager at Ohio Scientific) who initiated the project and has overseen each phase of its development should not go unrecognized. His persistence and direction have successfully guided the project to its conclusion.

# TABLE OF CONTENTS

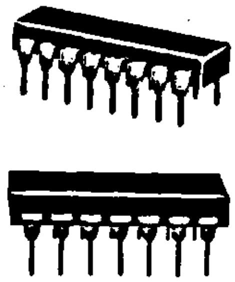
# PARTS AND TOOLS LIST

This manual is designed to be used with the Ohio Scientific CA-24 Solderless Interface Prototyping Board. Experiments 1 - 12 require only that the CA-24 board be connected to an adequate power supply. Section II, page 8 discusses the power supply requirements of the CA-24 board. Experiments 13 - 23 require that the CA-24 board be interfaced with either a C4P or C8P using a 16 pin ribbon cable. Section III, page 82 of this manual describes how to make this connection.

The following is a list of the components provided with this manual. You should identify each of these components before beginning to work through this manual.

IC Chips (The IC Chips provided with this manual have a black rectangular top approximately one or two inches long with a row of metal pins extending from both sides. The identification numbers on the IC chips are clearly marked on their top surface. Page 23 describes how to identify pin one on an IC chip.)

68B21 PIA Chip
MC685Ø ACIA Chip
74ØØ IC Chip
74Ø2 IC Chip
74Ø8 IC Chip
74121 IC Chip
74175 IC Chip
8T28 IC Chip
339A IC Chip



Resistors (The resistors provided with this manual are mounted in the center of wires approximately 2 1/2 inches long. The resistors themselves are approximately 1/4 inch long and 1/8 inch in diameter. The value of a resistor is specified by a pattern of color bands. There is a row of 32 resistors along the front edge of the CA-24 board.)

5 1 kilo Ohm resistors ( brown - black - red - gold )

4 2 kilo Ohm resistors ( red - black - red - gold )

1 33 kilo Ohm resistor ( orange - orange - orange - gold )

1Ø micro Farad capacitor (The metal cased capacitor is labeled 1Ø μF and has the + (or -) lead labeled.)

Jumper Wires (2Ø each)

Short wires (2 inches)
Medium wires (4 inches)
Long wires (8 inches)

5 kilo Ohm potentiometer (This is a duplicate of the part labeled R11 on the CA-24 board. See Fig. 1, page 2.)

DIP Switch (This is a block of 8 SPST switches. It is a duplicate of the parts labeled SW-A and SW-B on the CA-24 board. See Fig. 1, page 2.)

A small screwdriver is required for removing IC chips and adjusting the potentiometer. Experiment 23 (the last experiment) requires the use of a voltmeter. Experiment 3 describes how an (optional) oscilloscope can be attached to the CA-24 board if one is available. In some of the experiments you may find the use of a digital logic probe to be helpful in tracing the actual signals on the board. Inexpensive logic probes can be purchased at electronics stores for approximately $25.

# FIGURE LIST

# SECTION I

# DESCRIPTION AND USES OF THE CA-24 EXPERIMENTER'S BOARD

## STAND ALONE BREADBOARD

The CA-24 experimenter's board is a versatile device which includes active circuitry and a large breadboard area on which both digital and analog circuits can be constructed. The physical layout of the board is shown in Fig. 1 while Fig. 2 shows the connections between terminals on the under side of the breadboard area (strips A through G).

Terminal strips B, D, and F are intended for integrated circuit use. Each side of the strip, separated by the mid-strip indentation, consists of 5 interconnected terminals. There are 64 sets of these 5 interconnections on each side of the strip.

Strips C and E are used for power distribution. Each strip consists of terminals connected to 2 single conductors which run the length of the strip.

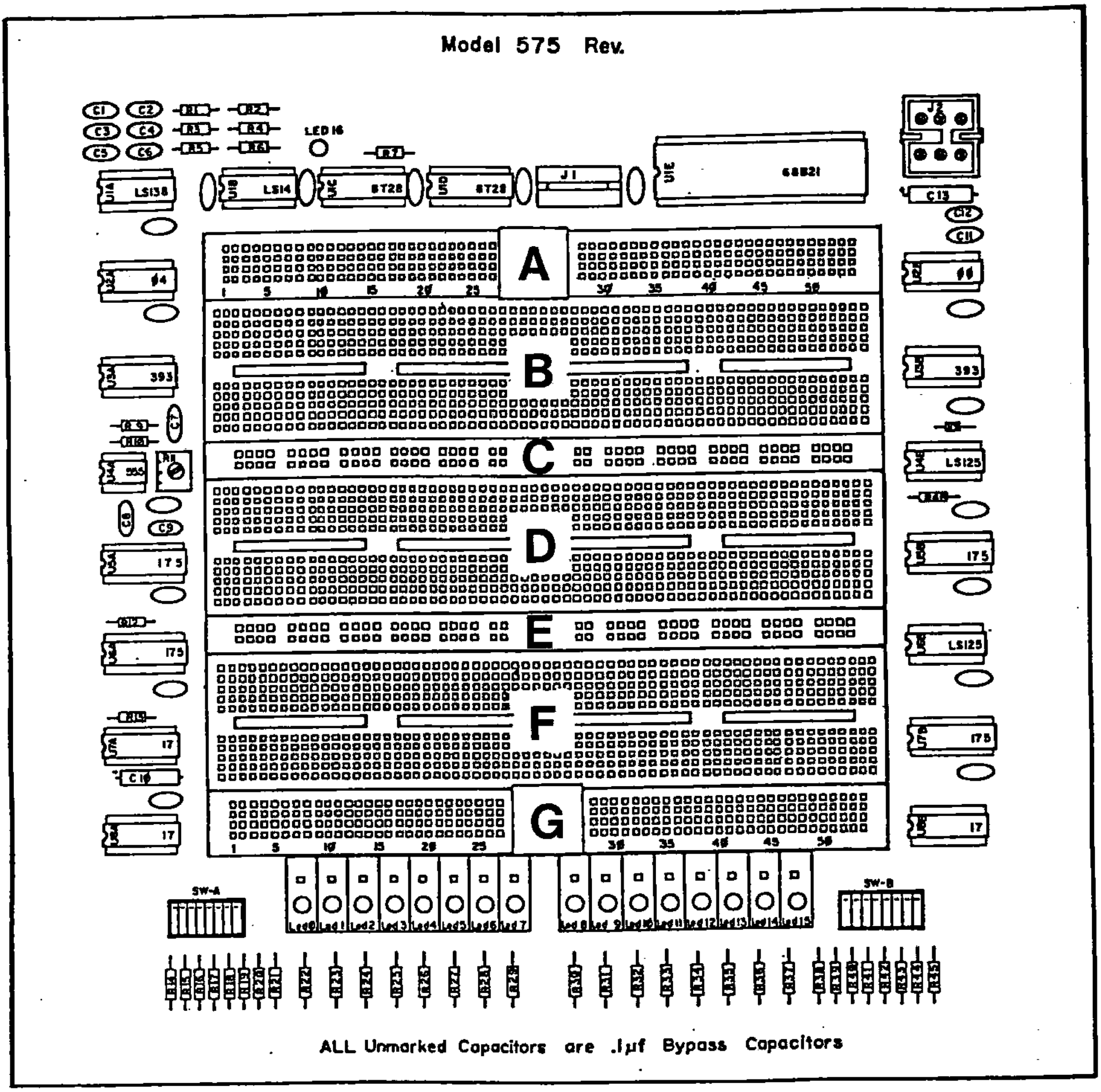
Terminals for signals from and to the active circuit elements are on strips A and G. Power for components which you will place on the breadboard is also available at these strips. Connections to the terminals should be made with single strand wire in the size range of AWG 2Ø to AWG 26. Wire size smaller then AWG 26 may not make good contact, and wire which is too large in diameter may damage the terminal housing. The recommended size is AWG 22.

The list of signals corresponding to the terminals on strips A and G is given in Table 1 and the schematic diagrams for the active circuit components are shown in Fig. 3a and 3b. On the far right of these diagrams are also the signal labels found in Table 1. Extra copies of Table 1 and Figures 3a and 3b have been included in Appendix 3. It will be helpful to remove these diagrams for easy reference as you work through the experiments in this manual.

When digital circuits are tested, logic levels (Ø V and +5 V) need to be available for input signals. Throughout this manual Ø V will be referred to as logic "Ø" or "FALSE" and +5 V will be referred to as logic "1" or "TRUE". These levels can be produced either by the set of 16 switches located at the left and right front of the board (SW-A and SW-B), or by a series of clock pulses generated by a 555 IC chip (left side of the board).

In addition to the clock whose frequency can be varied from 25 kHz to 7Ø kHz, a series of 16 divide-by-two circuits form a frequency reduction chain through which the base frequency of the clock can be reduced by factors of 2 raised to powers ranging from 1 to 16. The combination of the variable frequency clock and the divide-by-two chain allow for a continuous range of clock pulse rates from Ø.4 Hz to 7Ø kHz.

Rear



Front

Figure 1
Terminal Strip Labeling

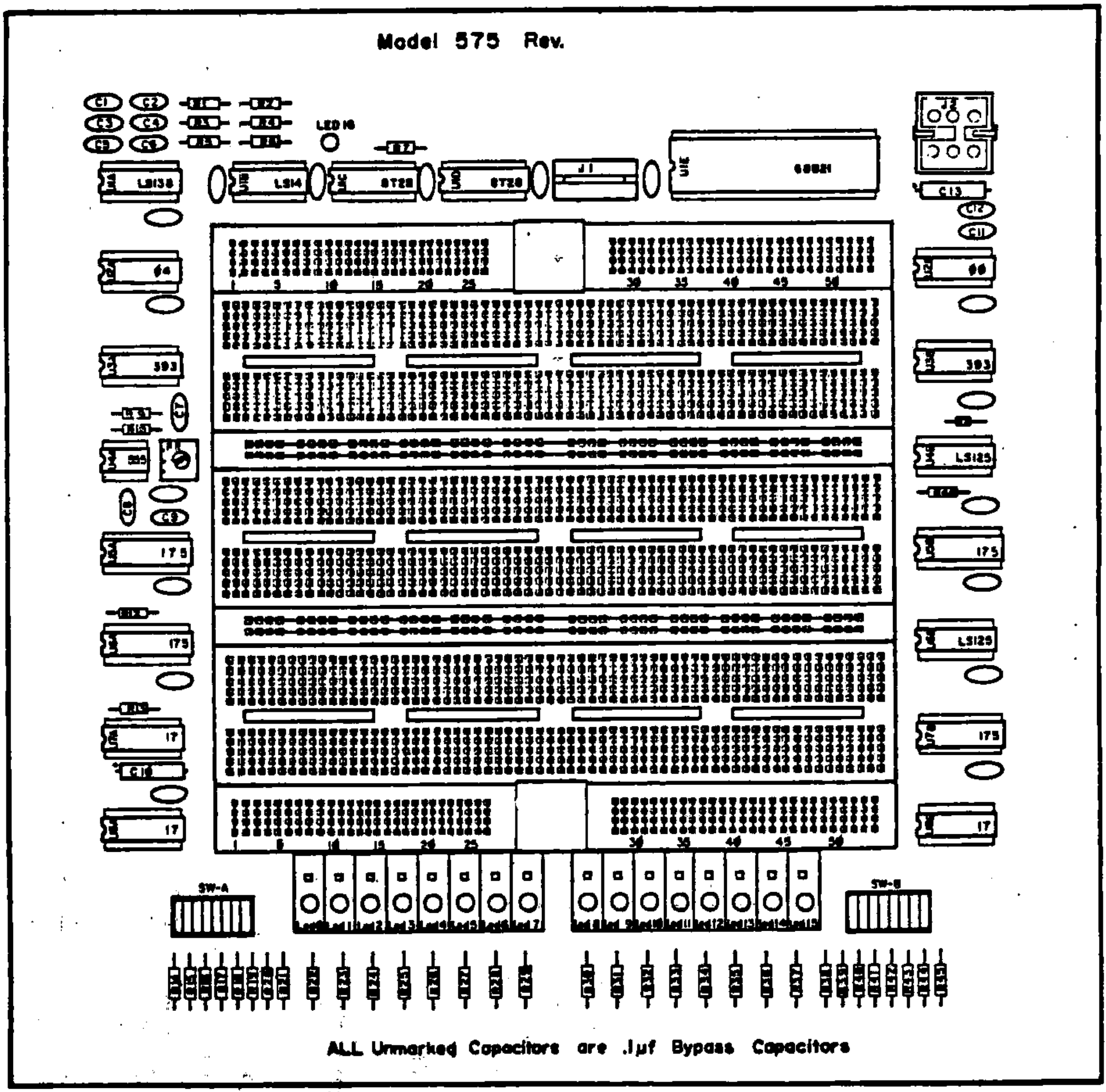


Figure 2
Under-Side Terminal Connections

TABLE 1  CA-24 On Board Terminal Strip Connections

| Terminal | STRIP(G) | Terminal | STRIP(A) |
|---|---|---|---|
| G1 | VCC (+5 V) | A1 | VCC (+5 V) |
| G2 | GND | A2 | GND |
| G3 | CLK4 | A3 | COUT |
| G4 | CLK8 | A4 | CIN |
| G5 | CLK7 | A5 | CLK1 |
| G6 | CLK6 | A6 | CLK2 |
| G7 | CLK5 | A7 | CLK3 |
| G8 | GND | A8 | N.C. |
| G9 | SW1A | A9 | N.C. |
| G1Ø | SW2A | A1Ø | N.C. |
| G11 | SW3A | A11 | N.C. |
| G12 | SW4A | A12 | AØ |
| G13 | SW5A | A13 | A1 |
| G14 | SW6A | A14 | $\overline{\text{POR}}$ |
| G15 | SW7A | A15 | DØ |
| G16 | SW8A | A16 | D1 |
| G17 | LODØ | A17 | D2 |
| G18 | LOD1 | A18 | D3 |
| G19 | LOD2 | A19 | D4 |
| G2Ø | LOD3 | A2Ø | D5 |
| G21 | LOD4 | A21 | D6 |
| G22 | LOD5 | A22 | D7 |
| G23 | LOD6 | A23 | Φ2·VMA |
| G24 | LOD7 | A24 | LA3 |
| G25 | $\overline{\text{EXTDW}}$ | A25 | $\overline{\text{EXTDE}}$ |
| G26 | CLK16 | A26 | LA2 |
| G27 | VCC | A27 | R/$\overline{\text{W}}$ |
| G28 | $\overline{\text{Φ2·VMA}}$ | A28 | CB1 |
| G29 | LOAD | A29 | CB2 |
| G3Ø | LIDØ | A3Ø | PBØ |
| G31 | LID1 | A31 | PB1 |
| G32 | LID2 | A32 | PB2 |
| G33 | LID3 | A33 | PB3 |
| G34 | LID4 | A34 | PB4 |
| G35 | LID5 | A35 | PB5 |
| G36 | LID6 | A36 | PB6 |
| G37 | LID7 | A37 | PB7 |
| G38 | $\overline{\text{A1}}$ | A38 | CA1 |
| G39 | $\overline{\text{EXTDR}}$ | A39 | CA2 |
| G4Ø | SW1B | A4Ø | PAØ |
| G41 | SW2B | A41 | PA1 |
| G42 | SW3B | A42 | PA2 |
| G43 | SW4B | A43 | PA3 |
| G44 | SW5B | A44 | PA4 |
| G45 | SW6B | A45 | PA5 |
| G46 | SW7B | A46 | PA6 |
| G47 | SW8B | A47 | PA7 |
| G48 | CLK16 | A48 | CLK9 |
| G49 | CLK15 | A49 | CLK1Ø |
| G5Ø | CLK14 | A5Ø | CLK11 |
| G51 | CLK13 | A51 | CLK12 |
| G52 | -V (-12 V or -15 V) | A52 | +V (+12 V or +15 V) |
| G53 | GND | A53 | GND |
| G54 | VCC (+5 V) | A54 | VCC (+5 V) |

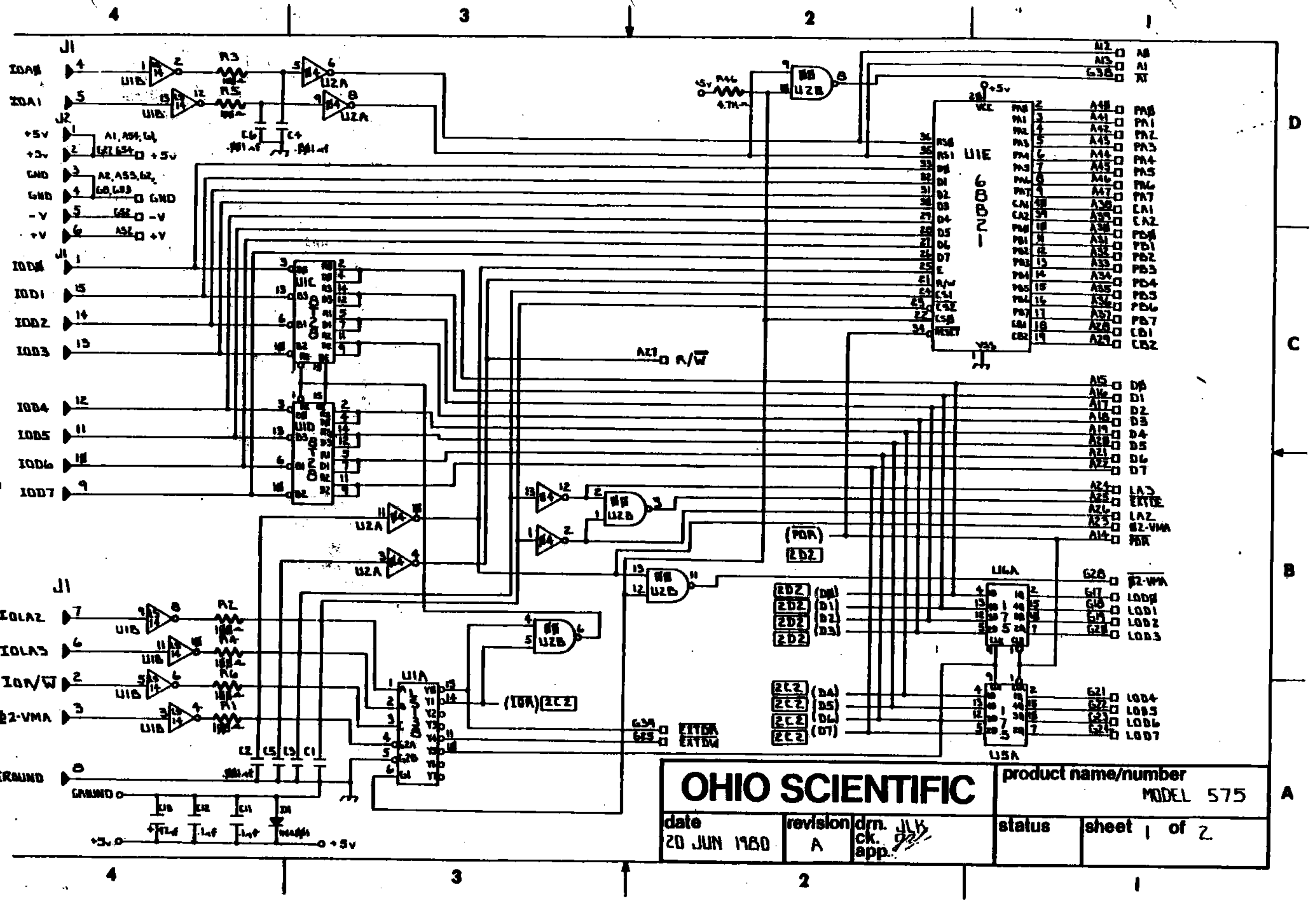


Figure 3a
CA-24 (575) Board Schematic

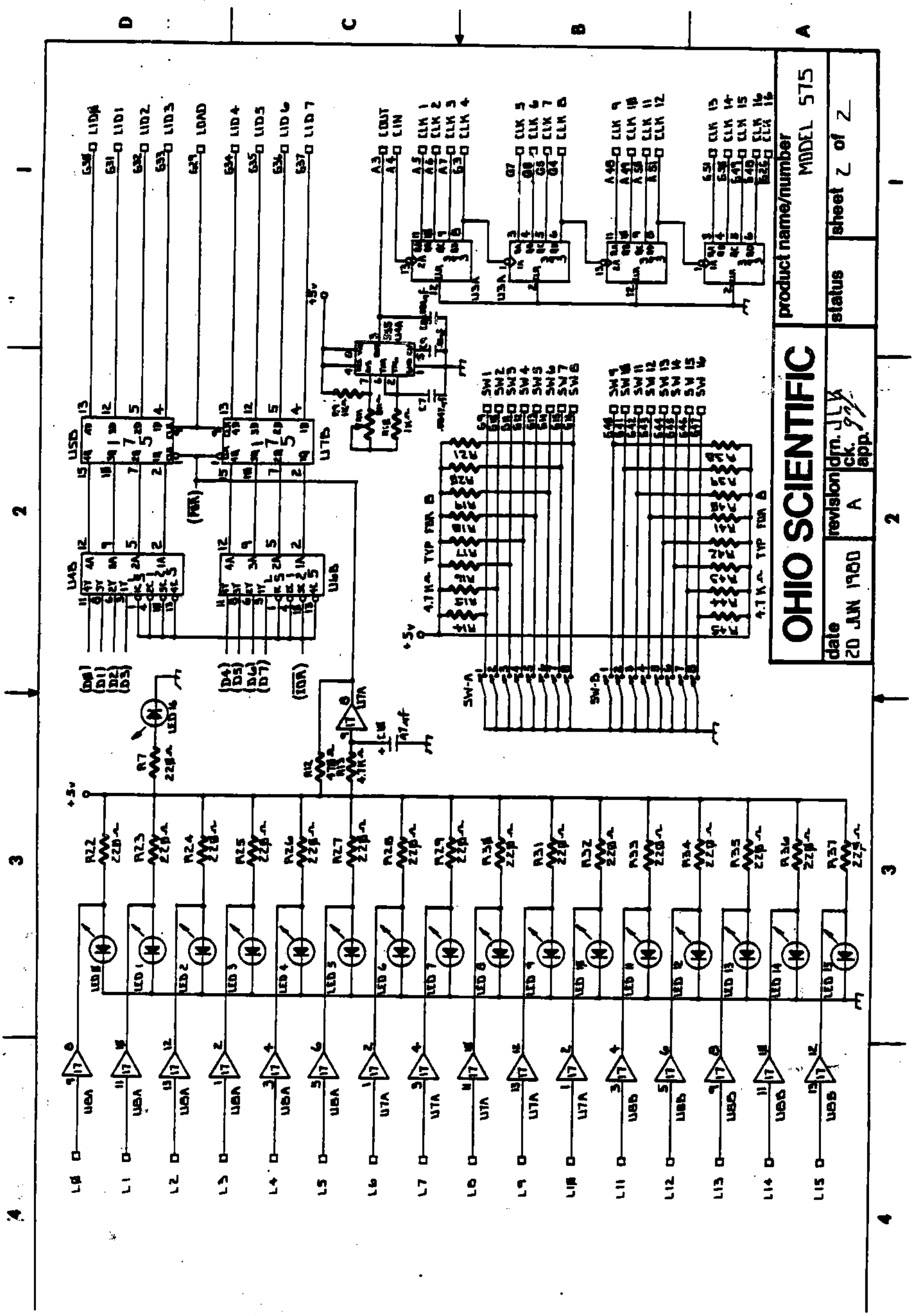


Figure 3b
CA-24 (575) Board Schematic

For digital circuits, it is frequently necessary to display logic levels both for input signals and for output signals. There are 16 light emitting diodes (LED's) for this purpose mounted at the front of the board. In Fig. 1, they are marked LEDØ through LED15.

If analog circuits are constructed on the board, they can be powered from terminals on strips A and G (provided that the proper power supply has been connected to the board). Varying voltage input signals, however, need to be generated by some device external to the board, and the display of the input and output signals must be done on instruments external to the board.

The CA-24 experimenter's board used as a stand-alone device is an excellent tool for the hobbyist or for the person working in electronic circuit design. On it, trial circuits can easily be constructed and evaluated. Furthermore, the board is an excellent teaching device. Experiments in this manual form a nucleus of material by which a person can learn about digital circuit performance. In addition, an educator can design experiments which suit the purposes of a course which includes basic electronic circuitry.

## COMPUTER INTERFACED BREADBOARD

More advanced circuits can take advantage of the computer interface circuity which resides on the CA-24 experimenter's board. The interface components include a 3-to-8 line decoder/multiplexer for processing control signals, bi-directional latches (or bus transceivers) for transferring information back and forth between the CA-24 board and a computer, TTL (Transistor-Transistor Logic) latches for transferring signals back and forth between the board and external devices, and a PIA (Peripheral Interface Adapter) which permits a computer to have a broad range of control over peripheral devices.

The interfacing feature of the CA-24 board will also be useful for the hobbyist, the person working in circuit design, and the educator. Additional interfacing components can be added to the breadboard and circuits built around them evaluated. The experiments in the latter portion of the manual introduce the user to interfacing techniques and give some practical examples.

We expect you will find your CA-24 board to be an interesting and worthwhile investment.

# SECTION II

## EXPERIMENTS USING ONLY THE CA-24 BOARD AND POWER SUPPLY

### OVERVIEW AND POWER SUPPLY CONNECTION

The following twelve experiments can be worked through using only the CA-24 board (with power supply attached) and the components selected from those supplied with the board. The experiments are designed to familiarize you with with the basic functions of the board and with some relevant digital electronic circuits. Experiments 10, 11, and 12 (particularly 10 and 11) will help you to understand the function of IC chips on the board which allow it to interact with a computer.

For the most part, the experiments are in order of increasing complexity. If you have some familiarity with digital electronics and integrated circuits, you may wish to work through Experiments 1, 2, and 3 and then jump ahead to the point where you feel the experiments would be of value.

In order to use the CA-24 board, it is necessary to connect a power supply to the board. This is done by way of connector J2 in the rear right corner of the board (Amphenol type 1-480270-0 with pins type 60619-1). The pin connections are shown in Fig. 4. Be certain that the power supply is off when the connection is made.

The voltage which is necessary for the operation of the digital circuit components mounted on the board is +5 V. All of the permanent circuit components of the board are powered by this voltage. In order to have sufficient current capability to meet the requirements of components used in various experiments throughout the manual, the power supply must be able to deliver about 1 Ampere.

If you should have occasion to use the breadboard area for constructing analog circuits, you will need also the +12 V (or +15 V) and the -12 V (or -15 V) power. Current requirements probably would not be greater than 100 mA. These voltages, however, are not necessary for the experiments which follow.

When you are certain that the appropriate power supply has been correctly connected to the CA-24 board, proceed to Experiment 1.

## CA-24 POWER CONNECTION and DISTRIBUTION

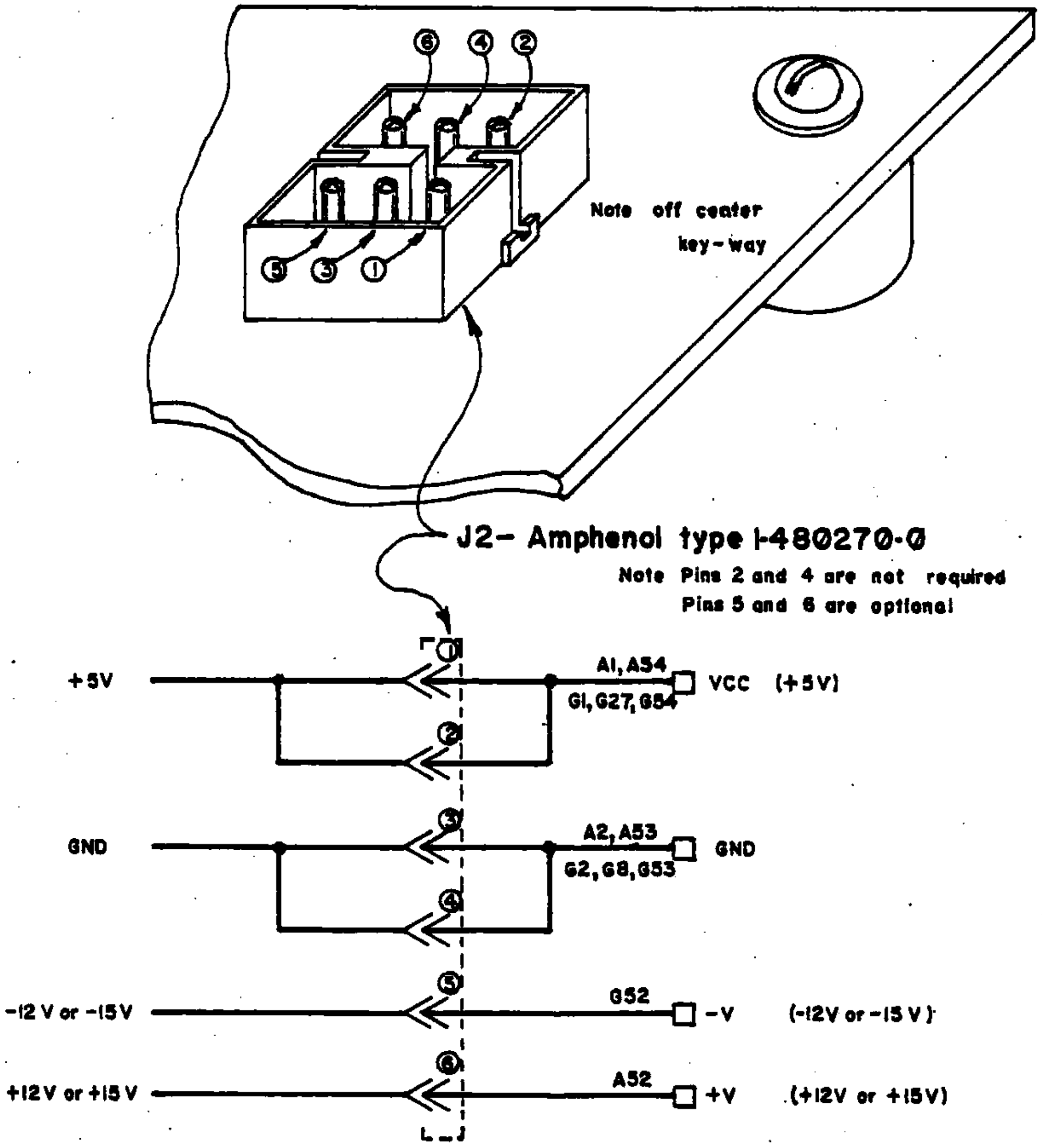


Figure 4
CA-24 (575) Board Power Connection and Distribution

EXPERIMENT 1

TITLE

LED Indicators

PURPOSE

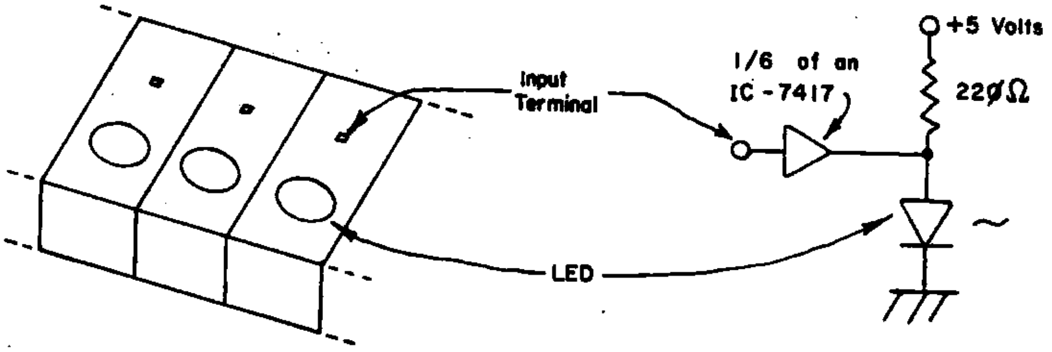
The purpose of this experiment is to investigate the function of the seventeen LED (light emitting diode) indicators on the CA-24 board.

EQUIPMENT

CA-24 board and power supply
One short (about 2 inch) jumper wire (22 gauge)

DISCUSSION

The light emitting diode (LED) is an inexpensive indicator or display device. On the CA-24 board, a single LED functions as a "power-on" indicator. It is located at the rear (power cable end) of the board and is marked LED16 in Fig. 5. Sixteen additional LED's are in two groups of eight at the front of the board and are marked LEDØ through LED15 in Fig. 5. These sixteen additional indicators are pre-wired as shown below.



Shown also is a sketch of three LED housings. Each 22Ø Ohm resistor is located in front of the LED to which it is wired. The resistors are labeled R22 through R37 on Fig. 5. The triangle symbol marked "one-sixth of an IC-7417" represents an identity element (same output as input) which acts as a buffer between the LED and a signal connected to it. The identity element is most commonly called a buffer or a driver. There are three 7417 chips on the CA-24 board. They are marked U7A, U8A, and U8B in Fig. 5. Each 7417 chip contains six identity elements or buffers.

Each LED will require a maximum current of 2Ø mA. When all of them are on, the power requirement is 17 times 2Ø mA or about 1/3 Ampere. This heavy power consumption (relative to other circuits on the board) should be kept in mind when a power supply for the CA-24 board is being considered and is the principal reason a 1 Ampere supply has been recommended.

PROCEDURE

Be certain that a 5 Volt power supply of sufficient current capability (about 1 A) has been connected to J2 of the CA-24 board as described earlier in this section. Turn on the power supply. Note that all seventeen LED's will glow. The indicators at the front of the board will be on because a floating input (no connection made to the input) will be equivalent to a +5 Volt signal (commonly referred to as logic "1" or high).

Insert one end of a short jumper wire into the input terminal hole of the LEDØ housing. Insert the other end into terminal G1 (one of the four holes at the far left marked 1 on the G terminal strip). The connection with the jumper wire is shown pictorially in Fig. 5. Reference to the terminal strip connections in Table 1 of SECTION I shows that G1 is +5 V. It will be noted that the LED remains on.

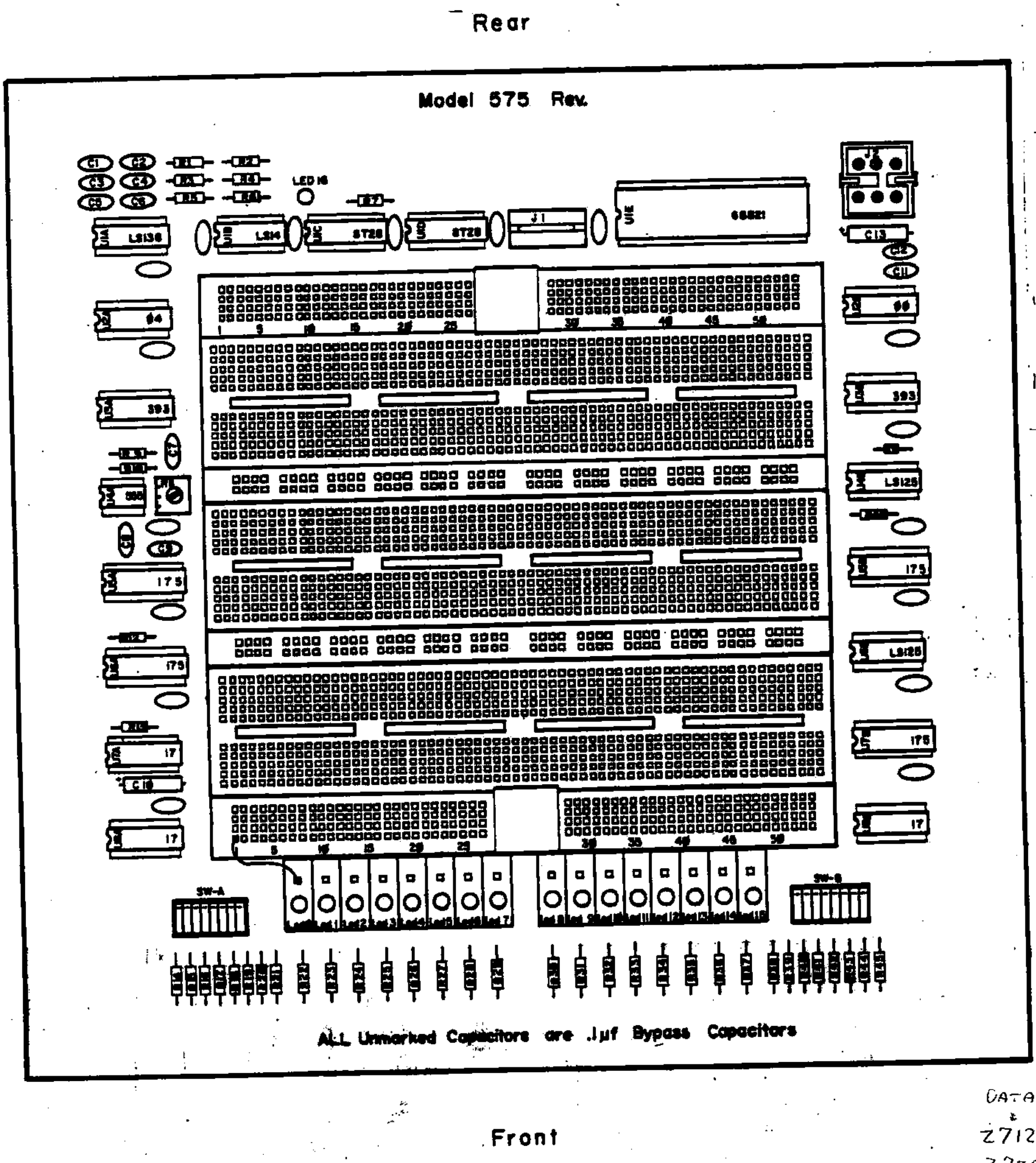
Recall that Appendix 3 contains extra copies of Table 1 and Figures 3a and 3b. It will be helpful to remove these diagrams for easy reference as you work through the experiments in this manual.

Next, shift the jumper wire from G1 to G2 as shown in Fig. 6. Table 1 shows that G2 is ground (GND) or Ø Volts. Note that the LED turns off indicating a Ø Volt signal at its input (commonly referred to as logic "Ø" or low).

Can you find other GND or Ø V terminals on strip G? (Refer to Table 1. Do not just start poking the jumper wire into holes.) Try other combinations of LED's and GND terminals on strip G.

Through this experiment you have found that the LED can be used as an indicator for logic levels. When the input to the LED is +5 Volts (called a logic "1" or "true"), the light is on. When the input is Ø Volts (called a logic "Ø" or "false"), the light is off. This simple indicator will serve as the basis for most of our display functions.

TERRY -
0131TW



.75
3.58
1.58
.19
.19
.39
2.58
10.50
.29
.24
13.90
5.95
40.14

4

3-17✓
4-175✓
2-393✓
1-00✓
1-04✓
1-LS14✓
2-8T28✓
3-68821✓
1-5406✓
1-7408✓

DATA
27128
27256

Figure 5
Logic "1" (+5 Volts) Connected to LEDØ

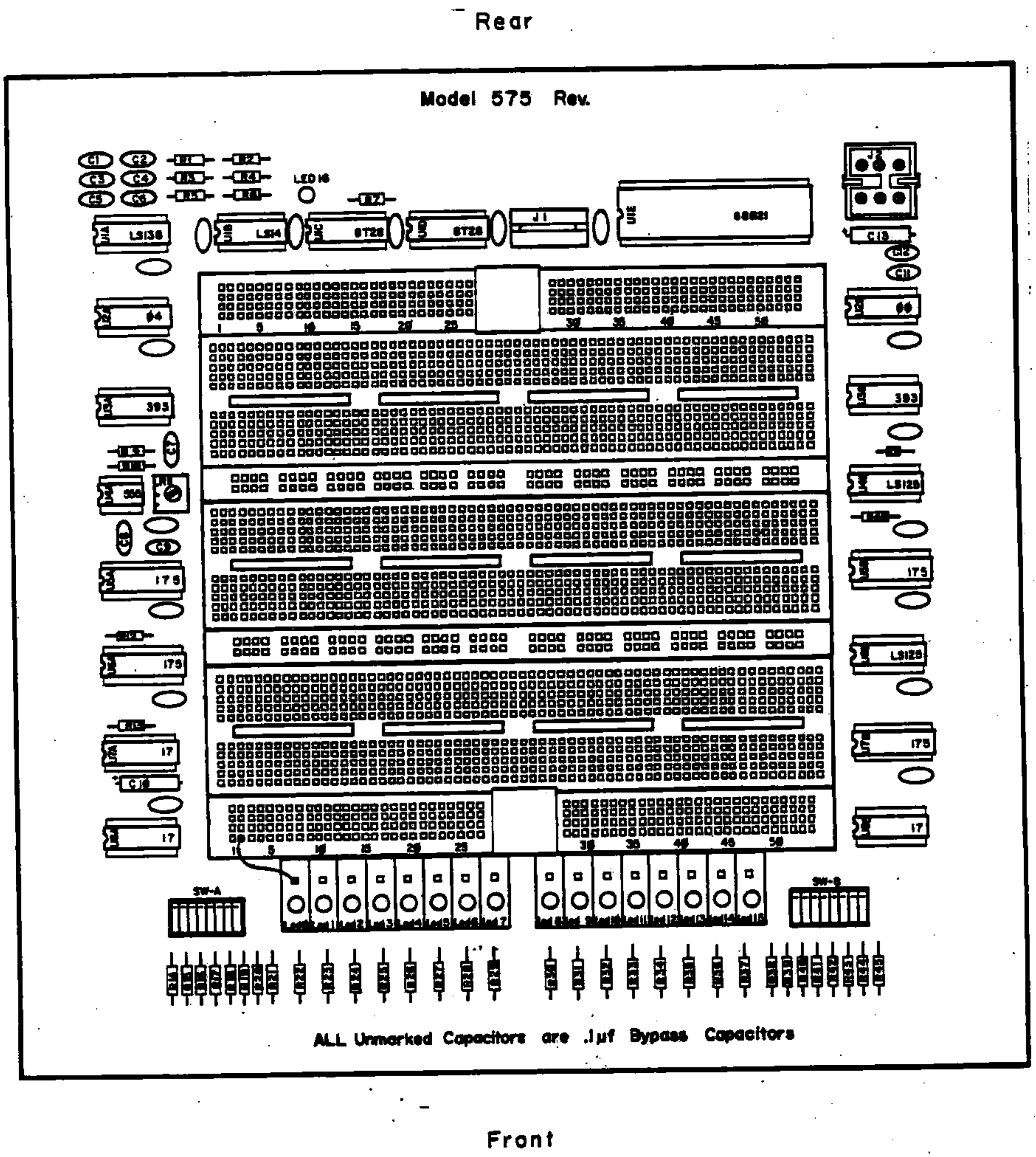


Figure 6
Logic "Ø" (Ø Volts) Connected to LEDØ

# EXPERIMENT 2

## TITLE

Logic Level Switches

## PURPOSE

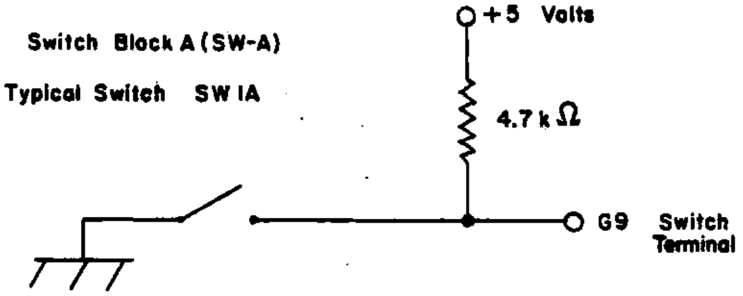
The purpose of this experiment is to investigate the use of the sixteen switches on the CA-24 board.

## EQUIPMENT

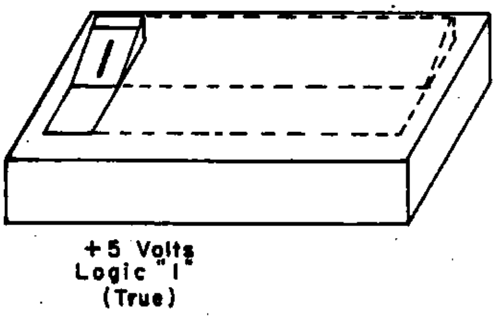
CA-24 board and power supply
One short jumper wire

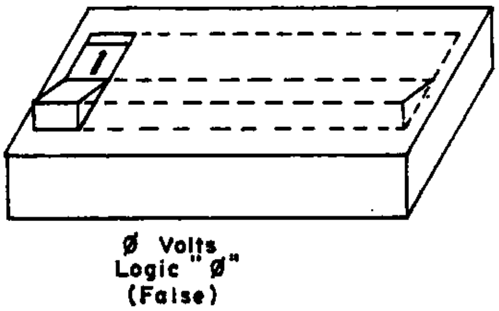
## DISCUSSION

There are 16 single-pole-single-throw (SPST) switches on the CA-24 board (see front left and right corners of Fig. 7) which can be used to set voltage (also called logic) levels. Each switch is pre-wired as shown below for switch 1A (SW1A) in switch block A (SW-A).



The diagrams below show the switch positions for the two output or logic level conditions.

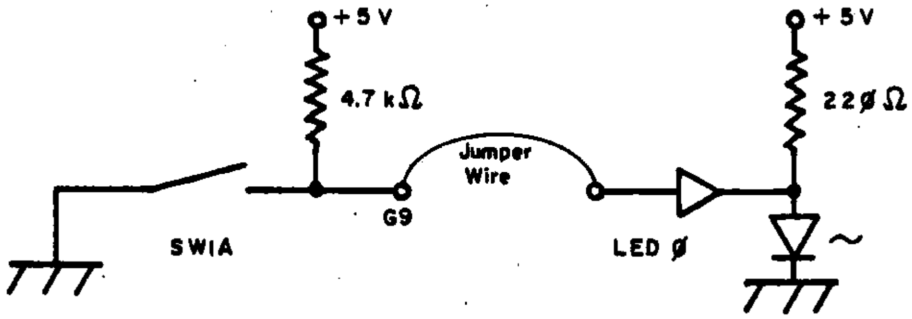




Note that when the rear end of the switch is up, the switch produces a +5 Volt (logic "1" or "true") signal. When the rear end is down, the switch signal is Ø Volts (logic "Ø" or "false"). These signals appear at terminals G9 through G16 for switch block A (the left set) and G4Ø through G47 for switch block B. (See Table 1 in SECTION I.)

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Connect SW1A to LEDØ as shown schematically below.



This is done by inserting one end of the jumper wire into the terminal hole on the LEDØ housing and the other end into one of the four G9 terminal holes. This connection is pictured in Fig. 7. Now turn on the power supply.

Move the far left switch in switch block A (SW1A) back and forth. Note that LEDØ turns on and off depending on the position of the switch. Now move the jumper wire from G9 to G1Ø. Note that now SW2A controls LEDØ. Try various combinations of switches and LED's to be certain you understand how the switches can be used.

In future experiments we will use switches instead of changing jumper wires to change an input signal. This method will be more convenient and it will allow us to tell at a glance which signals are logic "Ø" and which are logic "1".

In some circuits, the resistance between the power supply and the switch terminal is important. For those circuits, we must take the internal 4.7 kΩ resistance into account. We shall point out these critical cases as they arise in future experiments.

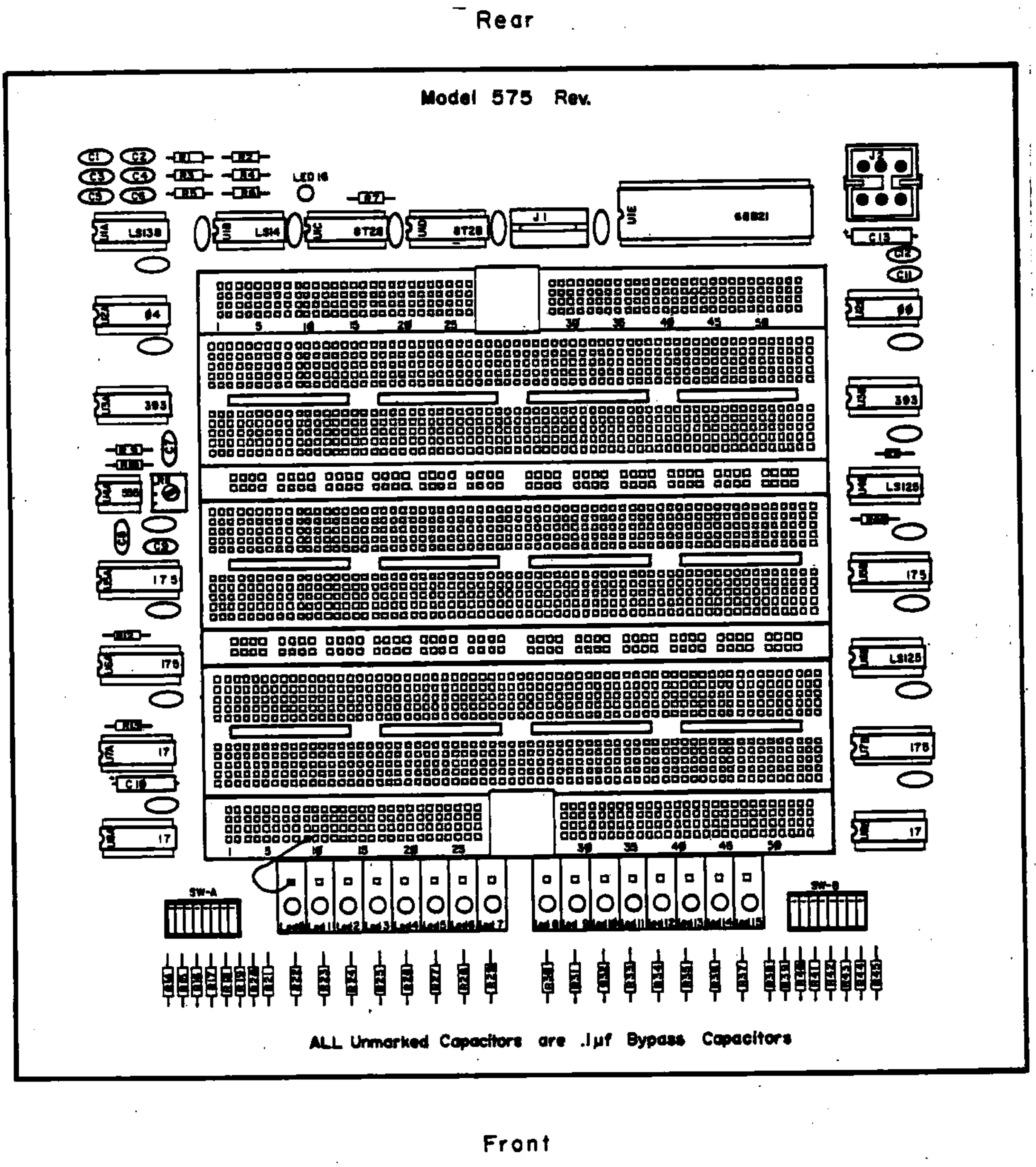


Figure 7
SW1A Connected to LEDØ

# EXPERIMENT 3

## TITLE

Adjustable Frequency Clock

## PURPOSE

The purpose of this experiment is to investigate the adjustable frequency clock which is a part of the CA-24 board, use it together with the divide-by-two circuits also on the board, and measure its maximum and minimum frequencies.

## EQUIPMENT

CA-24 board with power supply
Six short jumper wires
Four medium jumper wires (about 4 inches)
Seven long jumper wires (about 8 inches)
Screwdriver (narrow tip)
Oscilloscope (optional)

## DISCUSSION

An adjustable square wave oscillator, or clock, producing a series of Ø V to 5 V transitions, is a useful feature of the CA-24 board. This oscillator, together with a group of sixteen divide-by-two circuits provides a wide range of possible signal frequencies. Since the frequency range of the square wave oscillator is adjustable from 25 kHz to 7Ø kHz and since sixteen divide-by-two circuits will produce an overall division of 65536, the full range of available frequencies is from 7Ø kHz to about Ø.4 Hz. The duty cycle (the fraction of the total time spent at the 5 V level) is approximately 5Ø% except for the highest frequencies directly from the clock which are somewhat greater than 5Ø%.

The basic clock signal is produced by an IC chip type 555. The chip is located on the left side of the CA-24 board. It is marked U4A in Fig. 8. The wiring details for this chip are shown in Fig. 3b, slightly right of center. The adjustable resistor R11 varies the frequency of the output signal which is available at terminal A3 (on the strip farthest to the back).

The divide-by-two circuits are in the IC-74393 chips, one on each side of the CA-24 board, toward the rear, labeled U3A and U3B in Fig. 8. Each 74393 chip contains two divide-by-sixteen circuits, each of which is four divide-by-two circuits chained together with outputs after each division. The wiring details in Fig. 3b show how these circuits are strung together. As seen in Table 1 of SECTION 1, the input to the divide sequence is terminal A4 and the outputs after each division are A5 through A7, G3 through G7 (but note the reverse order, G7 through G4, for increasing division), A48 through A51, and a decreasing number sequence of G51 through G48. In addition to the terminal G48, G26 is the location of the lowest

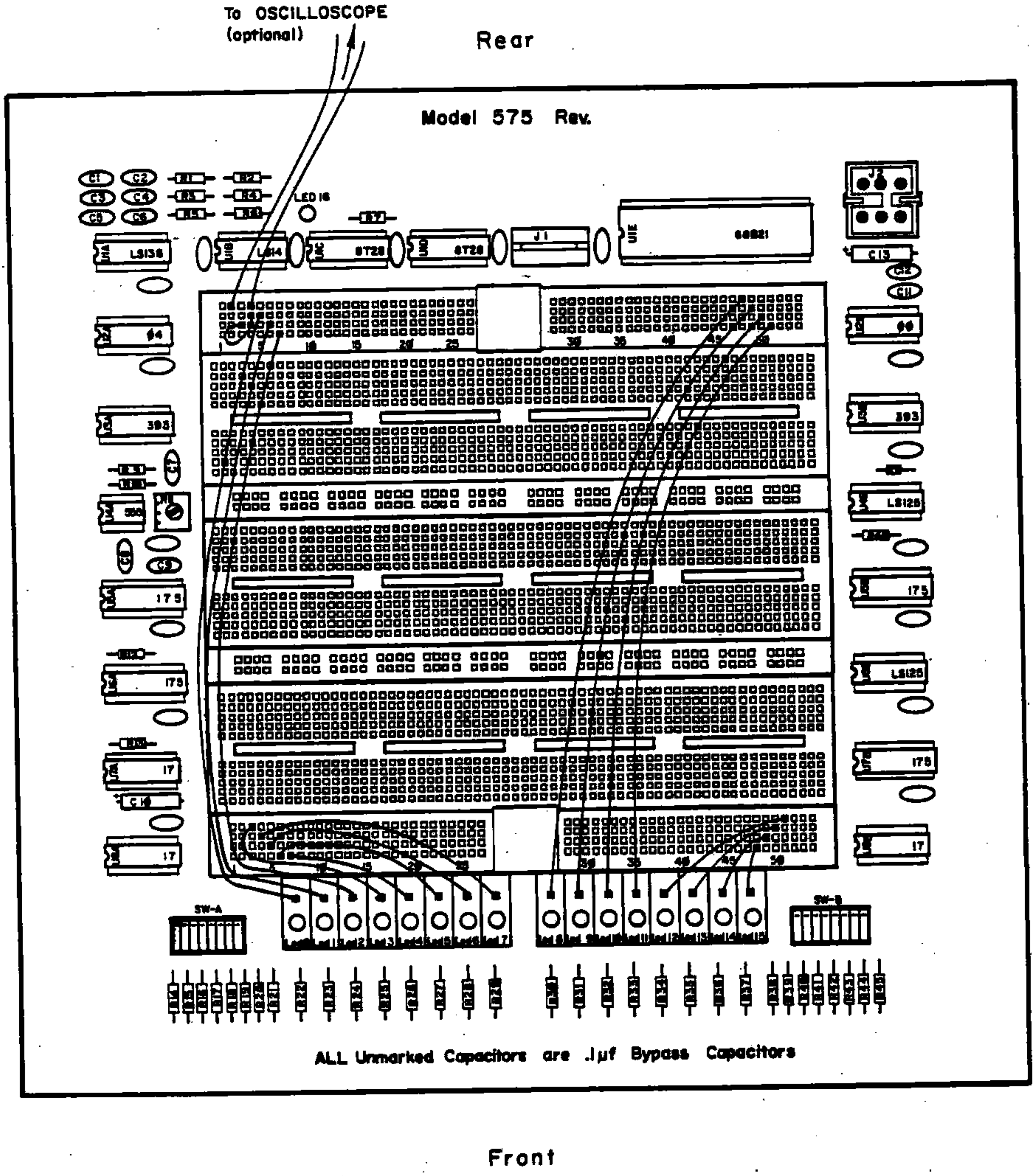
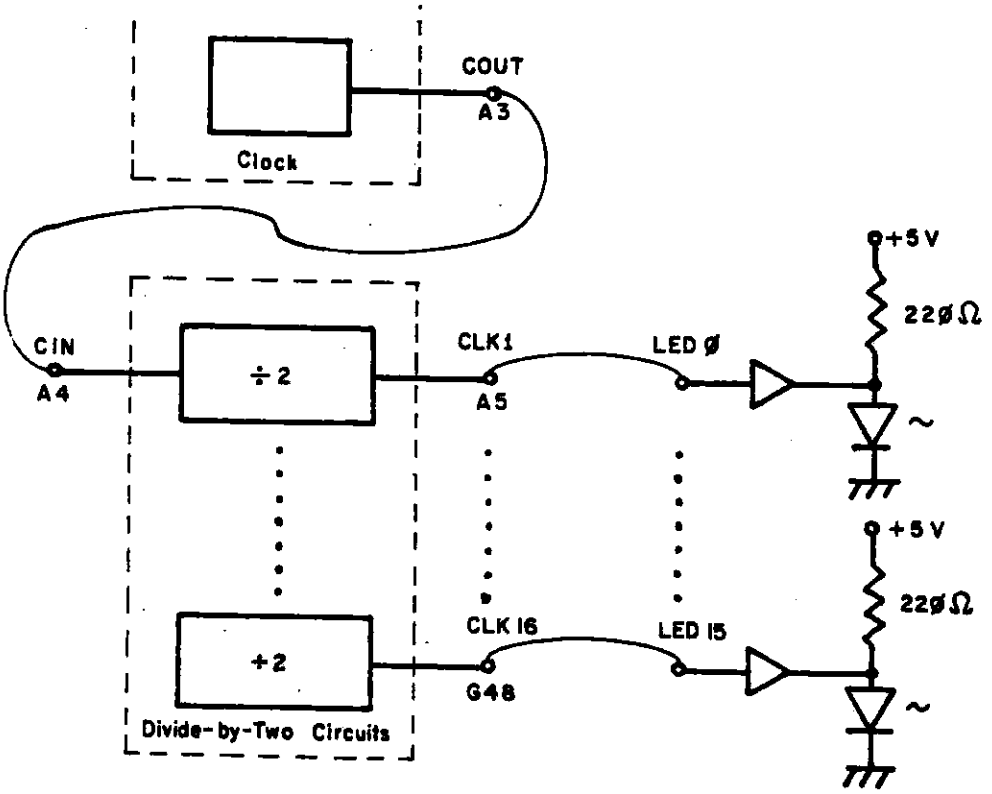


Figure 8
Clock Signals Connected to LED's

frequency output. Note that on the circuit diagram (Fig. 3b) and on the terminal connection list (Table 1) the labels for the output terminals of the divide chain are CLK1 through CLK16. To determine division factor at each output, use the label number as a power of two. For example, terminal A7 is labeled CLK3. Two raised to the third power is eight. A square wave connected to A4, then, will be reduced in frequency by a factor of eight at A7. Since two raised to the sixteenth power is 65536, the frequency of the signal at G48 or G26 is equal to the frequency of the signal at A4 divided by 65536.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Use a short jumper wire to connect A3 to A4. Next, connect the outputs of the divide chain to the LED indicators. While any sufficiently long jumper wires can be used to do this, one possibility is to use three long wires to connect A5 to LEDØ, A6 to LED1 and A7 to LED2. Then one short wire will reach from G3 to LED3 while four medium length wires can be used for G7 to LED4, G6 to LED5, G5 to LED6, and G4 to LED7. Now four long wires are needed to connect A48 through A51 to LED8 through LED11. Finally, the last four LED's can be connected to G51 through G48 with short jumper wires. A schematic representation of these connections is shown below.

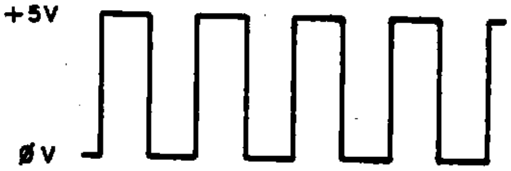


A picture of these connections is given in Fig. 8.

When you are certain that you have made all of the connections correctly, turn on the power supply. The far right

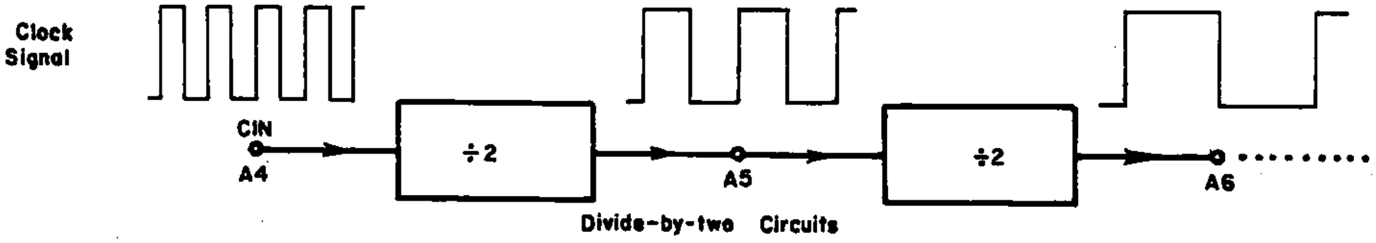
LED (LED15) should be turning on and off at the slowest rate while LED14 should be blinking twice as fast, etc. Most of the LEDs will flash too rapidly to detect.

If an oscilloscope is available, use it to observe the output voltage at either terminal A3 or A4. (See Fig. 8 for oscilloscope connections. Be certain to connect the ground of the oscilloscope to one of the GND terminals on the CA-24 board.) The oscilloscope trace should have the appearance something like the idealized diagram shown below. Note, however, that the actual oscilloscope trace probably will have an overshoot spike at the beginning of each high and low level. These overshoot spikes are not detrimental to the function of the clock circuit.



Use a screwdriver to turn the frequency adjust potentiometer, R11. Observe the change in the oscilloscope pattern. Each pulse width should be within the range of 7 microseconds (highest frequency) to 2Ø microseconds (lowest frequency).

After noting the width of each pulse for a particular setting of R11, connect the oscilloscope to terminal A5, then to A6. The change in trace should be as shown below where each pulse width is twice the previous one.



Divide-by-two Circuits

If an oscilloscope is not available (or after the oscilloscope has been used), the frequency of the clock can be measured by determining the rate at which the far right LED blinks on and off. For example, if that LED (LED15 connected to CLK16) blinks on and off at a rate of 1Ø blinks in 14.3 seconds (a frequency of Ø.7Ø Hz), then the divide-by-two raised to the sixteenth power or 65536 rate is Ø.7Ø Hz. The original oscillation frequency is 65536 times higher, or about 46ØØØ Hz. Alternatively, multiplying by 65.536 will give the original frequency in kilohertz. By using this method, the maximum and minimum clock frequencies can be measured. Fill in the table below, to confirm the frequency range over which the clock oscillates.

The value for the maximum frequency of the clock will be used in Experiment 12. Furthermore, you may at some later date need the maximum or minimum value as you use the CA-24 board in your own work. Consequently, we recommend that you make a careful measurement and record the values.

| Clock Adjust Resistor (R11) Position | Time for LED 15 to Blink 1Ø times (T) | Blink Frequency of LED 15 (F(15)=1Ø / T) | Clock Frequency (F(CLK)=F(15)* 65, 536) |
|---|---|---|---|
| Fully Clockwise | s | Hz | kHz |
| Fully Counter-Clockwise | s | Hz | kHz |

Note that upon disconnecting the CIN signal (wire removed from A4), the LED display stops changing and maintains the pattern it had at the instant that the wire was disconnected. When the wire is again connected to A4, the change in pattern resumes without a reset of the LED conditions. If it is desired to reset the LED's, turn off the power supply then turn it back on. Under most applications, continuous signal rates are needed, and the unavailability of a reset signal is not a problem.

This experiment used all 16 divide-by-two circuits connected, in numerical sequence, to the 16 LED's. The layout on the board for doing this is a bit messy. Since, as you observed, most of the LED's blinked too rapidly to count, connection to only the last two or three would have fulfilled the purpose of the experiment. (You probably were aware of this fact as you made the connections.) However, connections to all lights were suggested to emphasize the fact that the glowing of an indicator does not necessarily tell the whole story. What appears to be an "on" condition may be a rapidly varying "on-off" condition.

## EXPERIMENT 4

### TITLE

AND Gate

### PURPOSE

The purpose of this experiment is to investigate the function of the AND gate which is used in digital circuits.

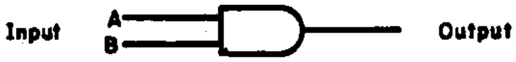
### EQUIPMENT

CA-24 board with power supply
Five short jumper wires
Two medium jumper wires
One 74Ø8 IC chip

### DISCUSSION

Logic circuits are the basis for computer operation. Understanding their functions permits the user of an interfaceable computer to perform time critical computer operations in hardware. Since you are now a CA-24 board user, you probably will wish to use a variety of logic circuits, such as the AND gate, the NAND gate, the OR gate, etc. If you are an accomplished user of these logic circuits and their combinations, you may wish to proceed directly to SECTION III where the experiments involving a computer begin.

Perhaps the simplest and certainly a very common logic circuit is the AND gate. The schematic representation is shown below.



The input signals to the AND gate would be on the wires represented by the two lines on the left and the output signal would be on the wire represented by the line on the right.

As has been mentioned earlier, a 5 V signal is called a "true" or logic "1" level. A Ø V signal is called a "false" or logic "Ø" level. The convention of referring to the more positive voltage level of the two possible levels as "true", is called the "positive" logic convention. We will work only with the "positive" logic convention.

The AND gate functions to produce a logic "1" at the output only when both inputs (input A AND input B) are at logic "1". If either input or both inputs are at logic "Ø", the output is at logic "Ø".

There is a standard notation associated with the binary system of logic (or Boolean algebra as it is called in honor of

George Boole, a nineteenth century Englishman) of which the AND gate is our first example. Some of the texts listed in Appendix 3 include detailed discussions of Boolean algegra. The AND operation is symbolized by a dot between the symbols for the input signals. Consequently, the function of the AND gate can be written as:

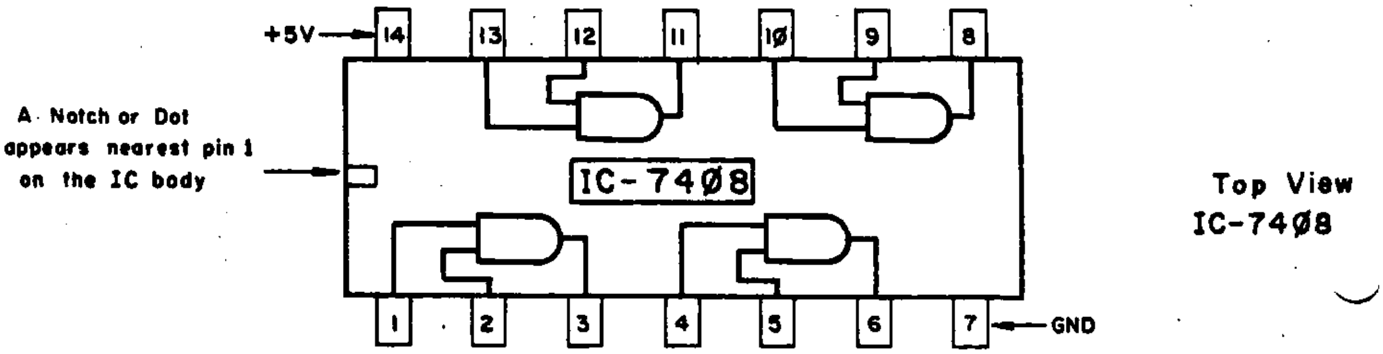
output = A•B

Sometimes the dot is left out in which case the notation becomes:

output = AB

Additional notation will be introduced in the following experiments.

There are four AND gates in the 7408 IC chip which is shown schematically below.

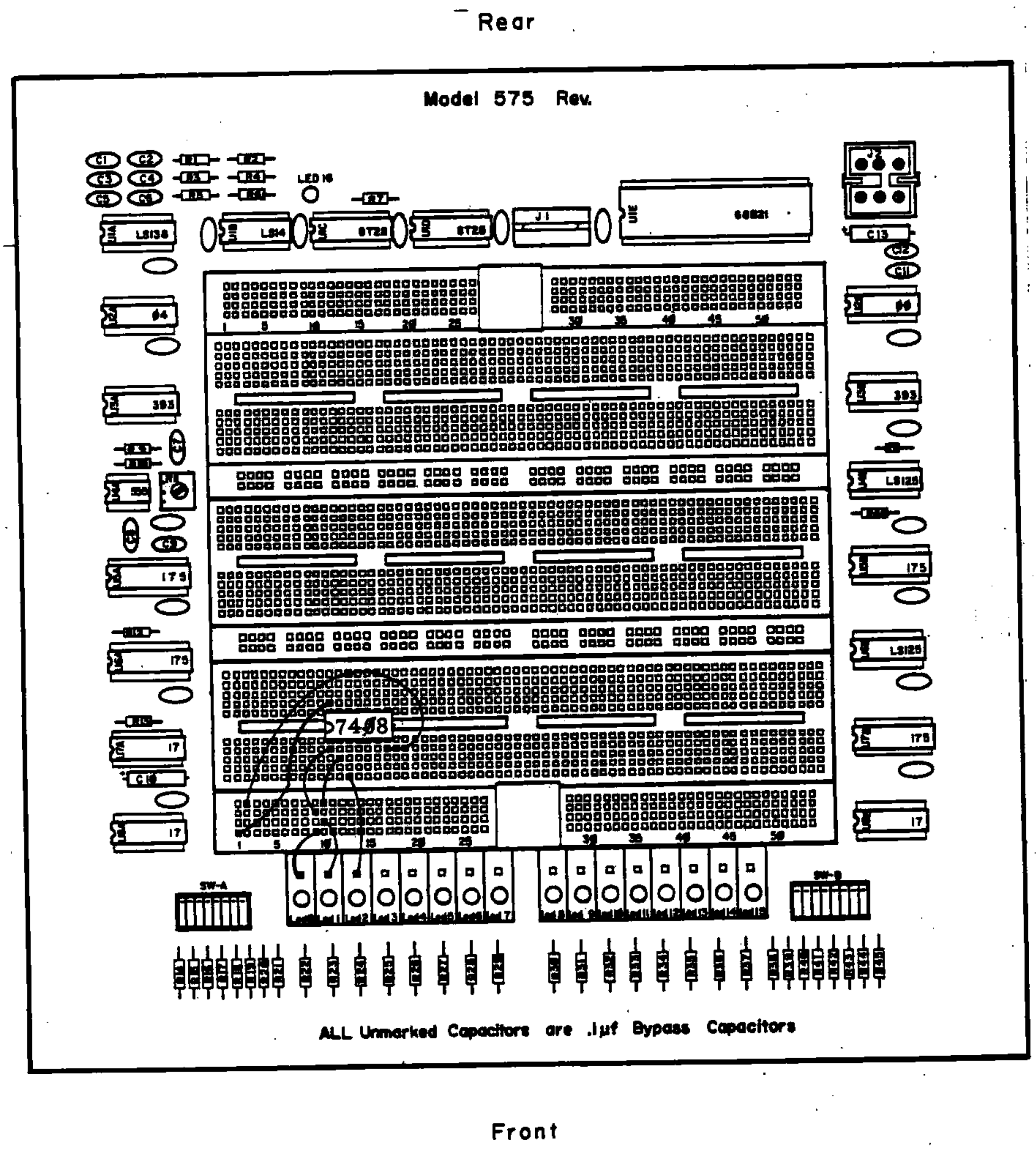


Reference to the diagram will allow determination of input and output pins for the various gates. Pin one is marked on IC chips in one of two ways. On some chips a small dot is placed at one end, off center and near pin one. If used, this is the clearest manner of marking pin one. A notch at one end of the chip (on the center line of the chip) is often used to indicate the end containing pin one (and also the highest numbered pin), see the drawing above. Finally, some manufacturers use both a dot and a notch to mark pin one. Plastic chips may have one or two circular mold marks on the centerline of the chip. Do not confuse these with the "pin one" marks described above.

PROCEDURE

Many of the experiments in this manual require that you insert IC chips on the CA-24 board. The following instructions describe how to remove IC chips from the board without damaging their pins.

> CAUTION - Use extreme care in removing any IC chip from a terminal strip. It is a very common experience to have one end or side of the chip suddenly come free as the chip is being removed. The result is that the pins remaining in the holes are badly bent and



ALL Unmarked Capacitors are .1µf Bypass Capacitors

Figure 9
Switch and LED Connections to an AND Gate

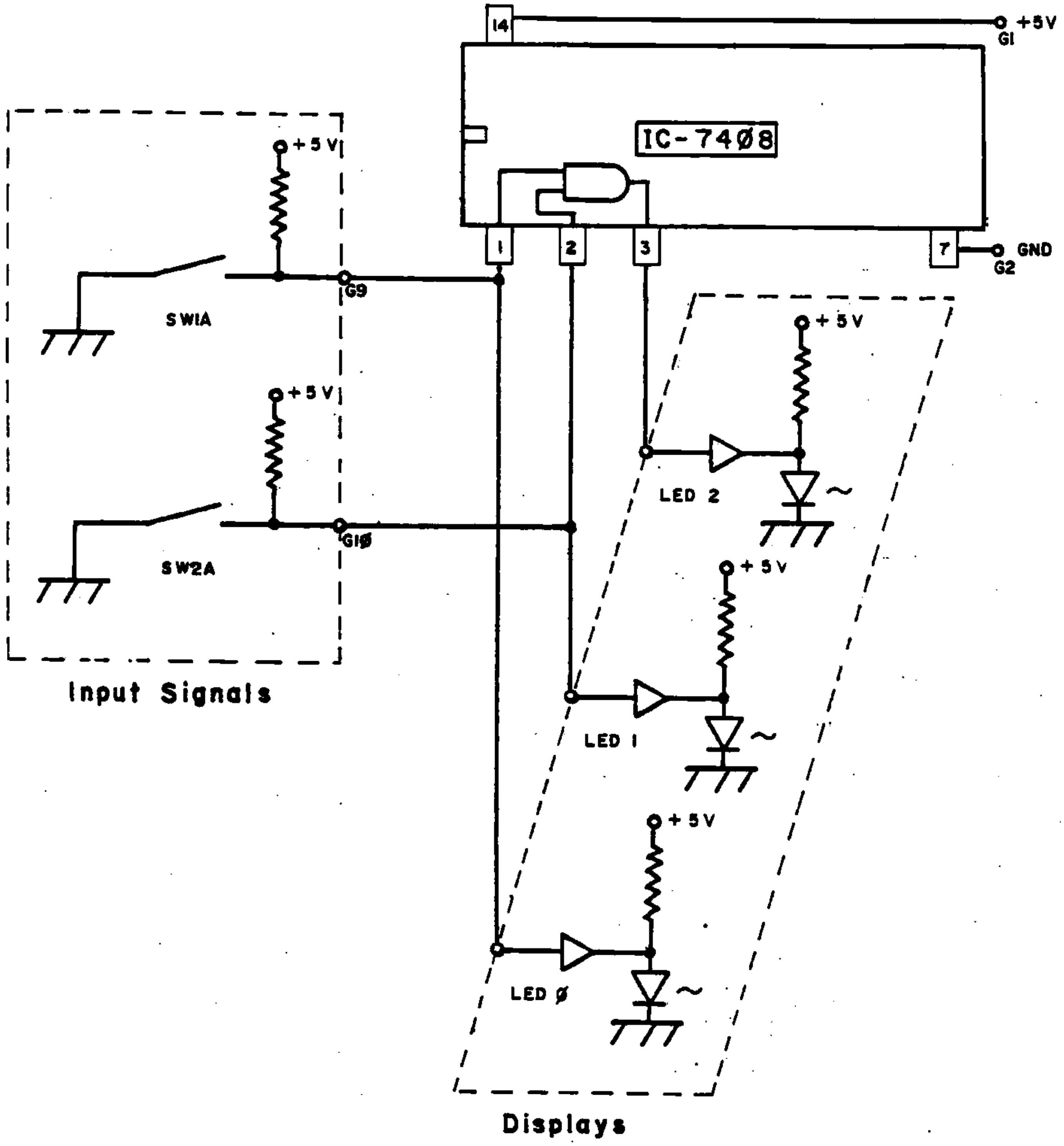
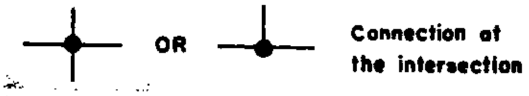


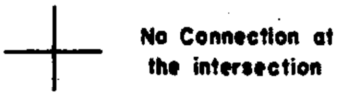
Figure 10
Schematic for AND Gate Circuit

may be broken off. A good technique is loosen the chip slightly then insert a small screwdriver or narrow knife blade under the chip and slowly pry up first one end and then the other without ever tilting the chip very much. BE SURE THE POWER IS OFF WHEN YOU DO THIS.

Be certain that the power supply to the CA-24 board is turned off. Insert a 74Ø8 IC chip into terminal strip F so that it spans the center groove, is left of center, and has pin 1 to the left (see Fig. 9). The chip can be placed anywhere on the board as long as it spans a center groove of strip B, D, or F. We have chosen the location shown in Fig. 9 because it is close, but not too close, to the LED's which will be used for display of logic levels. Use the two medium length jumper wires to connect pin 14 to terminal G1 and pin 7 to G2. These connections are the 5 V and GND power connections. Next, use four short jumper wires to connect SW1A (terminal G9) and SW2A (terminal G1Ø) to LEDØ and LED1 respectively, and to the input of the AND gate associated with pins 1 and 2 of the 74Ø8 chip. The connections are shown on the schematic diagram in Fig. 1Ø, and on the drawing in Fig. 9. Finally use a short jumper wire to connect the output of the AND gate to LED2 as shown in Fig. 9. You will note that on the schematic diagram, a connection is represented by a dot as shown below.



If there is to be no connection when lines cross, then the dot is omitted.



When you are certain that all connections have been made correctly, turn on the power supply. Set switches 1A and 2A to various combinations and observe the three LED's. Remember that when LEDØ is on, one gate input signal (from SW1A) is 5 V or logic "1" while LEDØ off means that the input is Ø V or logic "Ø". Similar statements hold for LED1 and SW2A. When LED2 is on, the gate output is 5 V or logic "1" while LED2 off means that the gate output is Ø V or logic "Ø".

The combination of input and output signals can be arranged in an organized fashion called a truth table. Such a table is shown below with the two input columns filled in. Position SW1A and SW2A so that LEDØ and LED1 correspond to the condition shown on each line, then fill in the last column. For example, set SW1A and SW2A so that both LEDØ and LED1 are off (logic "Ø"). Note the condition of LED2 and write Ø or 1 in the third column depending on whether LED2 is off or on.

| SW 1A<br>(LED Ø condition) | SW 2A<br>(LED 1 condition) | OUTPUT<br>(LED 2 condition) |
|---|---|---|
| Ø | Ø | |
| 1 | Ø | |
| Ø | 1 | |
| 1 | 1 | |

The output of the AND gate should be logic "1" only when both inputs are at logic "1". That is, LED2 should be on only when both LEDØ and LED1 are on. If either or both of the input indicators are off, LED2 should be off. In terms of the "true" and "false" designations, the output is "true" only when both inputs are "true". If either input is "false", the output is "false".

Another way of describing the gate function is in terms of signal transmission through the gate. If SW1A is kept in its logic "1" position, then the AND gate output display reports the status of SW2A. If SW1A is kept in its logic "Ø" position, the output display is off regardless of the position of SW2A. Consequently, it can be said that SW1A enables the gate, permitting the status of SW2A to be transmitted through the gate.

# EXPERIMENT 5

## TITLE

NAND Gate and Inverter

## PURPOSE

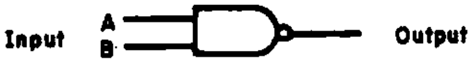
The purpose of this experiment is to investigate the function of the logic circuit NAND gate and to use it as an inverting element.

## EQUIPMENT

CA-24 board with power supply
Five short jumper wires
Two medium jumper wires
One 74ØØ IC chip

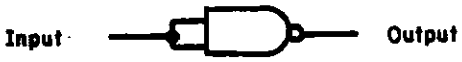
## DISCUSSION

The NAND (NOT AND) gate combines an AND gate and an inversion of the output signal. The schematic representation of the NAND gate is shown below.

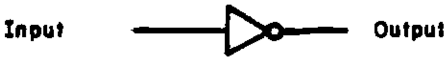


Note that the only difference between the AND gate symbol and the NAND gate symbol is the circle at the output of the NAND gate which represents inversion. For the NAND gate, therefore, when both inputs are at logic "1", the output is at logic "Ø" rather than at logic "1" which would be the case for an AND gate. Any other condition produces a logic "1" at the output.

A NAND gate can be used to change a 5 V or logic "1" signal into a Ø V or logic "Ø" signal or vice versa. This is done by connecting together the two inputs so that the same signal appears at both of them. Schematically, a NAND gate used as an inverting element looks like that shown below.



There is a general symbol for an inverting element which differs from the symbol for an identity element (see page 1Ø) in the same way that the NAND gate symbol differs from the AND gate symbol; a circle is added to the output end. Any circuit which inverts, then, can be represented as shown below.



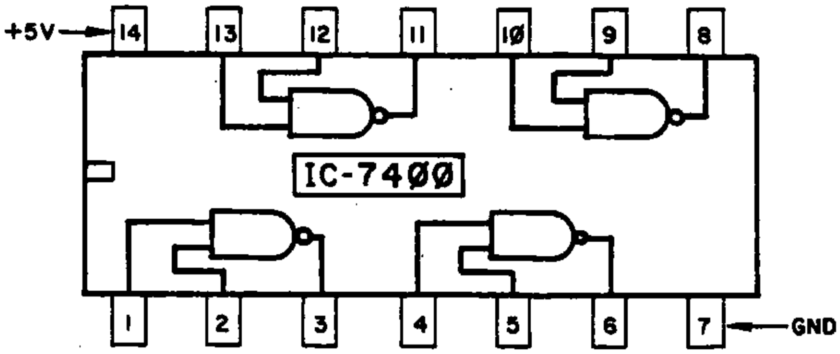
The standard notation which we began to develop in Experiment 4 uses a bar over a symbol or group of symbols to represent inversion. The simple inversion, then, is expressed as

$$\text{output} = \overline{\text{input}}$$

while the NAND gate function would be written as follows:

$$\text{output} = \overline{A \cdot B}$$

There are four NAND gates in the 74ØØ IC chip which is shown schematically below.



Top View
IC-74ØØ

Reference to the diagram will allow determination of input and output pins for the various gates.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. If the circuit of Experiment 4 is still hooked up on the board, simply replace the 74Ø8 chip with a 74ØØ chip.

> CAUTION - Use extreme care in removing the chip. As explained in Experiment 4, it is very easy to damage the pins as the chip is being removed. Use the procedure suggested in the CAUTION note found in that experiment.

The wiring will then be that shown schematically in Fig. 11. If the wiring of Experiment 4 has been changed, replace the jumper wires as described in that experiment and shown in Fig. 12.

When you are certain that all connections have been made correctly, turn on the power supply. Set switches 1A and 2A to various combinations and observe the three LED's. Remembering the convention of LED on being a 5 V signal or logic "1" and the LED off being a Ø V signal or logic "Ø", fill in the truth table at the top of page 32.

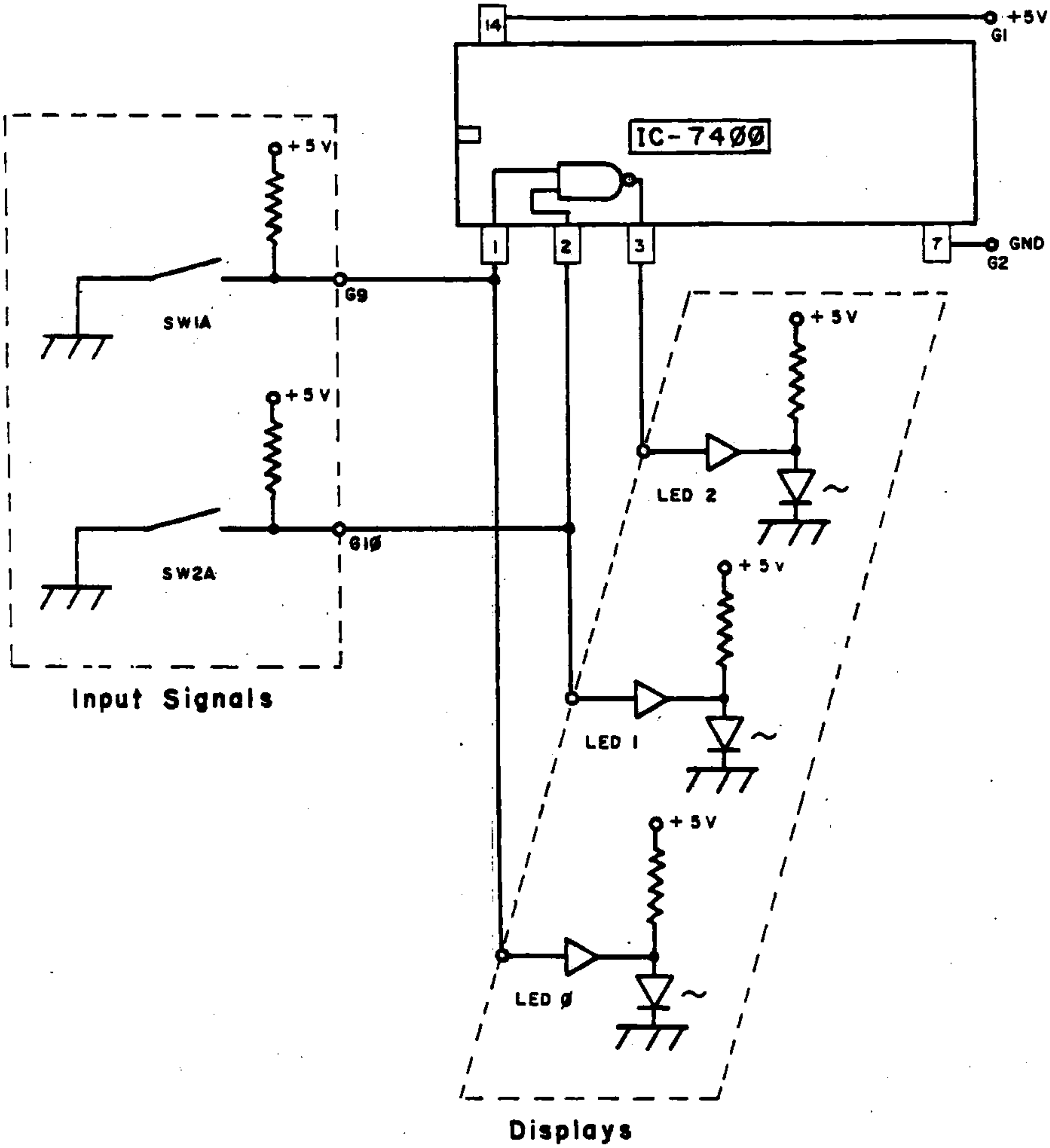


Figure 11
Schematic for NAND Gate Circuit

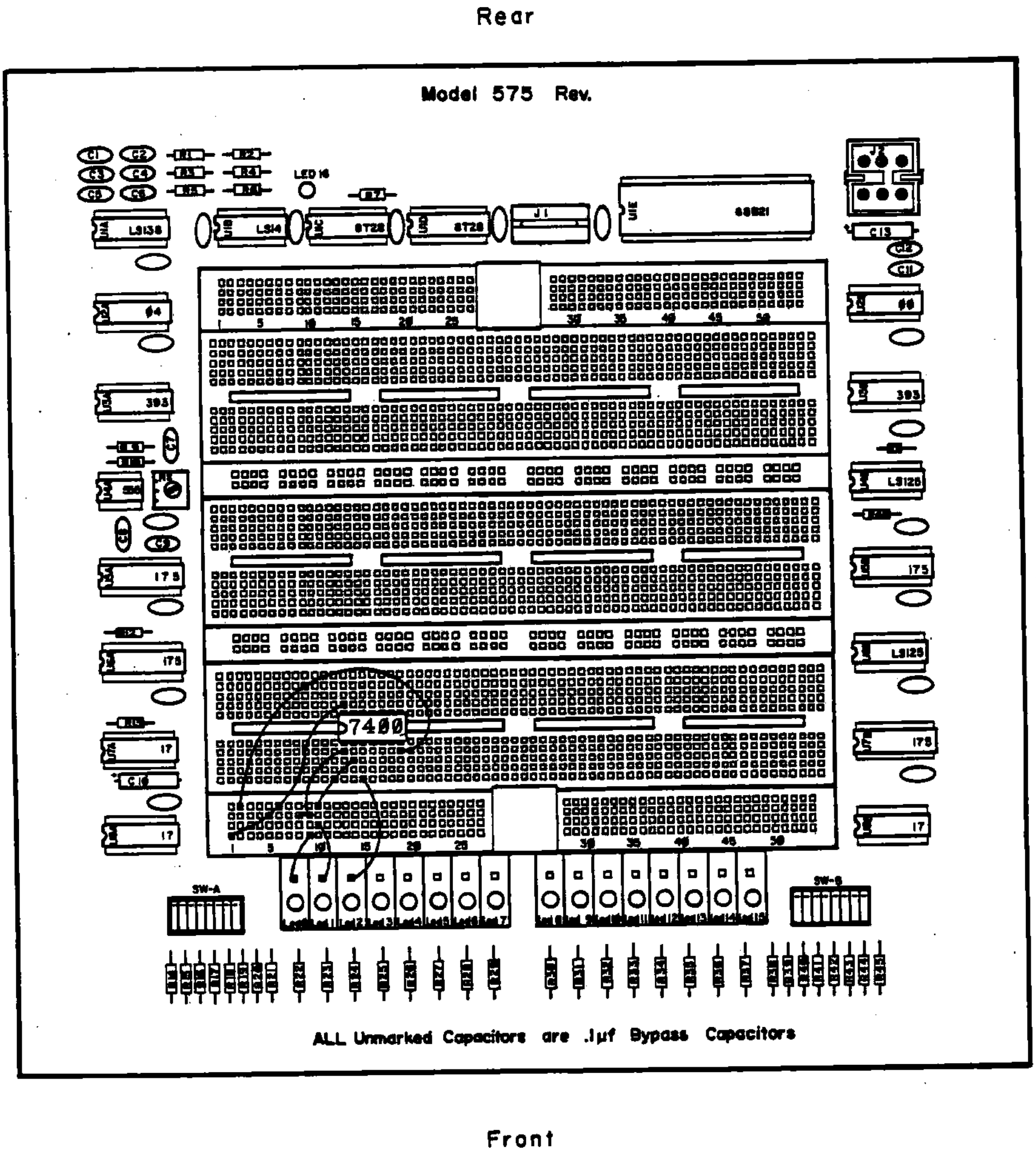


Figure 12
Switch and LED Connections to a NAND Gate

| SW 1A<br>(LED Ø condition) | SW 2A<br>(LED 1 condition) | OUTPUT<br>(LED 2 condition) |
|---|---|---|
| Ø | Ø | |
| 1 | Ø | |
| Ø | 1 | |
| 1 | 1 | |

The output of the NAND gate should be logic "Ø" only when both inputs are logic "1".

If you wish to observe the NAND gate functioning as an inverter, first turn off the power supply then remove the jumper wire connecting G1Ø to LED1. Next, for the wire connecting pin 2 to G1Ø, shift the end which is in a G1Ø terminal hole to pin 1 of the 74ØØ. The circuit will then be as shown schematically in Fig. 13 and pictorially in Fig. 14. Now turn on the power supply. Note that as SW1A is moved back and forth, LEDØ and LED2 will always be in the display conditions opposite to one another.

Returning the circuit to the NAND gate form will facilitate wiring changes for Experiment 6.

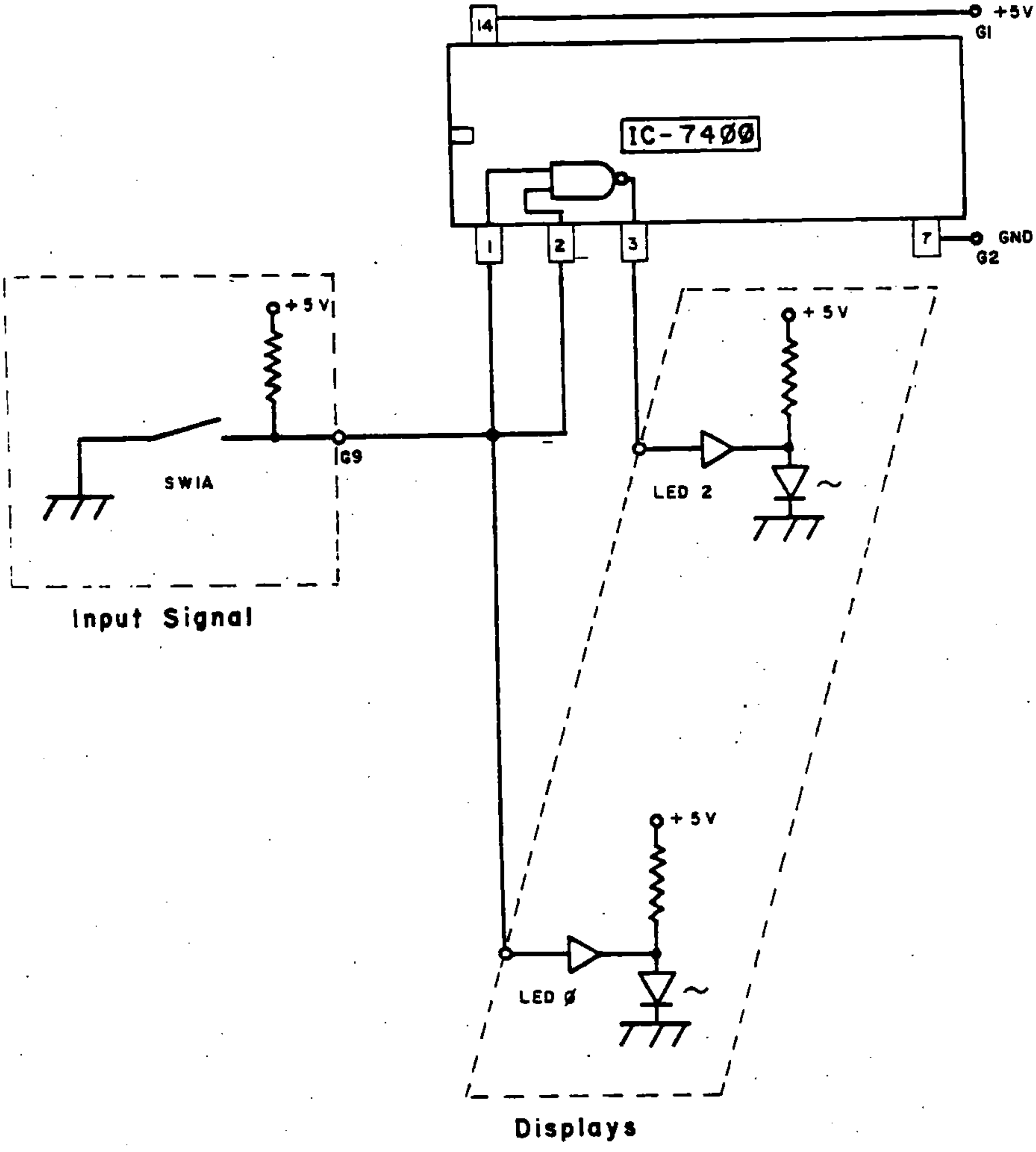


Figure 13
Schematic for a NAND Gate Used as an Inverter

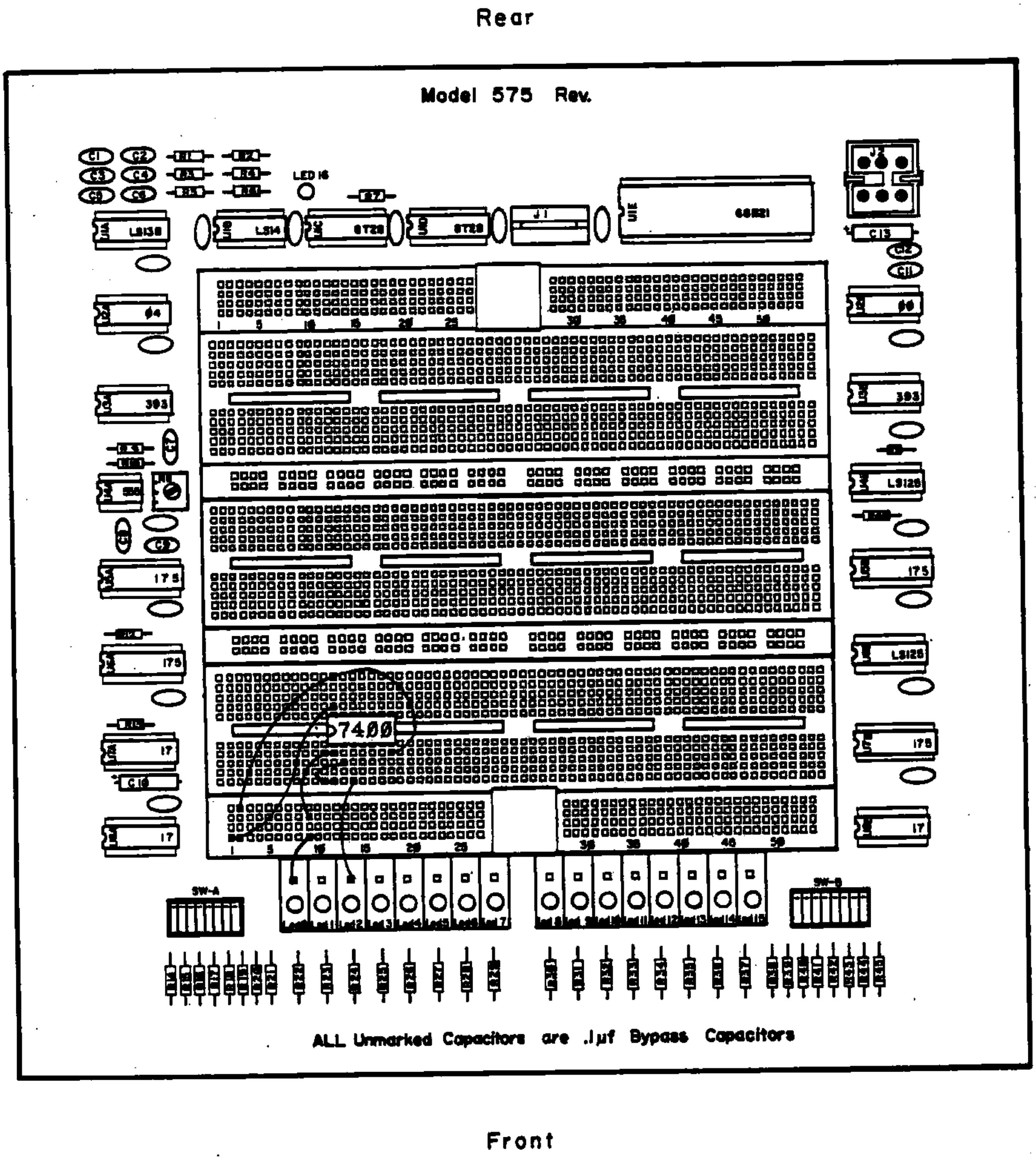


Figure 14
Switch and LED Connections to a NAND Gate Inverter

# EXPERIMENT 6

## TITLE

NOR Gate and Inverter

## PURPOSE

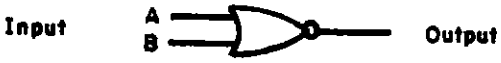
The purpose of this experiment is to investigate the function of the logic circuit NOR gate and to use it as an inverting element.

## EQUIPMENT

CA-24 board with power supply
Five short jumper wires
Two medium jumper wires
One 74Ø2 IC chip

## DISCUSSION

The NOR (NOT OR) gate combines an OR gate (which will be investigated in Experiment 7) and an inversion of the output signal. Since NOR gate IC chips are significantly less expensive than OR gate chips and since OR gates can easily be made from NOR gates, we will work with only the NOR gate chip. The schematic representation of the NOR gate is shown below.

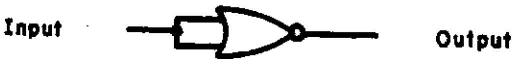


The function of the NOR gate is to produce a logic "Ø" (inverted logic "1") when either one of the inputs is logic "1". The "either/or" function is not, however, exclusive of the "both" condition for the inputs. That is, the output of the gate is also logic "Ø" when both inputs are high. The only condition which produces a logic "1" at the output is that of both inputs at logic "Ø".

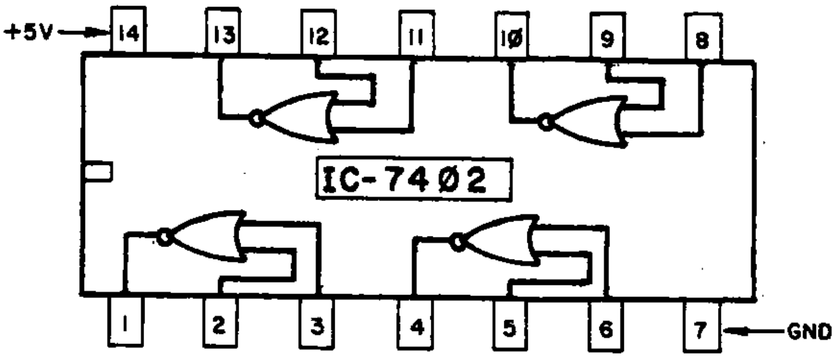
The standard notation representing a NOR gate function is a combination of the OR operation and an inversion. The + symbol represents the OR operation. Consequently, the NOR gate function is written as follows:

$$\text{output} = \overline{A + B}$$

A NOR gate can be used as an inverting element by connecting together the two inputs so that the same signal must always appear at both of them. The schematic diagram for using the NOR gate as an inverter is shown below.



There are four NOR gates in the 74Ø2 IC chip which is shown in the diagram below.



Reference to the diagram will allow determination of the input and output pins for the various gates.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. If the circuit for Experiment 5 is still on the CA-24 board, remove the 74ØØ chip and replace it with the 74Ø2 chip. Next, shift jumper wires until the circuit has been connected as shown schematically in Fig. 15 and pictorially in Fig. 16. Note that if the circuit in Experiment 5 was left in the NAND gate form, only two changes are needed. The jumper wire which connected G9 to pin 1 should now connect G9 to pin 3 and the jumper which connected LED2 to pin 3 should now connect LED2 to pin 1.

When you are certain that all connections have been made correctly, turn on the power supply. Set switches 1A and 2A to various combinations and observe the three LED's. Fill in the truth table below.

| SW 1A<br>(LED Ø condition) | SW 2A<br>(LED 1 condition) | OUTPUT<br>(LED 2 condition) |
|---|---|---|
| Ø | Ø | |
| 1 | Ø | |
| Ø | 1 | |
| 1 | 1 | |

The output of the NOR gate should be logic "1" only when both inputs are logic "Ø".

If you wish to observe the NOR gate functioning as an inverter, simply remove the jumper wire connecting G1Ø to LED1. Then for the wire connecting pin 2 to G1Ø, shift the end which is in a G1Ø terminal hole to pin 3 of the 74Ø2. The circuit will then be as shown schematically in Fig. 17 and pictorially in Fig. 18. Note that as SW1A is moved back and forth, LEDØ and LED2 will always be in the display conditions opposite to one another.

Returning the circuit to the NOR gate form will facilitate wiring changes for Experiment 7.

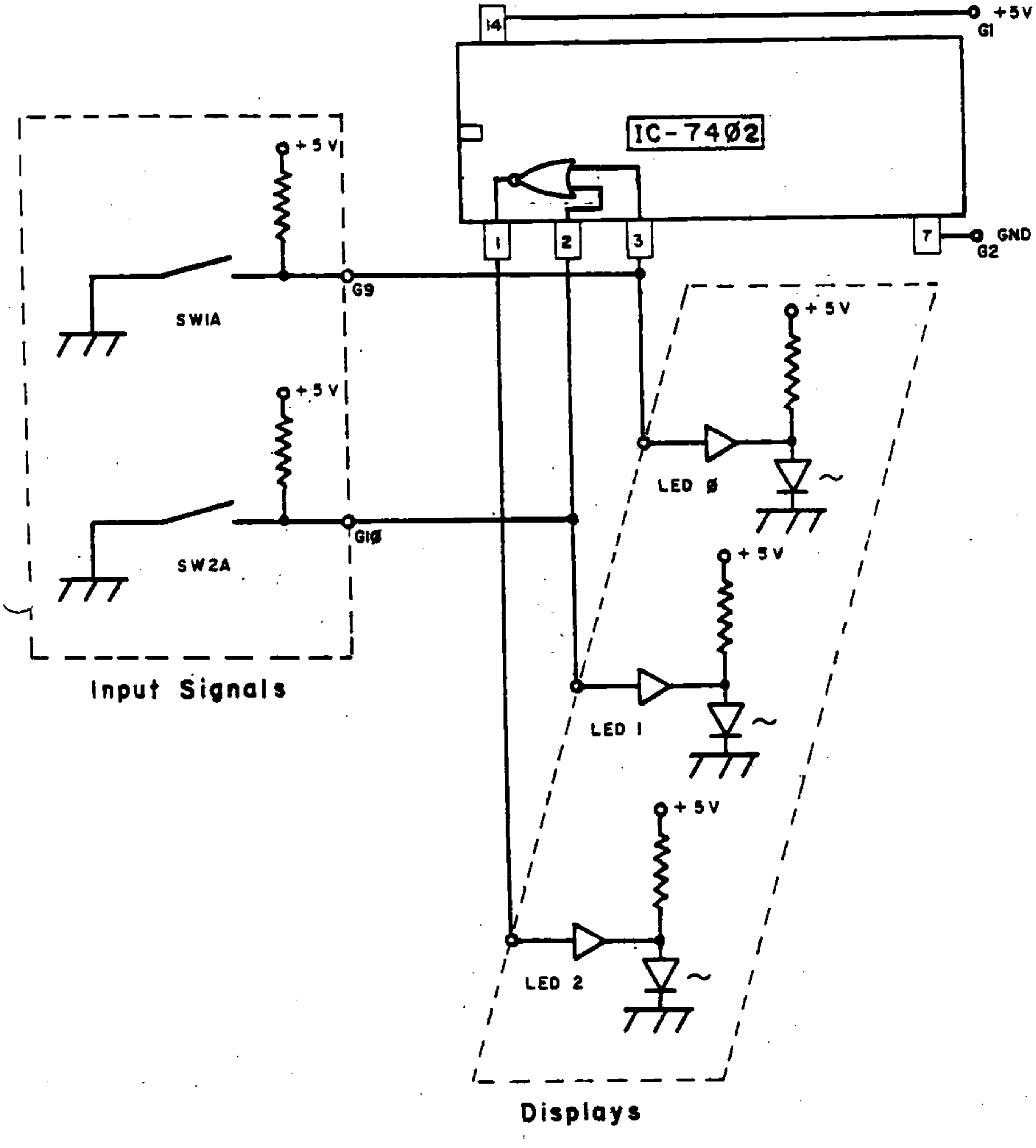


Figure 15
Schematic for NOR Gate Circuit

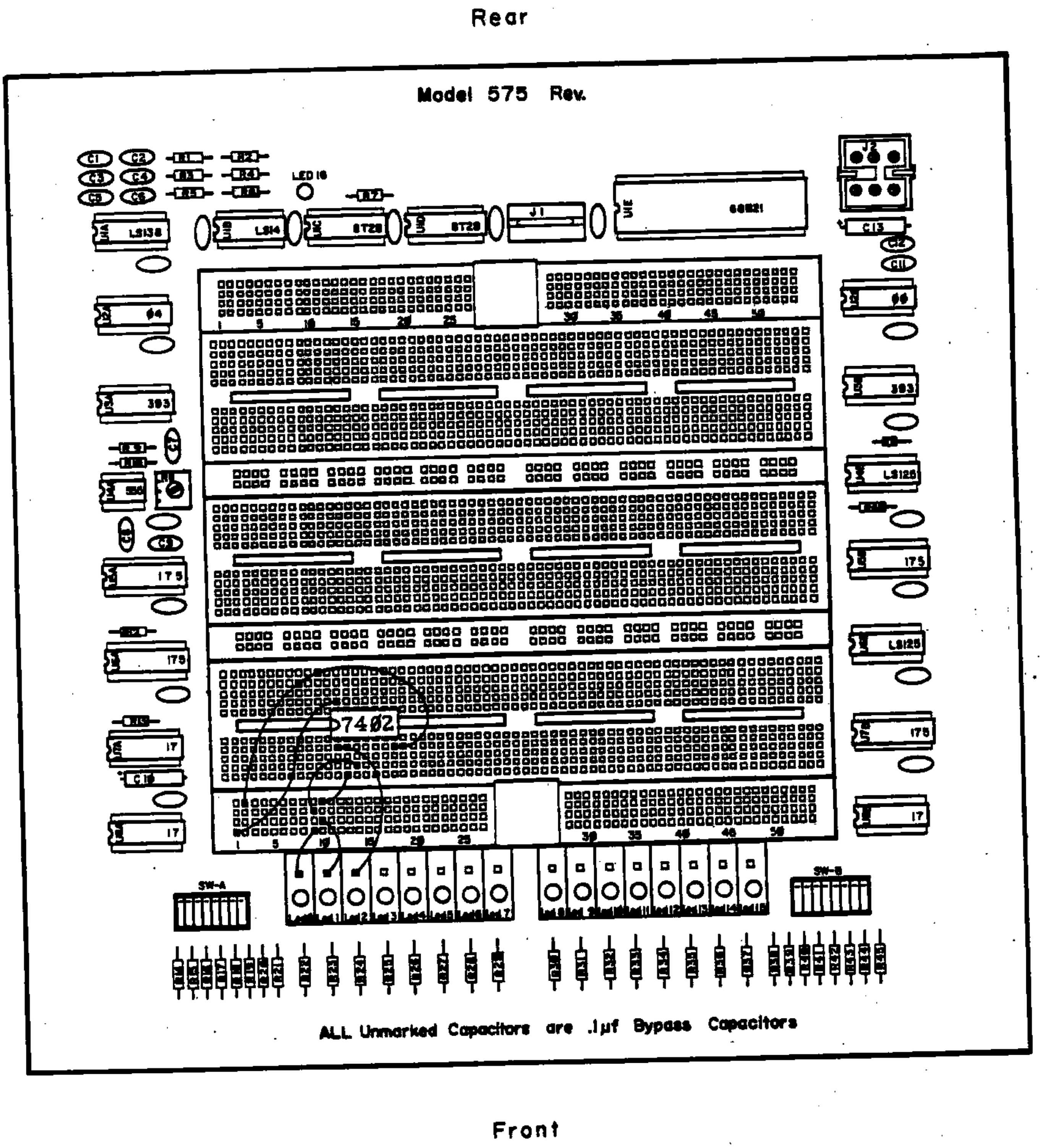


Figure 16
Switch and LED Connections to a NOR Gate

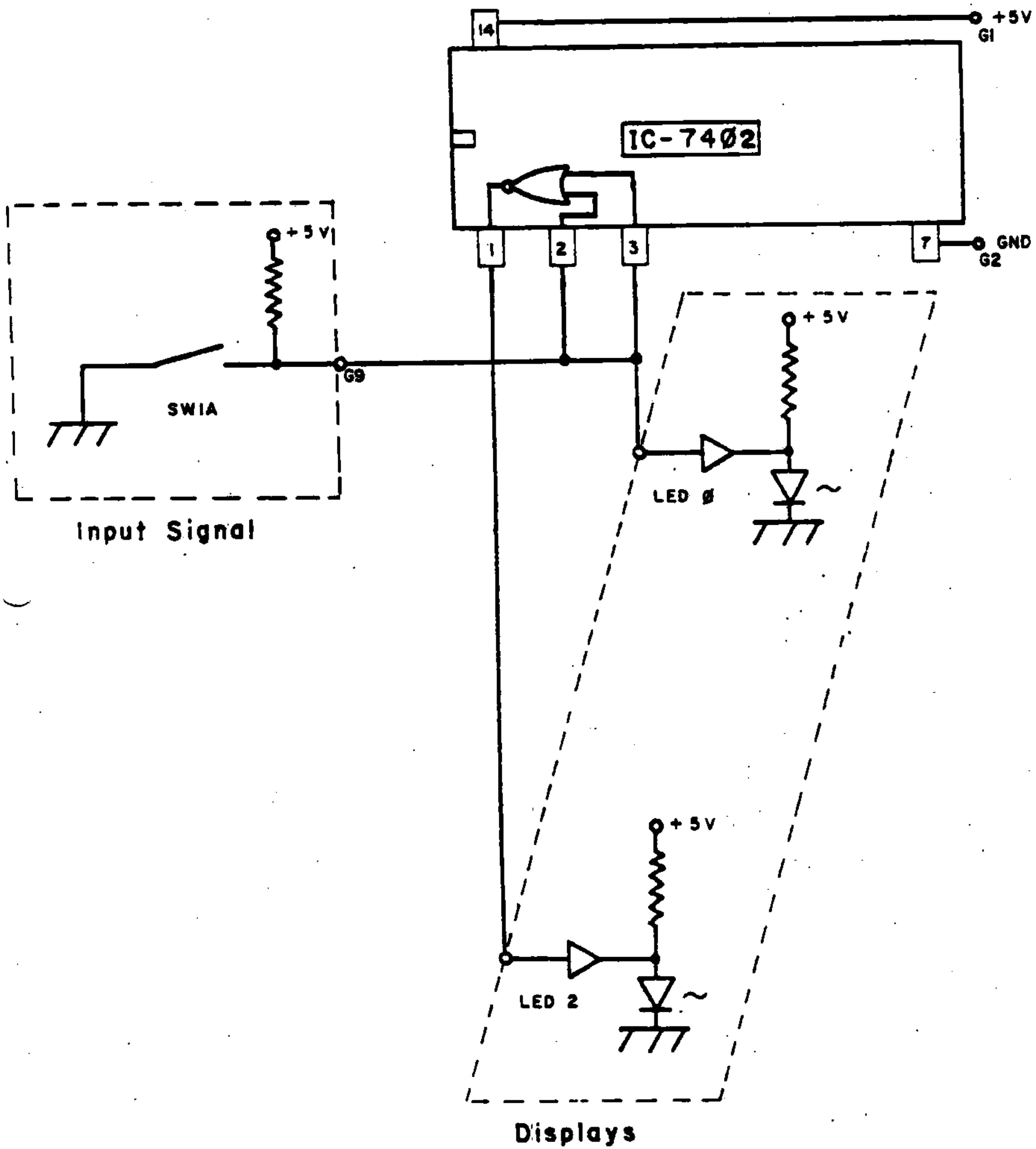


Figure 17
Schematic for a NOR Gate Used as an Inverter

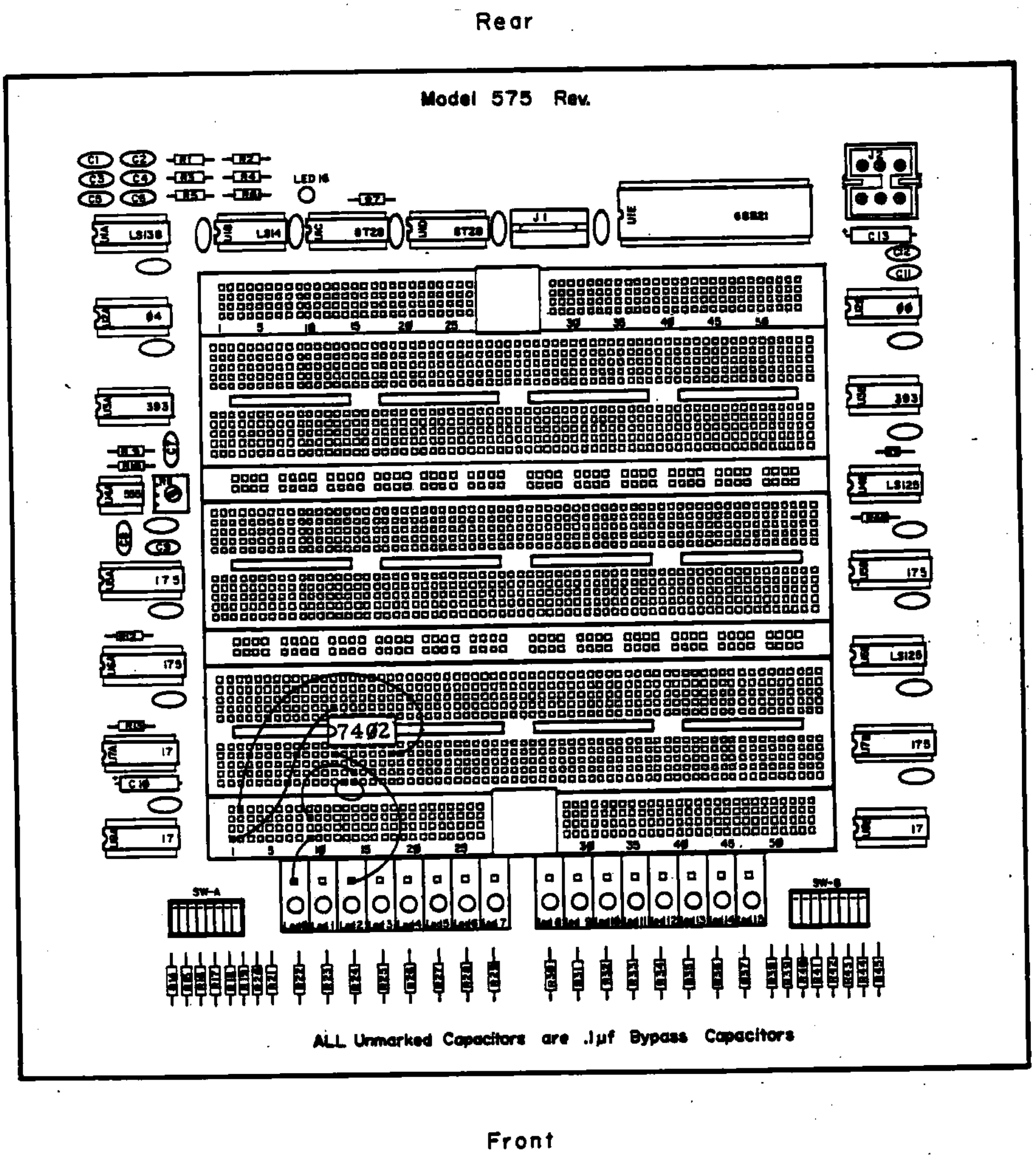


Figure 18
Switch and LED Connections to a NOR Gate Inverter

EXPERIMENT 7

## TITLE

OR Gate from NOR Gates

## PURPOSE

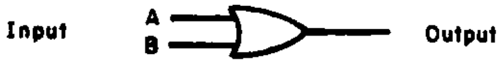
The purpose of this experiment is to construct an OR gate using two NOR gates and to investigate its function.

## EQUIPMENT

CA-24 board with power supply
Seven short jumper wires
Two medium jumper wires
One 74Ø2 IC chip

## DISCUSSION

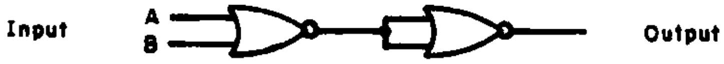
The function of an OR gate is to produce a logic "1" signal at the output when either of the two inputs (or both) is at logic "1". The symbol for an OR gate is shown below.



In terms of the standard notation of Boolean algebra, the OR gate function is written as follows:

output = A + B

Since NOR gates are easier to manufacture than are the OR gates, it is common practice to construct OR gates from NOR gates. This is done simply by sending the output of the NOR gate through an inverter. The combination is shown schematically in the following diagram. The output from the NOR gate is the inverse of A + B. When the output of the NOR gate is sent through an inverter, the result is A + B (that is the inverse of the inverse of A + B is just A + B itself).



## PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. If the NOR gate circuit of Experiment 6 is still on the board, shift the wires which currently connect terminals G9 and G1Ø to pins 2 and 3, so that they now connect G9 and G1Ø to pins 5 and 6. Next, use two additional jumper wires to connect together pins 2, 3, and 4. The circuit should then be as shown in the schematic diagram of Fig. 19 and the drawing of Fig. 2Ø.

When you are certain that all connections have been made correctly, turn on the power supply. Set switches 1 and 2 to various combinations and observe the three LED's. Fill in the truth table below.

| SW 1A | SW 2A | OUTPUT |
|---|---|---|
| (LED Ø condition) | (LED 1 condition) | (LED 2 condition) |
| Ø | Ø | |
| 1 | Ø | |
| Ø | 1 | |
| 1 | 1 | |

The output of the OR gate should be logic "Ø" only when both inputs are logic "Ø".

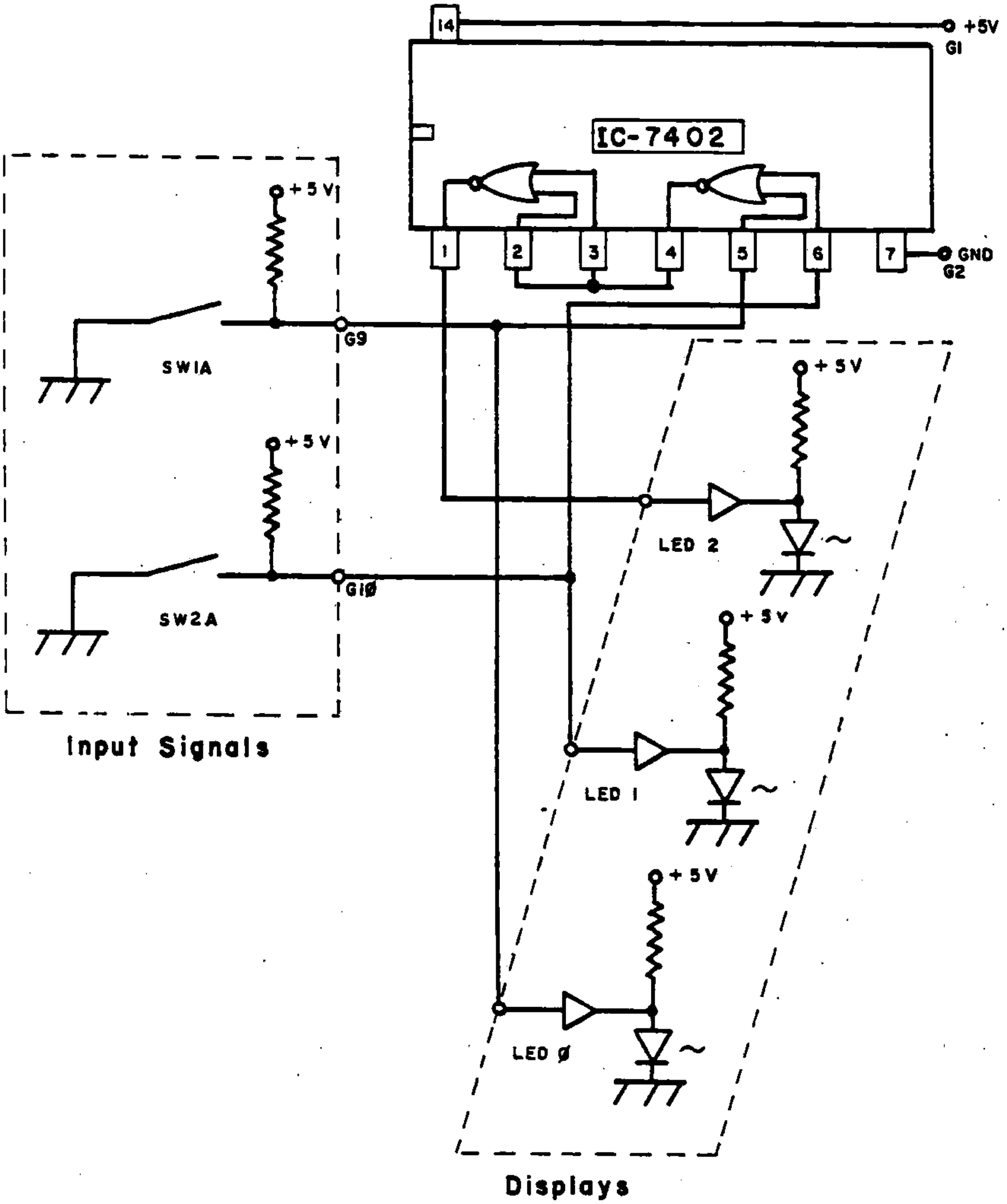


Figure 19
Schematic for an OR Gate Made From NOR Gates

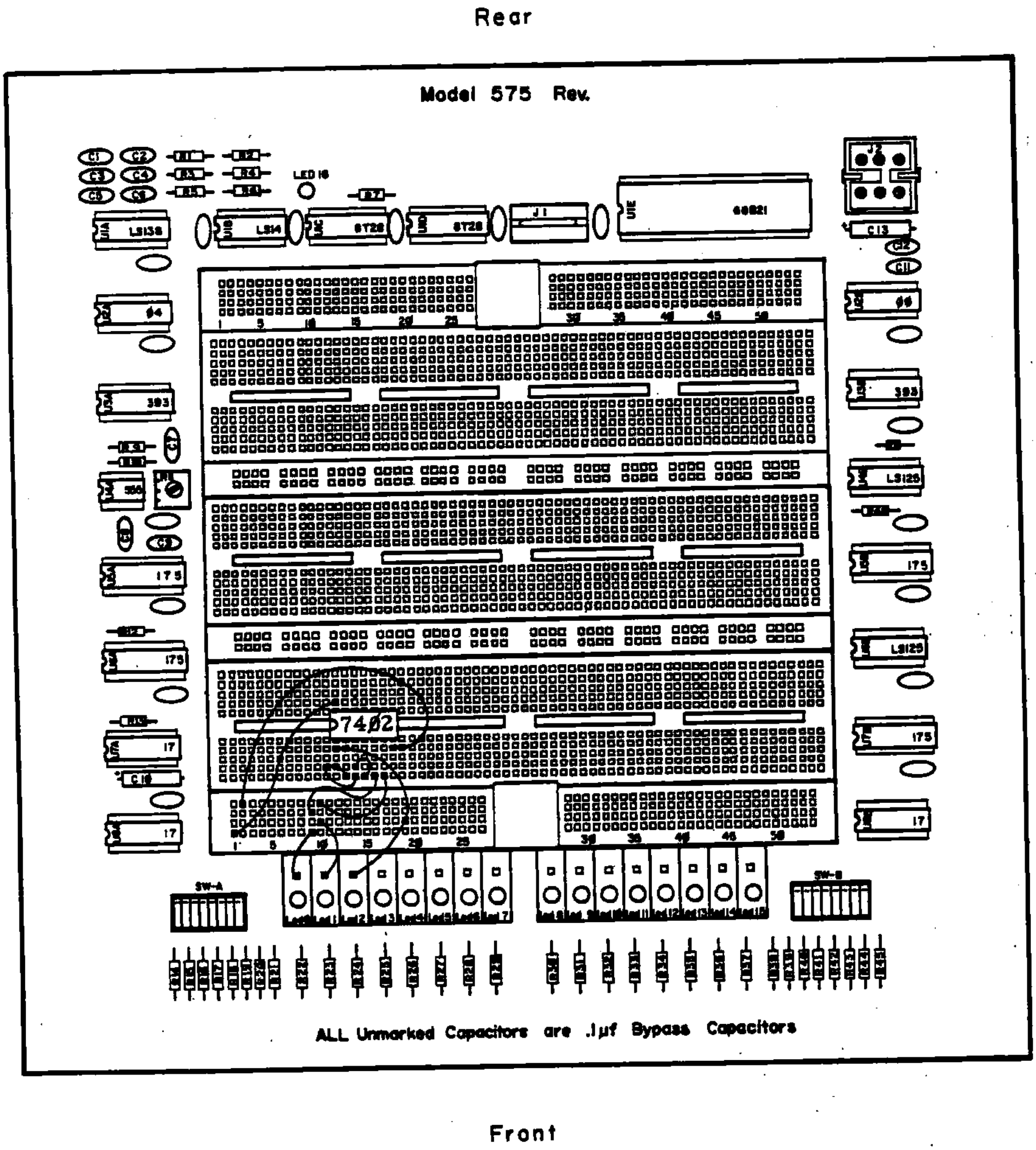


Figure 20
Switch and LED Connection to OR Gate

# EXPERIMENT 8

## TITLE

Exclusive OR Gate from AND, NAND, and NOR Gates

## PURPOSE

The purpose of this experiment is to construct an exclusive OR gate using the three gates which were studied in Experiments 4, 5, and 6 and to investigate its functions.

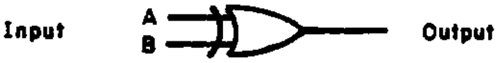
## EQUIPMENT

CA-24 board with power supply
Eight short jumper wires
Eleven medium jumper wires
One 74ØØ IC chip (NAND gates)
One 74Ø2 IC chip (NOR gates)
One 74Ø8 IC chip (AND gates)

## DISCUSSION

The function of an exclusive OR gate is to produce a logic "1" signal at the output when either of the two inputs is at logic "1", but not when both are at logic "1". For the exclusive OR gate, then, both inputs at logic "Ø" or at logic "1" will produce an output of logic "Ø".

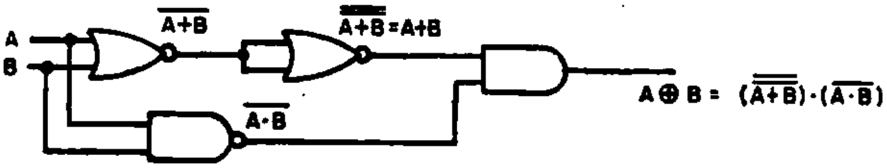
The symbol for the exclusive OR gate is shown below.



In the standard notation of Boolean algebra, the exclusive OR operation is represented by the symbol ⊕. The function of the exclusive OR gate, then, would be written as follows:

$$\text{output} = A \oplus B$$

It is possible to purchase an IC chip which has exclusive OR gates in it (the 7486). However, if AND, NAND, and NOR gates are already being used, it is frequently advantageous to construct the exclusive OR from the others. The diagram for doing this is shown below.


A
B
$\overline{A+B}$
$\overline{\overline{A+B}}=A+B$
$\overline{A \cdot B}$
$A \oplus B = (\overline{\overline{A+B}}) \cdot (\overline{A \cdot B})$

Note that the Boolean algebra notation has been included to show how equations could be used instead of diagrams to represent the circuit functions.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Remove any chips and wires left on the board from previous work and then position the three IC chips as shown in Fig. 21. Make all the necessary connections using the short and medium jumper wires where appropriate. It probably will be helpful to refer to the schematic diagram in Fig. 22 as well as the pictorial representation while inserting the jumper wires. Note that strip E is being used as a power distribution strip. This procedure helps to keep the board organized.

When you are certain that all connections have been made correctly, turn on the power supply. Set switches 1A and 2A to various combinations and observe the three LED's. Fill in the truth table below.

| SW 1A<br>(LED Ø condition) | SW 2A<br>(LED 1 condition) | OUTPUT<br>(LED 2 condition) |
|---|---|---|
| Ø | Ø | |
| 1 | Ø | |
| Ø | 1 | |
| 1 | 1 | |

The output (LED2) of this exclusive OR gate should be logic "Ø" (off) when both inputs (LED's Ø and 1) are at logic "Ø" (off) or at logic "1" (on). The output should be at logic "1" when either input is at logic "1".

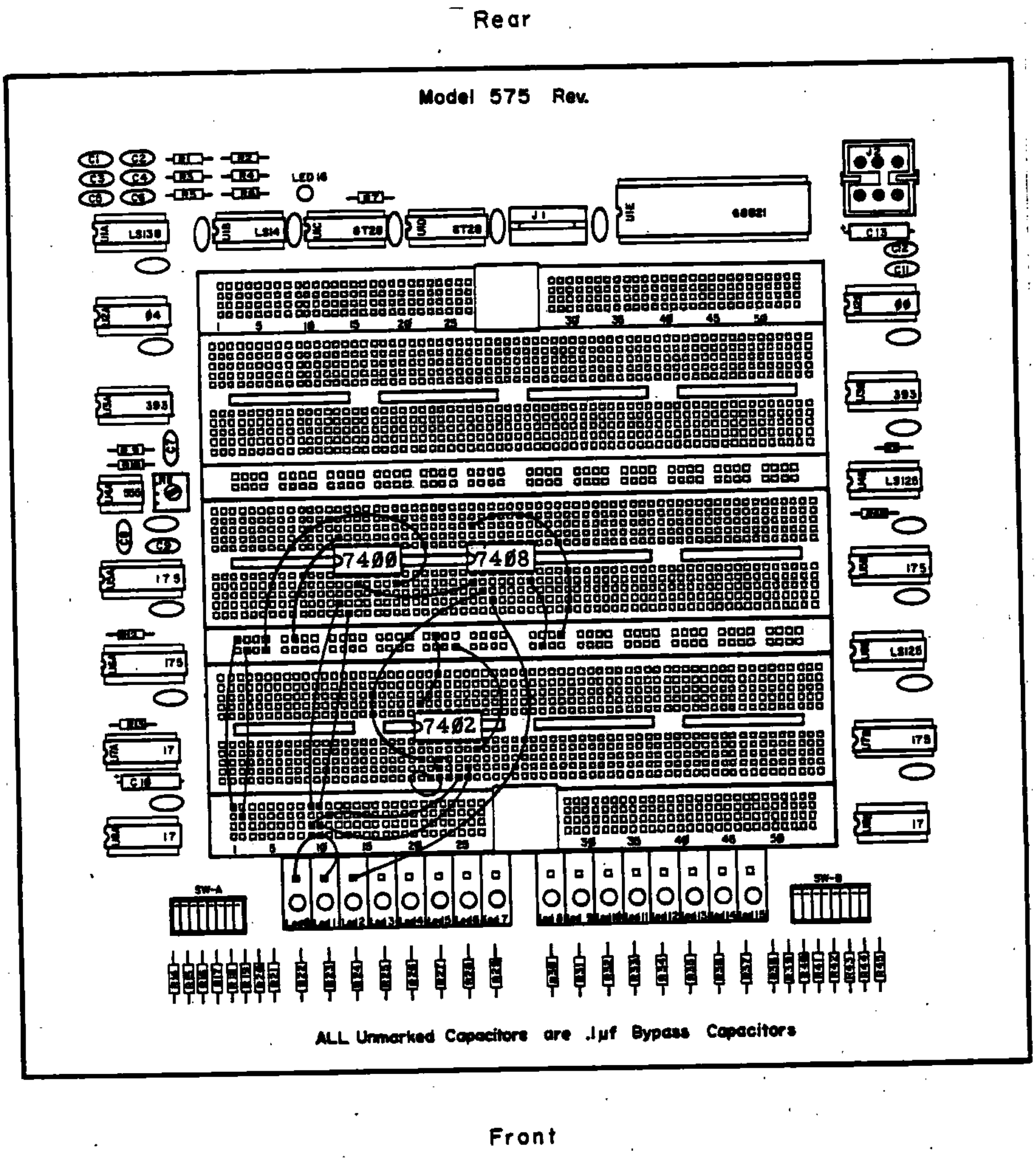


Figure 21
Switch, LED, and Inter-Chip Connections for an Exclusive OR Gate

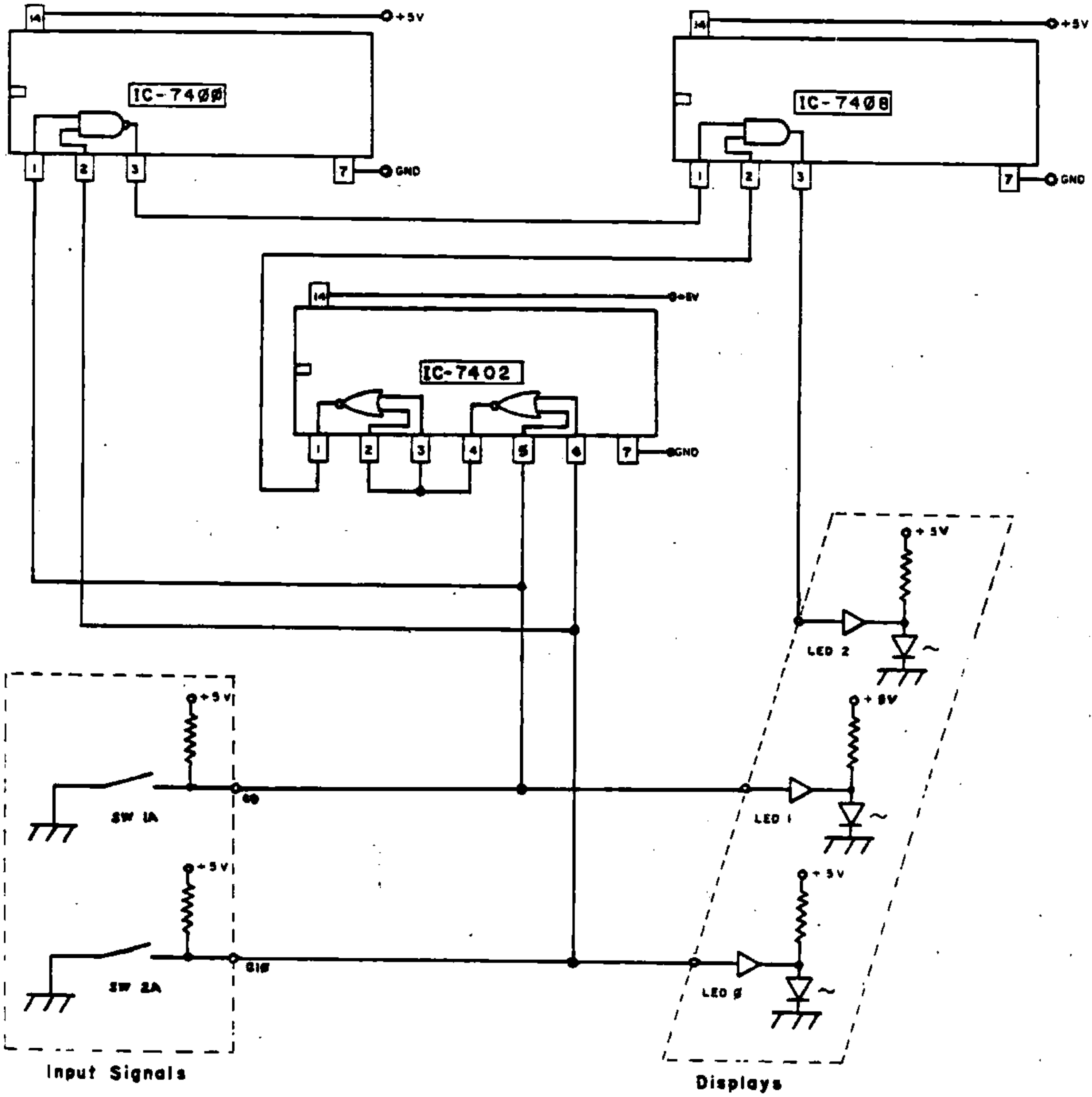


Figure 22
Schematic for an Exclusive OR Gate Made From AND, NAND, and NOR Gates

# EXPERIMENT 9

## TITLE

The R-S Flip-Flop (RSFF)

## PURPOSE

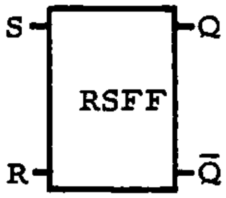
The purpose of this experiment is to become familiar with the basic digital flip-flop circuit and to construct and test an R-S (Reset-Set) flip-flop with an enable-disable input.

## EQUIPMENT

CA-24 board with power supply
Eleven short jumper wires
Five medium jumper wires
One 74ØØ IC chip

## DISCUSSION

The flip-flop circuit is one for which the output has two stable conditions which can be changed from one to the other by a proper combination of signals at the input. A simple flip-flop is the R-S flip-flop. The "black box" representation of this device is shown below together with its truth table.



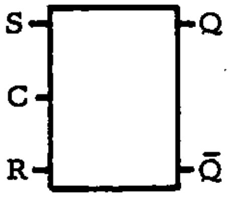
| S | R | Q | $\overline{Q}$ |
|---|---|---|---|
| Ø | Ø | No Change | |
| 1 | Ø | 1 | Ø |
| Ø | 1 | Ø | 1 |
| 1 | 1 | Indefinite | |

It is common practice to label the two output signals Q and $\overline{Q}$ where $\overline{Q}$ always has the opposite value from Q. This flip-flop is not used with the condition of both R and S being at logic "1".

The flip-flop circuit in this experiment can be compared to a fluorescent desk lamp (Q) controlled by two push button switches (R and S). Push button S (the set switch) is used to turn the lamp on. Once the lamp is on, pressing S has no further effect until the lamp is turned off again. Push button R (the reset switch) is used to turn the lamp off. Once the lamp is off, pressing R has no further effect until the lamp is turned on again. Pressing both buttons simultaneously yields indefinite results. As you read through the following discussion, it may help to identify the terminology low and logic "Ø" with "off" and high and logic "1" with "on".

The terms set and reset come from the condition that, from the starting condition of both inputs at logic "Ø", the set signal going high will set Q to logic "1" and the reset signal going high will reset Q to logic "Ø". With R low, then, S going high will produce at Q a high which will stay there when S is made low again. This is one of the two stable conditions. As long as R is low, changing S back and forth between high and low will not alter the logic "1" at Q. (The "No Change" notation in the truth table means Q stays high when S is made low.) When S is low, however, changing R to high will change Q to low, the second stable condition. As long as S remains low, changing R will not alter the logic "Ø" at Q. (The "No Change" notation means also that Q stays low when R is made low.)

Because the flip-flop has the capability of holding a signal at its output while certain changes are occuring at its input, the flip-flop is sometimes referred to as a latch. The latching feature is enhanced if an additional signal can enable or disable the entire flip-flop. The general diagram for a flip-flop with this capability is shown below.



If the signal at C (for control or clock) is low (logic "Ø"), the circuit will be disabled, holding whatever condition is at Q. If the signal at C is high (logic "1"), the circuit will function normally. Furthermore, the signal at C can be used to change the condition at Q. If the flip-flop has been reset (level at Q is logic "Ø") and the signal at C has been put at logic "Ø", then a high input at S will have no effect. However, if the signal on C is raised to +5 V while the logic "1" is on S, the signal at Q will become logic "1". In this function mode, the flip-flop can be used to transfer a logic "1" signal from S to Q and then hold it at Q.

The two flip-flop circuits described above can be constructed from NAND gates. The circuit diagrams are shown in Figs. 23a and 23b. (Pin numbers for the 74ØØ IC chip and other CA-24 board designations are also shown there as this figure will be referenced again in the PROCEDURE portion of this experiment.) If you wish to trace the logic level conditions in order to understand why the circuit behaves as it does, it will probably be helpful to begin with a logic "1" at S, a logic "Ø" at R, and a logic "1" at Q. $\bar{Q}$, of course, will be low because the NAND gate for which it is the output has both inputs high. Convince yourself that no change at Q will occur if S is switched low, then with S low, trace through what will happen when R is raised.

## PROCEDURE

Be certain that the power supply to the CA-24 board is off. Remove any chips and wires left on the board from previous work and then position a 74ØØ IC chip as shown in Fig. 24. Make all the necessary connections using the short and medium jumper wires where appropriate. It probably will be helpful to refer to the schematic diagram in Fig. 23a as well as the pictorial

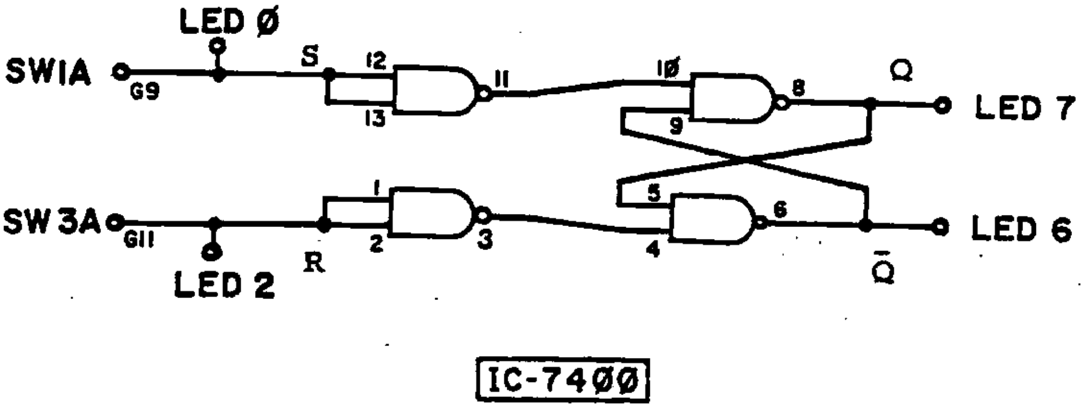


Figure 23a
Schematic for an R-S Flip-Flop Made From NAND Gates

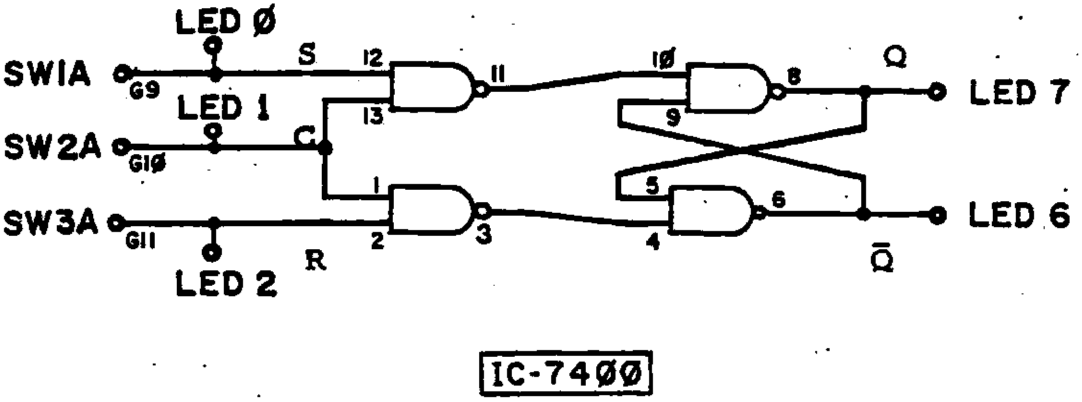


Figure 23b
Schematic for an R-S Flip-Flop with Clock Input

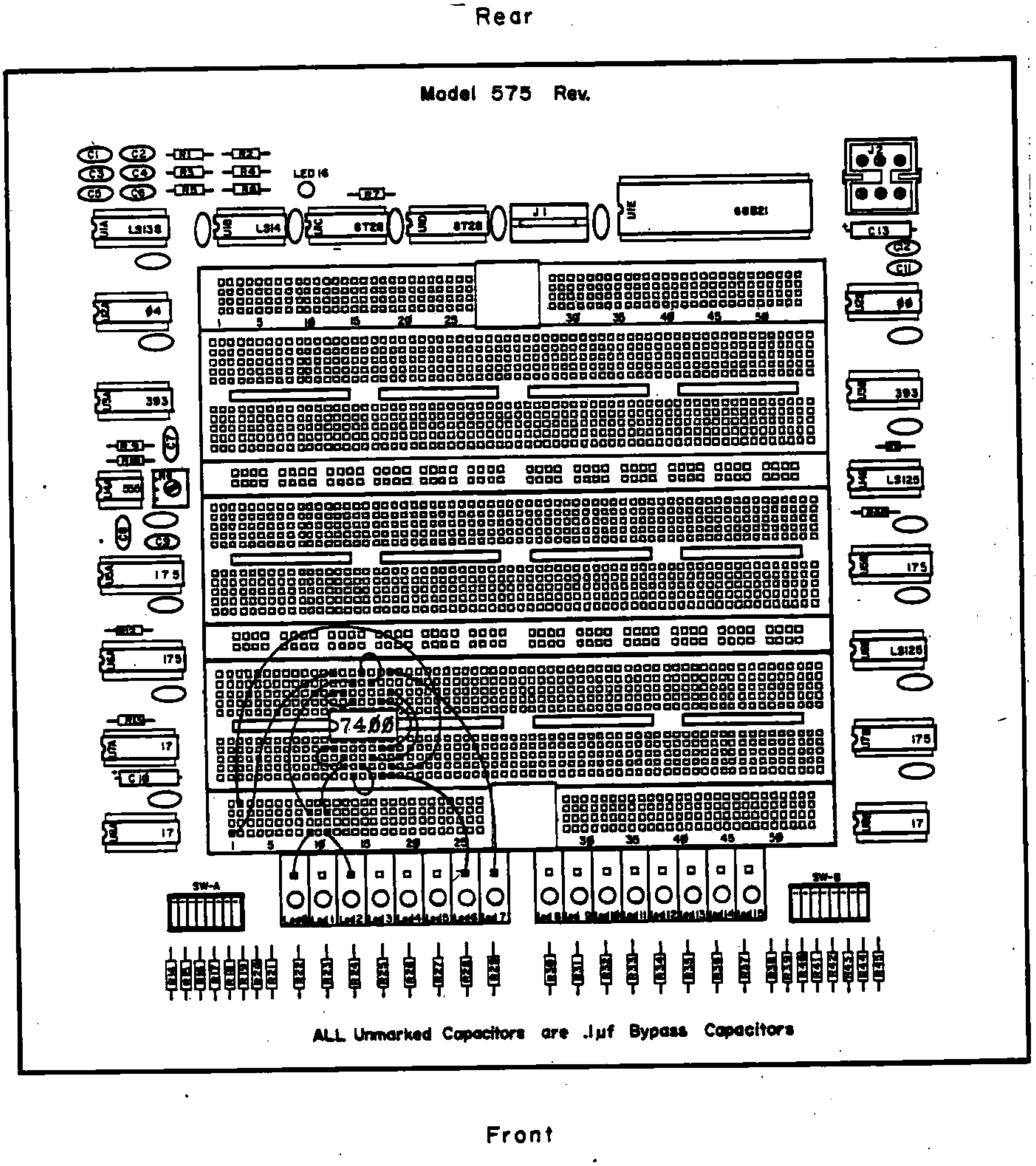


Figure 24
Switch, LED, and other Connections for an R-S Flip-Flop

representation while inserting the jumper wires. Note that power connections do not show on the schematic diagram. Be certain that you have G1 connected to pin 14 and G2 connected to pin 7.

When you are certain that all connections have been made correctly, set switches 1A and 3A to the logic "Ø" position (number end down) and turn on the power supply. Move switch 1A back and forth while noting the condition of LED6 and LED7. With switch 1A in the logic "Ø" position, move switch 3A back and forth while noting LED6 and LED7. Now go back to changing SW1A (with SW3A left at logic "Ø"). Finally, remembering that SW1A corresponds to S, SW3A corresponds to R, LED7 corresponds to Q and LED6 corresponds to $\overline{Q}$, verify the truth table on page 49. (It will not harm the circuit to have both inputs at logic "1". This condition, however, has no meaning in the functioning of the flip-flop.)

The next step is to add the control or clock feature. As can be seen from figures 23b and 25, this is quite simple to do. First, turn off the power supply and remove the wires connecting pin 12 to 13 and 1 to 2. Then use one of the wires to connect pin 13 to pin 1 and the other wire to connect the combination to terminal G1Ø. Finally, use an additional jumper wire to connect G1Ø to LED1. The result should be as shown schematically in Fig. 23b and pictorially in Fig. 25.

When the change has been made correctly, turn on the power supply. Use the following procedure to verify the enable and disable function of the signal at C. After setting switch 2A to its low condition, change the levels produced by SW1A AND SW3A and note that the signals at Q and $\overline{Q}$ do not change. (The flip-flop is disabled.) Set switch 2A to its high condition and note that now the flip-flop functions normally. Use the signal at C as a data transfer signal by resetting the flip-flop, disabling it, setting SW1A to logic "1", then moving SW2A to its logic "1" position.

The clock on the CA-24 board can be used to produce a continuous set of alternating enable-disable signals. Turn off the power supply, then remove the jumper wires which are connecting pin 1 to G1Ø and G1Ø to LED1. Use one of these to connect A3 to A4 (the clock output to the divide chain). Use the other to connect pin 1 directly to LED1. The last connection needs a short wire to connect pin 1 to G26 (the end of the divide chain). The circuit should look like that shown in Fig. 26.

When the changes have been made, turn on the power supply. LED1 should be blinking on and off. Use a screwdriver to adjust the clock frequency until the slowest blink rate is achieved.

Move switches 1A and 3A back and forth at various times during the clock cycle and observe the function of the signal at C. Note that with the flip-flop reset, a logic "1" placed at the S input (LEDØ on) when C is low (LED1 off) will not appear at Q (LED7) until C goes high. However, with the flip-flop reset, a logic "1" placed at S when C is high will immediately transfer to Q. This uncertainty as to whether the data will transfer immediately or at the beginning of the next clock cycle can be a problem. More sophisticated latch circuits remove this uncertainty. Such a latch circuit will be investigated in Experiment 11.

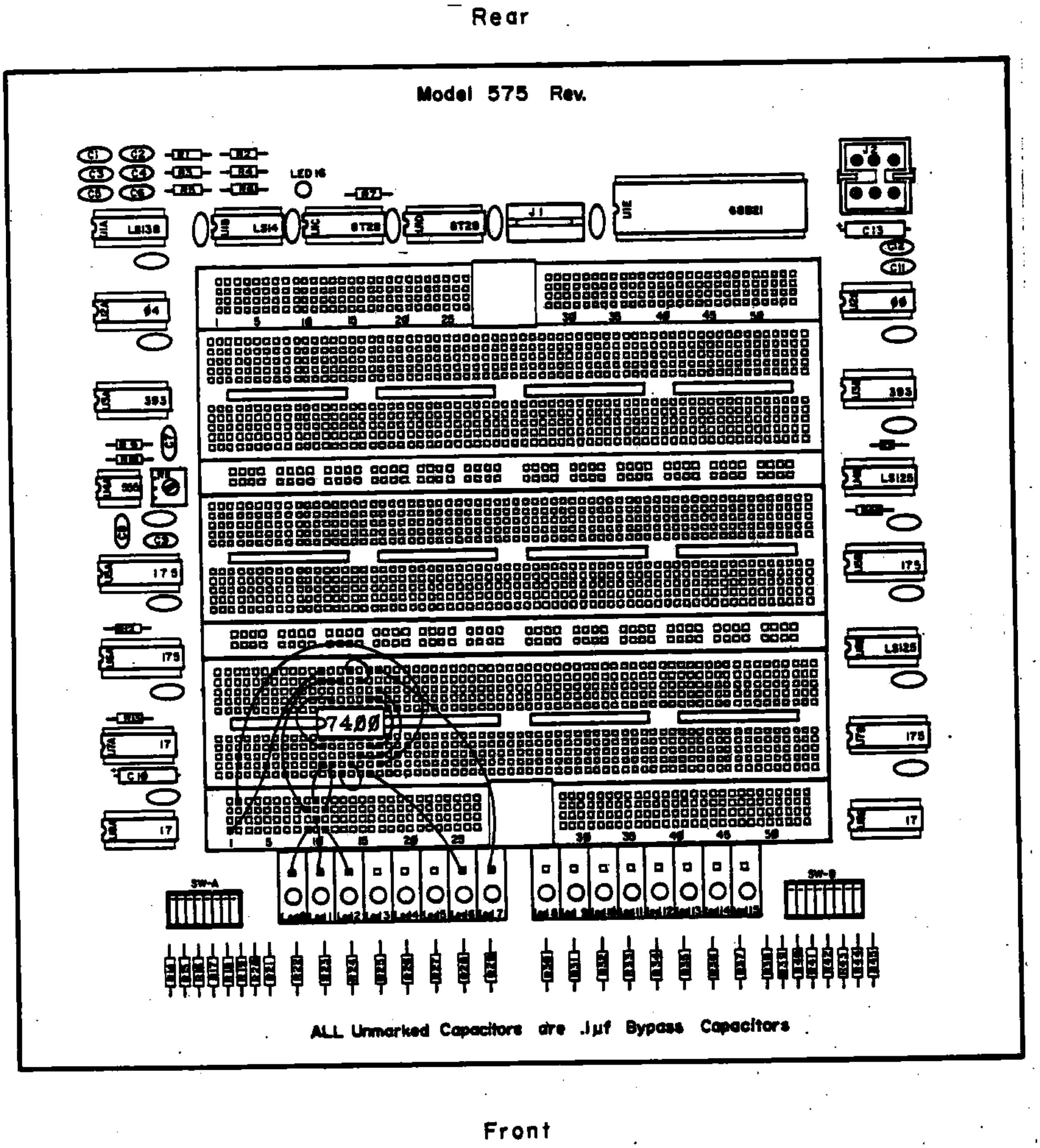


Figure 25
Switch, LED, and other Connections for an R-S Flip-Flop with Clock Input

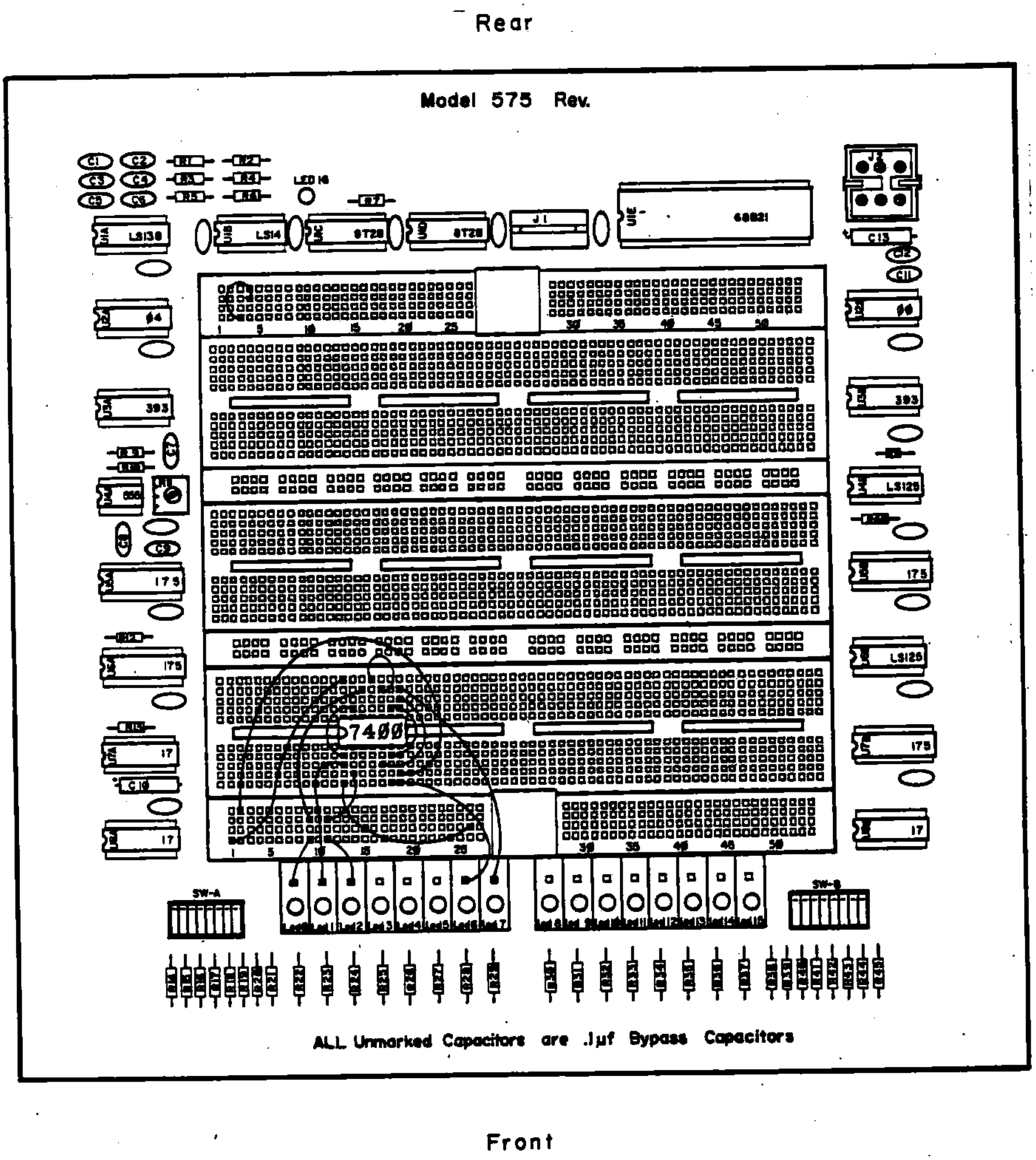


Figure 26
Connection for Use of the On-Board Clock with the R-S Flip-Flop

# EXPERIMENT 1Ø

## TITLE

The D Flip-Flop (74175)

## PURPOSE

The purpose of this experiment is to investigate the function of a D (delay) flip-flop using a flip-flop on the 74175 IC chip as an example.

## EQUIPMENT

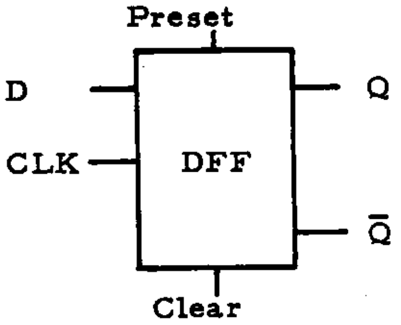
CA-24 board with power supply
Six short jumper wires
Five medium jumper wires
One 74175 IC chip

## DISCUSSION

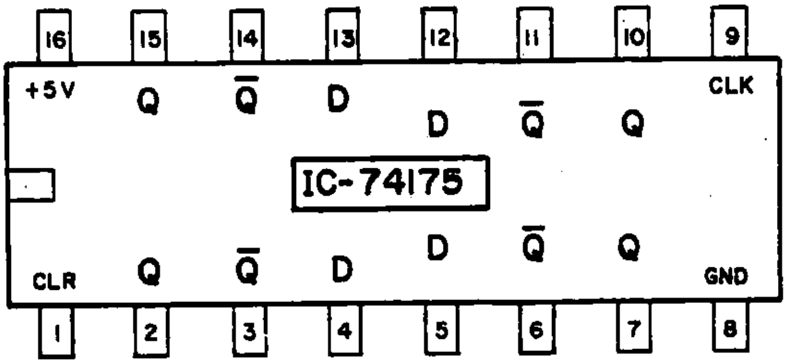
In Experiment 9 it was shown how a flip-flop circuit can transfer a logic level from an input line to an output line and hold (or latch) it there. It was noted that for the relatively simple flip-flop being studied, the logic "1" would be transferred if it appeared at S when the signal at C was high or else the transfer would wait until the signal C went high. A review of that experiment will show that it is only the logic "1" at S which can be transferred by the signal at C (unless the reset signal is changed to logic "1").

The D (delay) flip-flop (DFF) is a more complex circuit which will transfer either level of a data line when the C signal (from now on called clock signal) goes from low to high regardless of when in the clock cycle the data signal appears at the input (from now on referred to as the D input). Note that the transfer is always delayed until the next rise of the clock signal.

The general diagram for a flip-flop is shown below.

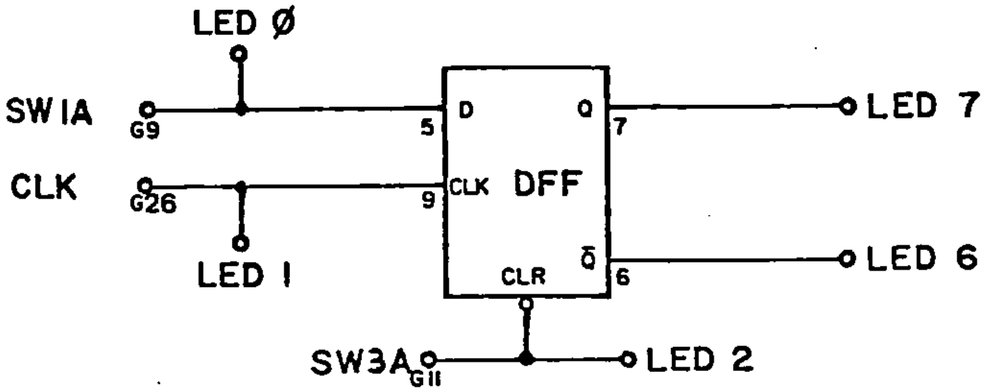


The normal transfer function occurs when both the preset and clear signals are high. If the clear signal is set low, Q is low regardless of the conditions at D and CLK. If preset is low (clear high), then Q is high regardless of the conditions at D or CLK. The CA-24 board contains four 74175 IC chips, each of which contains four D flip-flops or latches. In experiments 13 and 14 we shall examine the role these chips play in the I/O process when the CA-24 board is interfaced with the computer. In this experiment the extra 74175 IC chip provided with this manual is used to study the operation of a D flip-flop. The circuits in the 74175 chip are not completely independent but have a common clock line and a common clear line. A further modification is the elimination of the preset input, which is not needed for many applications. The following diagram shows which pins of the 74175 correspond to which signals for the flip-flops. Each set of Q, $\overline{Q}$, and D pins corresponds to one of the latches.



PROCEDURE

Be certain that the power supply to the CA-24 board is off. Remove any chips and wires left on the board from previous work. Insert the extra 74175 IC chip provided with this manual as shown in Fig. 27. The schematic diagram below shows how one of the latches in the chip is to be connected. Not shown on the diagram is the connection of terminal A3 to A4 which connects the clock to the divide chain and, or course, the power connections to the chip. Note that this chip has 16 pins, rather than 14 like the chips used in the previous experiments.



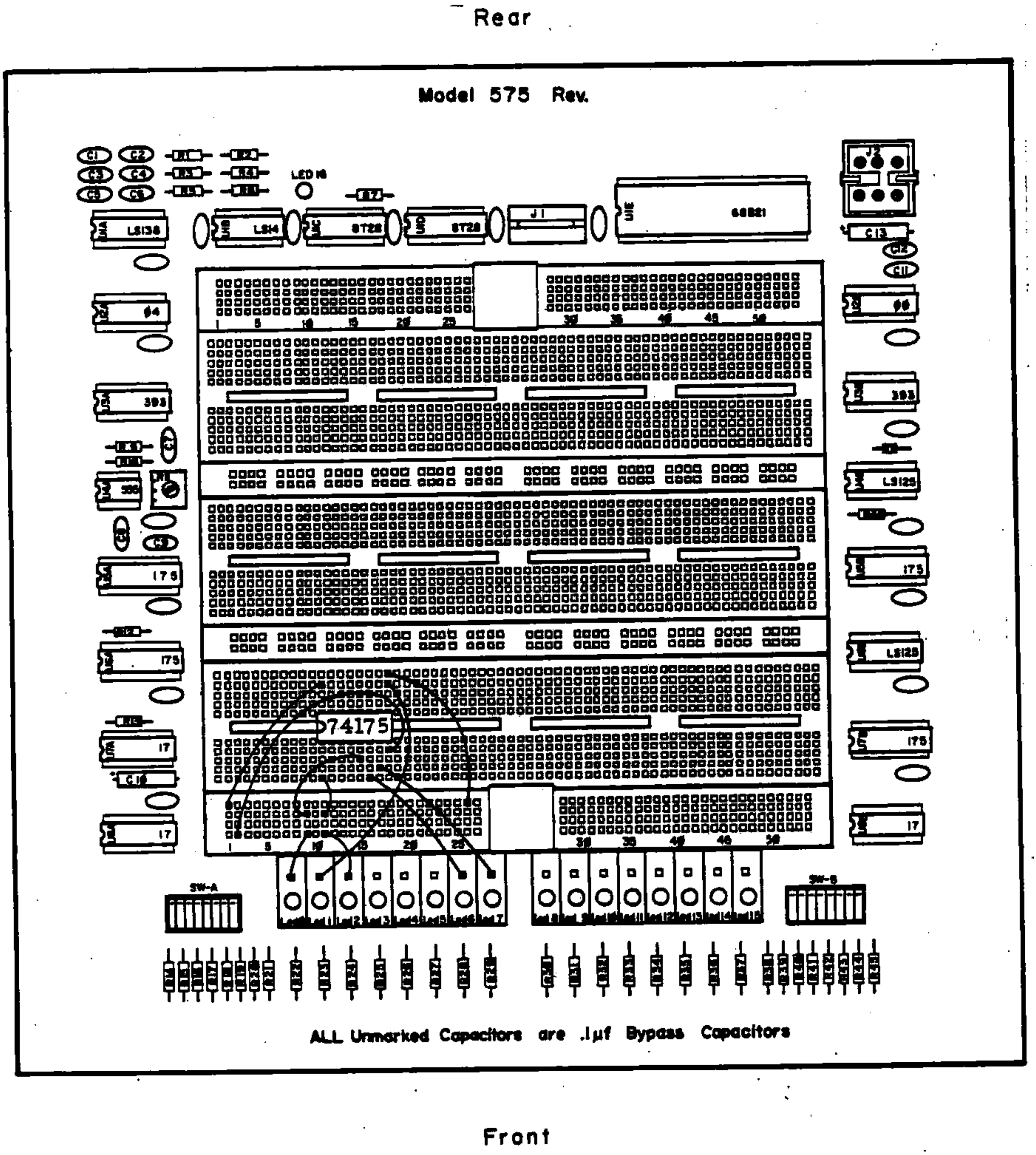


Figure 27
Switch and LED Connections to a D Slip-Flop

Use the various lengths of jumper wires where appropriate to make the connections. Fig. 27 shows how the wires should be positioned.

When you are certain that all connections have been made correctly, turn on the power supply. LED1 should begin blinking on and off.

Move switch 3 to its logic "1" or high position. The status of this switch will show on LED2 which should now be glowing. Move switch 1 from low to high and from high to low at various times during the clock cycle and note the behavior of LED7 (the indicator for Q) and LED6 (the indicator for $\overline{Q}$). Your observation should verify that the data signal (setting of SW1A) is transferred to Q only when the clock signal rises (i.e., only when LED1 turns on).

Set SW1A to its logic "1" condition. Move SW3A to its low position at any time during the clock cycle. Note that LED7 will turn off immediately. The clear (logic "Ø") signal will set Q to logic "Ø" as soon as it is applied regardless of when in the clock cycle it arrives.

Move SW3A back to its high position and watch LED7. If you have left SW1A in its logic "1" condition you will observe LED7 turn on the next time the clock signal rises.

The 74175 IC chip is one of two chips on the CA-24 board used for transfer of information. The other chip is the 8T28, two of which are located along the back edge of the CA-24 board. The function of this chip will be investigated in Experiment 11.

# EXPERIMENT 11

## TITLE

The 3-State Quad Bus Transceiver (8T28)

## PURPOSE

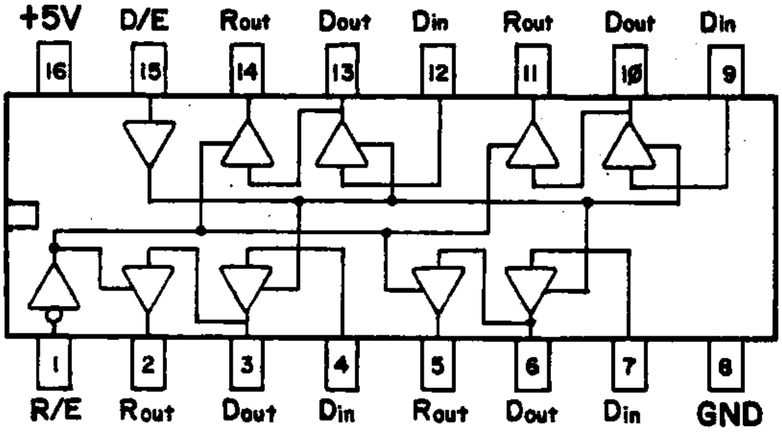
The purpose of this experiment is to examine the function of the 8T28 3-state quad bus transceiver as it is used on the CA-24 board.

## EQUIPMENT

CA-24 board with power supply
Five short jumper wires
Five medium jumper wires
One 8T28 IC chip

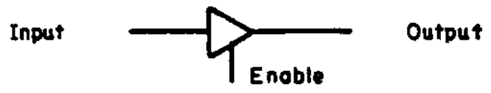
## DISCUSSION

The 8T28 3-state quad bus transceiver is used in bus oriented systems to route data from one location to another. A bus in digital systems is a path over which digital information is transferred from any of several sources to any of several destinations. The CA-24 board has two 8T28 chips located along the rear edge of the board. Each chip can handle four data lines. The two chips, then, can route data along eight data lines. As used on the CA-24 board, the two 8T28 chips permit transfer of data from the computer to an external device or transfer of data from the device to the computer. The schematic diagram of the chip together with pin numbers and manufacturer's labels is shown below.



In the schematic diagram of Fig. 3a, the 8T28 chips are at the left. It will be noted that the pin labels in Fig. 3a differ from the manufacturer's labels. In the PROCEDURE section of this experiment where you will set up the circuit as it is used on the CA-24 board for one data line, we will use the labeling found in Fig. 3a instead of that shown above.

On the schematic diagram of the chip, all but two of the identity elements (described in Experiment 1) are seen to have a third line associated with them. The symbol is shown below.



As the labels imply, the signal at the input appears at the output only when an enable signal (+5 V) is applied to the third line. If there is a Ø V signal on the enable line, the output appears as though it were an open circuit and so has no effect on a line to which it is attached.

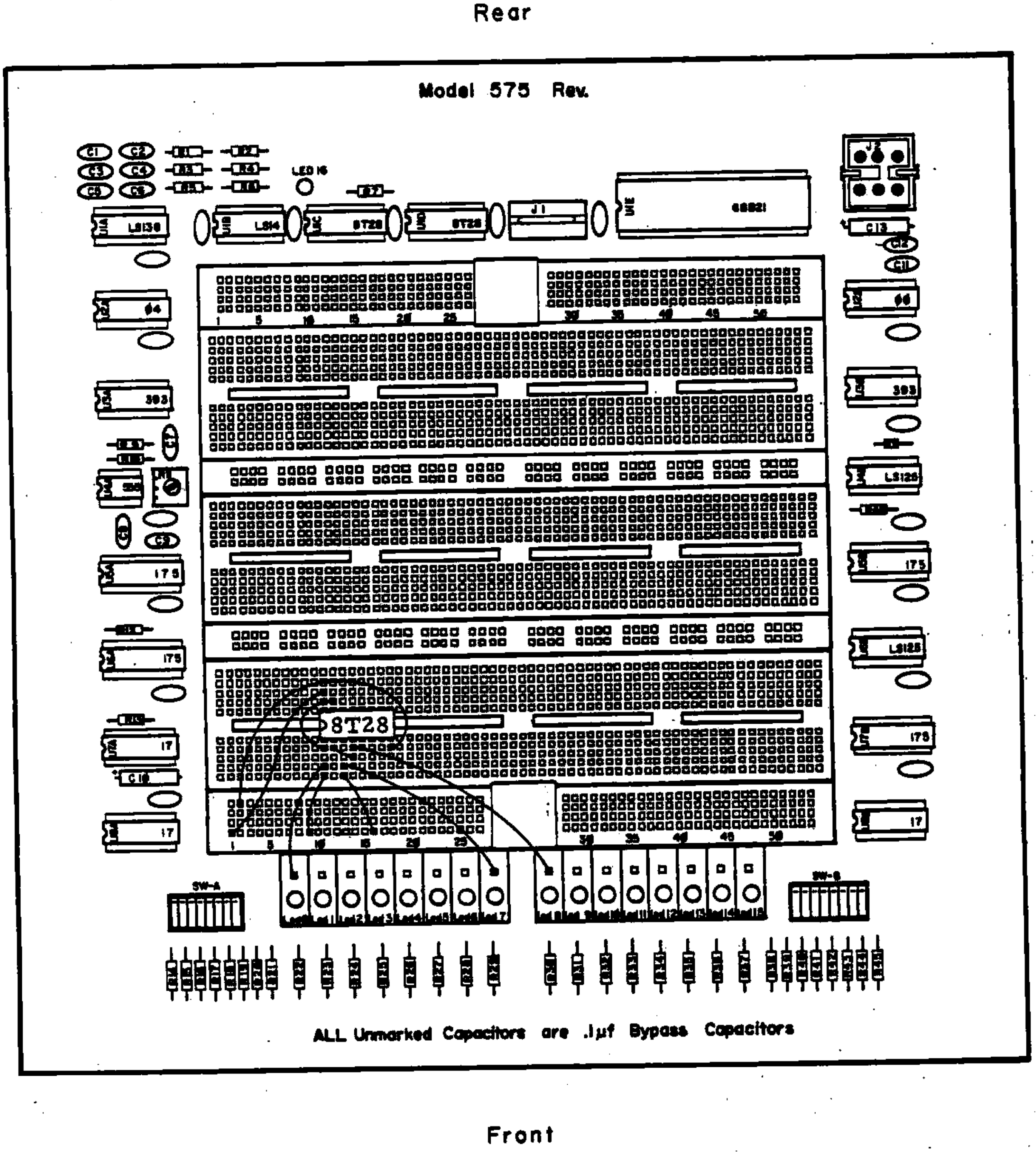
It will be noted that the enable lines for four of the identity elements come from pin 1 and the enable lines for the other four come from pin 15. However, the signal at pin 1 is inverted before going to the identity elements, whereas the signal at pin 15 is not inverted. As we will see, this feature makes it possible for the two pins (1 and 15) to be connected together so that a signal of Ø V at the combination will permit the 8T28 to receive data from the computer for use on the CA-24 board while a signal of 5 V at the combination will allow the 8T28 to pick up data from the CA-24 board for use in the computer.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Remove all wires and circuit components left on the board from previous work. Insert the extra 8T28 IC chip provided with this manual as shown in Figs. 28 and 29. Use the appropriate lengths of jumper wires to make the connections shown pictorially in Fig. 28 and schematically in Fig. 3Øa.

Note that pins 1 and 15 are connected together and receive the signal from SW1A which will be displayed on LEDØ. This signal will be the enable level which will determine the direction of data transfer. LED8 will represent the data line going to or from the peripheral device.

The circuit of Figs. 28 and 3Øa is set up to show transfer of data from the computer to a peripheral device which can be the LED displays on the CA-24 board or some device connected to the CA-24 board. Set SW1A to its logic "Ø" condition, then move SW8A back and forth. Note that LED8 follows the changes of LED7 showing that the signal on the "computer line" is being transferred to the "peripheral line" (compare Fig. 3Øa).



ALL Unmarked Capacitors are .1µf Bypass Capacitors

Figure 28
Switch and LED Connections to a Three-State BUS Transceiver:
Computer to Peripheral Transfer

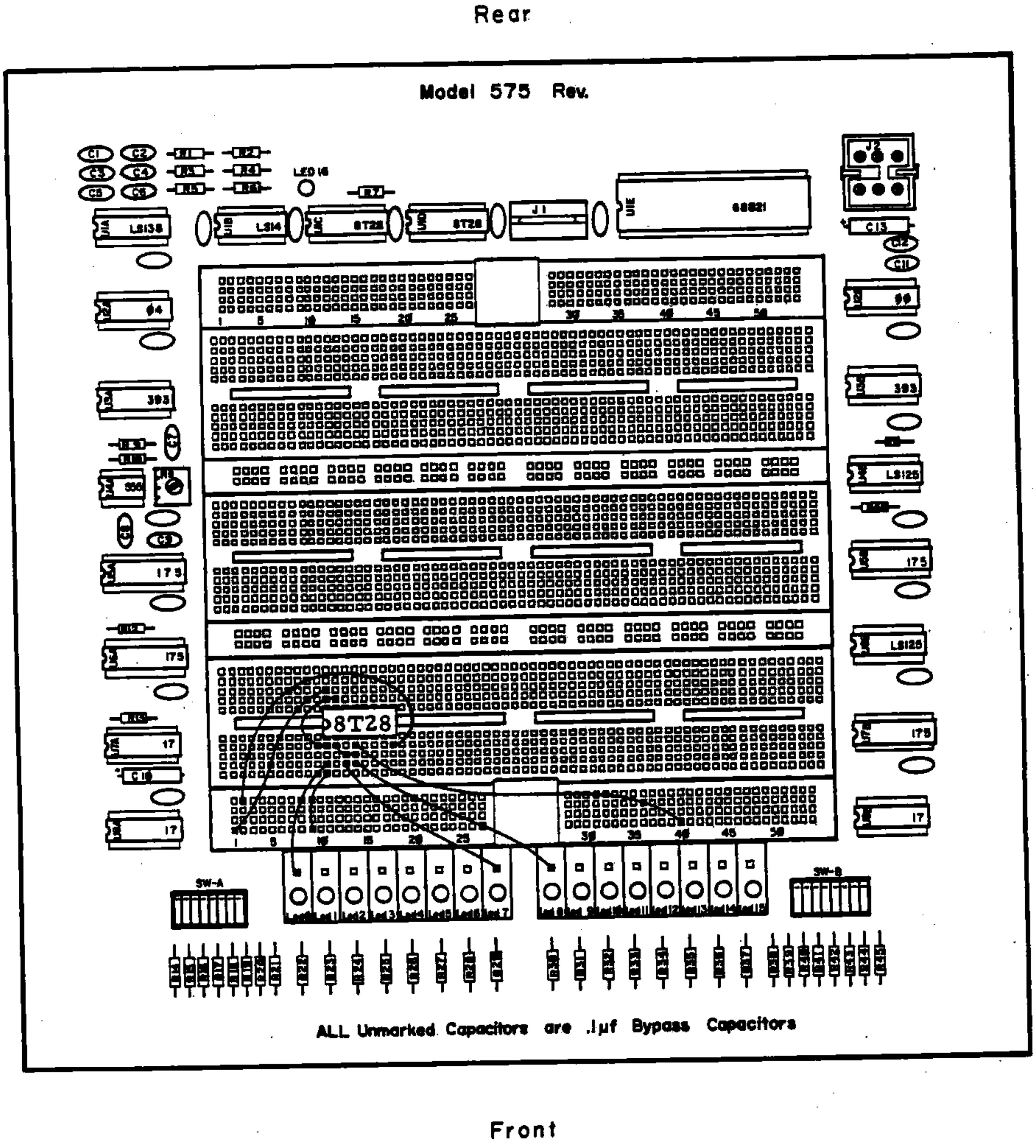


Figure 29
Switch and LED Connections to a Three-State BUS Transceiver:
Peripheral to Computer Transfer

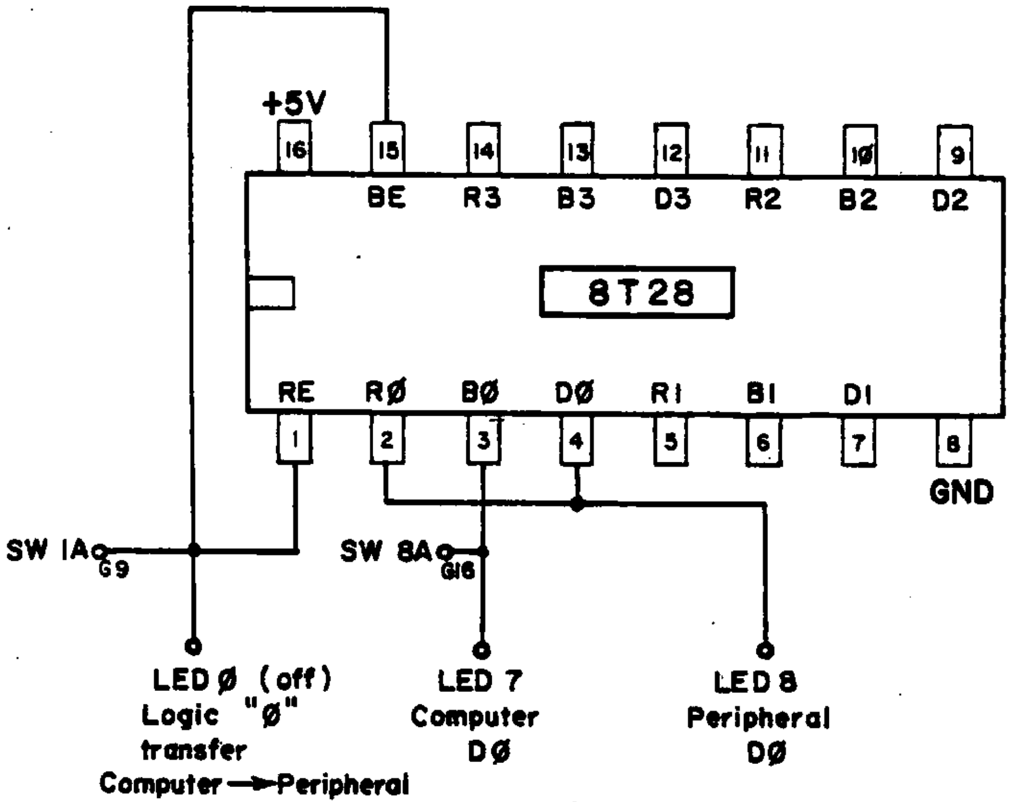


Figure 30a
Schematic for Computer to Peripheral Transfer Across a
Three-State BUS Transceiver

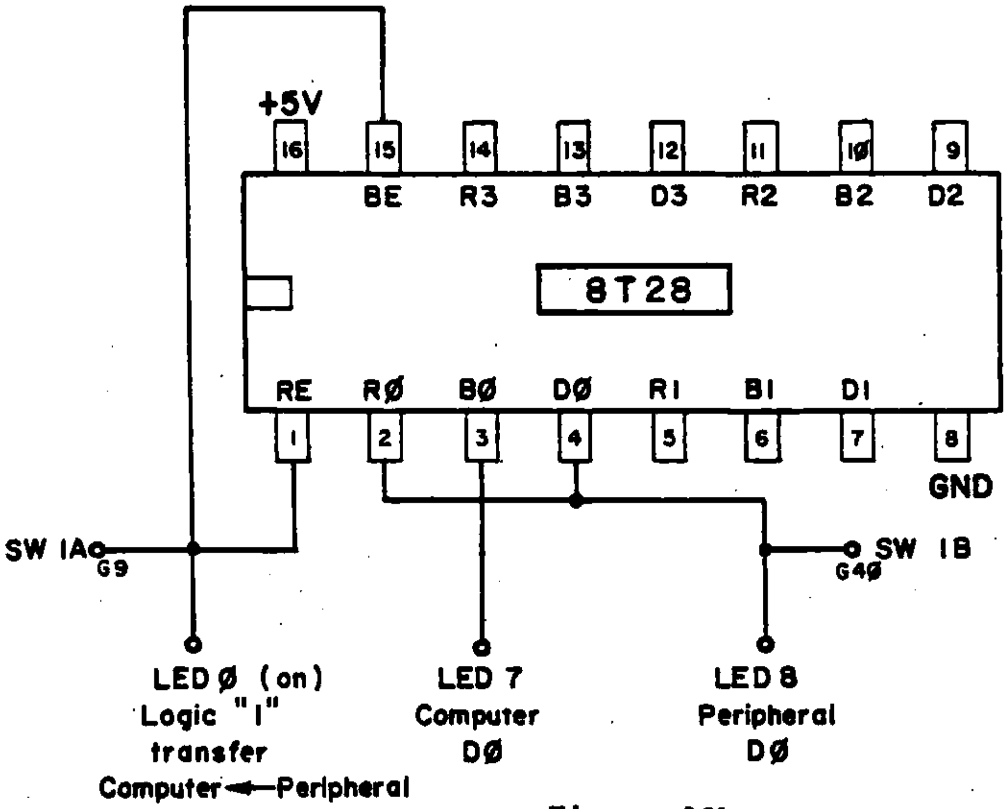


Figure 30b
Schematic for Peripheral to Computer Transfer Across a
Three-State BUS Transceiver

Now move SW1A to its logic "1" position and again move SW8A back and forth. Note that LED8 no longer follows the changes of LED7 but stays turned on. The signal on the "computer line" is now blocked.

Next, change the circuit by removing the wire connecting pin 3 to terminal G16 and inserting a wire to connect pin 4 to terminal G4Ø (see Figs. 29 and 3Øb). With SW1A still in the logic "1" position, move SW1B back and forth. Note that LED7 follows the changes of LED8 showing that the signal on the "peripheral line" is being transferred to the "computer line" (compare Fig. 3Øb).

Finally, move SW1A back to the logic "Ø" position and again move SW1B back and forth. Note that LED7 no longer follows the changes of LED8 but stays turned on. The signal on the "peripheral line" does not get through.

This experiment illustrated the transfer conditions for only one of the four data lines possible with the 8T28 IC chip. Reference to Fig. 3a will show that for the two 8T28 IC chips on the CA-24 board we have modeled the signal transport on either data line DØ or D4. You may wish to make additional connections to construct a circuit which will illustrate the ability of the 8T28 chip to route data along as many as four lines.

EXPERIMENT 12

TITLE

Contact Bounce and the Monostable Multivibrator or One-Shot (74121)

PURPOSE

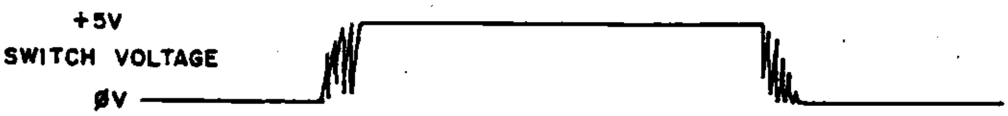
The purpose of this experiment is to "observe" switch contact bounce, construct a monostable multivibrator (one-shot) circuit to mask the contact bounce, and investigate the function of the one-shot inluding measuring the time width of its output pulse.

EQUIPMENT

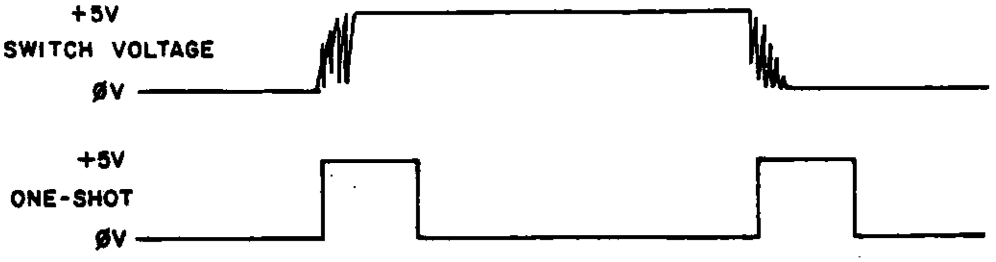
CA-24 board with power supply
Twelve short jumper wires
Nine medium jumper wires
Nine long jumper wires
One 74121 IC chip
One 74ØØ IC chip
One 1Ø$\mu$F capacitor
One 33 k$\Omega$ resistor

DISCUSSION

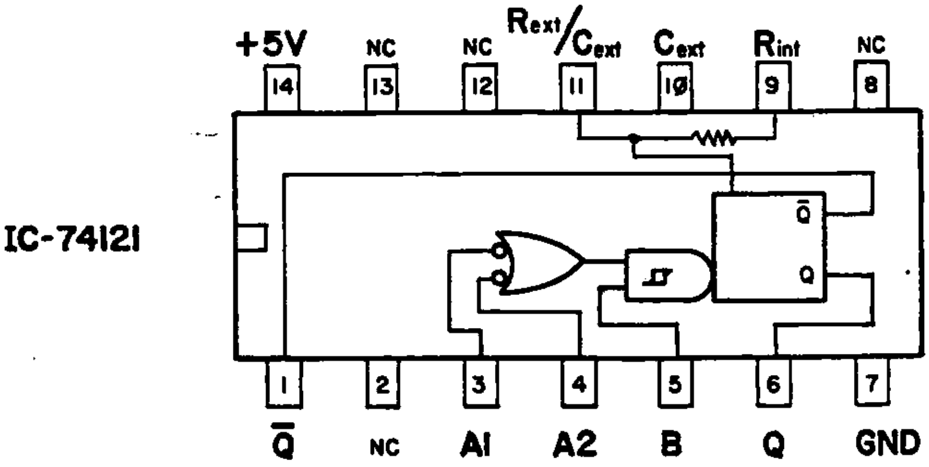
Whenever a mechanical device is coupled with an electrical circuit, timing problems arise. Mechanical devices are slow and frequently are erratic in behavior compared to electronic components. A particular problem is found with switches. During either contact closure or separation, a series of rapidly varying open and closed conditions always occurs which generates many voltage pulses. This situation is known as contact bounce. A graphical representation of the problem is shown below as it would appear on an oscilloscope trace (time coordinate along the horizontal). The series of pulses during contact closure or separation may last for several milliseconds. In this experiment we will use the divide-by-two chain on the CA-24 board to count the number of pulses during contact bounce.



In digital circuits where the transfer of information depends on clean voltage rises and falls between each logic state, contact bounce must somehow be eliminated. A common way of doing this is to use an electronic circuit called a monostable multivibrator or one-shot. The function of this circuit is to generate a clean Ø to 5 V output transition (for positive logic) at the first rise of input signal and then hold the 5 V level at the output for a duration of time determined by a resistance-capacitance combination which is a part of the circuit. Ideally, the switch can be returned to its Ø V condition after the cycle of the one-shot without a re-triggering of the output. In practice, however, as we shall see, the "bounce" at contact separation may be so bad that one of the spikes is interpreted by the one-shot as a rising activation signal. The timing sequence is shown below.



Monostable multivibrators can be constructed from NAND gates. However, the 74121 IC chip contains a versatile one-shot which is convenient to use and serves well the purposes of this experiment. The diagram for the 74121 chip is shown below.



Inputs A1 and A2 (pins 3 and 4) are inverted before entering an OR gate. The output of the OR gate together with the B input (pin 5) form the input for a Schmitt trigger AND gate. The Schmitt trigger gate differs from the gates we have studied previously in that activation of the gate depends only on the voltage level of the incoming signal and so is independent of how rapidly the signal is rising or falling. For purposes of this experiment, inputs A1 and A2 will be connected to GND so that the output of the OR gate is always high. Consequently, a logic "Ø" to logic "1" transition at B will activate the one-shot.

The duration (time width) of the output pulse at Q or $\overline{Q}$ will be determined by a capacitor connected between pins 1Ø and 11, and a resistor connected between pin 11 and +5 V. For resistance values (R) and for capacitance values (C) in the range used in this experiment, the pulse width is approximately given by the following equation:

pulse width = .7*R*C

For the resistance and capacitance values suggested for this experiment (33 kΩand 1Ø μF) the pulse width should be somewhat over Ø.2 sec.

The clock on the CA-24 board will be used for two purposes. We will use the clean transitions of the clock signals to show that the one-shot triggers only on transition from low to high. Secondly, we will use the known frequency of the clock (determined in Experiment 3) to measure the time duration of the output pulse.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Remove any chips and wires left on the board from previous work.

As indicated earlier, the divide-by-two chain on the CA-24 board can be used as a counting device. If a series of pulses is fed into the chain at A4, and the output from each divide-by-two circuit (A5, A6, A7, etc.) is connected to an LED, then the lights can be used as a binary number display.

Use eight long jumper wires to connect A5 through A7 to LED8 through LED1Ø, and G3 through G7 to LED11 through LED15 as shown in Fig. 31. Note the reverse order of connection for G4 through G7 (i.e., G4 connects to LED15, G5 to LED14, etc.). The ninth long wire connects G9 with A4 and the short wire connects G9 with LEDØ.

Before turning on the power supply, carefully read the next two paragraphs.

When the board is first turned on, all LED's will light. (Of course, LED1 through LED7 will be on because there is no connection to them, and the condition of LEDØ will depend on the setting of SW1A.) When SW1A is changed, a series of pulses will enter the divide chain. The first pulse in the series will cause LED8 through LED15 to turn off. This first pulse can be considered as a reset signal. The next pulse in the series will light LED8. When the next pulse enters, LED8 will turn off and LED9 will turn on. Then, with the next pulse, LED8 will come on, and next both 8 and 9 will turn off and 1Ø will come on, etc. All this will happen in a few milliseconds and so you will not see the individual changes.

After the entire pulse train has activated the divide chain, an on-off pattern will be displayed on LED8 through LED15. If each higher LED is interpreted as the next higher power of two starting with LED8 as two to the zero power, then the sum of the powers of two for the lighted LED's will be the decimal equivalent of the binary display. For example, if after

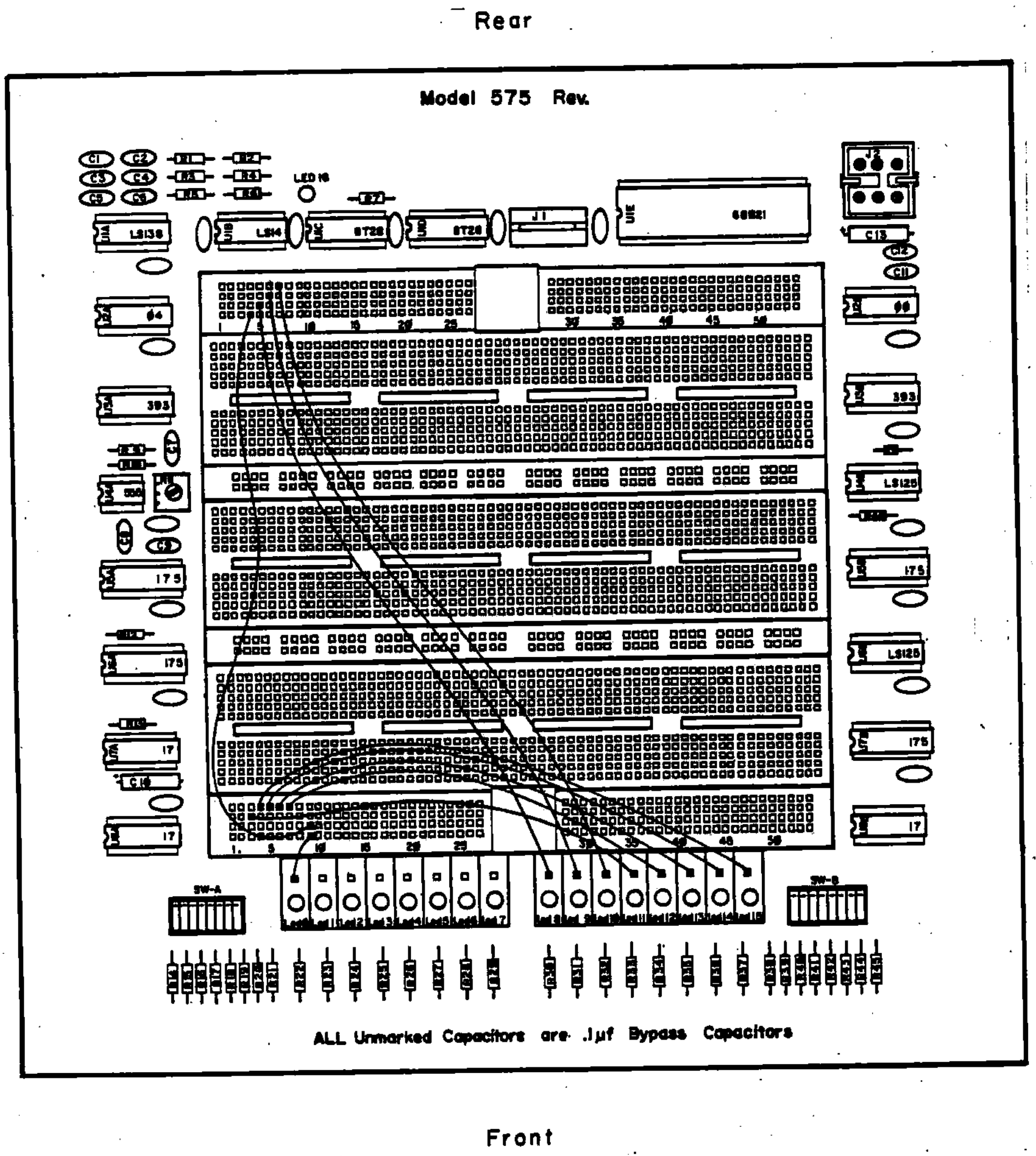


Figure 31
Connections for Use of LED's to Count Contact Bounce

turning on the power supply and changing SW1A (which enters a series of pulses), LED's 9, 11, and 12 are glowing, the number of pulses entered is 2 + 8 + 16 or 26 plus the reset pulse.

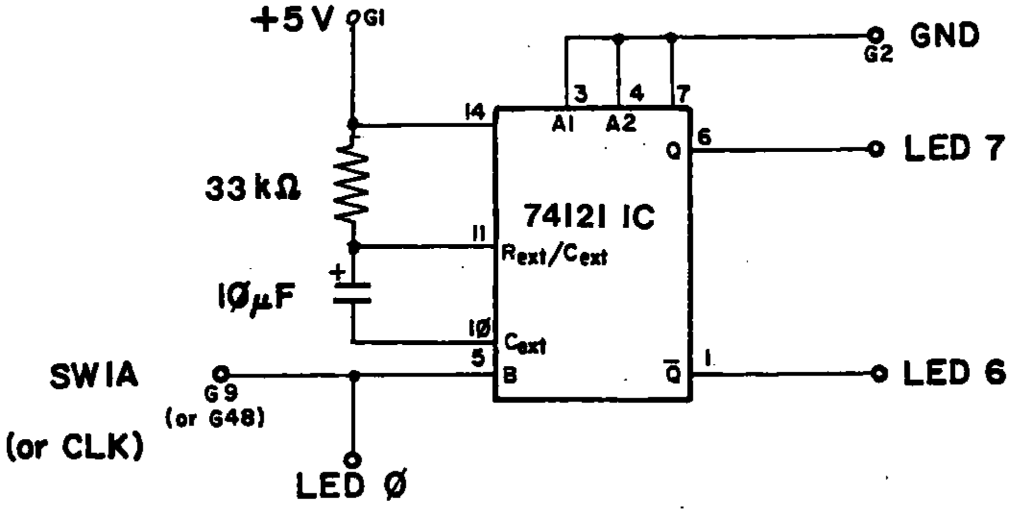
Now turn on the power supply to the CA-24 board and change the position of SW1A. Note which LED's are glowing (ignoring, of course, LED's Ø through 7) and determine how many pulses entered. Turn the power supply off, then back on again to set all LED's on. Again change the position of SW1A and determine the number of "bounces".

If you wish to measure the contact bounce for other switches, simply change the wire which connects A4 to G9 so that it connects A4 to G1Ø, then G11, etc. (The LEDØ wire can also be changed, although that connection is not necessary for counting.)

Note that the set of LED's can count up to 256 pulses. It is not likely that your switches will produce that many.

Since the counting circuit will be used later, you may wish to leave the long wires in place while making connections for the 74121 IC chip.

Turn off the power supply. Insert the 74121 chip as shown in Fig. 32. Use the short and medium wires as appropriate to connect the one-shot as shown in the schematic diagram below and pictorially in Fig. 32. (Note that the left end connections of the wires from LED's 11 through 15 are not shown in Fig. 32. They still go to G3 through G7 as shown in Fig. 31).



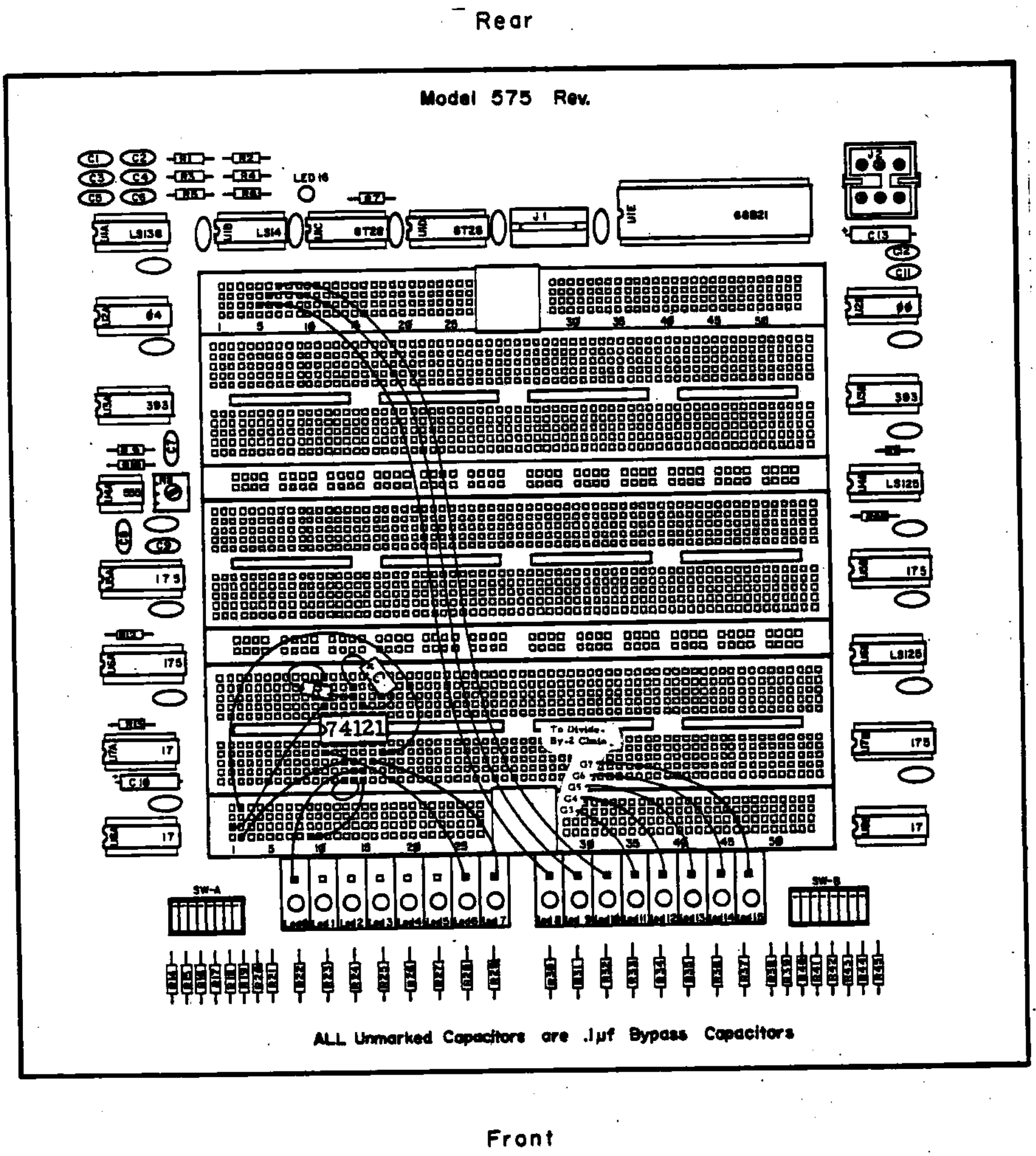


Figure 32
Connections for Use of SW1A with a Ø.2 Sec Monostable Multivibrator

Insert the 33 kΩ resistor from pin 11 to +5 V and the capacitor from pin 1Ø to pin 11. Note the polarity for the electrolytic capacitor (the plus end should be connected to pin 11 and the negative end should be connected to pin 1Ø).

When you are certain that the connections have been made correctly, turn on the power supply. Change the position of SW1A and note the behavior of LED7 and LED6. LED7 should turn on and LED6 off for about Ø.2 sec, then revert to their former condition.

Move SW1A back and forth and note that the one-shot is activated whether the switch is moved from logic "Ø" to logic "1" or the other way.

Now test the one-shot using the lowest frequency signal from the CA-24 clock. Turn off the power supply, then move the wire from pin 5 to G9 so that it connects pin 5 to G26. Then insert a jumper between A3 and A4 (see Fig. 33). Rotate the R11 screw full clockwise (lowest frequency).

Turn on the power supply and observe LEDØ (the condition of the activating signal), LED7 (signal Q), and LED6 (signal $\overline{Q}$). Note that now the one-shot output pulse occurs only for transition from logic "Ø" to logic "1".

We can now use the one-shot to obtain a clean pulse from a mechanical switch and thereby use the switch to advance our counting circuit one light at a time.

Turn off the power supply. Re-connect pin 5 to G9. Remove the A3 to A4 jumper wire and move the wire which goes to LED7 so that it connects pin 6 with A4. The new circuit is shown in Fig. 34.

Turn on the power supply. LED's 8 through 15 will glow. Change the position of SW1A once. LED8 through LED15 should turn off. Now move the switch back and forth and observe the display advance by one with each change.

We are now ready to set up the timing circuit for the one-shot pulse width. The circuit is shown schematically in Fig. 35 and pictorially in Fig. 36. From the schematic it can be seen that the one-shot pulse is being used as an enabling signal for a NAND gate in a 74ØØ chip. While enabled, the gate will pass the clock pulse train to the divide-by-two circuits which together with the LED's form a sixteen digit binary display counter which can store a decimal number up to 65535 (two raised to the sixteenth power less one).

Turn off the power supply. Many changes are necessary for this last circuit, so you may wish to remove all wires except those immediately surrounding the 74121 chip and then re-wire. Refer both to the schematic and to the pictorial diagram as you make the connections.

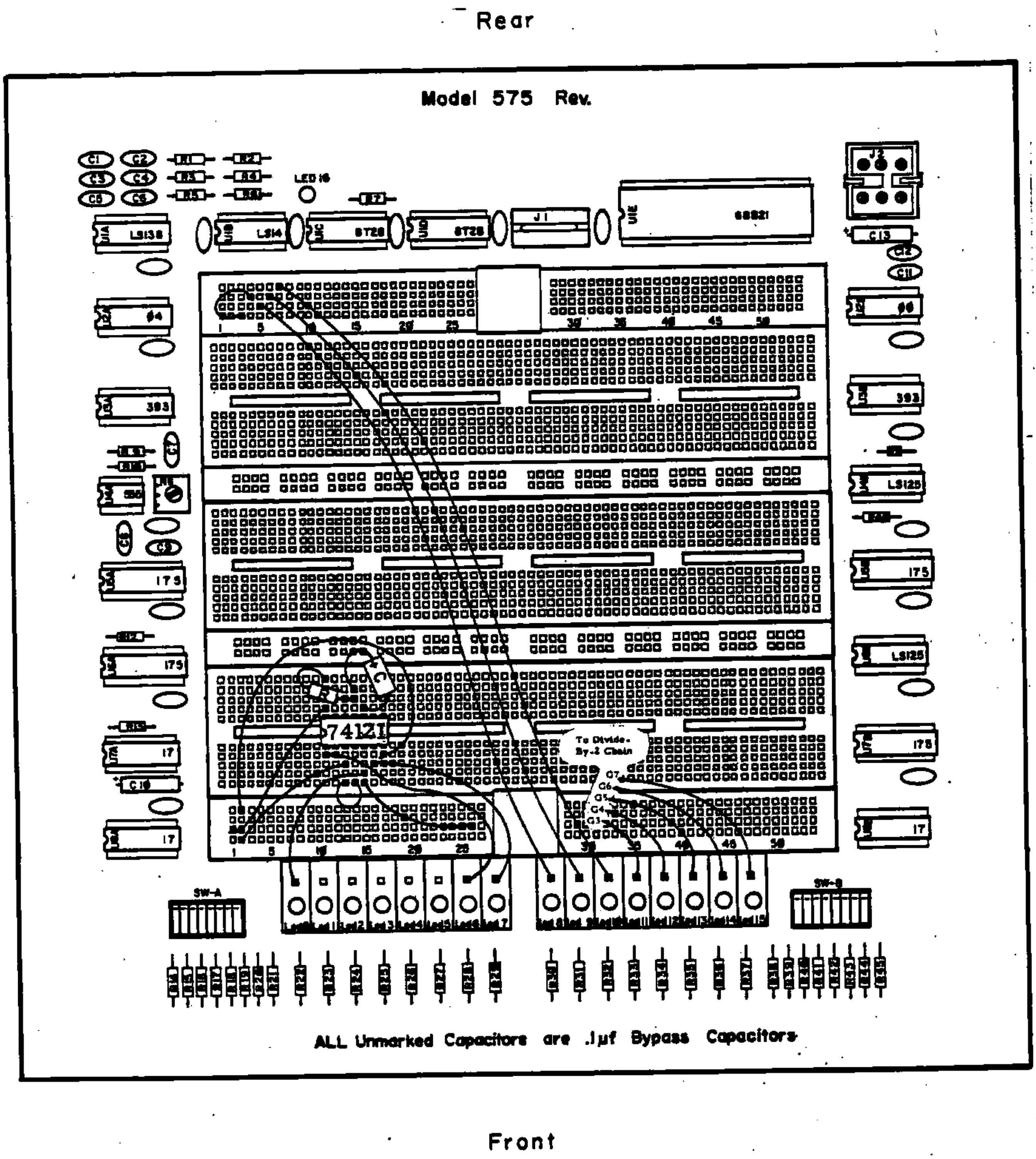


Figure 33
Connections for Use of Clock Signal with a Ø.2 Sec Monostable Multivibrator

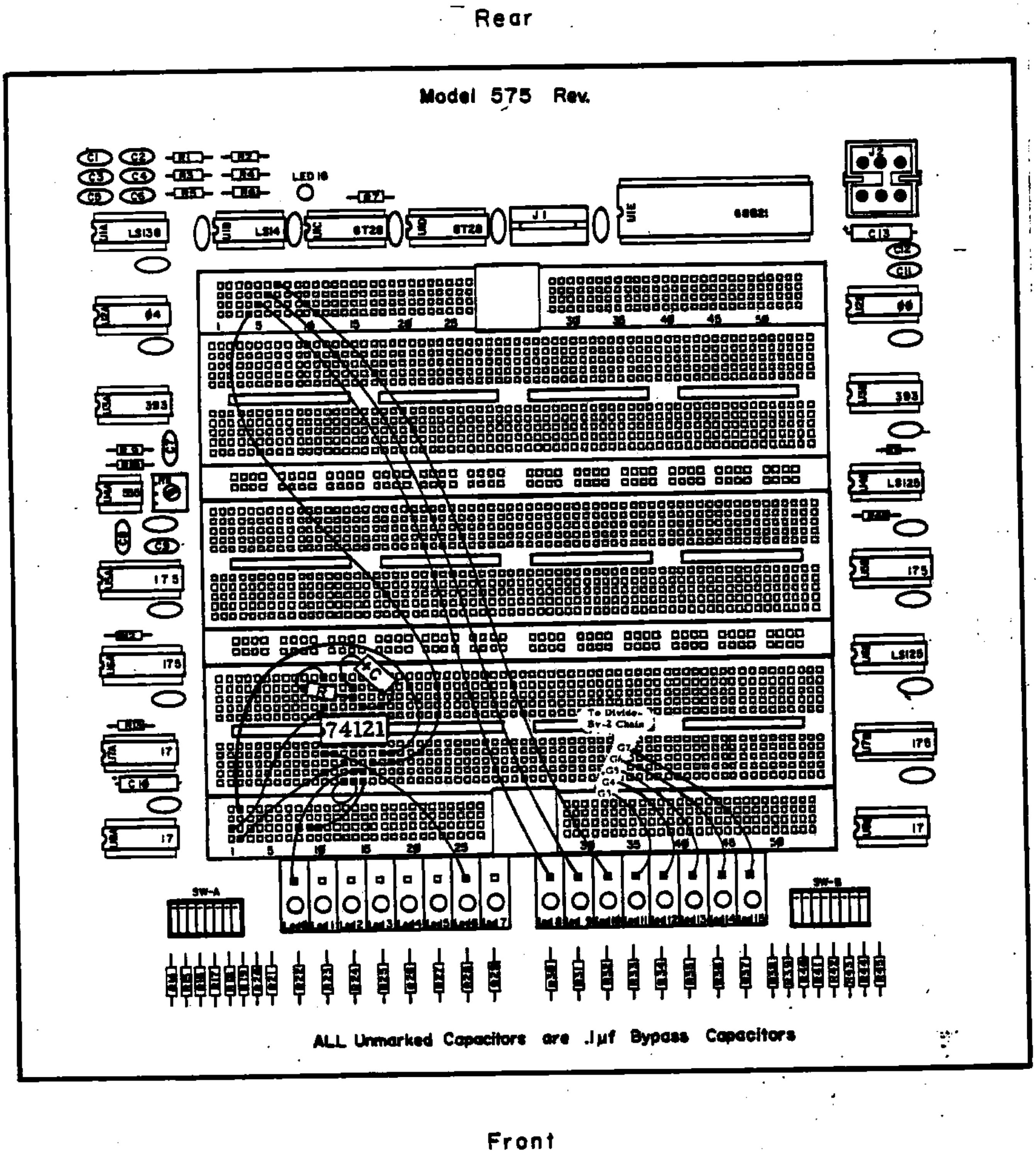


Figure 34
Connections for Use of SW1A and the Monostable Multivibrator to Advance the Counting Circuit

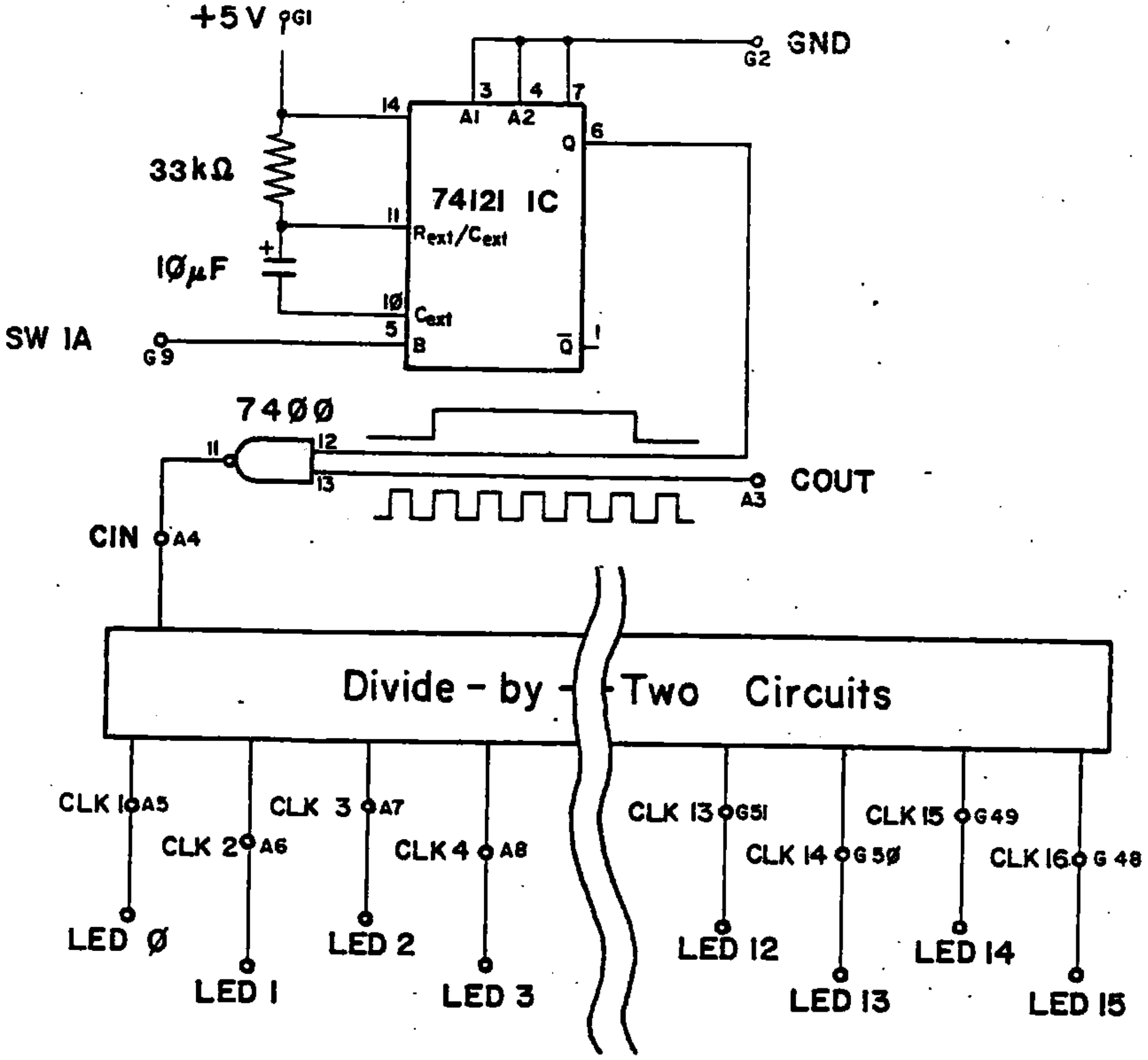


Figure 35
Schematic for Timing the Monostable Multivibrator Pulse Width

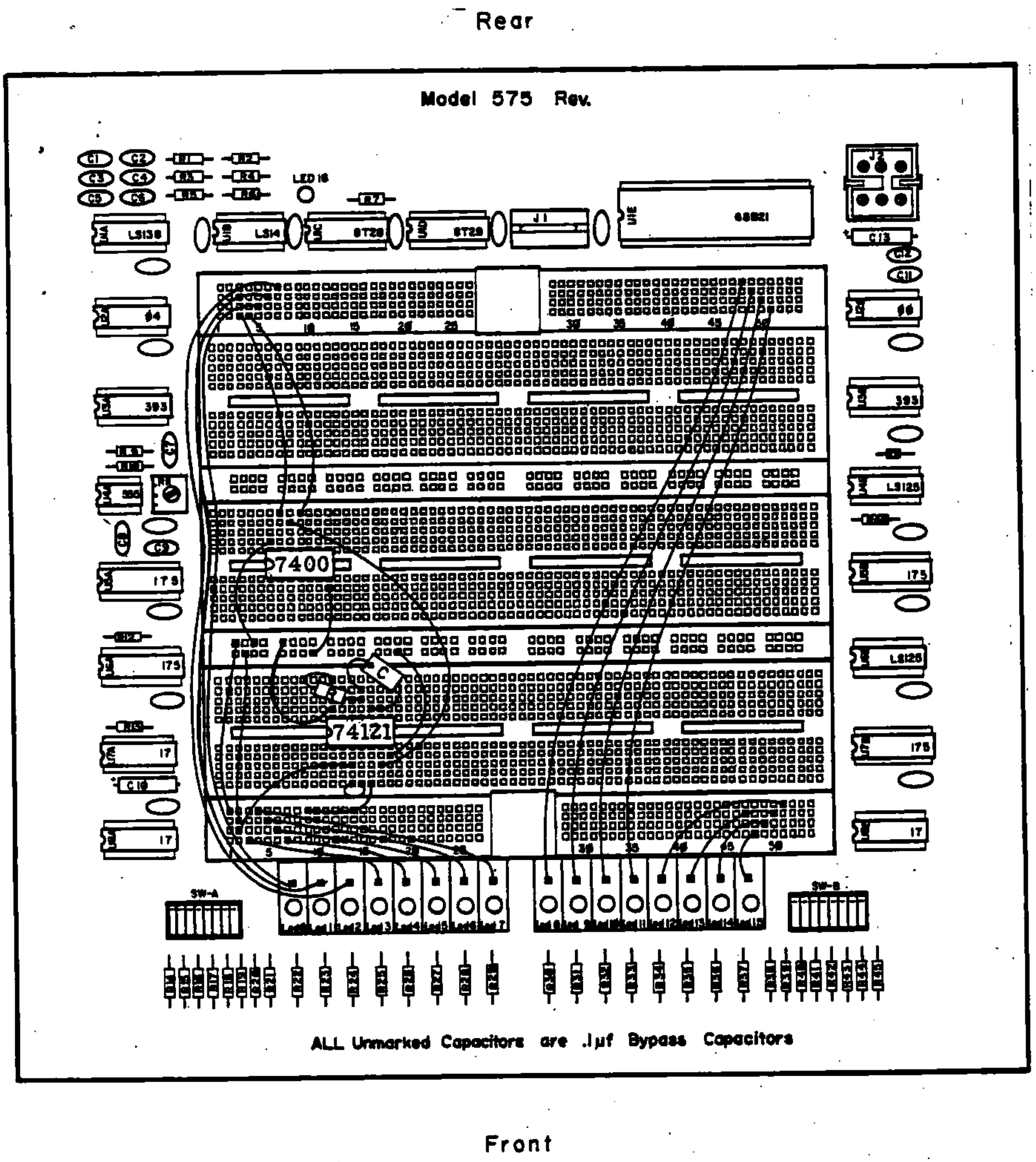


Figure 36
Connections for Circuit Used to Time the Monostable
Multivibrator Pulse Width

When you are certain that all connections have been made correctly, turn on the power supply and turn the frequency adjust resistor, R11, full counter-clockwise for maximum frequency. All of the LED's will be glowing. Change SW1A to trigger the one-shot. Note which LED'S are glowing. From the table below, write down the numbers corresponding to the glowing LED's, then add the numbers.

| LED | VALUE | LED | VALUE | LED | VALUE | LED | VALUE |
|---|---|---|---|---|---|---|---|
| Ø | 1 | 4 | 16 | 8 | 256 | 12 | 4Ø96 |
| 1 | 2 | 5 | 32 | 9 | 512 | 13 | 8192 |
| 2 | 4 | 6 | 64 | 1Ø | 1Ø24 | 14 | 16384 |
| 3 | 8 | 7 | 128 | 11 | 2Ø48 | 15 | 32768 |

To be accurate, of course, you should add 1 for the first reset pulse. However, you will find that subsequent trials will not yield exactly the same number. This is due to the small variability inherent in resistance-capacitance (RC) timing circuits.

For each new trial, turn off the power supply, then turn it back on to start again with all LED's glowing. Alternatively, without resetting, you can take the difference between successive trials to determine the count for each trial.

In order to determine the one-shot pulse duration, you must know the pulse rate or frequency of the clock. This was determined in Experiment 3 and, if you completed that experiment, the value for the maximum pulse rate is recorded in the table toward the end of the experiment. Divide the number of pulses counted by the pulse rate value to determine the one-shot pulse width.

Do not be concerned if the value measured is significantly larger than the calculated value of Ø.23 sec. Electrolytic capacitors can have tolerances of -1Ø% to +75%.

# SECTION III

## INTERFACING WITH THE COMPUTER

### OVERVIEW

The experiments which follow will illustrate the way in which your Ohio Scientific computer can interact with the CA-24 board to give you extensive data handling and control capabilities.

When the CA-24 board is connected to the computer with a 16 wire ribbon cable (as described at the end of this section), circuit elements on the board (latches and a Peripheral Interface Adapter) are controlled by signals from the computer and route data to and from your computer. In addition, the circuit elements on the board generate control signals which, when used in conjunction with circuit elements placed on the breadboard or in a device external to the board, provide versatile data and device control capabilities.

The control signals from the computer and those generated on the board are briefly described below. They will be used and further explained in the experiments of this section. The abbreviations for the signals are also found in Table 1, page 4, and on the schematic diagrams of Figs. 3a and 3b (pages 5 and 6).

| Signal | Pin | Description |
|---|---|---|
| AØ | A12 | Address bit Ø (lowest) |
| A1 | A13 | Address bit 1 |
| $\overline{A1}$ | G38 | Address bit 1 inverted |
| LA2 | A26 | Address bit 2 |
| LA3 | A24 | Address bit 3 |
| $\Phi 2 \cdot VMA$ | A23 | Phase 2 AND Valid Memory Address |
| $\overline{\Phi 2 \cdot VMA}$ | A28 | $\Phi$2·VMA Inverted |
| R/$\overline{W}$ | A27 | Read/Write |
| $\overline{EXTDE}$ | A25 | External Device Enable |
| $\overline{EXTDR}$ | G39 | External Device Read |
| $\overline{EXTDW}$ | G25 | External Device Write |
| $\overline{POR}$ | A14 | Power On Reset |

At this point, the descriptions above probably will not have much meaning. However, as you work through the experiments, the meaning and use of the signals should become clear. Most of the signals are used for information transfer along the data bus when latches on the CA-24 board are used (Experiments 13,14, and 15) and when the Peripheral Interface Adapter (PIA) on the board is used (Experiment 16). The other signals make it possible to interface with the CA-24 data bus line using components placed on the breadboard or external to the CA-24 board.

There are three sets of addresses in your Ohio Scientific computer which are reserved for transfer of data to and from the computer by way of port J2, the 16-pin connector to which the CA-24 board attaches (see Figs 37 and 38). These addresses together with a description of their uses are given below.

| Addresses | | Uses |
|---|---|---|
| Decimal | Hexadecimal | |
| 5Ø948-5Ø951 | C7Ø4-C7Ø7 | Data handling by way of the on-board PIA. |
| 5Ø952-5Ø955 | C7Ø8-C7ØB | Data handling by way of the on-board latches or user designed circuits. |
| 5Ø956-5Ø959 | C7ØC-C7ØF | Data handling by way of an additional PIA which connects directly with terminal strip A, by way of an ACIA, or by way of user designed circuits. |

Additional data handling capability using the PIA in your computer is available through the input/output ports J3 and J4 on the back of your Ohio Scientific computer. These ports which are completely independent from the CA-24 board are associated with addresses 63232 through 63235. Refer to your User's Manual for connections to these PIA ports.

The selection of the 16-pin connector which connects to your CA-24 board is determined by the bit pattern of the high byte of a computer address. The pattern which selects the connector J2 is 11ØØØ111. The leftmost four bits are represented by hexadecimal C ($C where the $ symbol means that the numbers and/or letters following are hexadecimal notation) and the rightmost four bits are represented by hexadecimal 7 ($7). This leaves the low byte of the address for additional selection and control. However, only the rightmost four bits of this byte are available at J2. These are the AØ, A1, LA2, and LA3 found in the earlier list of signals. A particular combination of logic levels on LA2 and LA3 (seen as IOLA2 and IOLA3 in Fig. 3a) corresponds to a set of four addresses. For example, LA2 = 1 and LA3 = Ø are common to the bit patterns of Ø1ØØ, Ø1Ø1, Ø11Ø, and Ø111 which are the rightmost bits for addresses $C7Ø4 through $C7Ø7 (or decimal 5Ø948 through 5Ø951). As will be illustrated in Experiment 16, these four addresses are used with the data and control registers for the PIA mounted on your CA-24 board.

The following listing shows how the CA-24 board is associated with the various addresses and indicates which experiments introduce you to a particular use. Note that the last four addresses can be used to select a device which is external to the CA-24 board. The experiments listed for these addresses deal only with the PIA and ACIA use. A brief description of external device selection and control is given below the listing.

| Decimal | Hex | Use | Experiment |
|---|---|---|---|
| 5Ø948 | C7Ø4 | On-Board PIA Data/Data Direction Register (Port A) | 16 |
| 5Ø949 | C7Ø5 | On-Board PIA Control Register (Port A) | 16 |
| 5Ø95Ø | C7Ø6 | On-Board PIA Data/Data Direction Register (Port B) | 16 |
| 5Ø951 | C7Ø7 | On-Board PIA Control Register (Port B) | 16 |
| 5Ø952 | C7Ø8 | Input/Output (8 Bit) Using On-Board Latches | 13,14,15 |
| 5Ø953 | C7Ø9 | Input/Output (8 Bit) Using On-Board Latches | 13,14,15 |
| 5Ø954 | C7ØA | Input/Output (8 Bit) Using On-Board Latches | 13,14,15 |
| 5Ø955 | C7ØB | Input/Output (8 Bit) Using On-Board Latches | 13,14,15 |
| 5Ø956 | C7ØC | Secondary PIA Data/Data Direction Register (Port A), ACIA Control and Status Registers, or External Device Select #Ø | 17,21 |
| 5Ø957 | C7ØD | Secondary PIA Control Register (Port A), ACIA Data Registers, or External Device Select #1 | 17,21 |
| 5Ø958 | C7ØE | Secondary PIA Data/Data Direction Register (Port B), ACIA Control and Status Registers, or External Device Select #2 | 17,21 |
| 5Ø959 | C7ØF | Secondary PIA Control Register (Port B), ACIA Data Registers, or External Device Select #3 | 17,21 |

In addition, the CA-24 board generates a read or write signal for the external device. The listing below shows which levels correspond to which address (i.e., which signals are generated when a particular address is used for data transfer).

| Decimal | Hex | LA3 | LA2 | A1 | AØ | Operation | $\overline{\text{EXTDR}}$ | $\overline{\text{EXTDW}}$ | $\overline{\text{EXTDE}}$ |
|---|---|---|---|---|---|---|---|---|---|
| 5Ø956 | C7ØC | 1 | 1 | Ø | Ø | READ | Ø | 1 | Ø |
| 5Ø957 | C7ØD | 1 | 1 | Ø | 1 | READ | Ø | 1 | Ø |
| 5Ø958 | C7ØE | 1 | 1 | 1 | Ø | READ | Ø | 1 | Ø |
| 5Ø959 | C7ØF | 1 | 1 | 1 | 1 | READ | Ø | 1 | Ø |
| 5Ø956 | C7ØC | 1 | 1 | Ø | Ø | WRITE | 1 | Ø | Ø |
| 5Ø957 | C7ØD | 1 | 1 | Ø | 1 | WRITE | 1 | Ø | Ø |
| 5Ø958 | C7ØE | 1 | 1 | 1 | Ø | WRITE | 1 | Ø | Ø |
| 5Ø959 | C7ØF | 1 | 1 | 1 | 1 | WRITE | 1 | Ø | Ø |

The four addresses $C7ØC through $C7ØF are the only ones which will produce the logic "Ø" signal at the $\overline{\text{EXTDR}}$, $\overline{\text{EXTDW}}$, and $\overline{\text{EXTDE}}$ terminals (G39, G25, and A25). For all other addresses, the levels there remain at logic "1". Note that the AØ and A1 signals (at terminals A12 and A13) are used to select one of the four devices. The signal $\overline{\text{A1}}$ (at G38) may also be used depending on the circuitry of the device.

If you are already familiar with the interfacing of computers, the previous discussion will have given you an overview of the capabilties of the CA-24 board as it is used with your Ohio Scientific computer. If you have never worked with computer interfacing, you may at this point feel hopelessly confused. Do not give up! Connect the CA-24 board to your computer and begin working through the experiments. You may find it helpful to re-read this section after completing Experiments 13 through 15, then again after Experiment 16 and yet a third time after completing Experiment 17. By that time, you should be prepared not only to understand the applications of the CA-24 board as illustrated in Experiments 18 through 25, but also to design your own uses.

## COMPUTER CONNECTION

The CA-24 board can be used with an Ohio Scientific C4PMF, C4PDF, or C8PDF. In order for the CA-24 board to be used with a C1PMF, it is necessary first to install a 63Ø board in the C1PMF. The connection between the experimenter's board and the computer is by way of the standard 16-pin ribbon cable supplied with your CA-24 board. (If you wish to replace the cable with a longer one, note that the maximum length is 4 feet.)

Be certain that the power both to the CA-24 board and to the computer are off before making the connection. Insert one end of the cable into the J1 connector at the rear of the CA-24 board (see Fig. 1) so that the ribbon extends away from the board. Then insert the other end into the J2 connector, the left most of the three 16-pin connectors on the back of your computer, so that the ribbon extends to the left as shown in Fig. 37 or Fig. 38. The orientation of the cable is critical. Be certain that you have made the connection properly before proceeding to Experiment 13.

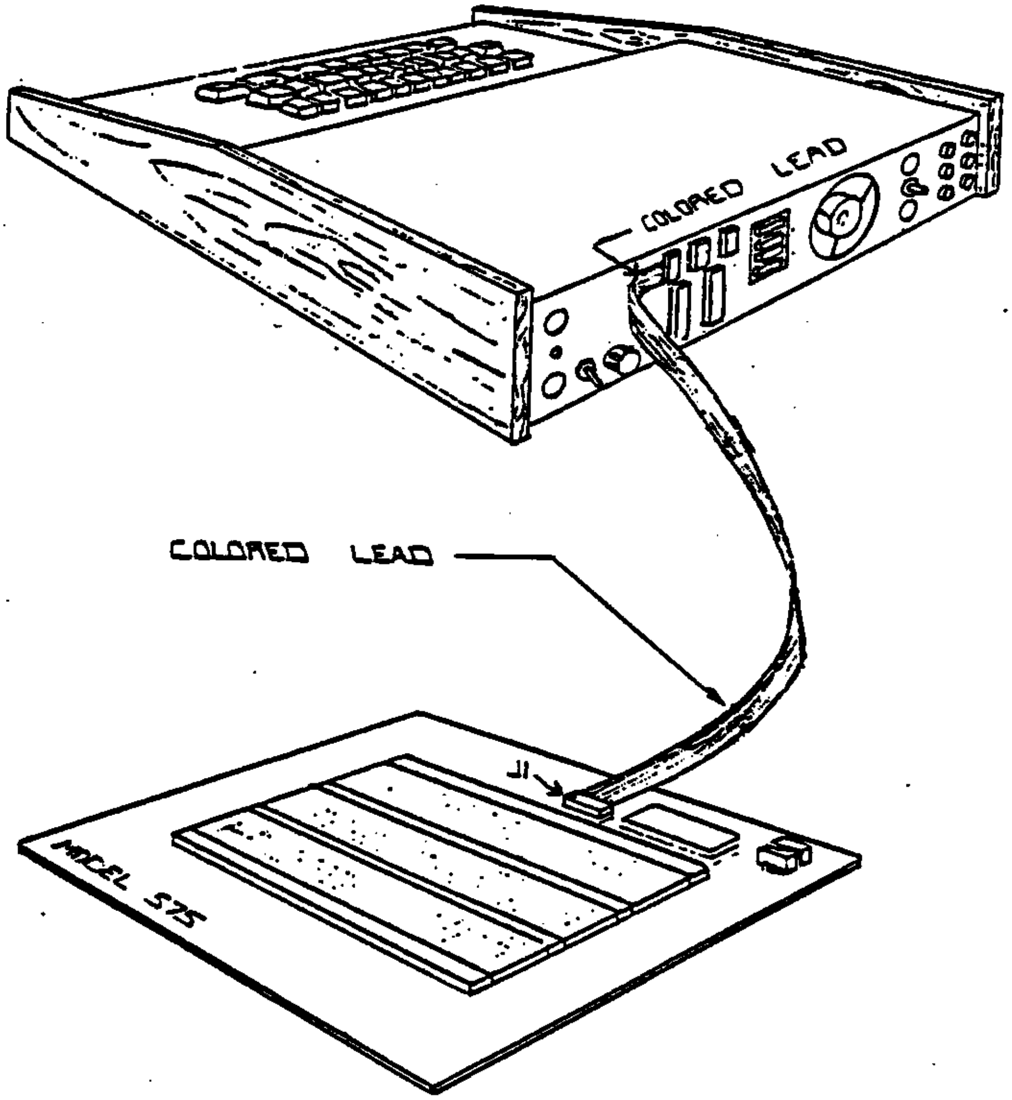


Figure 37
Connection of CA-24 Board to C4PMF or C4PDF Computer

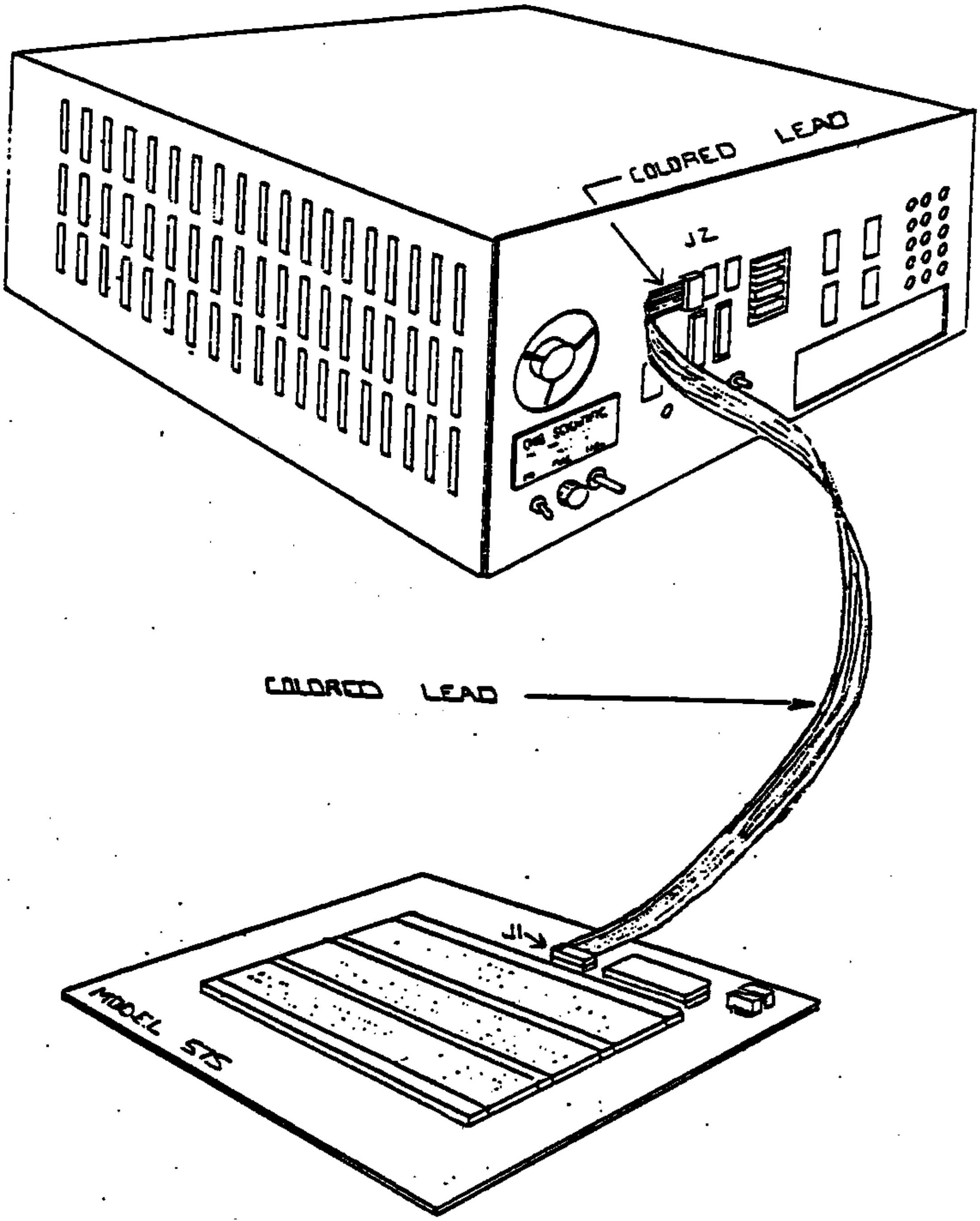


Figure 38
Connection of CA-24 Board to C8PDF Computer

# EXPERIMENT 13

## TITLE

Use of CA-24 Board Latches: Output

## PURPOSE

The purpose of this experiment is to investigate the function of the CA-24 board bi-directional 8T28 latches (3-state quad bus transceivers) and two of the 74175 latches (D type flip-flops) when the POKE command is used to output (write) information from the computer.

## EQUIPMENT

CA-24 board with power supply
OSI Computer (C4PMF, C4PDF, C8PDF)
Eight short jumper wires

## DISCUSSION

As much as possible, software is used to solve timing problems when data is transferred back and forth between a computer and some external device. However, there are occasions when it is advantageous to use hardware. A relatively simple aid to data transfer is the latch (flip-flop circuit) which, when presented with varying signals, can transfer a particular set of data across its circuitry at the appropriate time. The latch then can hold the transferred data for later pick-up even though the original data signals have changed. Since the latch waits for the appropriate command, it can be controlled through a computer program written in BASIC. Later (Experiment 2Ø), when speed is important, assembly language programs will be used.

## OVERVIEW OF DATA TRANSMISSION

Data transmission between the computer and the CA-24 board is routed through the lines IODØ - IOD7 (see Fig. 3a). POKEs to the CA-24 board place levels on these lines. PEEKs to the CA-24 board allow your computer to receive levels through these lines. In this experiment we investigate output to the CA-24 board with the POKE command. Experiment 14 investigates input from the CA-24 board with the PEEK command.

This experiment examines the data path associated with a POKE to memory location 5Ø952 ($C7Ø8 in hexadecimal). First, the high byte $C7 of the address selects port J2 on the back of your computer, so that the value being POKEd is routed through the 16 pin connector to the lines IODØ - IOD7. The route taken by the data once it arrives on these lines is determined by the last hexadecimal digit in the address (in this case a $8). The binary representation of this digit is 1ØØØ. As pointed out previously, these four bits are available at pin J2. In this case we have IOAØ = AØ = Ø, IOA1 = A1 = Ø, IOLA2 = LA2 = Ø and

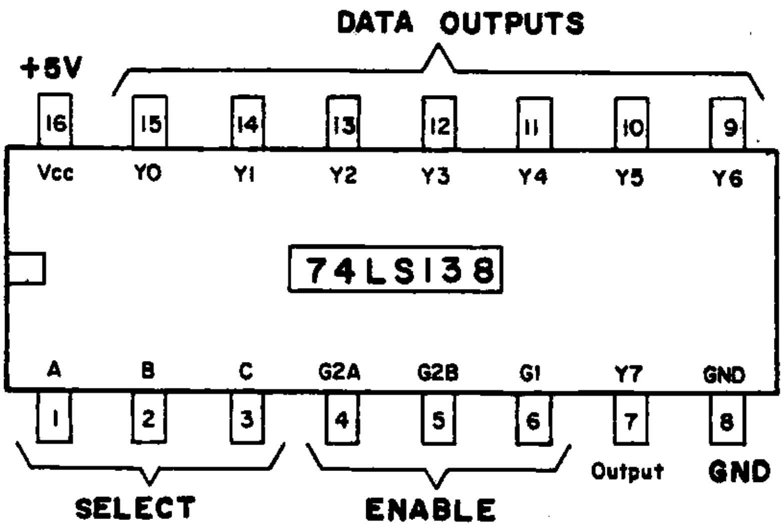
IOLA3 = LA3 = 1. The following technical discussion describes how these values of IOLA2 and IOLA3 route the data at IODØ - IOD7 first through the 8T28 latches and then through the 74175 latches to LODØ - LOD7 (pins G17 - G24).

The discussion is fairly technical. Even if you have difficulty following all the details you will be able to perform the experiment and observe the results.

## DETAILED TIMING DESCRIPTION

This experiment uses a set of latches to transfer data from the computer to the CA-24 board's LED display. The LED's could be the final display or represent a recording device connected to the CA-24 board. The latches involved are in the 8T28 3-state quad bus transceiver (see Experiment 11) and in the 74175 D flip-flop chip (see Experiment 1Ø). These chips are shown on the schematic diagram of Fig. 3a. (The 74175 chips are labeled 175.)

It should be helpful to begin tracing the path of the data in Fig. 3a. At the far left, the eight connections labeled IODØ through IOD7 are where the levels for bits Ø through 7 appear when output from the computer. To get data onto the CA-24 board data lines, the 8T28 bi-directional latch must be enabled when output is sent to the CA-24 board from the computer. The proper timing and address selection is accomplished through the 74LS138 IC chip (labeled LS138 in Fig. 3a) which is a 3-to-8 line decoder/multiplexer. The connections to the chip together with its function table are shown below.



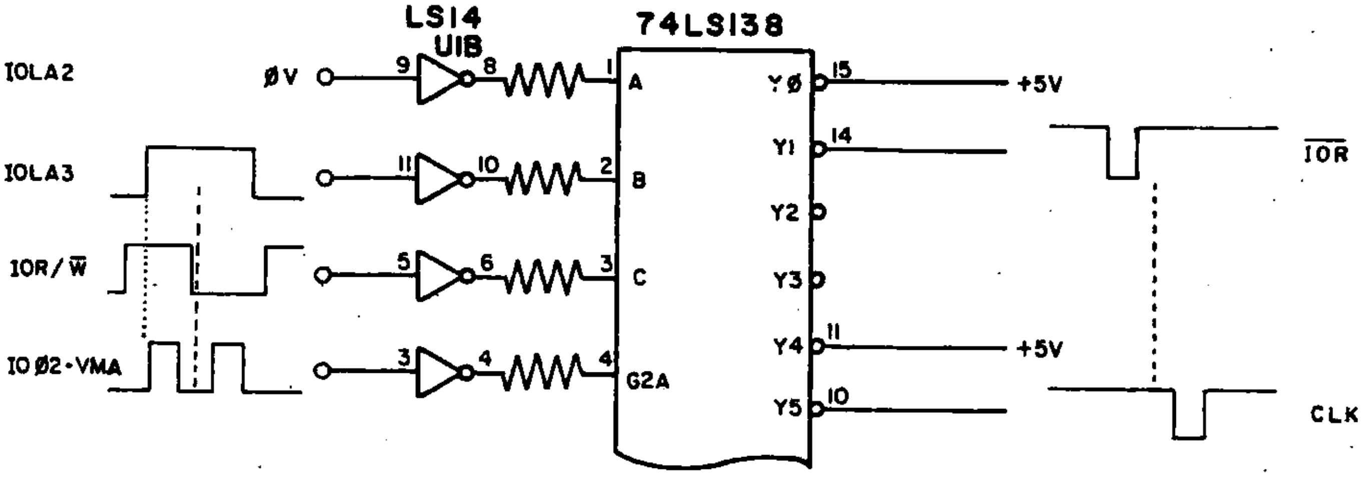
## 74LS138 Function Table

| INPUTS | | | | | OUTPUTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ENABLE | | SELECT | | | | | | | | | | |
| G1 | G2* | C | B | A | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
| X | H | X | X | X | H | H | H | H | H | H | H | H |
| L | X | X | X | X | H | H | H | H | H | H | H | H |
| H | L | L | L | L | L | H | H | H | H | H | H | H |
| H | L | L | L | H | H | L | H | H | H | H | H | H |
| H | L | L | H | L | H | H | L | H | H | H | H | H |
| H | L | L | H | H | H | H | H | L | H | H | H | H |
| H | L | H | L | L | H | H | H | H | L | H | H | H |
| H | L | H | L | H | H | H | H | H | H | L | H | H |
| H | L | H | H | L | H | H | H | H | H | H | L | H |
| H | L | H | H | H | H | H | H | H | H | H | H | L |

H = High level, L = Low level, X = Irrelevant *G2 = G2A + G2B

It is important to note in Fig. 3a that the selection and timing levels sent to the CA-24 board from the computer (IOLA2, IOLA3, IOR/W̄, and IOØ2·VMA) are inverted before going to the 74LS138 chip.

Normally IOLA2 and IOLA3 are low while a continuous train of pulses exists for IOR/W̄ and IOØ2·VMA. When a bit pattern is transferred into any one of the addresses 5Ø952 through 5Ø955 by way of a POKE command (or by way of a machine code program), IOLA3 goes high while IOLA2 remains low. (A decimal to binary conversion will show that IOLA3 high and IOLA2 low are common to addresses 5Ø952 through 5Ø955.) This occurs at the proper time in the pulse trains of IOR/W̄ and IOØ2·VMA so that while IOLA3 is high, IOR/W̄ is first high with IOØ2·VMA high then IOR/W̄ is low when IOØ2·VMA again goes high. The sequence is shown in the diagram below.



The function table for the 74LS138 chip shows that, since G2B on the 74LS138 chip is grounded and G1 is connected to 5 V, the combination of levels at A, B, C, and G2A will produce first a low at Y1 and then a low at Y5. The schematic diagram of Fig. 3a shows that the low at Y1 is inverted and sent to the 8T28 chip. A review of Experiment 11 will show that a high level at pins 1 and 15 of the 8T28 bi-directional latch enables transfer to the computer, not from it. However, the circuitry in your computer prevents any levels on the CA-24 board data lines from being transferred into a memory location during a write operation. Consequently, the high which is sent to the enable pins of the 8T28's has no effect. The important transition is the dropping back to the low state which occurs at the same time that a low is generated at Y5. At this moment the bit pattern from the computer address is transferred to the CA-24 board data lines which are connected to the two 74175 output latches as seen in the lower right corner of Fig. 3a. But since pin 9 (CLK) of each 74175 latch is low, the latch is disabled and transfer does not occur. Finally, when Y5 goes high, an enabling signal is sent to the 74175 output latches and the bit pattern is transferred to terminals G17 through G24. If these terminals are connected to LED's, the bit pattern will be displayed. It should be noted at this point that the flip-flops in the 74175 chips are capable of driving ten TTL loads per data line. Each LED is one such load.

## PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Use eight short jumper wires to connect terminals G17 through G24 to LEDØ through LED7 as shown in Fig. 39 (i.e., connect G17 to LEDØ, G18 to LED1, etc. until G24 is connected to LED 7).

If the CA-24 board has not yet been connected to your computer, refer to page 8Ø for instructions. If the board is connected and the computer is turned on, use an OS-65D development disc to boot up the system and then answer "UNLOCK" to the FUNCTION question. Next, if you wish, enter NEW, although initializing the work space is not necessary for this experiment. Turn on the CA-24 board power supply.

LEDØ through LED7 will be off when the power supply is first turned on. Now type

```
POKE 5Ø952,4
```

and watch the LED display as you press the RETURN key. LED2 will turn on. Try using the POKE command for other numbers from Ø through 255.

Alternatively, you can use the 65V Machine Monitor for transferring a bit pattern to the LED's. (If you are not familiar with the use of the 65V Machine Monitor, the 65V Primer manual available from Ohio Scientific will be helpful.) Press

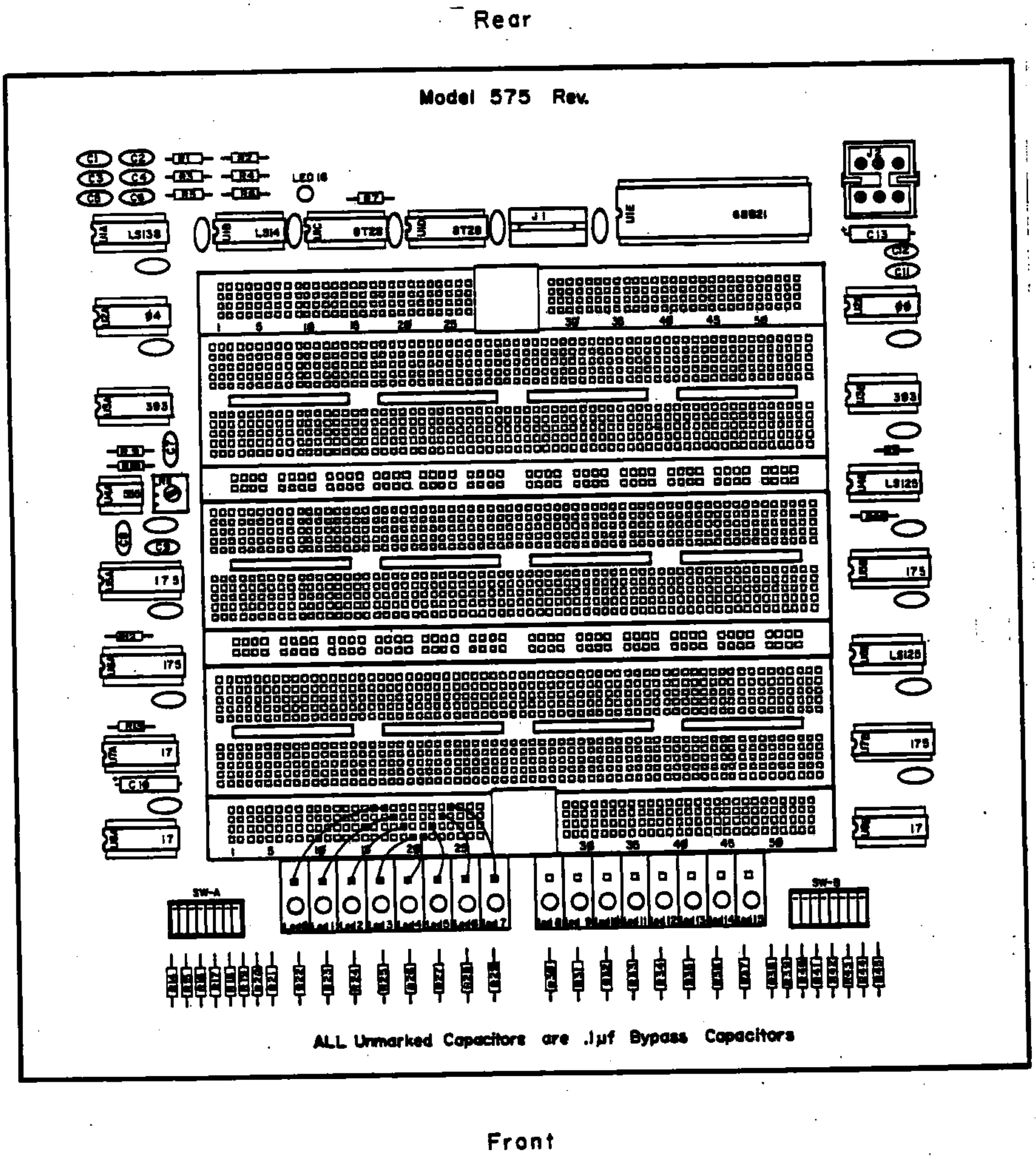


Figure 39
Connections for Latch Output to LED's

the BREAK key, then press the M key. Next type C7Ø8 (the hexadecimal equivalent of 5Ø952) so that this address number is displayed on the video screen. Now press the key with the slash (/) symbol on it and type Ø then 4. Again LED2 should be on. Type Ø then 8 and LED3 will be on.

Finally, you can repeat the use of the POKE command or the Machine Monitor with address 5Ø953, 5Ø954, and 5Ø955 as any one of these will properly activate the latches.

Leave the jumper wires in place. They will be used in Experiment 15.

# EXPERIMENT 14

## TITLE

Use of the CA-24 Board Latches: Input

## PURPOSE

The purpose of this experiment is to investigate the function of the CA-24 board bi-directional 8T28 latches and two of the 74175 latches when the PEEK command is used to input (read) information into the computer.

## EQUIPMENT

CA-24 board with power supply
OSI computer (C4PMF, C4PDF, C8PDF)
One short jumper wire
Eight medium jumper wires

## DISCUSSION

This experiment is designed to follow Experiment 13. That experiment should be completed prior to working through this one. As stated there, latches are a relatively simple way of transferring data back and forth between a computer and some external device. Experiment 13 illustrated the transfer of data from the computer to the CA-24 board LED display.

## OVERVIEW OF DATA TRANSMISSION

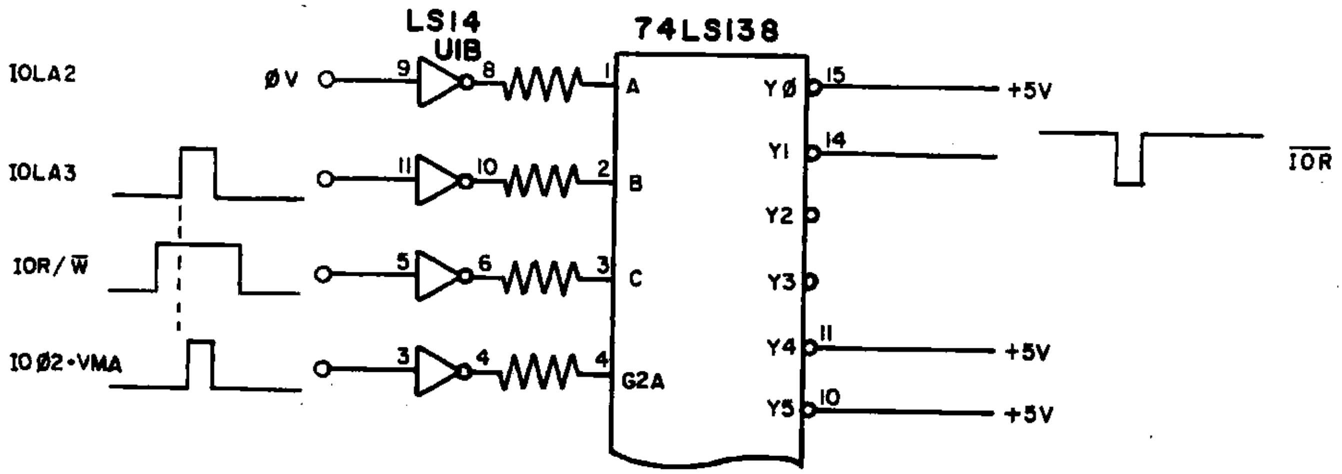
This experiment examines the data path associated with a PEEK of memory location 5Ø952 ($C7Ø8 in hexadecimal). As in experiment 13, the high byte $C7 of the address selects port J2 on the back of your computer for input/output. The PEEK command generates signals instructing the CA-24 board to internally transfer certain levels to IODØ - IOD7. The resulting bit pattern is returned as the value of the PEEK. In experiment 13 it was seen that bits 3 and 4 of the address, IOLA2 and IOLA3, dictated the route followed by the levels placed at IODØ - IOD7 by a POKE to 5Ø952. In this experiment we will see that, for a PEEK, these same two bits determine which data is routed to lines IODØ - IOD7 for transfer to the computer. The following technical discussion describes how the values IOLA2 = Ø and IOLA3 = 1 cause the levels at LIDØ - LID7 (pins G3Ø - G37) to be routed first through the 74175 latches and 74LS125 buffers on the right hand side of the CA-24 board and then through the bi-directional 8T28 latches to lines IODØ - IOD7. Even if you have difficulty following all the details you will be able to perform the experiment and observe the results.

## DETAILED TIMING DESCRIPTION

This experiment uses eight switches on the CA-24 board to set levels at LIDØ - LID7 (pins G3Ø - G37). These levels are then transferred to the computer, via a series of latches, with a PEEK command. The latches involved are in the two 74175 chips shown in the upper portion of Fig. 3b and in the two 8T28 chips shown in Fig. 3a. The two 74LS125 chips next to the 74175 chips in Fig. 3b contain 3-state buffers which function to isolate the 74175 chips from the data lines when the latches are not being used. (Isolation here means a very high impedance coupling to the data lines so that the latches do not act as loads on the lines).

If signals are connected to terminals G3Ø through G37 (upper right of Fig. 3b), these signals (or levels) can be transferred to the CA-24 board data lines by way of the 74175 latches and 74LS125 buffers. The signals can be sent to the computer by way of the bi-directional 8T28 latches. These latches and buffers must be enabled in the appropriate sequence whenever data is to be transferred to the computer from LIDØ - LID7. (When tracing the path of the signals, note that the lines marked (DØ) through (D7) in the upper middle of Fig. 3b are the same as the equivalently marked lines in the lower right portion of Fig. 3a.)

Once again, the 74LS138 3-to-8 line decoder/multiplexer (lower left of Fig. 3a) is the critical element in selection and timing. But before looking at the function of this chip, it is important to see what has happened to the LIDØ through LID7 levels which are being brought in through terminals G3Ø through G37. Clearly, they can get no farther than the 74175 latches unless an enable signal is sent to pin 9 of each chip. But that can happen only if a signal is connected to terminal G29. We do this by connecting G29 to G28 where the $\overline{\Phi 2 \cdot VMA}$ signal can be picked up. But recall from Experiment 13 that Φ2 · VMA and consequently also $\overline{\Phi 2 \cdot VMA}$ is a train of pulses which is always present. Consequently, a level connected to any one of the input data lines will almost immediately be transferred to the other side of the latch and be waiting to continue when the 74LS125 3-state buffer is enabled. Keep this in mind as we look at the function of the 74LS138 decoder.



As stated in Experiment 13, IOLA2 and IOLA3 are normally low while a continuous train of pulses exists for IOR/$\overline{W}$ and for IOØ2·VMA. When a bit pattern is to be picked up at any one of the addresses 5Ø952 through 5Ø955 by way of a PEEK command (or by way of a machine code program), IOLA3 goes high (IOLA2 remaining low) at the proper time in the pulse trains of IOR/$\overline{W}$ and IOØ2·VMA so that the level of IOR/$\overline{W}$ is high and the level of IOØ2·VMA goes high shortly after IOLA3 goes high. The timing sequence is shown above.

From the function table shown in Experiment 13 (page 87), it can be seen that a low level is generated at Y1 at the moment IOØ2·VMA goes high. But this level is on the line (IOR) which enables the 74LS125 3-state buffer. The signal generated at Y1, then, transfers the input bit pattern onto the CA-24 data lines. But this signal, in addition, is inverted and sent to the 8T28 bi-directional latch. A review of Experiment 11 will show that the latch transfers data to the computer when the enable signal is high. Consequently, as the levels are transferred to the CA-24 data lines, they are transferred through the 8T28 bi-directional latch to the computer.

The PEEK command, as mentioned earlier, picks up a bit pattern from a memory location. In this case, the pattern which is picked up is the one which has been transferred to IODØ - IOD7.

Finally, a comment needs to be made concerning the use of $\overline{\text{Ø2·VMA}}$ as the CLK signal for the 74175 input latch chips. Since $\overline{\text{Ø2·VMA}}$ makes pins 9 low while data transfer to the computer is being accomplished and since the 74175 is enabled only when the CLK input is high, transfer across the 74175 latches occurs while the 74LS125 3-state buffers are enabled and levels are being transferred to the computer. This insures that any level changes occuring at G3Ø through G37 cannot have an effect on the data bus lines while transfer to the computer is in progress.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Without removing the jumper wires used in Experiment 13, insert a short jumper wire to connect G28 with G29. Then use the eight medium wires to connect G9 to G3Ø, G1Ø to G31, etc., until G16 has been connected to G37. Fig. 4Ø shows these connections.

When you are certain all connections have been made correctly, turn on the power supply. (We assume that the CA-24 board is still connected to your computer and that the computer has been booted up.)

Again, LEDØ through LED7 will be off. Note that changing the switches will not affect the LED's. Set the switches so that SW4A is at logic "1" (rear end up, or switch open) while all other switches are at logic "Ø". Now type

```
PRINT PEEK(5Ø952)
```

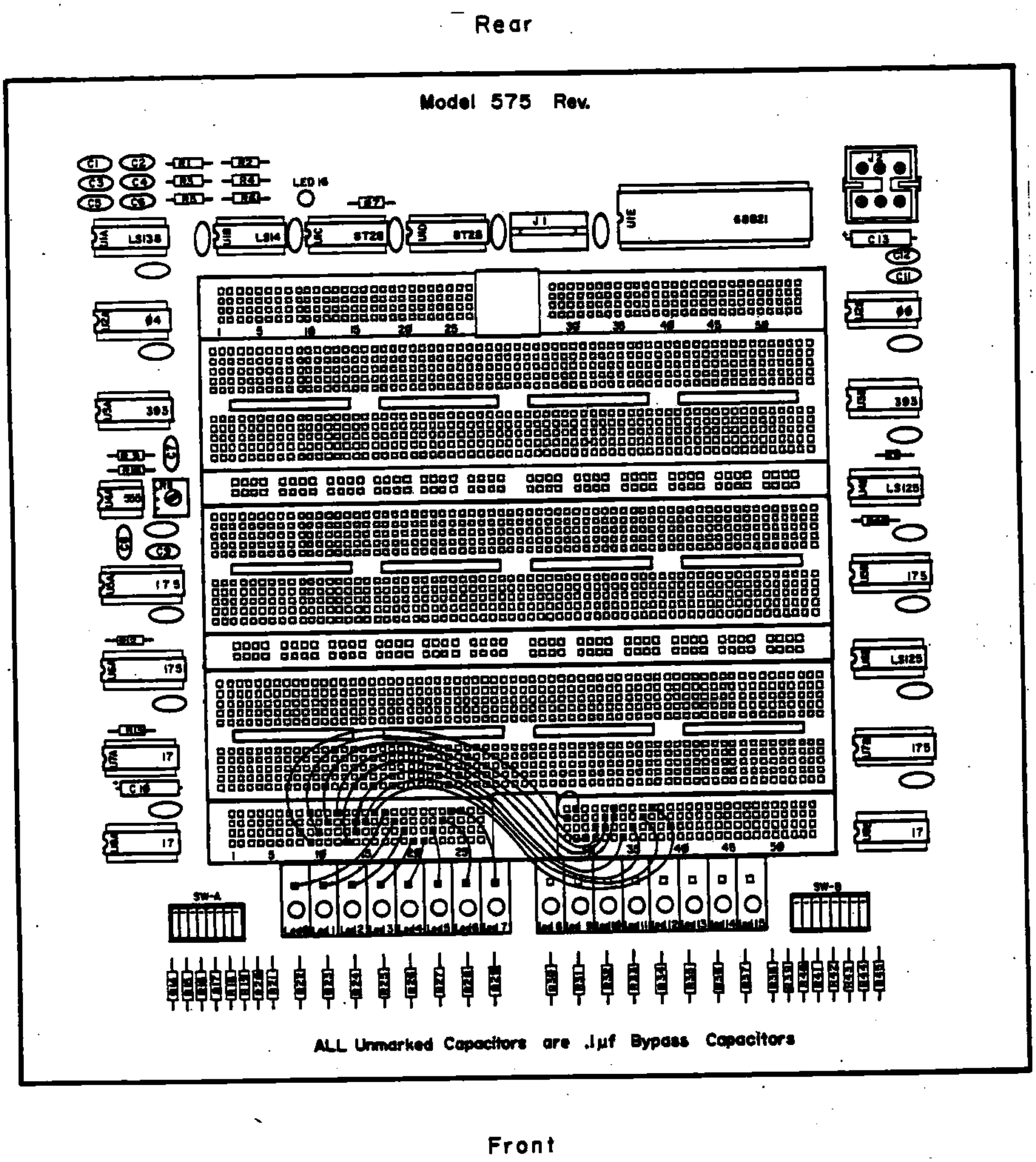
then press the RETURN key. The number 8 should appear on the video display. (The LED's still will not be affected.) Set the switches at various bit patterns and repeat the use of PEEK in conjunction with PRINT. Note that 5Ø953, 5Ø954, or 5Ø955 can be used in place of 5Ø952. Note also that the sequence of commands

```
SW=PEEK(5Ø952) <RETURN>
PRINT SW <RETURN>
```

will have the same effect as the earlier command.

You can use the Machine Monitor to read the switch bit pattern into memory location 5Ø952 (or $C7Ø8). Press BREAK then M. Set a bit pattern on the switches such as SW1A, SW2A, and SW3A at logic "1" with all others at logic "Ø". Now type in the address C7Ø8. You will find Ø7 displayed as the contents of the address. Change SW3A to its logic "Ø" position and again type C7Ø8. Now the contents will be displayed as Ø3.

Leave the jumper wires used in this experiment in place. They and those of Experiment 13 will be used in Experiment 15.



ALL Unmarked Capacitors are .1µf Bypass Capacitors

Figure 40
Connections for Switch Level Entry by Way of On-Board Latches
(Latch output is also shown)

# EXPERIMENT 15

## TITLE

Computer Routing of Data

## PURPOSE

The purpose of this experiment is to use a computer program to pick up input data from the CA-24 board, display the data on the video screen, and output the data onto the CA-24 board LED display.

## EQUIPMENT

CA-24 board with power supply
OSI computer (C4PMF, C4PDF, C8PDF)
Nine short jumper wires
Eight medium jumper wires

## DISCUSSION

This experiment is designed to follow experiments 13 and 14. They should be completed prior to working through this one.

Experiments 13 and 14 illustrated the enabling of CA-24 board latches through the use of the POKE and PEEK commands. In this experiment, you will write a short program which will monitor the bit pattern set on switch block A. When the program begins running, it will pick up the bit pattern set on the switches, display the decimal value of that bit pattern on the video monitor, and output the pattern to the LED display (with LEDØ being the lowest bit and LED7 being the highest bit). The program will wait for (and ask for) a switch change. As soon as a switch is set to a new value, the program will display the decimal value and change the LED display.

Data transfer by way of the latches requires eight lines dedicated to input and eight lines dedicated to output. All eight lines for each function are treated simultaneously in the same way. In Experiments 16 and 17, you will be introduced to hardware which can control sixteen lines, and software which can set each line individually as an input or an output line.

## PROCEDURE

If the wiring of Experiment 13 and 14 has been removed, turn off the CA-24 board power supply and connect the jumper wires as shown in Fig. 4Ø. The power supply can now be turned back on.

If you have left the wiring of Experiments 13 and 14 on the board, you are ready to write the program. (Note that the power supply to the CA-24 board does not need to be turned off while you enter the program into the computer.)

Boot up the computer, enter UNLOCK, then enter NEW (or if the computer is already booted up, simply enter NEW). Type in the program shown below. The REM statements, of course, are not necessary but have been included as an aid to understanding the program.

```
10 REM DATA I/O USING LATCHES
20 SS = 300:REM INSURES FIRST OUTPUT
30 SW = PEEK(50952):REM READ FIRST SWITCH PATTERN
40 IF SW = SS THEN 30:REM WAIT FOR SWITCH CHANGE
50 SS = SW
60 PRINT"THE DECIMAL SWITCH PATTERN VALUE IS";SW
70 PRINT"SET A NEW SWITCH PATTERN."
80 POKE 50952,SW:REM OUTPUT SWITCH VALUE
90 GOTO 30
```

When the program has been entered and you are certain that it is correct, set the switches so that SW1A, SW2A, and SW3A are at logic "1" and the others are at logic "Ø". Note that the switches have had no effect on the LED display. Now RUN the program. The message

```
THE DECIMAL SWITCH PATTERN VALUE IS 7
SET A NEW SWITCH PATTERN
```

will appear on the video display and LEDØ through LED2 will be on while LED3 through LED7 will be off.

Change one of the switches while watching the video display. Note that as soon as a switch is changed, a new message will appear. The LED display will also be changed to correspond to the new switch pattern.

The program can be terminated by using CTRL-C.

# EXPERIMENT 16

## TITLE

The CA-24 Board Peripheral Interface Adapter (PIA)

## PURPOSE

The purpose of this experiment is to investigate the function of the CA-24 board Peripheral Interface Adapter (IC chip 68B21) and to use it for data transfer.

## EQUIPMENT

CA-24 board with power supply
OSI computer (C4PMF, C4PDF, C8PDF)
Sixteen short jumper wires
Sixteen long jumper wires

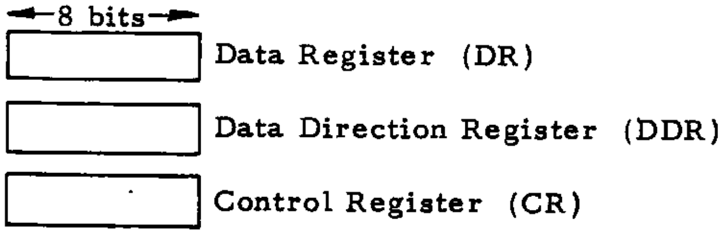
## DISCUSSION

As stated in Experiment 15, the transfer of data by means of the latches on the CA-24 board requires that eight lines be dedicated to input and eight lines be dedicated to output. In this experiment, you will use an IC chip which can utilize sixteen data lines, each one of which can be independently specified through software as an input or output line. The chip which does this is called a Peripheral Interface Adapter or PIA.

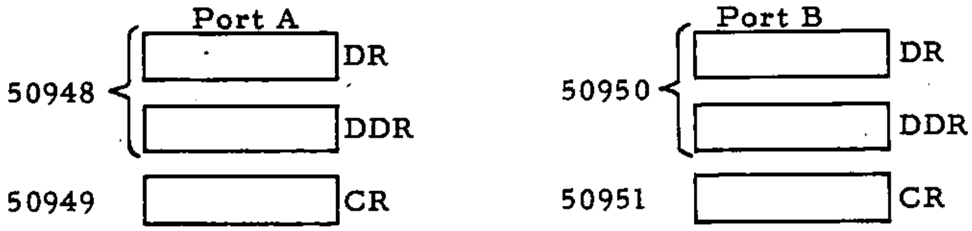
Your CA-24 board is equipped with a type 68B21 PIA. The PIA connects to the 8-bit (also referred to as 8-line) data bus of your OSI computer and has two 8-bit ports (port A and port B) each line of which can be used as either an input line or an output line. (One bit, of course, is transferred along one line.) Although the bi-directional property of the PIA permits more flexible use, each line of a PIA port can drive only two TTL loads. (Recall that the 74175 latch can drive ten TTL loads per line.)

The CA-24 board PIA is shown in the upper right corner of Fig. 3a. There it can be seen that pins 26 through 33 (labeled D7 through DØ) connect to the computer data bus. Pins 2 through 9 (labeled PAØ through PA7) are the eight lines of port A and connect to the CA-24 board terminals A4Ø through A47. Pins 1Ø through 17 (labeled PBØ through PB7) are the eight lines of port B and connect to the CA-24 board terminals A3Ø through A37.

Each PIA port has three 8-bit registers associated with it. These are the Data Register (DR), the Data Direction Register (DDR), and the Control Register (CR). These registers can be visualized as shown below.



There are, however, only two computer addresses associated with each port and hence with each set of three registers. This is possible because the Control Register, associated with one of the two computer addresses, is used to control whether the other address is used for the Data Register or for the Data Direction Register. The addresses used for the on-board PIA registers are shown below.



The POKE command will be used to transfer bit patterns to the PIA registers.

Addresses 5Ø949 ($C7Ø5) and 5Ø951 ($C7Ø7) can be thought of as the control addresses while 5Ø948 ($C7Ø4) and 5Ø95Ø ($C7Ø6) can be thought of as the data addresses. Whether the data addresses are used for data direction (determining whether the transfer of data will be input or output) or whether the data addresses are used for actual data transfer is determined by the bit pattern put into the Control Register by way of the control address. All zeros in the Control Register selects the data direction function for the data address. The pattern ØØØØØ1ØØ (decimal 4) selects the data transfer function for the data address.

If the data direction function for address 5Ø948 has been selected (Ø POKEd into 5Ø949), then the bit pattern placed into 5Ø948 will determine which lines are input lines and which are output lines. A Ø specifies input and a 1 specifies output.

For example, if, after selecting the data direction function for 5Ø948, the decimal value 15 is POKEd into 5Ø948, the bit pattern there will be ØØØØ1111. This means that for port A, the high four lines have been designated as input lines while the low four lines have been designated as output lines.

Next the use of the POKE command to place the bit pattern ØØØØØ1ØØ in 5Ø949 will revert 5Ø948 to its data transfer function (while maintaining the data transfer directions on each line as just specified). Now POKE and PEEK commands can be used with 5Ø948 for data transfer through port A.

Be careful, though. Since only the low four lines are output lines, only these can be changed by the POKE command; i.e., both

```
POKE 5Ø948,1
```

and

```
POKE 5Ø948,17
```

will have the same effect.

On the other hand, the PEEK command will pick up all eight bits even though only the high four are input. The PEEK command can be made selective by using it together with the logical AND. The way in which this is done will be shown in the PROCEDURE section of this experiment.

The previous discussion has implied that port A and port B are identical in function. This is not quite the case. For port A (unlike port B), the PIA lines assigned to output will not be in the high impedance mode (characteristic of 3-state devices as explained in Experiment 14 for the 74LS125 3-state buffer) and consequently an output bit in the data register can be reset by a Ø voltage level applied to that data line. Port B does have 3-state operation and so a voltage level on an output line cannot change the data register output bit pattern. This feature will be illustrated in the PROCEDURE section of this experiment.

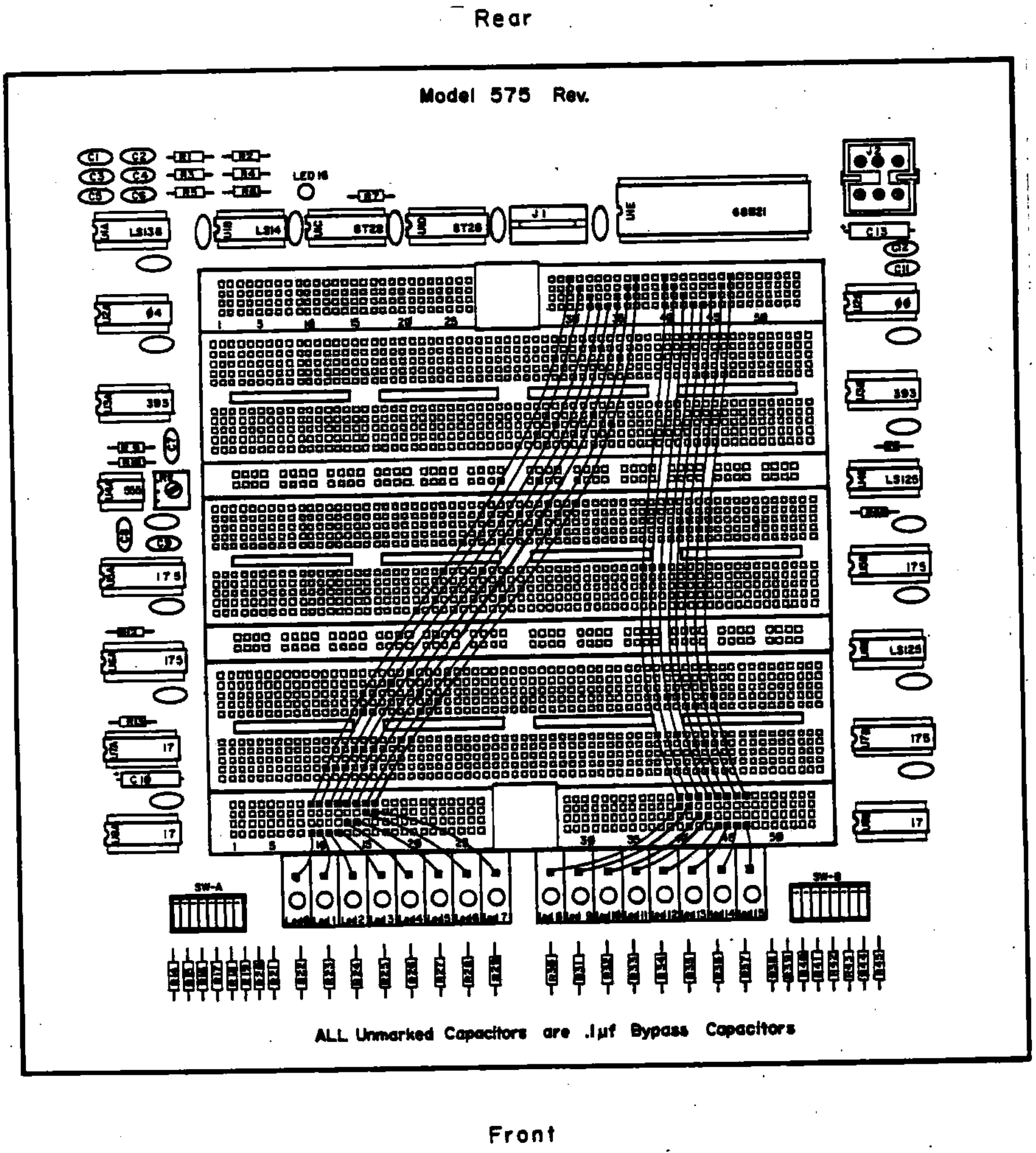
There are control features of the PIA which will not be illustrated in this experiment. The specification sheets for the 68B21 are included in the appendix of this manual. Some understanding of the additional control features can be obtained by reading through those sheets.

If the previous discussion has been confusing, take heart! Learning to use a PIA is most easily done by actually using one, and this you will do as you work through the rest of this experiment.

## PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Remove all previous wiring from the breadboard.

Use eight long jumper wires to connect A3Ø through A37 to G9 through G16 and then eight short jumper wires to connect G9 through G16 to LEDØ through LED7 as shown in Fig. 41. This completes the connection of port B (refer again to Fig. 3a) to the switches which can set input levels and to the LED's which will be an output display. Note that the leftmost LED (LEDØ) represents the lowest bit in the bit pattern while LED7 represents the highest bit for port B. Similarly, switch 1 of switch block A will set the lowest bit input level for port B



ALL Unmarked Capacitors are .1µf Bypass Capacitors

Figure 41
Connections for PIA Input and Output

while switch 8 will set the highest bit level.

Next, use eight long jumper wires to connect A4Ø through A47 to G4Ø through G47 and then eight short jumper wires to connect G4Ø through G47 to LED8 through LED15 as shown in Fig. 41. Note that SW1B and LED8 will be associated with the lowest bit of port A while SW8B and LED15 will be associated with the highest bit of port A.

Set all switches to their logic "1" position. Turn on the power supply. All LED's should be on.

If you have not already done so, boot up the computer, UNLOCK, and enter NEW.

We will begin by working with port A. The first step is to set which lines are input and which lines are output. Two POKE commands are required to do this. First, enter

```
POKE 5Ø949,Ø
```

The bit pattern of ØØØØØØØØ which you have placed into the Control Register selects the data direction function for address 5Ø948. Now enter

```
POKE 5Ø948,15
```

The bit pattern of ØØØØ1111 which you have placed into the Data Direction Register designates the high four lines of port A as input and the low four lines as output. This completes the setting of direction for the port A lines.

Note that LED8 through LED11 (representing the low four output bits of port A) were turned off when the bit pattern was initially entered into the Data Direction Register. LED12 through LED15 (representing the high four input bits of port A) were left on.

Now we must switch address 5Ø948 into its data transfer function. One POKE command will do this. Enter

```
POKE 5Ø949,4
```

The bit pattern of ØØØØØ1ØØ which you have placed into the Control Register selects the data transfer function for address 5Ø948.

We are now ready to transfer data. Enter

```
POKE 5Ø948,2
```

and note that LED9 turns on. Next, enter

```
POKE 5Ø948,16
```

and note that LED8 through LED11 are off and LED12 through LED15 have not been affected. You have tried to output data on an input line.

Now use the PEEK command to pick up the input bit pattern. With LED8 through LED11 off and LED12 through LED15 on, enter

```
PRINT PEEK(50948)
```

Note that the number 240 appears on the video display. This is the decimal equivalent of the bit pattern 11110000. Change SW8B, SW7B, and SW6B to their logic "0" position (LED15, LED14, and LED13 will now be off). The bit pattern which will be picked up is 00010000. Enter

```
PRINT PEEK (50948)
```

and note that the number 16 is now on the video display.

Next, enter the two commands

```
POKE 50948,2
PRINT PEEK(50948)
```

The number on your video screen will now be 18 instead of 16 indicating that you have picked up a bit which is on an output line. To avoid doing this, it is necessary to mask the output lines by using the logical AND in conjunction with the PEEK command. To do this, enter

```
PRINT PEEK(50948) AND 240
```

and note that again the number displayed is 16. Since 240 is the decimal equivalent of the bit pattern 11110000, all of the low four bits in address 50948 will be ignored while any of the high four bits which are logic "1" will be picked up. Note that in the above example, 240 could have been replaced by 112, 48, or 16 and the same result would have been obtained. However, if we wish the PEEK command to look at all of the high four bits, then 240 must be used.

One more note of caution should be introduced here. If no connection to a PIA input line is made, the default value is logic "1". This can be observed by removing the wires from terminals A44 through A47 and again entering

```
PRINT PEEK(50948) AND 240
```

The number 240 will be displayed indicating that all of the input lines are high. If you make this test, replace the wires before proceeding.

At this point, you should have LED8 through LED11 monitoring output lines with LED9 glowing. (SW1B through SW4B should be in their logic "1" position so that the output levels will control the LED's.) SW5B should be in its logic "1" position so that a high level is on the D4 input line of port A and LED12 is glowing. SW6B, SW7B, and SW8B should be set at logic "Ø" so that low levels are on the D5, D6, and D7 input lines. The PEEK command (without the AND), then, will result in the number 18 being displayed on the video screen.

Now set SW2B to its logic "Ø" position so that LED9 is turned off. Now enter

```
PRINT PEEK(5Ø948)
```

and note that the value 16 comes up on the video screen. As stated toward the end of the DISCUSSION section of this experiment, port A does not have a 3-state operation and consequently a low level on an output line will be reflected in the Data Register.

Now set SW2B back to its logic "1" position. LED9 will again glow, and the PEEK command will again produce the number 18. The alteration in the Data Register was not permanent.

Next, we will try this same alteration on port B. Enter the series of commands

```
POKE 5Ø951,Ø
POKE 5Ø95Ø,15
POKE 5Ø951,4
POKE 5Ø95Ø,2
```

to set up port B so that the low four lines are input with the next to the lowest bit high. Now set SW6A, SW7A, and SW8A to their logic "Ø" position so that LEDØ through LED7 show the same display as LED8 through LED15. Enter the command

```
PRINT PEEK(5Ø95Ø)
```

and note that the number 18 is displayed on the screen. Now set SW2A to its logic "Ø" position so that LED1 goes out. Again enter

```
PRINT PEEK (5Ø95Ø)
```

and note that for port B the output bit in the data register has not been altered. Return SW2A to its logic "1" position.

You now should be able to set your own pattern of input and output lines for the two ports and transfer data to the LED's or from the switch settings. Be sure to remember that the LED's will correctly monitor the output lines only if the corresponding switches are in their logic "1" position.

In this experiment, the input and output lines were set up by entering sequentially the necessary POKE and PEEK commands. Alternatively, a BASIC program can be written for this purpose. A sample program which sets all lines of either port to output or input is given below. You may wish to try this program or write your own.

```
10 REM PIA INITIALIZATION
20 X = 50948:REM SET STARTING ADDRESS OF PIA
30 INPUT"A PORT I OR O";A$
40 INPUT"B PORT I OR O";B$
50 POKE X+1,0:POKE X+3,0:REM SELECT DATA DIRECTION REGISTER
60 IF A$ = "I" THEN POKE X,0:REM SET A AS INPUT
70 IF A$ = "I" THEN 90
80 POKE X,255:REM SET A AS OUTPUT
90 IF B$ = "I" THEN POKE X+2,0:REM SET B AS INPUT
100 IF B$ = "I" THEN 120
110 POKE X+2,255:REM SET B AS OUTPUT
120 POKE X+1,4:POKE X+3,4:REM SELECT DATA REGISTER
130 REM INPUT AND OUTPUT PROGRAM
140 INPUT"PORT (A OR B)";C$
150 IF C$ = "A" THEN 180
160 IF C$ = "B" THEN 240
170 GOTO 140
180 IF A$ = "I" THEN 220
190 INPUT"OUTPUT TO A";K
200 POKE X,K
210 GOTO 140
220 PRINT"INPUT TO A IS";PEEK(X)
230 GOTO 140
240 IF B$ = "I" THEN 280
250 INPUT"OUTPUT TO B";K
260 POKE X+2,K
270 GOTO 140
280 PRINT"INPUT TO B IS";PEEK(X+2)
290 GOTO 140
```

Leaving the wires on the board will facilitate the procedure of Experiment 17.

## EXPERIMENT 17

### TITLE

Use of an Additional PIA

### PURPOSE

The purpose of this experiment is to investigate the use of a second PIA which can be positioned on the CA-24 board such that no jumper wires are needed for control or data bus line connections.

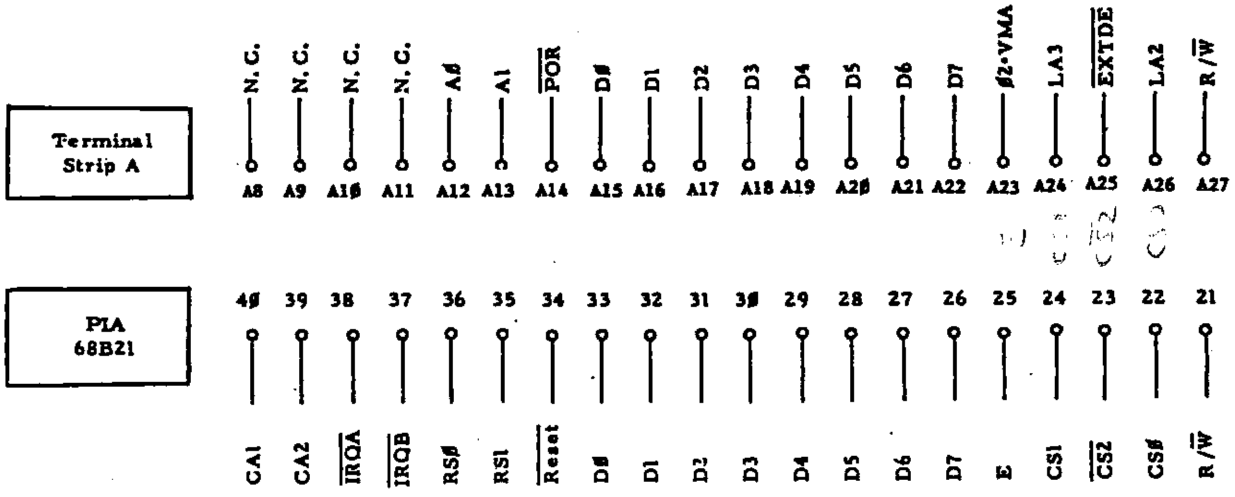
### EQUIPMENT

CA-24 board with power supply
OSI computer (C4PMF, C4PDF, C8PDF)
One 68B21 IC Chip
Sixteen long jumper wires
One medium jumper wire
Seventeen short jumper wires

### DISCUSSION

This experiment is designed to follow Experiment 16. That experiment should be worked through before proceeding with this one.

In Experiment 16, the PIA mounted on your CA-24 board was used as a data transfer device. In addition to this PIA, a second PIA can be positioned on the breadboard in a manner which allows it to directly pick up the control and data bus signals from terminal strip A. The proper pin and terminal orientation which enables this is shown below.



When the PIA pins are inserted in strip A with this match, the two input/output ports are available on strip B and can be used in the fashion described in Experiment 16. This breadboard PIA, however, is completely independent from the PIA mounted at the rear of the board. Addresses 5Ø956 ($C7ØC) through 5Ø959 ($C7ØF) are used for the breadboard PIA while 5Ø948 ($C7Ø4) through 5Ø951 ($C7Ø7) were used for the PIA in experiment 16.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off.

Position the extra 68B21 PIA chip provided with this manual so that pins 4Ø through 21 line up with terminals A8 through A27 as shown in Fig. 42 and press it into place. Use a short jumper wire to connect pin 1 to A2 and a medium jumper wire to connect pin 2Ø to A1. Next, move the long jumper wires left on the board from Experiment 16 so that G9 through G16 connect to pins 2 through 9 and G4Ø through G47 connect to pins 1Ø through 17 as shown in Fig. 42. Finally, set all switches to their logic "1" position.

When you are certain that the connections have been made correctly, turn on the power supply.

Start with port A. Enter the commands

```
POKE 5Ø957,Ø
POKE 5Ø956,15
POKE 5Ø957,4
```

and note that it is LEDØ through LED3 which turn off instead of LED8 through LED11 as was the case in Experiment 16. A review of the PIA pin listing in the specification sheets of the appendix will show that port A has been connected to LEDØ through LED7 while port B has been connected to the right-hand set of LED's.

You can now go through the steps outlined in Experiment 16 to investigate the functioning of the PIA or devise your own procedure.

Line 2Ø on the program listed at the end of experiment 16 should be changed to the address for the breadboard PIA as follows

```
2Ø X = 5Ø956:REM SET STARTING ADDRESS OF PIA
```

When you are satisfied that you understand the functioning of the breadboard mounted PIA, turn off the power supply, then CAREFULLY remove the PIA by prying up gently on both ends. Remember -- the larger the chip, the easier it is to damage the pins. Be certain that the PIA moves up slowly and evenly.

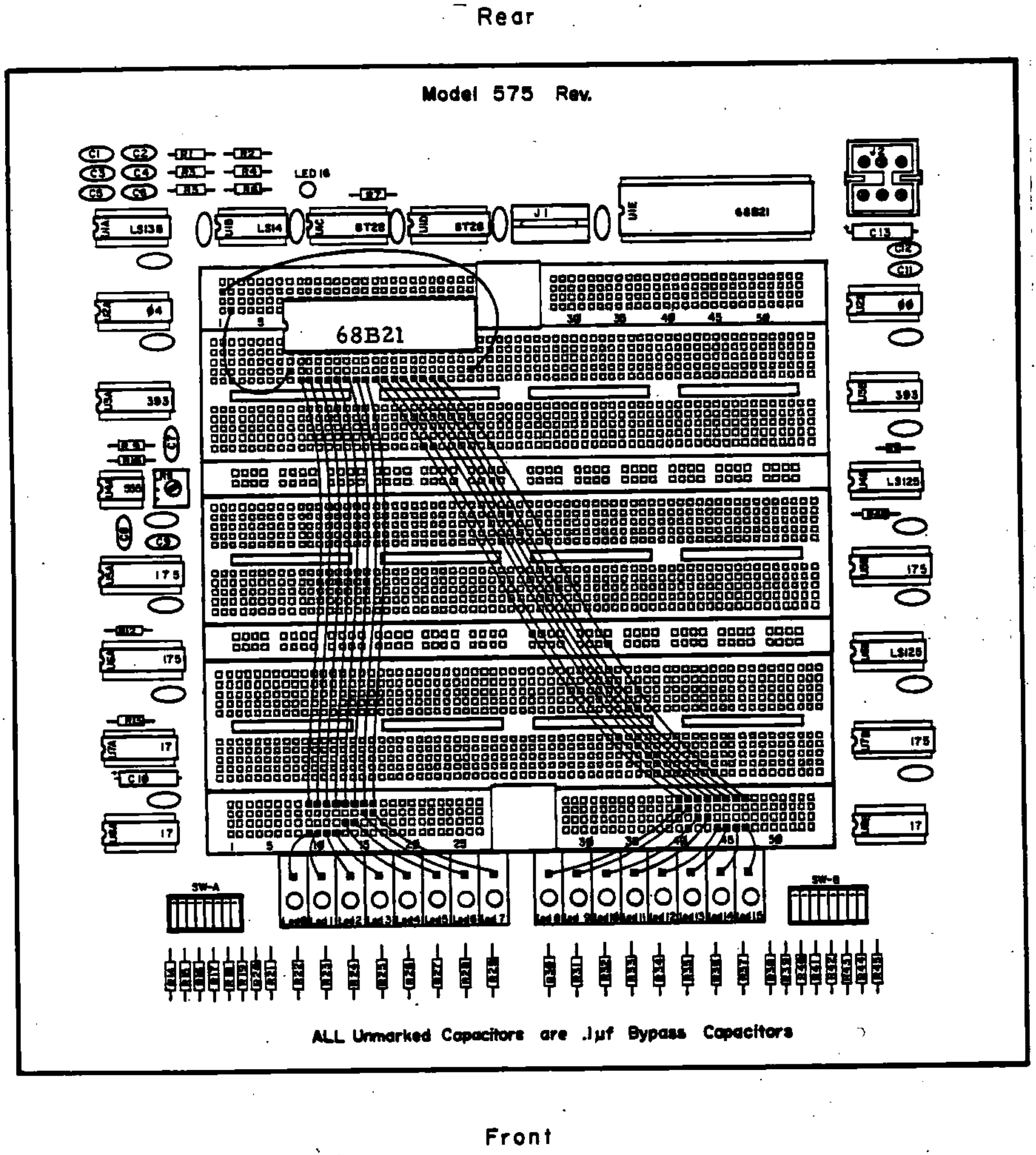


Figure 42
Connection for Use of Additional PIA

# EXPERIMENT 18

## TITLE

Interfacing the Computer With a Weathervane

## PURPOSE

The purpose of this experiment is to illustrate how the ports of the CA-24 board PIA can be used to pick up and display wind direction information from a weathervane.

## EQUIPMENT

CA-24 board and power supply
OSI computer (C4PMF, C4PDF, C8PDF)
Sixteen short jumper wires
One medium jumper wire
Eight long jumper wires
One DIP switch
or
One weathervane with position sensors (optional)

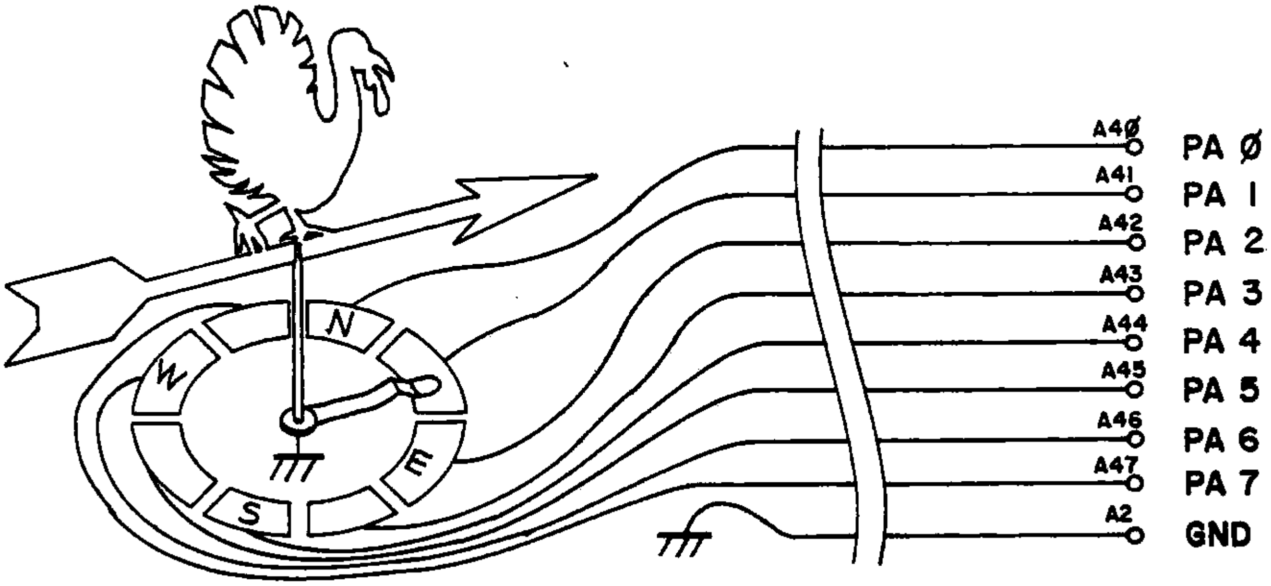
## DISCUSSION

The following illustration of weathervane position sensing uses straightforward switch contact grounding of a PIA port line for the input signal. This approach simplifies the connection to the PIA port but requires nine wires from the weathervane to the CA-24 board. It should be possible to reduce the number of wires but only at the expense of decoding circuitry between the weathervane and the PIA port.

Long wires between your computer and some external device can give problems. If you plan to monitor an actual weathervane, get the system working first with the weathervane positioned next to the CA-24 board and connected with a short cable. When the longer cable is installed, it may be necessary to use additional circuitry to drive the line.

If you use this experiment simply as an illustration of a practical use for the PIA, then a set of eight switches like those installed on the CA-24 board can be used as a substitute for the weathervane position sensing contacts. Consequently, the weathervane itself is optional equipment for this experiment.

If a weathervane is to be used, it must have some type of mechanical position sensing device attached to the rotating shaft. This could be a set of slide contacts or magnetically activated reed switches. In any case, there should be only small gaps between the closure conditions. Such an arrangement is shown in the diagram below.



The diagram has been drawn so that the closure when the weathervane is pointing north will be sensed by PAØ, the lowest bit line of port A, northeast will be sensed by PA1, etc.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Remove all previous circuitry from the board. Position the extra DIP switch supplied with this manual on strip B as shown in Fig. 43. The switch should be mounted with the same orientation as SW-A and SW-B at the front of the CA-24 board.

Use the one medium jumper wire to connect terminal A2 to a bus line on strip C as shown in Fig. 43. Then use eight short jumper wires to connect the front pins of the switch block to the strip C bus line which was connected to A2 (GND). Next, use the additional eight short jumper wires to connect the rear switch block pins to terminals A4Ø through A47 as shown in Fig. 43. Finally, connect A3Ø through A37 directly to LEDØ through LED7.

Reference to the terminal strip signal list (page 4) or to Experiment 16 will show that the LED's have been connected to port B while the switches have been connected such that each switch can ground a line of port A of the PIA.

When you are certain that all connections have been made correctly, turn on the power supply.

Boot up, UNLOCK your computer, and enter NEW. Type in the program shown below.

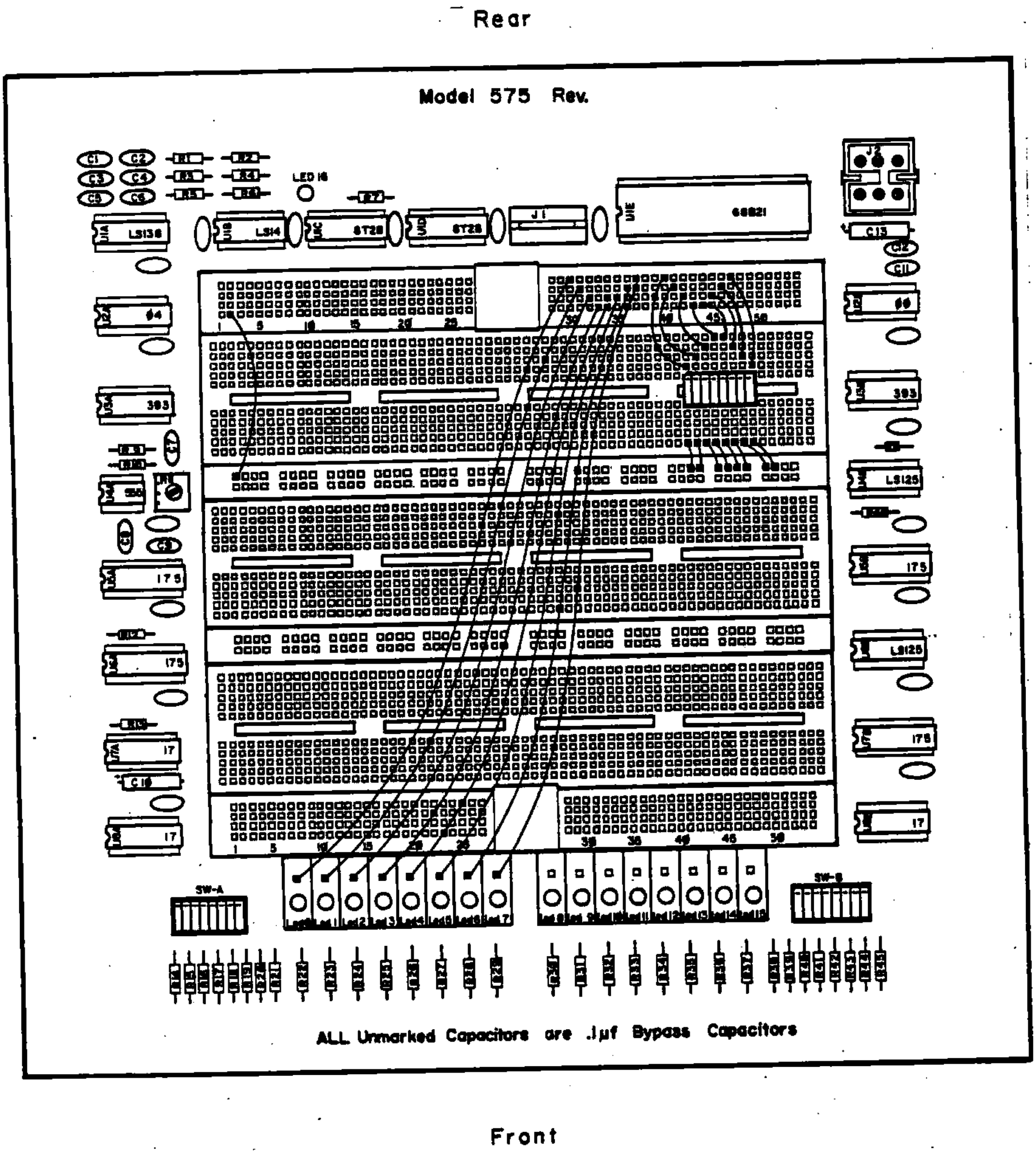


Figure 43
Connections for Switch Simulation of Weathervane Positions
(PIA Interfacing)

```
10 REM WEATHERVANE PROGRAM
20 H = 300
30 POKE 50949,0:POKE 50951,0
40 POKE 50948,0:POKE 50950,255
50 POKE 50949,4:POKE 50951,4
60 DIR = PEEK(50948)
70 IF DIR = 254 THEN W$ = "NORTH"
80 IF DIR = 253 THEN W$ = "NORTH EAST"
90 IF DIR = 251 THEN W$ = "EAST"
100 IF DIR = 247 THEN W$ = "SOUTH EAST"
110 IF DIR = 239 THEN W$ = "SOUTH"
120 IF DIR = 223 THEN W$ = "SOUTH WEST"
130 IF DIR = 191 THEN W$ = "WEST"
140 IF DIR = 127 THEN W$ = "NORTH WEST"
150 IF H = DIR OR DIR = 255 THEN 60
160 H = DIR:DS = 255 - DIR
170 POKE 50950,DS
180 PRINT"WIND DIRECTION IS ";W$
190 GOTO 60
```

Note that the program has been written to accommodate break-before-make switches by testing for no switch closure (line 15Ø). If your weathervane has make-before-break switches, this portion of the program will need revision.

Set the left-hand switch to its closed position (rear end down) and all other switches open. RUN the program.

The message

WIND DIRECTION IS NORTH

should appear on the video display and LEDØ should light while LED1 through LED7 are off.

Note that moving the left switch to its open position will not affect the output. Leave the left switch in its open position and close one of the other switches. Note the video and LED displays. Test each of the switches, returning it to its open position before closing the next.

If you close a second switch before opening the first, a new output will be given but it will have no meaning as the type of weathervane suggested for use cannot produce this type of signal.

If you plan to do Experiment 19 right after this experiment, and you do not have four momentary contact switches which can be used to simulate a closure on a keyboard, leave the DIP switch block in place.

## EXPERIMENT 19

### TITLE

Interfacing the Computer With a Keyboard

### PURPOSE

The purpose of this experiment is to illustrate how the two ports of the CA-24 board PIA can be used to pick up data from a keyboard which has as many as 64 characters.

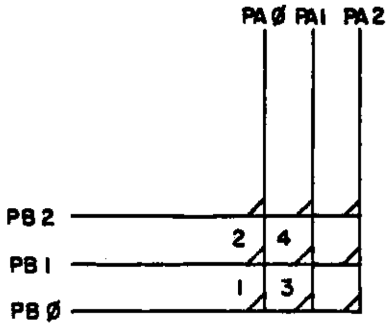
### EQUIPMENT

CA-24 board with power supply
OSI computer (C4PMF, C4PDF, C8PDF)
Four short jumper wires
Four medium jumper wires
One DIP switch
or
Four momentary contact switches (optional)

### DISCUSSION

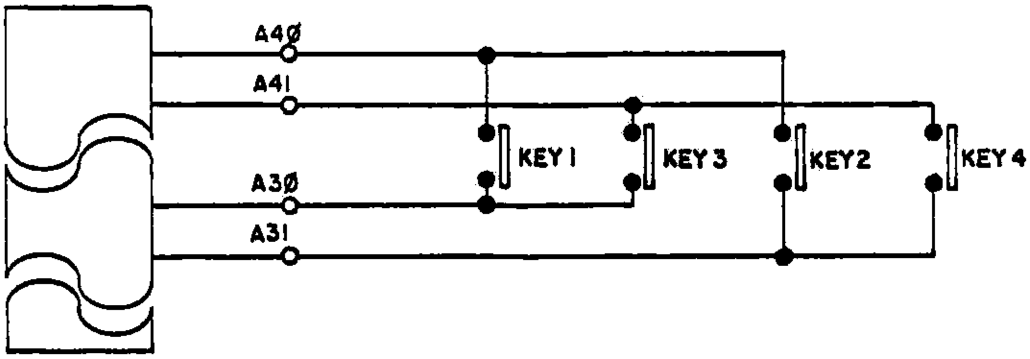
In Experiment 18, information was picked up by an input port of the CA-24 board PIA through the technique of simply grounding one of the lines. If this technique were used for a keyboard, one input line would be required for each character.

An alternate method of reading a keyboard utilizes both ports of the PIA and accomodates the sensing of up to 64 characters with only 16 lines. One port serves to output a high level on one line (other output lines low), while the other port is used to sense which line is high. If the output port is used to address rows, and the second port is used to address columns of a keyboard matrix, the key corresponding to a row-column intersection can be identified. A schematic diagram of such an arrangement is shown below.



Only three lines for each port have been shown, and only the four intersections used in this experiment have been labeled.

The wiring diagram which will be referred to in the PROCEDURE section is shown below. Reference to it here will help in understanding how the PIA selects an intersection.



When port A is specified as an input port, the default bit pattern (no connection made to the input lines) is all bits high. If a logic "Ø" is placed on the PBØ line (all other lines logic "1") then a closure of key 1 will pull the PAØ line low while a closure of key 3 will pull the PA1 line low. Closure of keys 2 and 4 will have no effect (and if more port B lines are used, closures of keys associated with those lines will have no effect).

With a logic "Ø" on only the PBØ line, then, the PEEK command is used to find the bit patterns of port A. In this experiment where only the first two lines of port A are used, the PEEK command will be used in conjunction with AND 3 (see Experiment 16). If line PAØ is found to be low, then key 1 is closed. If line PA1 is found to be low, then key 3 is closed. If both lines are high, then neither key 1 nor key 3 is closed.

Next, line PB1 is set at logic "Ø", while all other output port lines are high. Again, the bit pattern of port A is examined. This time a selection can be made between key 2 and key 4.

## PROCEDURE

If the circuitry for Experiment 18 is still on the board and you do not have four momentary contact switches to simulate key closure, the DIP switch block on the breadboard can be used. The procedure description will be for this situation.

Be certain that the power supply to the CA-24 board is turned off. Remove all of the jumper wires, but leave the DIP switch block in place. Use four short jumper wires and four medium jumper wires to connect the switches to the PIA port lines as shown in the wiring diagram above and in Fig. 44. Set all switches to their open position (rear end up).

When you are certain that all connections have been made properly and that the switches have been set to their open condition, turn on the power supply.

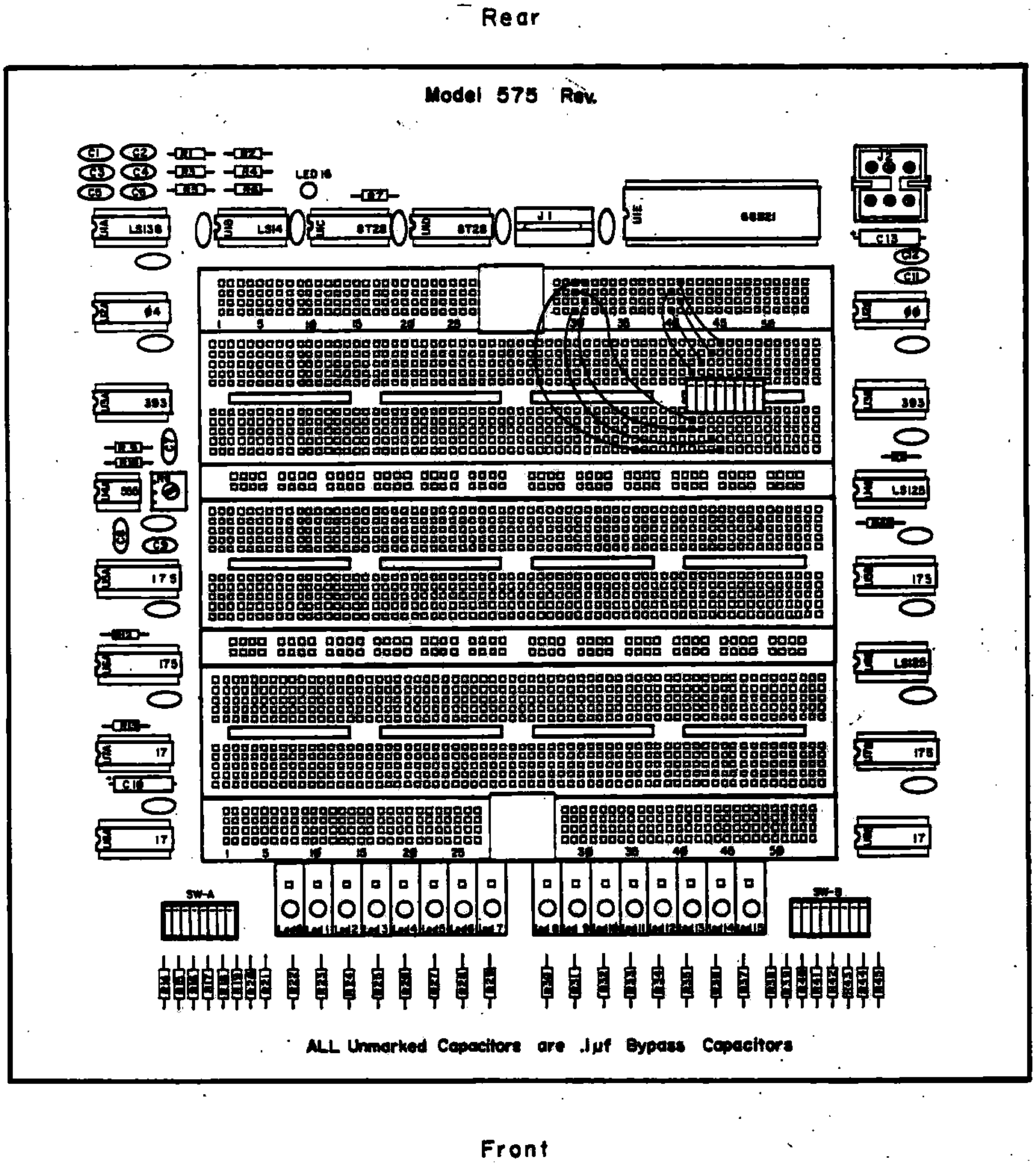


Figure 44
Connections for Switch Simulation of Four Keys of a Key Board
(PIA Interfacing)

Boot up your computer, UNLOCK, and enter NEW. Type in the program shown below.

```
10 REM USE OF PIA FOR KEYBOARD SENSING
20 A = 50948
30 POKE A+1,0:POKE A+3,0:REM SELECT DATA DIRECTION REGISTER
40 POKE A,0:POKE A+2,255:REM PORT A INPUT; PORT B OUTPUT
50 POKE A+1,4:POKE A+3,4:REM SELECT DATA REGISTER
60 KY = 0
70 POKE A+2,1:REM PUT 1 ON PB0
80 VA = PEEK(A) AND 3
90 IF VA = 2 THEN KY = 2
100 IF VA = 1 THEN KY = 4
110 POKE A+2,2:REM PUT 1 ON PB1
120 VA = PEEK(A) AND 3
130 IF VA = 2 THEN KY = 1
140 IF VA = 1 THEN KY = 3
150 REM NO TEST MADE FOR TWO KEYS DEPRESSED
160 IF KY = 0 THEN 60:REM NO KEY DEPRESSED
170 IF KY = ST THEN 60
180 ST = KY
190 PRINT"KEY DEPRESSED IS";KY
200 GOTO 60
```

When you are certain the program has been entered correctly, enter RUN. If all is working properly, nothing should happen. That is, no message should appear on the video screen as the computer is waiting for you to make a key closure. (The KY = Ø of line 6Ø has not been changed and so line 16Ø sends the program sequence back to line 6Ø.)

Move SW1 (the left most switch) to its closed position and then back open. The message

KEY DEPRESSED IS 1

should appear on the video screen. Rock SW2 to closure and back and note that the message now indicates a closure for key 2. Momentary closure of switch 3, then momentary closure of switch 4, will produce the messages for keys 3 and 4.

The key sensing technique illustrated in this experiment does not involve very sophisticated software. No test is made for two keys depressed at the same time, nor has provision been made for sensing a long closure which can be turned into repetitive use of that key. You may wish to try building these features into the program you have just entered.

When you feel satisfied with your understanding of this experiment, carefully remove the DIP switch block from the breadboard.

# EXPERIMENT 20

## TITLE

Interfacing the Computer with a Linear Air Track Photogate: Machine Language Programming

## PURPOSE

The purpose of this experiment is to illustrate the manner in which your computer can be used as a high resolution timing device.

## EQUIPMENT

CA-24 board with power supply
OSI computer (C4PMF, C4PDF, C8PDF)
Two long jumper wires
Two short jumper wires
Air track with photogate (optional)

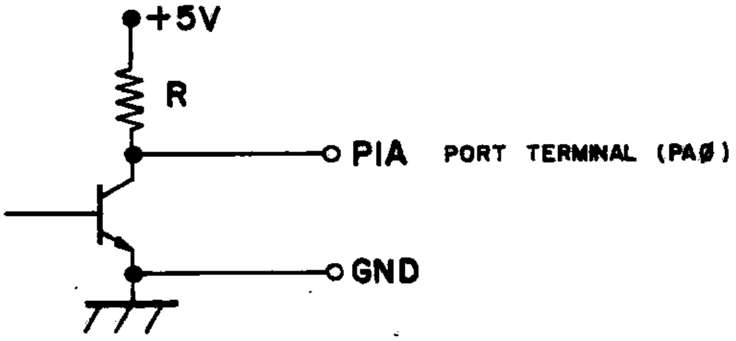
## DISCUSSION

Many physics laboratories make use of photogates in experiments where time information is needed. In some cases, such as measuring the time of free fall or the period of a pendulum, an initial signal from a light beam interruption starts a timer (usually a digital clock of some sort) and a second signal turns off the timer. In other cases, such as a glider on a frictionless linear air track, the interruption of a light beam turns on a timer which stays on until the beam is no longer interrupted. It is the second type of timing which is the dominant theme of this experiment. However, the two-signal timing technique will be illustrated as well.

An air track is a device which produces a nearly frictionless surface by creating a cushion of air under a glider which then is free to move along the track on this cushion of air. If a strip of metal (called a flag) of known length is mounted on the glider so that the known length extends horizontally, then the flag can be used to interrupt the light beam of a photocell gate as the glider moves along the track. The length of the flag divided by the time of light beam interruption gives the speed with which the glider was moving.

As stated earlier, it is common to use a digital clock in conjunction with the photocell gate to get the time information. If several air tracks are to be used in the laboratory at the same time, then as many clocks as air tracks are required and that can get expensive. Instead of clocks, the CA-24 board and the OSI computer can be used for up to 16 tracks (32 if an extra PIA is utilized). Of course, timing at different tracks cannot go on simultaneously. However, the video monitor can display a time-of-flight and speed message which is keyed to the air track being used.

Crucial to the use of the computer as a timing device is the compatibility of the photogate with the input signal requirements of the CA-24 board. Photogates have various designs, but most use either a photo resistor or a photo transistor as a sensing device which then either has with it or requires additional circuitry to produce high and low voltage levels which are TTL compatible. If the gate requires power from the timer with which it is associated, power can be supplied by the CA-24 board instead. If the circuitry of the gate has its own power, then probably the ground and gate level leads can be connected directly to the CA-24 board ground and a PIA port terminal. In any case, the final stage of the gate should be similar to that shown below.



The value of R depends on the final stage transistor and typically is between 1kΩ and 4.7kΩ.

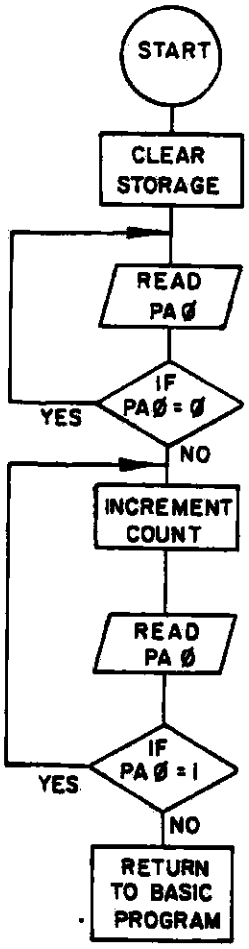
When the light beam of the photogate is interrupted, the final stage transistor must turn off causing the signal level to the PIA to go high. Alternatively, if you do not have a photogate to produce this transition, the signal from the CA-24 board clock can be used. By choosing a terminal near the end of the divide chain at which the frequency is about 1 Hz, a high level with a duration of about Ø.5 sec can be sent to the PIA. This will simulate a flag interrupting the photogate light beam for about Ø.5 sec.

Most of the effort in this experiment will go into writing and understanding a program which will produce time information and compute speed. Time information will be produced by having a memory location store the number of times a machine code loop is executed while the gate level stays high. The number of loop executions multiplied by the time for one loop gives the beam interrupt duration. The shorter the time required for one loop, the better is the resolution of the timer. Consequently machine code will be used instead of the FOR-NEXT loop of BASIC.

Since programming in machine language is complex and confusing at worst and time consuming at best, as much software as possible will be worked out in BASIC. Then the programming which requires speed will be imbedded in the BASIC program by way of the USR(X) function.

We will start by examining the machine code program at three levels. The first level will be that of the flow chart which will give the overall picture. Next we will see the program written in assembly language. Finally, we will see the program written in hexadecimal machine code.

For the flow chart shown at the right, we have chosen the first data line of port A (PAØ) as the photogate signal input line. After setting the contents of the count storage addresses (those which will be incremented in the second loop) to zero, the program will test PAØ to see if it has gone high. As long as it stays low, the program will go back and test again. This loop takes about 5 microseconds to execute. Consequently there cannot be more than a 5 microsecond delay between the level transition and the beginning of the timing loop.



When PAØ goes high, the contents of a count storage address are increased by one. Then PAØ is tested again. If it is still high, the count storage address is incremented again. This process continues until PAØ goes low. Then control is returned to the program written in BASIC which will pick up the stored counts, convert them to time information, and calculate the speed of the glider.

As we will see, two addresses are required to store counts sufficient to time a beam interrupt of up to one second with high resolution. Only 255 counts can be stored in one address, and since the timing loop has been adjusted to take about 15 microseconds, an interval of approximately 4 milliseconds could be timed. In two addresses, 65535 counts can be stored which will allow the timing of slightly under one second. More addresses could be used to increase the timing capability with the same resolution. Alternatively, the timing loop can be expanded. For example, if one millisecond resolution is tolerable, the timing loop can be expanded to one millisecond in which case an interval of somewhat over one minute can be timed.

The assembly code for this program is shown below together with explanatory comments.

```
        COUNT = $5FØØ  ;LABEL LOW COUNT STORAGE ADDRESS
        CNTHI = $5FØ1  ;LABEL HIGH COUNT STORAGE ADDRESS
        PORTIN = $C7Ø4 ;LABEL PIA PORT ADDRESS
        LDA   #$ØØ     ;LOAD ACCUMULATOR (IMMEDIATE)
        STA   COUNT    ;CLEAR LOW COUNT STORAGE ADDRESS
        STA   CNTHI    ;CLEAR HIGH COUNT STORAGE ADDRESS
START   LDA   PORTIN   ;LOAD PIA PORT
        AND   #$Ø1     ;TEST FOR PAØ HIGH
        BEQ   START    ;CONTINUE TEST UNTIL BEAM INTERRUPT
LOOP    INC   COUNT    ;ADD ONE TO LOW COUNT STORAGE
        BNE   SKIP     ;BYPASS HIGH STORAGE INCREMENT UNTIL
                       ;LOW STORAGE ZERO
        INC   CNTHI    ;ADD ONE TO HIGH COUNT STORAGE
SKIP    LDA   PORTIN   ;LOAD PIA PORT
        AND   #$Ø1     ;TEST FOR PAØ LOW
        NOP
        NOP
        NOP            ;LENGTHEN TIMING LOOP
        NOP
        NOP
        NOP
        BNE   LOOP     ;REPEAT TIMING LOOP UNTIL PAØ LOW
        RTS            ;RETURN TO BASIC PROGRAM
```

The program begins by loading zeroes into the microprocessor accumulator. Then the contents of the accumulator are stored in the loop count storage addresses in order to clear them. These have been chosen as $5FØØ and $5FØ1. Next, the contents of the PIA data register for port A are loaded into the accumulator. The test for PAØ high is done with the logical AND. The contents of the accumulator are ANDed with the bit pattern ØØØØØØØ1. The result will be zero if the first (right most) bit of the accumulator is Ø (regardless of the other bits). When the first bit of the accumulator is high, the result is not zero. Since BEQ means branch on zero result, a high level for the first bit (picked up from PAØ) will cause the program to move into the timing loop.

The first step in the loop is the incrementing of the contents of the low storage address. The result will not be zero again until loop execution number 256. During the first 255 passes, then, the program skips the incrementing of the high storage address. For each 256th pass, however, address $5FØ1 is incremented.

In either case, the next operation is to look at the level on PAØ. If it is still high, the program goes back to the storage address increment step. If PAØ is low, control is returned to the BASIC program.

The NOP commands are for lengthening the timing loop so that a period of just under one second can be timed.

We are now ready to see the machine code listing. It is shown below together with the memory addresses where the operation codes will be stored and the assembly code which we examined earlier. Since the imbedding of the machine language program in the BASIC language program involves POKE commands, the decimal equivalents of the hexadecimal addresses and operation codes have been included.

| PROGRAM STORAGE | | OP CODE | | ASSEMBLER |
|---|---|---|---|---|
| DECIMAL | HEX | DECIMAL | HEX | |
| 24064 | 5E00 | 169 | A9 | LDA #$00 |
| 24065 | 5E01 | 0 | 00 | |
| 24066 | 5E02 | 141 | 8D | STA $5F00 |
| 24067 | 5E03 | 0 | 00 | |
| 24068 | 5E04 | 95 | 5F | |
| 24069 | 5E05 | 141 | 8D | STA $5F01 |
| 24070 | 5E06 | 1 | 01 | |
| 24071 | 5E07 | 95 | 5F | |
| 24072 | 5E08 | 173 | AD | LDA $C704 |
| 24073 | 5E09 | 4 | 04 | |
| 24074 | 5E0A | 199 | C7 | |
| 24075 | 5E0B | 41 | 29 | AND #$01 |
| 24076 | 5E0C | 1 | 01 | |
| 24077 | 5E0D | 240 | F0 | BEQ |
| 24078 | 5E0E | 249 | F9 | |
| 24079 | 5E0F | 238 | EE | INC $5F00 |
| 24080 | 5E10 | 0 | 00 | |
| 24081 | 5E11 | 95 | 5F | |
| 24082 | 5E12 | 208 | D0 | BNE |
| 24083 | 5E13 | 3 | 03 | |
| 24084 | 5E14 | 238 | EE | INC $5F01 |
| 24085 | 5E15 | 1 | 01 | |
| 24086 | 5E16 | 95 | 5F | |
| 24087 | 5E17 | 173 | AD | LDA $C704 |
| 24088 | 5E18 | 4 | 04 | |
| 24089 | 5E19 | 199 | C7 | |
| 24090 | 5E1A | 41 | 29 | AND #$01 |
| 24091 | 5E1B | 1 | 01 | |
| 24092 | 5E1C | 234 | EA | NOP |
| 24093 | 5E1D | 234 | EA | NOP |
| 24094 | 5E1E | 234 | EA | NOP |
| 24095 | 5E1F | 234 | EA | NOP |
| 24096 | 5E20 | 234 | EA | NOP |
| 24097 | 5E21 | 234 | EA | NOP |
| 24098 | 5E22 | 208 | D0 | BNE |
| 24099 | 5E23 | 235 | EB | |
| 24100 | 5E24 | 96 | 60 | RTS |

Note that we have chosen to store the program in memory locations beginning with 24064 ($5E00). Since for a 24K computer the end of memory is address 24575 ($5FFF), our program will be stored at the beginning of the last 512 addresses or at the beginning of the last two pages (one page is 256 locations).

It is important to protect the machine language program from inadvertently being written over when the BASIC program in which it is imbedded stores a string variable picked up from an INPUT statement. (The BASIC interpreter uses high memory for such storage.) The last two pages can be protected by tricking the computer into believing it does not have these pages as it goes about its own business. The BASIC interpreter uses addresses 132 and 133 to specify the end of memory value. For a 24K machine, the high eight bits, represented by $5F, are stored in 133 while the low eight bits ($FF) are stored in 132. To change the end of memory to $5DFF, the contents of only 133 need to be changed from $5F to $5D. That can be done by a POKE command. Since the decimal equivalent of $5D is 93, the command

```
POKE 133,93
```

will tell the BASIC interpreter that the memory ends at $5DFF and prevent it from using the top two pages of actual memory. This POKE command will begin the BASIC program.

Finally, we are ready to see the BASIC program which, with the embedded machine language program, will time the air track glider.

```
10 REM SINGLE PHOTOGATE TIMER PROGRAM
20 POKE 133,93:REM RESERVE HIGH MEMORY
30 POKE 8955,0:POKE 8956,94:REM SET USR(X) ENTRY POINT
40 POKE 50949,0:POKE 50948,0:POKE 50949,4:REM PORT A AS INPUT
50 P = 24064:REM FIRST ADDRESS FOR USR(X) PROGRAM
60 FOR I = 1 TO 37
70 READ M:POKE P,M
80 P = P + 1
90 NEXT I
100 INPUT"READY (Y/N)";A$
110 IF A$ <> "Y" THEN 100
120 X = USR(X)
130 L = 0.02
140 CS = PEEK(24320):CO = PEEK(24321)
150 CT = CS + CO * 256
160 TT = CT * 15E-6 + CO * 2.5E-6
170 VEL = L / TT
180 PRINT"TIME MEASURED IS ";TT;" SEC
190 PRINT"SPEED FOR 20 CM FLAG IS ";VEL;" M/SEC"
200 GOTO 100
210 DATA 169,0,141,0,95
220 DATA 141,1,95,173,4,199
230 DATA 41,1,240,249
240 DATA 238,0,95,208,3
250 DATA 238,1,95,173,4,199
260 DATA 41,1
270 DATA 234,234,234,234,234,234
280 DATA 208,235,96
```

When a BASIC program encounters the USR(x) function, it goes to addresses 8955 and 8956 to find where the machine code program has been stored. Consequently, the value of the address where the machine language program begins must be placed in those two addresses. This is done in line 3Ø. The low eight bits are in 8955 and are all zeroes. The high eight bits are in 8956. Since the high eight bit pattern is $5E which is 94 in decimal representation, 94 is POKED into 8956.

Line 4Ø sets up port A of the PIA as an input port.

Lines 5Ø through 9Ø load the machine code program into memory beginning with address 24Ø64 ($5EØØ). This is done by reading the code from the DATA statements in lines 21Ø through 28Ø and POKEing it into memory. (Note that the sequence of numbers in the DATA statements is exactly that of the machine code listing shown earlier.)

Since the loading process is relatively slow and needs to be done only once, the program has been designed to allow the user to start the machine code timing program without repeating the first part of the BASIC program. When Y is entered in response to the "READY (Y/N)" message, control is transferred to the machine code timing program and loop counts are stored in addresses 2432Ø and 24321. When control is returned to the BASIC program, the loop counts are picked up and totaled (lines 14Ø and 15Ø). Next, the calculations of time and velocity are made, and the results displayed on the video monitor. Finally, control is returned to line 1ØØ so that a second run can be made.

It was stated earlier that each timing loop takes about 15 microseconds. This time is determined by summing the number of cycles which are taken by the operations in the timing loop. (Most sources which list the 65Ø2 operation codes give the number of cycles for each operation.) Since the Ohio Scientific computers are basically 2 MHz machines, each cycle will take Ø.5 microseconds. However, to achieve enhanced reliability, certain operations are slowed to 1.Ø microsecond. The result is that the calculation for time made on the basis of Ø.5 microseconds per cycle (which is how the 15E-6 in line 15Ø was obtained) is low by about 1.5%. The 2.5E-6 also in line 15Ø takes account of the extra time needed to increment the high count address every 256 counts. Clearly, this addition is small compared to the 1.5% and could be left out. If you plan to use your OSI computer for very accurate timing, the simplest procedure is to adjust the number in line 15Ø so that the program correctly calculates a time interval known to a high degree of accuracy. Such a time interval could be obtained using a square wave signal generator. (The PROCEDURE section of this experiment explains how a square wave can be used to simulate the photogate beam interrupt.)

PROCEDURE

Be certain that the power supply to the CA-24 board has been turned off. Remove all previously used wires and components.

We will begin by simulating a photogate signal with a divide chain clock pulse of about Ø.5 sec. Use a long jumper wire to connect A4Ø to G49 as shown in Fig. 45. Next, use the two short jumper wires to connect A3 to A4 and G49 to LED15.

Now turn on the power supply. LED15 should be blinking on and off. Adjust R11 (the clock frequency variable resistor) so that the rate of blink is about one per second. This will produce alternate low and high levels at G49 (and so at A4Ø, the terminal for PAØ) each of which has a time duration of about Ø.5 seconds.

If you have not already done so, boot up the computer, UNLOCK and enter NEW. Enter the BASIC program shown earlier.

When you are certain that the program has been typed in correctly, enter RUN. In response to the READY message, type Y but do not press RETURN. With your finger on the return key, watch LED 15. For one of the cycles, just after the light turns off press RETURN. At the moment the LED turns off again, the time and speed messages which have been written into the program should appear on the video display. Additional measurements of the time interval can be made simply by typing Y, then, as the LED turns off, pressing RETURN.

If you have a photogate and plan to connect it to the CA-24 board PIA, a word of caution is in order. The breaking of a light beam will produce signal "bounce" just as will mechanical switches. Rather than using hardware to solve this problem, a delay loop can be inserted into the machine language program such that the test for low level does not occur until a stable high has been achieved. The OP CODE and ASSEMBLER CODE for such a loop is listed below.

```
      OP CODE              ASSEMBLER

  DECIMAL    HEX

    169      A9            LDA #$??
      ?      ??
    141      8D            STA $5FØ2
      2      Ø2
     95      5F
    2Ø6      CE            DEC $5FØ2
      2      Ø2
     95      5F
    234      EA            NOP
    234      EA            NOP
    234      EA            NOP
    2Ø8      DØ            BNE (DEC)
    248      F8
```

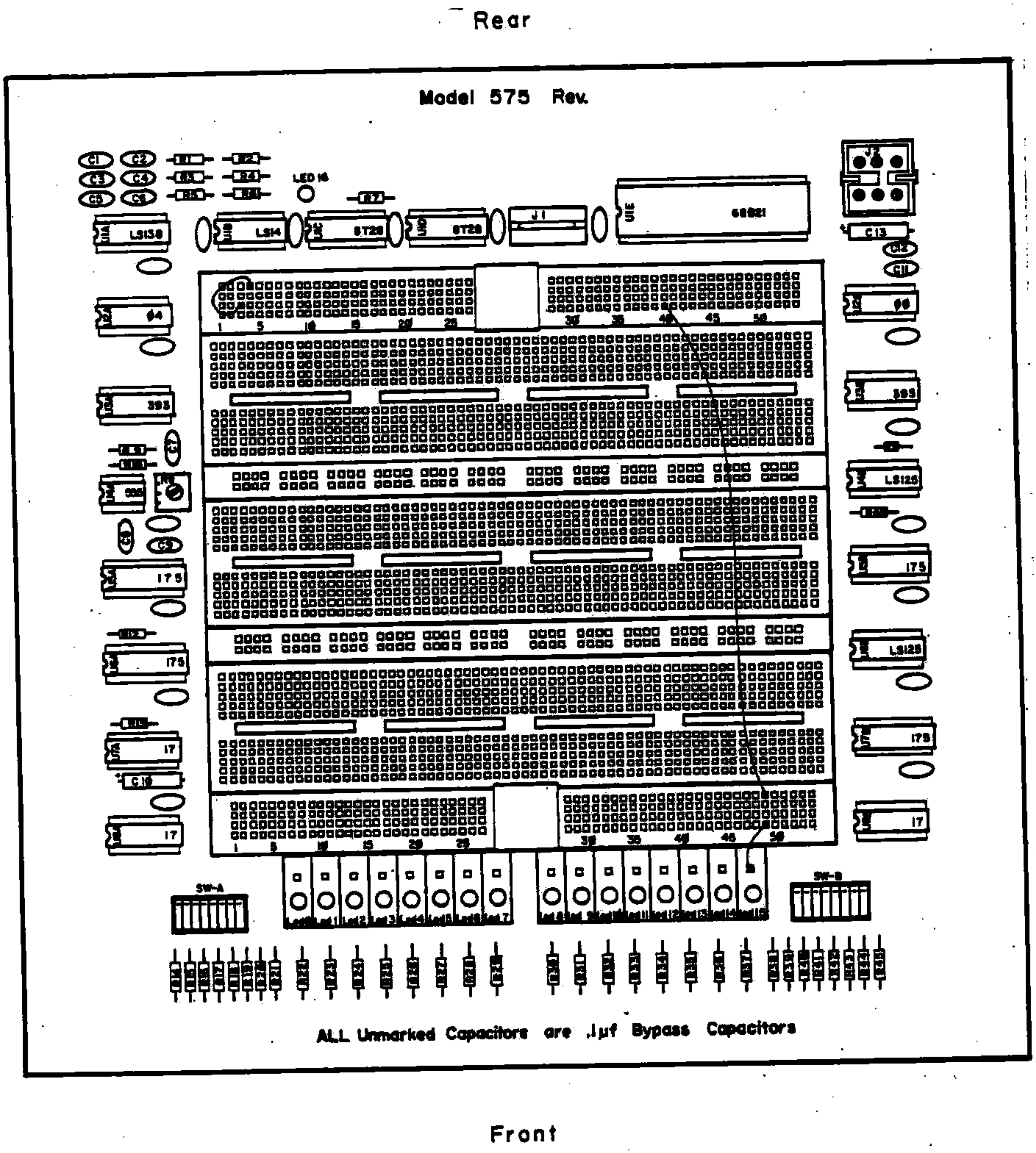


Figure 45
Connections for Producing a Clock Pulse to Simulate a Single Photogate Interrupt Signal

This loop can be inserted into the basic program by changing line 6Ø to

```
6Ø FOR I = 1 to 5Ø
```

and adding the lines:

```
221 DATA 169
222 DATA ?
223 DATA 141,2,95,2Ø6,2,95
224 DATA 234,234,234
225 DATA 2Ø8,248
```

The number which replaces the ? in line 222 will determine the number of times the loop is executed. Each time through takes 7.5 microseconds. You probably will not need a number larger than 5Ø.

We can now change the wiring and the program to illustrate the use of two photogates spaced a known distance apart. Turn off the CA-24 board power supply and remove the two short jumper wires. Change the G strip end of the long jumper wire so that it connects to terminal G4Ø, and add a long jumper wire to connect A41 to G47. These changes are shown on Figure 46. Set all switches on switch block B to their closed position (front end up).

Reference to Table 1 (page 4) will show that you have connected SW1B to PAØ and SW8B to PA1. SW1B will simulate the first photogate and SW8B will simulate the second photogate. Initially both are producing a low level on the input lines PAØ and PA1.

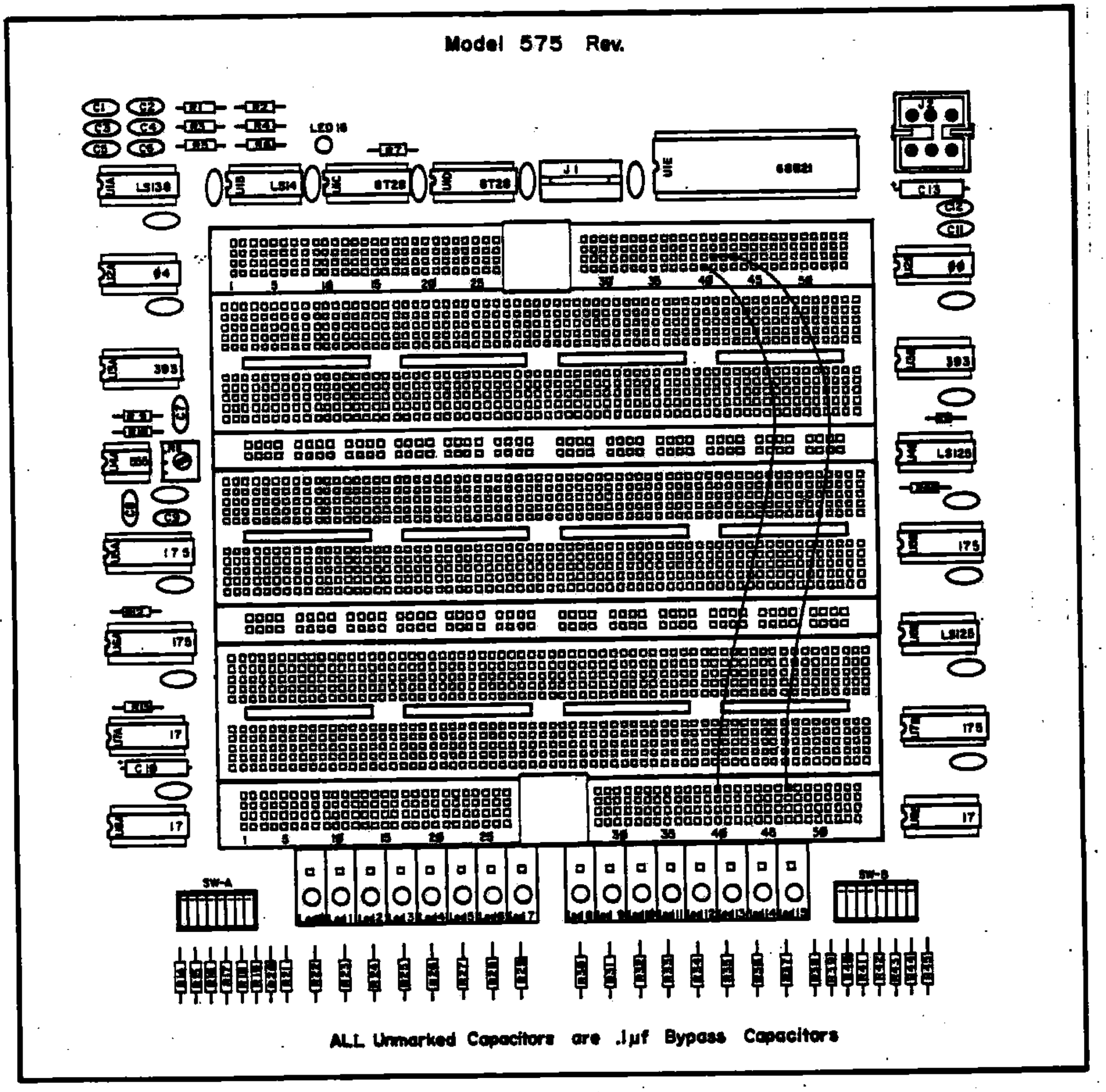
Next, change the BASIC program lines 1Ø, 19Ø, 26Ø, and 28Ø as follows:

```
1Ø REM TWO PHOTOGATE TIME PROGRAM
19Ø PRINT"SPEED FOR 2Ø CM GATE SEPARATION IS ";VEL;" M/SEC"
26Ø DATA 41,2
28Ø DATA 24Ø,235,96
```

The changes in lines 1Ø and 19Ø, of course, are only cosmetic. They bring labeling into accord with the different situation. The critical changes are in lines 26Ø and 28Ø. The change of 1 to 2 in line 26Ø means that now PA1 is being tested for signal change at the end of each timing loop instead of PAØ. The change of 2Ø8 to 24Ø is the change from BNE (branch on not equal to zero) to BEQ (branch on equal to zero). This will result in the termination of the timing loop when the signal at PA1 goes high.

When you have made the changes both in wiring and in the program, turn on the CA-24 board power supply. RUN the program, type Y, and press return. Next, you will need to switch SW1B to its open position and then, within one second, switch SW8B to its open position. The end switches of the switch block were chosen to make this task as easy as possible. Position both

Rear



Front

Figure 46
Connections for Use of SW1B and SW8B to Simulate
Interrupt Signals from Two Photogates

hands so that the switching can be done in fairly rapid succession and move the switches. Note that the time and speed messages appear on the video display as soon as the second switch is changed.

For the two-photogate timing, signal "bounce" is not a problem. The first rise at PAØ begins the timing loop and the first rise at PA1 terminates the loop. The fact that there are falls and additional rises on each line makes no difference as the test has been completed after the first rise.

Before entering Y to run the timing program again, be certain to return the two switches to their closed position.

# EXPERIMENT 21

## TITLE

The Asynchronous Communications Interface Adapter (ACIA)

## PURPOSE

The purpose of this experiment is to become familiar with the function of an Asynchronous Communications Interface Adapter which will perform parallel to serial and serial to parallel conversion.

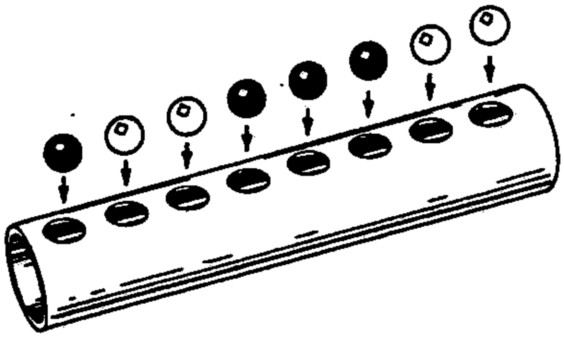
## EQUIPMENT

CA-24 board with power supply
OSI Computer (C4PMF, C4PDF, C8PDF)
One MC685Ø ACIA chip
Twenty short jumper wires
Seven medium jumper wires
Three long jumper wires

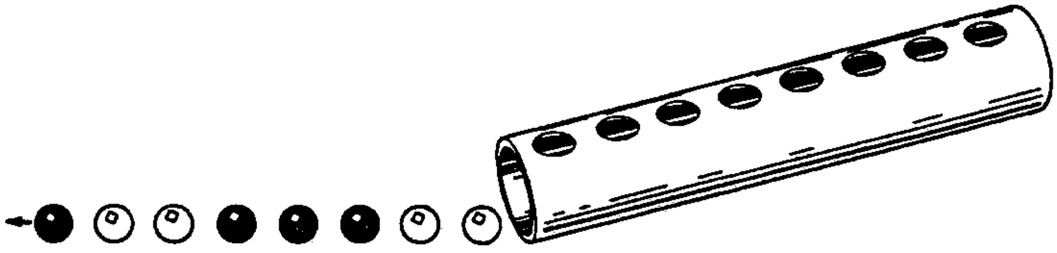
## DISCUSSION

Many mechanical devices which operate at relatively slow speeds require single bit data streams. To meet the need for serial (sequential) data, it is necessary to convert from parallel form (all bits of a data word simultaneously present). The conversion can be done with software (as will be shown later) but it is most easily done with the Asynchronous Communications Interface Adapter (ACIA). The ACIA will convert an ordered set of 1's and Ø's into a time sequence of 1's and Ø's. It will also convert a time sequence into an ordered set.

A situation analogous to this process is the placing of black and white marbles simultaneously into the holes of a pipe as shown below.



A time sequence can be achieved by rolling the marbles out the left end as shown below.



The electrical equivalent of the above example is achieved by the ACIA for which the high and low levels are equivalent to the black and white marbles. The rate at which the sequential highs and lows are sent to the receiving device can be tailored to match the speed of that device. While this method of transferring data is relatively slow, it does have the advantage of requiring only two wires for the transmission instead of a minimum of nine (for eight bits).

The ACIA used in this experiment is the Motorola type 685Ø. This versatile integrated circuit is capable of sending and receiving bit sequences in eight software selectable patterns. These patterns include the bracketing (called framing) of the character bit pattern (seven or eight bits which represent a number or a letter or some other character) by a start bit at the beginning and one or two stop bits at the end. In addition, some of the options include a parity bit (for error checking) just before the stop bit or bits.

The rate at which the bit sequence is transmitted or received (called Baud rate) is determined by an external clock signal together with three rate adjustment options which are software selectable. The sequence rate can proceed at the frequency of the clock, at 1/16 that frequency, or at 1/64 the clock frequency. For the latter two cases, the required data are internally synchronized at the reception of the start bit. External synchronization is required when the full clock frequency is used.

The most common clock frequency used with an ACIA is 19.2 kHz. The divide-by-16 option, then, will produce 12ØØ Baud while the divide-by-64 option will produce 3ØØ Baud.

An 8 bit control register (write only) in the ACIA selects the options described above. By means of a POKE command, a bit pattern can be written into the control register. Only five of the eight bits are needed to select Baud rate and bit pattern. In addition, then, three bits are available for a variety of transmitter control and interrupt features.

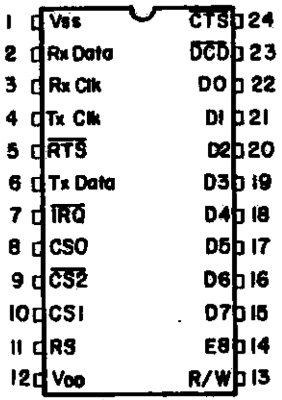
A summary of the control register options is shown below, followed by the ACIA pin assignment diagram and label listing.

| CR1 | CRØ | Function |
|---|---|---|
| Ø | Ø | ÷1 |
| Ø | 1 | ÷ 16 |
| 1 | Ø | ÷ 64 |
| 1 | 1 | Master Reset |

| CR4 | CR3 | CR2 | Function |
|---|---|---|---|
| Ø | Ø | Ø | 7 Bits + Even Parity + 2 Stop Bits |
| Ø | Ø | 1 | 7 Bits + Odd Parity + 2 Stop Bits |
| Ø | 1 | Ø | 7 Bits + Even Parity + 1 Stop Bit |
| Ø | 1 | 1 | 7 Bits + Odd Parity + 1 Stop Bit |
| 1 | Ø | Ø | 8 Bits + 2 Stop Bits |
| 1 | Ø | 1 | 8 Bits + 1 Stop Bit |
| 1 | 1 | Ø | 8 Bits + Even Parity + 1 Stop Bit |
| 1 | 1 | 1 | 8 Bits + Odd Parity + 1 Stop Bit |

| CR6 | CR5 | Function |
|---|---|---|
| Ø | Ø | $\overline{RTS}$ = low, Transmitting Interrupt Disabled |
| Ø | 1 | $\overline{RTS}$ = low, Transmitting Interrupt Enabled |
| 1 | Ø | $\overline{RTS}$ = high, Transmitting Interrupt Disabled |
| 1 | 1 | $\overline{RTS}$ = low, Transmits a Break label on the Transmit Data Output, Transmitting Interrupt Disabled |

| CR7 | Receive Interrupt Enable Bit |
|---|---|
| Ø | Receive Interrupt Disabled |
| 1 | Receive Interrupt Enabled<br>Receive Data Register Full, Overflow, or $\overline{DCD}$ |



| | |
|---|---|
| Vss | GND |
| Rx D | Receive Data |
| Rx CLK | Receive Clock |
| Tx CLK | Transmit Clock |
| RTS | Request-to-Send |
| Tx D | Transmit Data |
| IRQ | Interrupt Request |
| CSØ<br>CS2<br>CS1 | Chip Select |
| RS | Register Select |
| Vdd | +5V |
| R/W | Read/Write Control |
| E | Enable |
| DØ - D7 | Data Lines |
| DCD | Data Carrier Detect |
| CTS | Clear To Send |

Note that the two least significant bits of the control register (CRØ and CR1) also activate the master reset. These two bits must be set high after power-on to properly initialize the ACIA. Then the bit pattern needed to achieve the desired functioning of the ACIA is written into the control register.

Two 8 bit registers in the ACIA are used for data handling. The Transmit Data Register (write only) receives an 8 bit word from the data bus and holds it until the ACIA is ready to serialize it. The Receive Data Register (read only) stores the bit pattern which the ACIA receives in serial mode and upon command puts that pattern on to the data bus.

Finally, a fourth 8 bit register functions as a status register (read only). As the name implies, the contents of this register tell the status of the ACIA at any time during its operation.

A summary of the four registers is given below.

| Data Bus Line Number | Buffer Address | | | |
|---|---|---|---|---|
| | RS R / W<br>Transmit Data Register | RS R / W<br>Receive Data Register | RS R / W<br>Control Register | RS R / W<br>Status Register |
| | (Write Only) | (Read Only) | (Write Only) | (Read Only) |
| Ø | Data Bit Ø * | Data Bit Ø | Counter Divide Select 1 (CRØ) | Receive Data Register Full (RDFR) |
| 1 | Data Bit 1 | Data Bit 1 | Counter Divide Select 2 (CR1) | Transmit Data Register Empty (TDRE) |
| 2 | Data Bit 2 | Data Bit 2 | Word Select 1 (CR2) | Data Carrier Detect ($\overline{\text{DCD}}$) |
| 3 | Data Bit 3 | Data Bit 3 | Word Select 2 (CR3) | Clear to Send ($\overline{\text{CTS}}$) |
| 4 | Data Bit 4 | Data Bit 4 | Word Select 3 (CR4) | Framing Error (FE) |
| 5 | Data Bit 5 | Data Bit 5 | Transmit Control 1 (CR5) | Receiver Overrun (OVRN) |
| 6 | Data Bit 6 | Data Bit 6 | Transmit Control 2 (CR6) | Parity Error (PE) |
| 7 | Data Bit 7*** | Data Bit 7** | Receive Interrupt Enable (CR7) | Interrupt Request (IRQ) |

* Leading Bit = LSB = Bit 0

** Data Bit Will be Zero in 7 Bit Plus Parity Modes

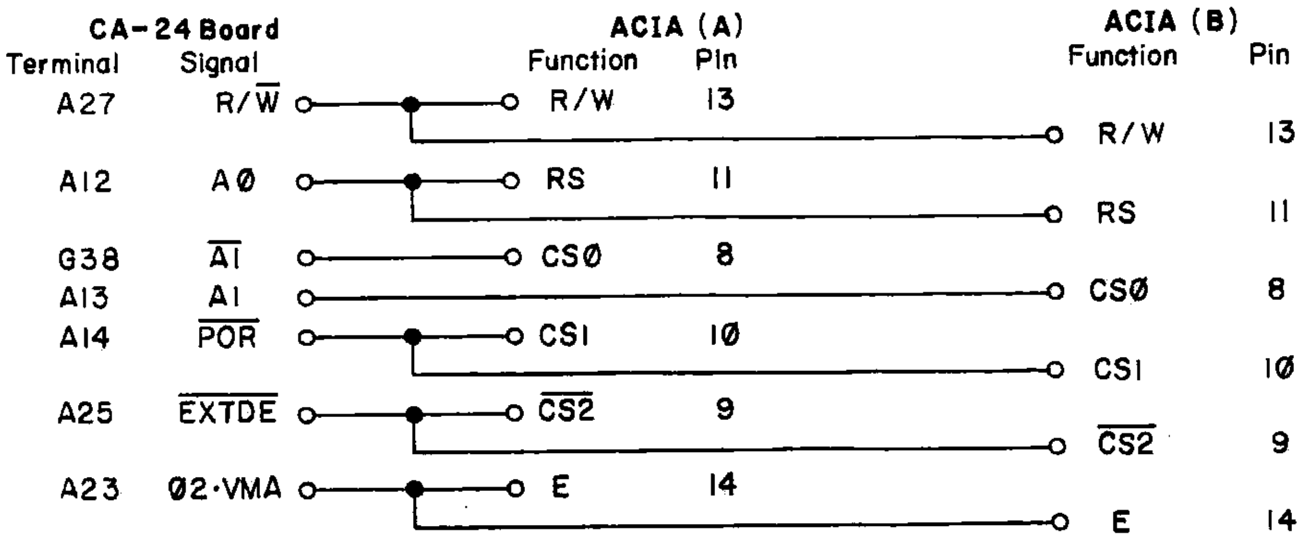
*** Data Bit is "Don't Care" in 7 Bit Plus Parity Modes

The registers are selected by a combination of levels at RS (pin 11) and R/W (pin 13). Only two memory locations in your computer are needed in conjunction with the four registers. One memory location writes to the control register and reads from the status register. The other memory location writes to the Transmit Data Register and reads from the Receive Data Register.

The computer addresses used for this are the same as those used for the extra PIA (see SECTION III, page 79 and Experiment 17). Since four addresses will activate the correct select lines, two ACIA's can be controlled at the same time. The assignment of addresses and ACIA registers is shown below.

| Addresses | | Uses | |
|---|---|---|---|
| Decimal | Hex | | |
| 5Ø956 | C7ØC | ACIA (A) | Control and Status Registers |
| 5Ø957 | C7ØD | ACIA (A) | Data Registers |
| 5Ø958 | C7ØE | ACIA (B) | Control and Status Registers |
| 5Ø959 | C7ØF | ACIA (B) | Data Registers |

Selection of ACIA (A) or ACIA (B) is accomplished through the connections shown below.



When either 5Ø956 or 5Ø957 is used in a POKE or PEEK command, A1 is low and consequently $\overline{A1}$ is high. Furthermore, $\overline{POR}$ is always high (see Figs. 3a and 3b, pages 5 and 6) and $\overline{EXTDE}$ will be low (see Fig. 3a noting that IOLA2 and IOLA3 will be high). Since an ACIA is selected when CSØ and CS1 are high and $\overline{CS2}$ is low, ACIA (A) will be selected. Processing of information begins when the ø2·VMA transition from low to high occurs at pin 14.

When either 5Ø958 or 5Ø959 is used in a POKE or PEEK command, A1 is high and consequently $\overline{A1}$ is low. All other conditions are the same as for 5Ø956 and 5Ø957. Consequently, ACIA (B) is selected and processing begins with the arrival of the ø2·VMA signal.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Remove all previously installed wires and components from the breadboard area. Carefully insert the 685Ø ACIA chip into strip B as shown in Fig. 47 so that pin 1 is front-left.

We will begin by setting up the circuit for transmitting serial data when parallel data are given to the ACIA. The schematic diagram for the connections is shown below and the wiring diagram is shown in Fig. 47. Note that the CA-24 board terminal labels are shown closest to the lines from the chip while the labels for the signals are shown next.

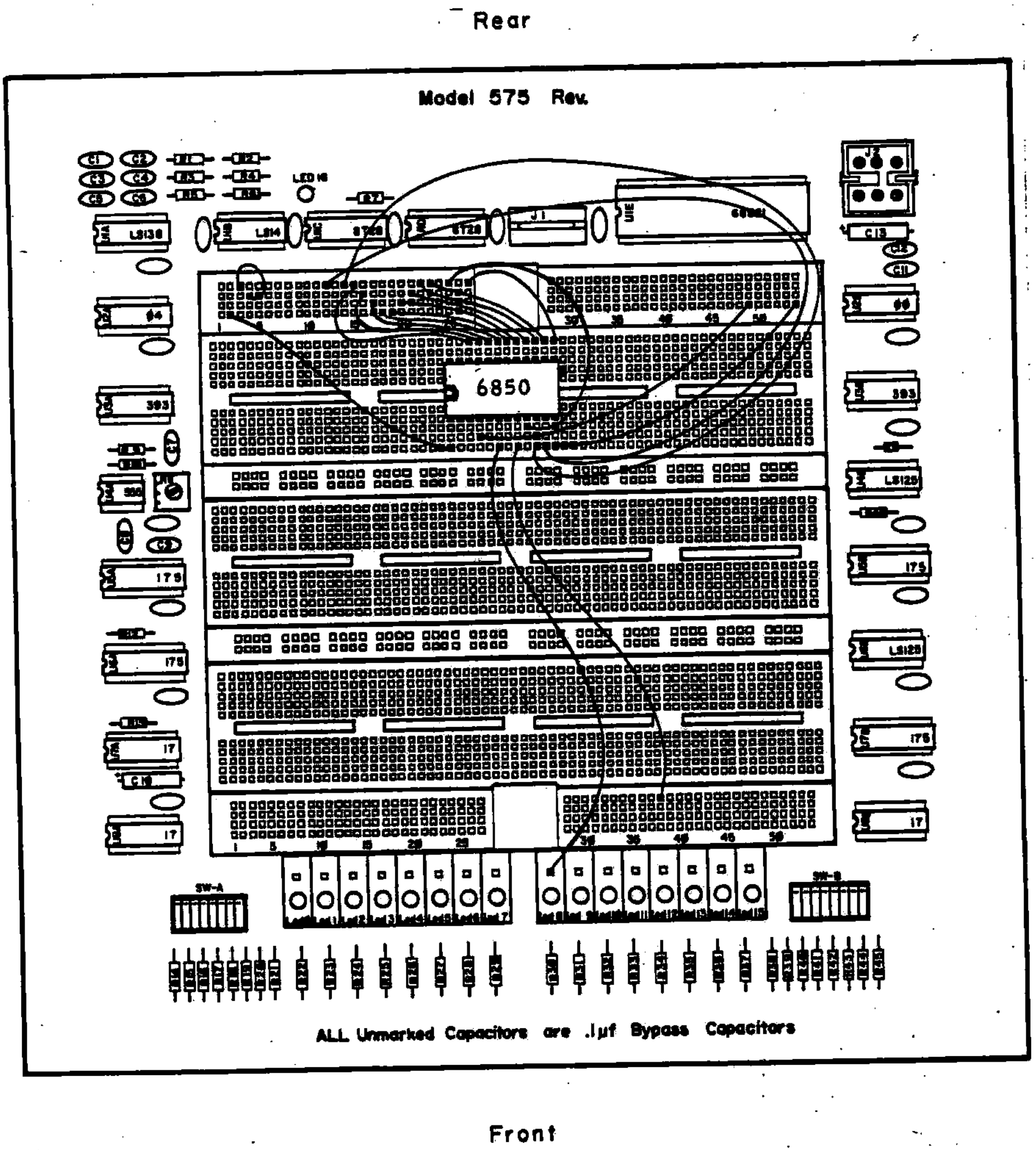
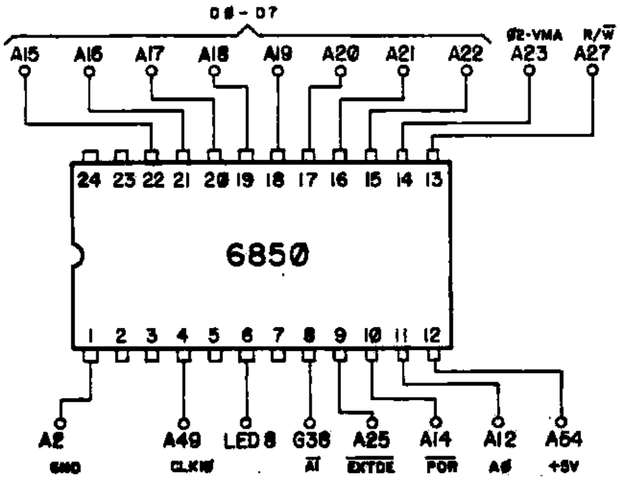


Figure 47
Connections for ACIA Serial Output to LED8
(Approximate Baud Rate of 1)



Use a short jumper wire to connect A3 to A4. This connects the clock OUT to the divide chain. Next use eight jumper wires to connect the eight data bus lines A15 through A22 to pins 22 through 15 (note the reverse order of bit sequence numbering). Two more short jumper wires will connect terminal A23 to pin 14 and terminal A27 to pin 13. Now, six medium jumper wires can be used for the connection of pins 1, 4, 9, 1Ø, 11, and 12. Pin 1 connects to A2, 4 to A49, 9 to A25, 1Ø to A14, 11 to A12, and 12 to A54. Finally, two long jumper wires can be used to connect pin 6 to LED8 and pin 8 to G38.

When you are certain that all of the connections have been made correctly, turn on the power supply to the CA-24 board. If you have not already done so, boot up the computer, UNLOCK, and enter NEW.

You are now ready to begin the parallel to serial conversion. A number POKEd into the Data Register will be sent out to LED 8 one bit at a time. The Terminal A49 on the clock divide channel (CLK1Ø) has been chosen so that when the divide-by-64 option in the Control Register is selected and frequency adjust resistor (R11) is turned full counterclockwise, the bits will be sent out at approximately one per second (the exact time sequence will vary from board to board). Be sure to set R11 full counterclockwise before proceeding.

Initialize the ACIA by using the command

```
POKE 5Ø956,3
```

Reference to the Control Register summary given earlier will show that the master reset has been entered.

Next, enter

```
POKE 5Ø956,22
```

Reference to the Control Register summary will show that the sequence of 8 bits + stop bit together with the divide-by-64 option, has been selected.

Now enter

POKE 5Ø957,21

and watch LED8. Approximately one second after pressing RETURN, it should turn off for about one second. Then, a sequence of one second on, off, on, off, on followed by a three second off period should occur. Then LED8 will come back on. Now try

POKE 5Ø957, 85

and note that instead of the three second off period, the one second on-off sequence will continue. These two time sequences together with several others are shown in Fig. 48.

At this point it will be helpful to enter a short program which will allow you to investigate various output formats and bit patterns by simply entering numbers. Be certain that NEW has been entered, then type in the program shown below.

```
10 REM ACIA INITIALIZATION AND USE
20 POKE 50956,3:REM MASTER RESET
30 INPUT"ENTER CONTROL REGISTER SELECTION";S
40 POKE 50956,S
50 INPUT"ENTER INTEGER 0 - 255";N
60 POKE 50957,N
70 GOTO 30
```

You now can use this program to observe the various combinations of serial output shown in Fig. 48. RUN the program, enter one of the values shown in the column headed Control Number (which will always select the divide-by-64 together with one of the serial format options), then enter either 21 or 85 and watch for the serial pattern shown. You will find that you can observe the condition of one or two stop bits by entering a number, then, shortly after the serial output has begun, entering the number again. The Data Register will hold the second entry and send it out immediately after the first.

When you have observed all of the serial output formats shown in Fig. 48, you should be ready to set up the circuitry which will allow the ACIA to receive serial information. The schematic diagram shown below and the wiring diagram in Fig. 49 show what is required. As you will note, most of the connections shown on the diagrams are already in place. The additional connections are described in the next paragraph.

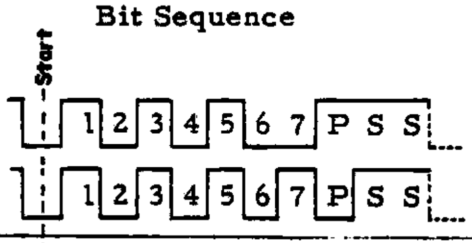
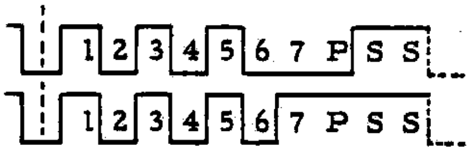
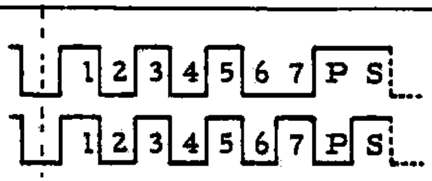
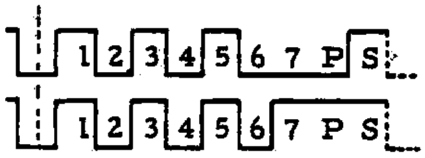
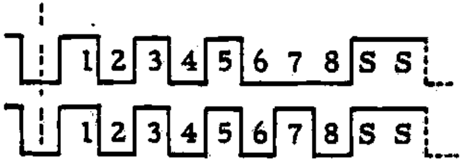
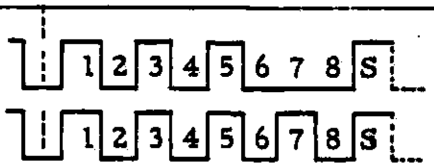
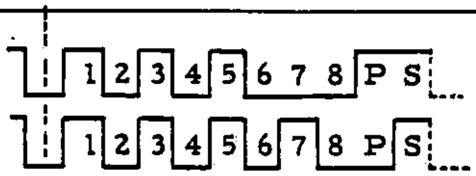
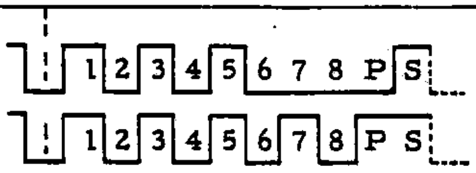
| Control Number | Function | Bit Sequence | Data Number |
|---|---|---|---|
| 2 | ÷ 64<br>7 Bits<br>Even Parity<br>2 Stop Bits |  | 21<br>85 |
| 6 | ÷ 64<br>7 Bits<br>Odd Parity<br>2 Stop Bits |  | 21<br>85 |
| 10 | ÷ 64<br>7 Bits<br>Even Parity<br>1 Stop Bit |  | 21<br>85 |
| 14 | ÷ 64<br>7 Bits<br>Odd Parity<br>1 Stop Bit |  | 21<br>85 |
| 18 | ÷ 64<br>8 Bits<br>No Parity<br>2 Stop Bits |  | 21<br>85 |
| 22 | ÷ 64<br>8 Bits<br>No Parity<br>1 Stop Bit |  | 21<br>85 |
| 26 | ÷ 64<br>8 Bits<br>Even Parity<br>1 Stop Bit |  | 21<br>85 |
| 30 | ÷ 64<br>8 Bits<br>Odd Parity<br>1 Stop Bit |  | 21<br>85 |

Figure 48
Serial Patterns (Bit Sequences) for
Various Control Register Settings and Data Numbers

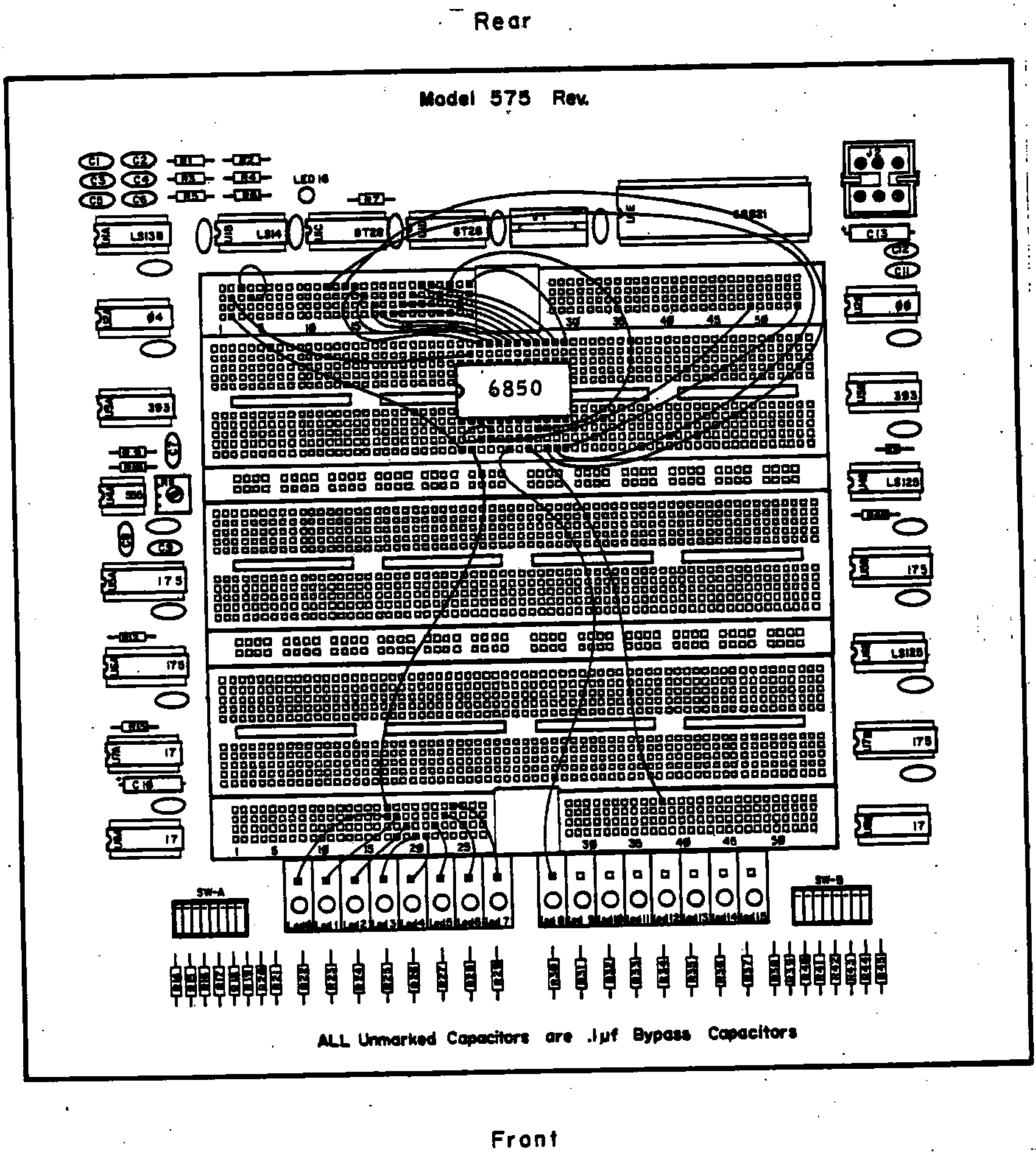
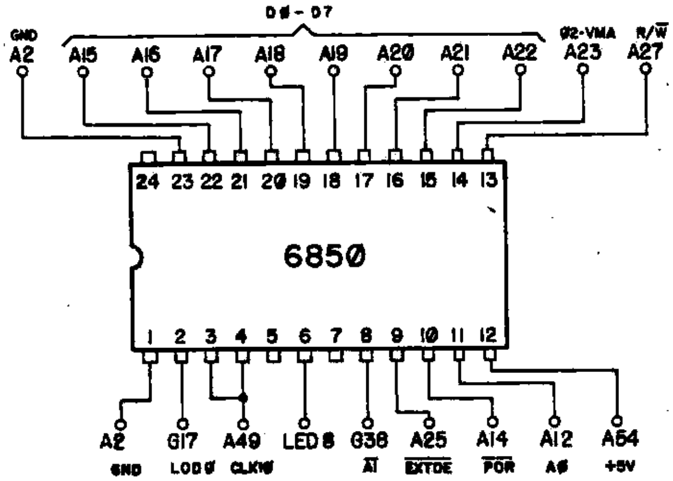


Figure 49
Connections for Serial Input to the ACIA
(Approximate Baud Rate of 1)



Turn off the power supply to the CA-24 board. Use a medium jumper wire to connect A2 to pin 23. This places a low on the DCD input. The ACIA will not receive serial data unless pin 23 is low. Next, use a long jumper wire to connect pin 2 to G17 and then eight short jumper wires to connect G17 through G24 to LEDØ through LED7. Finally, use a short jumper wire to connect pin 3 to pin 4.

Serial information for the ACIA to receive will be generated with software. The program has been set up to send one bit per second. Consequently, the clock signal being sent to the ACIA must be adjusted so that the one bit per second rate will be processed properly. This can be done by adjusting the clock frequency so that the ACIA gives a serial output of exactly one bit per second. The following BASIC program will help you do this.

```
10 REM ACIA-BASIC INTERACTIVE OUTPUT
20 POKE 50956,3:REM MASTER RESET
30 INPUT"ENTER CONTROL REGISTER SELECTION";S
40 POKE 50956,S
50 INPUT"ENTER INTEGER 0 - 255";N
60 POKE 50957,N
70 PRINT"NUMBER HAS BEEN LOADED"
80 T = PEEK(50956) AND 2:REM LOOK AT STATUS REGISTER TDRE
90 IF T <> 2 THEN 80:REM KEEP LOOKING UNTIL TDR IS EMPTY
100 GOTO 60
```

When the program is RUN, entering 22 for the Control Register selection and 85 for the integer will produce a sequential bit pattern of high-low to be sent to LED8 until CTRL-C is used to stop the program. This occurs because, in line 8Ø, the Status Register is read and tested for the condition of the TDRE bit. This tells the condition of the Transmit Data Register (TDR). When it is Empty (the E of TDRE), the bit goes high. As soon as this happens, the ACIA is ready to receive more parallel data, and the program sends it another 85.

Turn on the power supply to the CA-24 board, then enter the BASIC program shown above and RUN it, using 22 for the Control

Register selection and 85 for the integer. Time the on-off sequence of LED8 (preferably using a stop watch) and adjust R11 until the rate is within +/- 5% of one bit per second (i.e., LED8 is on 1 second, off 1 second, on 1 second, etc). On most CA-24 boards, full counterclockwise rotation of R11 will produce too rapid a rate, and turning the potentiometer shaft clockwise will slow the rate. If the rate is already too slow, move the pin 4 connection to A48 (from A49) and proceed with the R11 adjustment.

The serial bit sequence, which is received by the ACIA at pin 2, must include a start bit which is a transition from high to low. Then, after the data bit sequence, there must be whatever combination of parity and stop bits is specified by the bit pattern in the control register. For purposes of this experiment, we have chosen to work with the 8 bit + 1 stop bit (no parity) combination. With the divide-by-64 timing included, the decimal number which needs to be POKEd into 5Ø956 is 22.

The bit pattern will be generated by a BASIC program and sent to the CA-24 board by way of the latches (see Experiment 13). Except for the case of the start bit and stop bit, the time sequence of bits will be produced by shifting the pattern initially stored in memory location 5Ø952 (the address associated with latch controlled output and input). The shift operation will cause a sequence of high and low levels to appear at terminal G17, which has been connected to pin 2 of the ACIA. The wires which you have connected to LEDØ through LED7 are not needed for the data transfer process, but they do allow you to watch the shifting of the bit pattern as the BASIC program generates the sequence being received by the ACIA.

At this point, you should enter the BASIC program shown below.

```
10 REM LATCHED SERIAL OUTPUT TO ACIA (PIN 23 MUST BE GROUNDED)
20 POKE 50952,1:REM SET RECEIVE DATA LINE HIGH
30 POKE 50956,3:REM MASTER RESET
40 POKE 50956,22:REM 8 BITS + 1 STOP BIT, DIVIDE BY 64 TIMING
50 D = 1415:REM SET DELAY TIME FOR ONE BIT PER SECOND
60 INPUT"ENTER INTEGER 0 - 255";N
70 POKE 50952,0:REM PUT START BIT ON RECEIVE DATA LINE
80 FOR T = 1 TO D:NEXT T:REM HOLD FOR ONE SECOND
90 FOR B = 1 TO 8
100 POKE 50952,N:REM LOAD 8 BIT CHARACTER
110 FOR T = 1 TO D:NEXT T:REM HOLD FOR ONE SECOND
120 N = INT(N/2):REM SHIFT THE 8 BITS
130 NEXT B
140 POKE 50952,1:REM PUT STOP BIT ON RECEIVE DATA LINE
150 F = PEEK(50956) AND 1:REM LOOK AT STATUS REGISTER RDRF
160 IF F <> 1 THEN 150:REM KEEP LOOKING UNTIL RDR IS FULL
170 PRINT PEEK(50957)
180 PRINT PEEK(50956)
190 GOTO 60
```

Since the start bit in the sequence which will be received by the ACIA is a transition from high to low, the first step in the program is to set the serial data line high by POKEing 1 into 5Ø952. The line is then ready to receive the start bit. Next comes the master reset and then the setting up of the control register. The 1415 found in line 5Ø will produce a one bit per second sequence at G17 on a 2 MHz computer. After a number is entered as called for in line 6Ø, lines 7Ø through 13Ø will first generate a one second start bit, then place the number in address 5Ø952, shifting the bit pattern toward the least significant bit at the rate of one shift per second. When the shifting is completed and the most significant bit has been at G17 for one second, the stop bit is generated. Rather than timing the stop bit for one second, the status register is PEEKed to determine when the Receive Data Register is full (RDRF bit high). As soon as this occurs, the contents of the Receive Data Register is shown on the video display. In addition, the contents of the status register will be shown. If everything has gone correctly, the contents of the Receive Data Register will be the number entered at line 6Ø and the contents of the Status Register will be 2. If the clock has not been adjusted properly and is running too fast (R11 too far counterclockwise), the contents of the Status Register will be 18 indicating a framing error. (The contents of the Receive Data Register may or may not be the number entered.) If the clock is running too slow (R11 too far clockwise), the contents of the Status Register will be 2, but the contents of the Receive Data Register will be too large when small numbers are entered.

When you are certain that the program has been entered correctly, RUN it. Enter a small number (less than 128). If, after about 1Ø seconds, the number entered and then 2 are displayed on the video monitor, everything is functioning correctly. If an incorrect display (as described earlier) occurs, appropriately adjust R11 until the correct display is obtained.

In the latter portion of this experiment, only one serial bit pattern has been used to illustrate the serial reception function of the ACIA. You may wish to modify the BASIC program to produce patterns which include a parity bit. The parity error function of the status register could then be investigated.

You will note, furthermore, that this experiment has illustrated the use of only one of two possible ACIA's. By changing the connection to pin 8 from G38 to A13 (see page 129), the ACIA could be used in conjunction with addresses 5Ø958 and 5Ø959.

# EXPERIMENT 22

## TITLE

Analog to Digital Conversion (ADC)

## PURPOSE

The purpose of this experiment is to illustrate one of several ways in which your CA-24 board together with your Ohio Scientific computer can be used to sense a voltage level and display the value of that level on the video monitor.

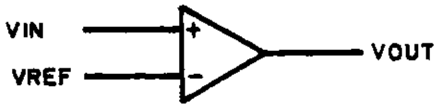
## EQUIPMENT

CA-24 board with power supply
OSI computer (C4PMF, C4PDF, C8PDF)
One 339A IC voltage comparator
Five 1kΩ resistors
One 5kΩ potentiometer
Eight medium jumper wires
Fourteen short jumper wires

## DISCUSSION

The circuits used in all of the previous experiments involved only two voltage levels, +5 Volts and Ø Volts. It is very common, however, for electrical circuits to involve a continuum of voltage levels. If a computer (or some other digital device) is to be used in conjunction with such circuits, then the continuous (analog) voltages must be converted into discrete level (digital) representation. Fortunately, integrated circuits are available which can perform this conversion. Their costs, however, directly relate to the resolution required (i.e., how small an increment of voltage change will give a different digital value) and the speed with which conversions are performed (i.e., how often the varying voltage is sampled). A sixteen channel high speed analog to digital converter is available from Ohio Scientific on the CA-22 analog I/O board.

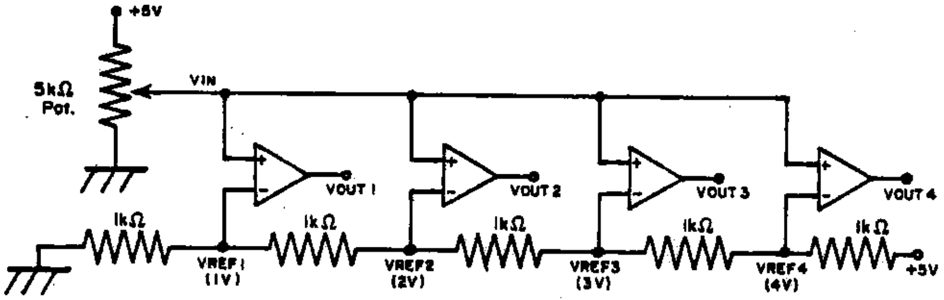
This experiment will illustrate a very low cost (but also very low resolution and relatively slow) method of sensing changes in a varying voltage. For this purpose, four voltage comparators will be used. All four are in the 339A IC chip. Each comparator can be represented schematically as shown below.



COMPARATOR

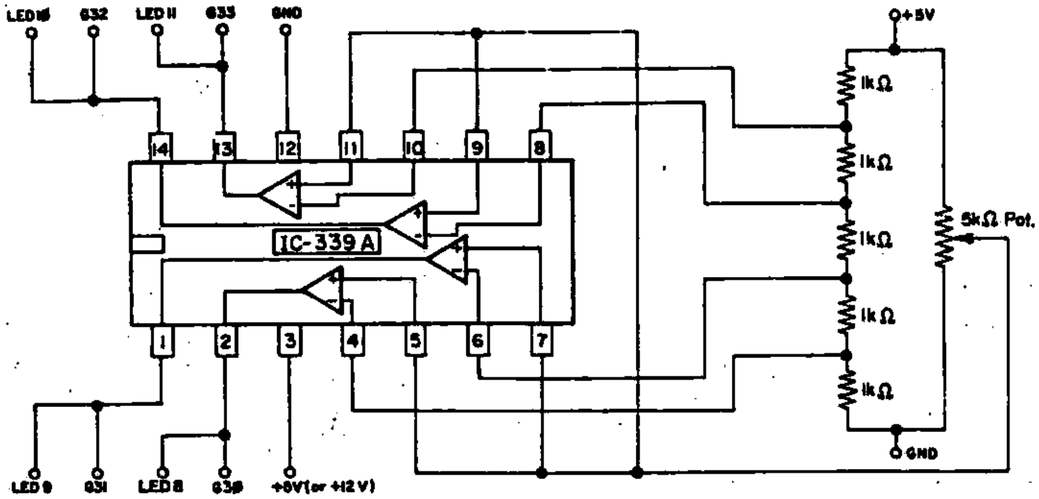
A reference voltage level (VREF) is placed on the inverting (-) input. The voltage to be tested (VIN) is placed on the other input line.

For this experiment, we will use a resistor chain to set the four reference voltages of 1 V, 2 V, 3 V, and 4 V. This is done very simply by connecting five 1kΩ resistors in series, placing the entire chain between +5 V and ground, then picking off the voltage at each junction between two resistors. Each voltage will be connected to the inverting input of a comparator in the 339A IC chip. The schematic diagram for doing this is shown below.



Note that the variable input voltage which is to be tested is generated by the 5kΩ potentiometer connected between +5 V and ground. The voltage level at the wiper contact is sent to the noninverting inputs of all of the comparators.

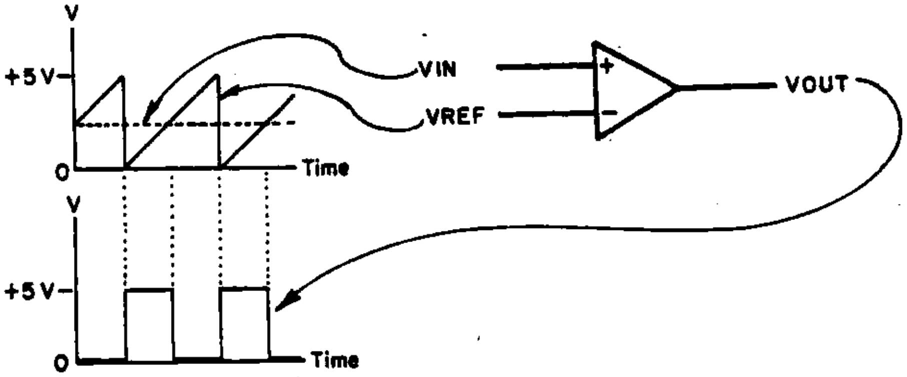
The schematic diagram showing chip pin and CA-24 board terminal connections is shown below.



Note that the power to the chip (pin 3) is specified as either +5 V or +12 V. The comparators in the 339A chip will not function properly if the reference voltage is too close to the voltage for powering the chip. When a 5 Volt power voltage is used, the reference voltage cannot be above 4 Volts. VREF4, then, is on the borderline unless a higher power voltage is used. If your power supply is delivering +12 V (or +15 V) to the board (pin A52) then this higher voltage should be used.

The resolution for the above circuit is only one Volt. However, the components of the circuit cost very little and, as you will see, the BASIC program needed is simple.

An alternate technique for analog to digital conversion can provide considerably better resolution. However, more circuit components are needed and the software is more involved. This technique uses a linear ramp voltage as the reference signal, VREF, instead of a constant voltage. At the beginning of the ramp, the comparator output will be high (assuming that VIN is a positive non-zero value). As soon as the voltage value of the ramp rises above the VIN value, the comparator output will go low. The length of time the output stays high then, is a measure of the VIN value. The diagram below illustrates the timing relationship between the various signals.



Measuring how long VOUT stays high can be accomplished by using VOUT to gate a clock signal such as that from the CA-24 board clock. The output of the gate, then, could be sent to the computer by way of the CA-24 board latches (e.g., G3Ø) and a counting program written to sum the sequential highs which are PEEKED in address 5Ø952. (Latch controlled input using address 5Ø952 is explained in Experiment 14.) If the ramp period for such an arrangement is Ø.1 sec (sawtooth signal of 1Ø Hz) and the clock is set at 5Ø kHz, then a maximum of 5,ØØØ counts could be recorded when VIN is 5 Volts. This means we would have a 1 millivolt resolution with a sample taken every Ø.1 sec.

Alternatively, VOUT could be connected directly to G3Ø or to one of the PIA port A lines and a machine language timing program written as described in Experiment 2Ø. This alternative timing method would allow for much faster sampling times or much better resolution.

## PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. Remove all previous components and wiring from the breadboard area. Insert the 339A IC voltage comparator chip as shown in Fig. 5Ø.

Before you forget (or overlook it) use a short jumper wire to connect G28 to G29. (This is needed for latch controlled input.) Now use four medium jumper wires to connect pin 1 to G31, pin 2 to G30, pin 13 to G33 and pin 14 to G32 as shown in Fig. 50 (and on the wiring schematic diagram). Then use four short jumper wires to connect G30 through G33 to LED8 through LED 11.

Next, use two short jumper wires to connect pin 3 to the rear line (which will be either +5 V or +12 V) of the front power bus (strip E) and pin 12 to the front line (which will be GND) of the rear power bus (strip C). Now two short jumper wires can be used to connect pin 5 to pin 7 and pin 9 to pin 11, and two more to bring the 5-7 and 9-11 pairs out to a conveniently common point on the breadboard (far right terminal set of strip D as shown in Fig. 50).

You are now ready to connect the resistor chain. Place one 1kΩ resistor (brown - black - red - gold) from GND on strip E to pin 4, a second from pin 4 to pin 6, a third from pin 6 to pin 8, a fourth from pin 8 to pin 10, and the fifth from pin 10 to +5 V on strip C (the rear line). Be sure that the exposed leads of the resistors do not touch each other.

Position the 5 kΩ potentiometer on strip B (as shown in Fig. 50) by inserting its terminal wires into a horizontal set of holes such that the terminal wires are to the front of the board. Use two short jumper wires to connect +5 Volts to the left terminal and GND to the right terminal. This connection will result in increasing voltage when the potentiometer shaft is rotated clockwise. Use a short jumper wire to connect the middle potentiometer terminal to the breadboard terminal set common to pins 5, 7, 9 and 11.

Finally, connect the power terminals (A1 and A2) to strip C, the ground jumper between strip C and strip E, and either +5 V or the alternate voltage of +12 V or +15 V to the rear line of strip E. Note that Fig. 50 shows both the +5 V and the higher voltage wires. Be certain that you connect only one of them.

When you are certain that all connections have been made correctly, rotate the 5kΩ potentiometer shaft full counterclockwise and turn on the CA-24 board power supply. LED8 through LED11 should be off (all others will be on).

Rotate the potentiometer shaft clockwise and watch the LED's. At about 1/5 of full clockwise rotation, LED8 should come on. At about 2/5 rotation, LED9 will light, etc.

If you are using the 12 V or 15 V power to the 339A chip, LED11 is certain to turn on strongly as the potentiometer shaft is rotated full clockwise. If you are using the 5 V power, LED11 may turn on completely, partially, or not at all depending on the value of the chain resistor closest to the 5 Volt level relative to the other resistors. If LED11 is not turning on and if you have an ohmmeter, measure each of the five resistors, choose the largest one, and insert it between the +5 V on strip C and pin 10. If you do not have an ohmmeter, simply use trial

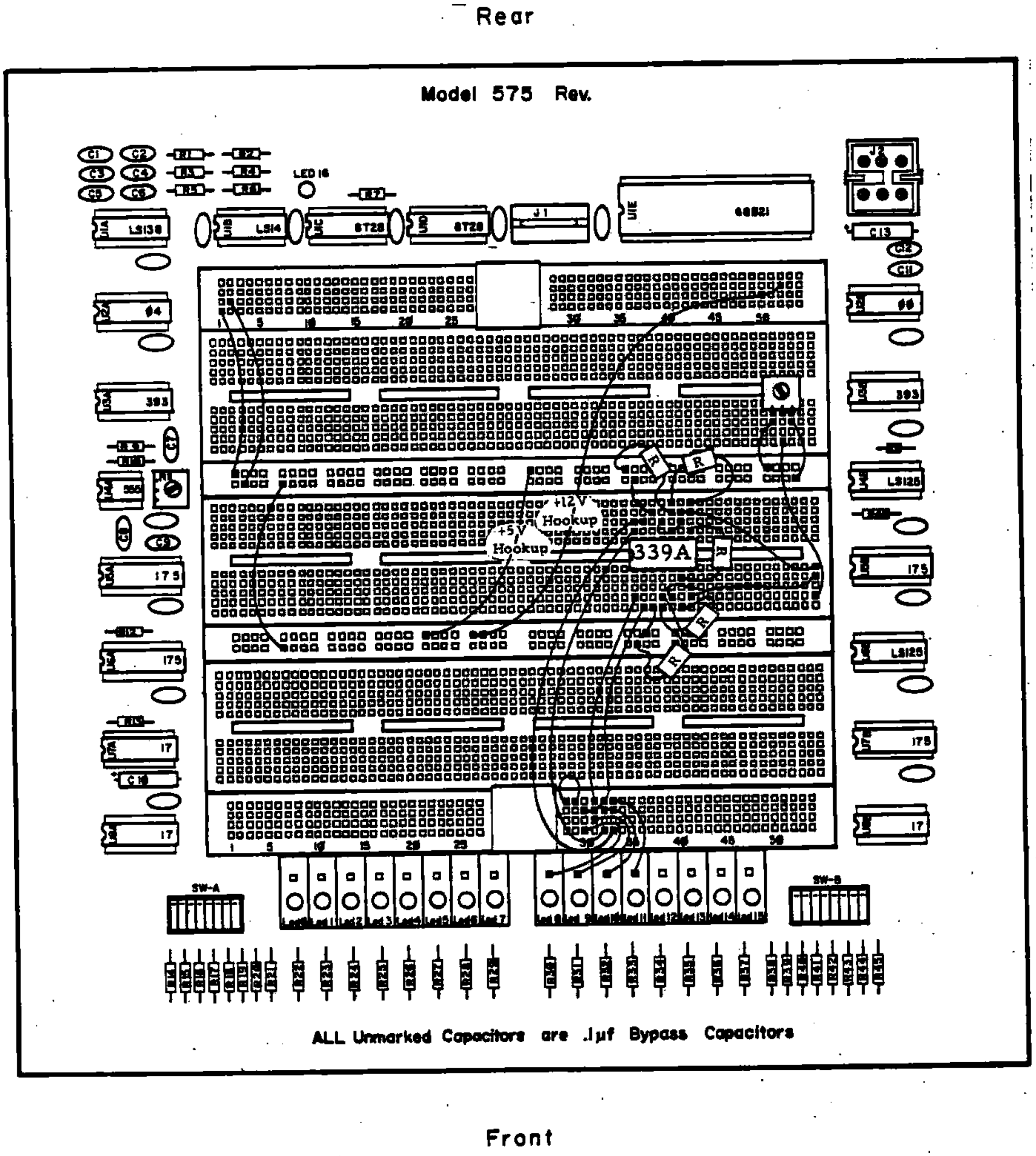


Figure 50
Connections for Constructing a Very Low Resolution Analog to Digital Converter

and error, replacing the last resistor in the chain with one of the first four. (Turn off the power supply before changing the resistors.)

Now boot up your computer, UNLOCK, and enter NEW. Type in the program shown below.

```
10 REM VOLTAGE LEVEL SORT PROGRAM (ADC)
20 VT = 20
30 V = PEEK(50952) AND 15
40 IF V = VT THEN 30
50 IF V = 0 THEN PRINT"VOLTAGE IS < 1 VOLT"
60 IF V = 1 THEN PRINT"VOLTAGE IS > 1 BUT < 2 VOLTS"
70 IF V = 3 THEN PRINT"VOLTAGE IS > 2 BUT < 3 VOLTS"
80 IF V = 7 THEN PRINT"VOLTAGE IS > 3 BUT < 4 VOLTS"
90 IF V = 15 THEN PRINT"VOLTAGE IS > 4 VOLTS"
100 VT = V
110 GOTO 30
```

When you are certain the program has been entered correctly, set the 5kΩ potentiometer to produce a voltage midway between two reference voltage values and RUN the program. Now rotate the potentiometer shaft slowly and note the screen display.

Note that there is a small range of ambiguity (which can be observed as a rapid screen display alternation if you rotate the potentiometer shaft very slowly) as VIN passes VREF. This range is only a few hundredths of a Volt and could be dealt with by way of software timing loops depending on the application of this type of circuit.

If Experiment 23 is to be worked through as the next use of the CA-24 board, the 339A IC chip should be left in its place on the breadboard.

## EXPERIMENT 23

### TITLE

Digital to Analog Conversion (DAC)

### PURPOSE

The purpose of this experiment is to illustrate the principle of digital to analog conversion through the construction of a simple circuit which will convert a bit pattern into a voltage level.

### EQUIPMENT

CA-24 board with power supply
OSI computer (C4PMF, C4PDF, C8PDF)
One 339A IC voltage comparator
Two 1kΩ resistors
Four 2kΩ resistors
One 5kΩ potentiometer
Six medium jumper wires
Six short jumper wires
One voltmeter

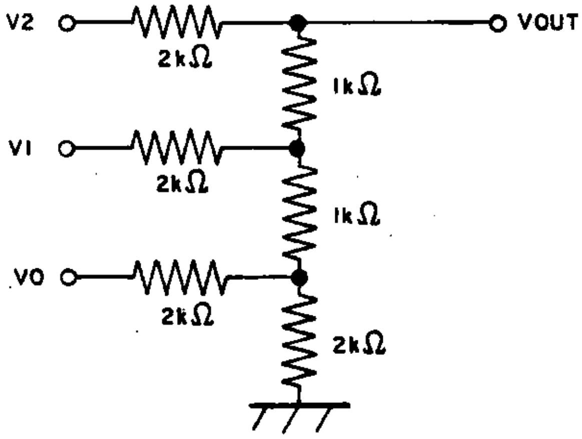
### DISCUSSION

In the previous experiment, techniques for turning an analog (continuous) voltage signal into digital representation were described. It is frequently necessary to turn a computer bit pattern into a voltage level. Such digital to analog conversion can be done with standard IC chips which are easily obtainable from electronics supply outlets. The price depends on the speed, accuracy, resolution, and range of the device. Two identical high speed digital to analog converters are available from Ohio Scientific on the CA-22 analog I/O board.

This experiment will illustrate a very common digital to analog conversion technique. Simplicity will be emphasized at the expense of accuracy, resolution, and range.

The combination of resistors shown below is called a ladder network. If this ladder network is to be used with the standard +5 V and Ø V logic levels, then each of VØ, V1, and V2 will be either +5 V or Ø V. Circuit analysis will show that the voltage appearing at VOUT can be calculated from the following equation.

$$VOUT = 5(V2/2 + V1/4 + VØ/8)$$



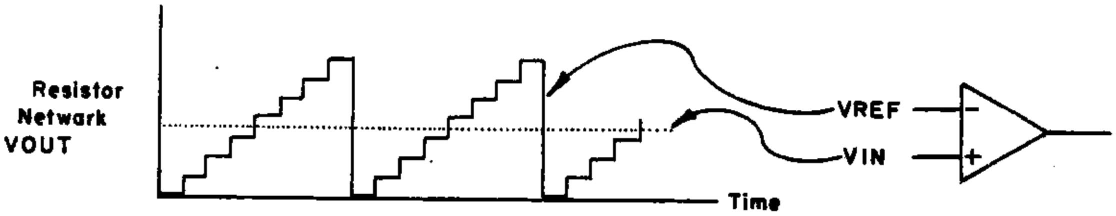
Clearly, if VØ, V1, and V2 all are Ø, then VOUT = ØV. If VØ is 5 V while the other two are Ø, then VOUT = .625 V. If V1 is 5 V while the others are Ø, VOUT = 1.25 V, etc. A bit pattern (in this case only three bits) at the input (the three 2kΩ resistors) will appear as a voltage level at VOUT.

The linearity of the ladder network (i.e., each voltage value at VOUT being exactly an integer multiple of the smallest non-zero value) depends on the precision of the resistors. The values do not have to be 1kΩ and 2kΩ. However, the smaller resistors must all be the same value, the larger resistors must all be the same value, and the larger resistor value must be exactly twice the smaller value. The resistors supplied with this manual are not precision resistors. Consequently, you will observe some deviation from linearity.

It is important, of course, that the conditions of the ladder network not be altered by circuits connected to it. Consequently, it is common practice to couple VOUT to recording or control circuits by way of a buffer amplifier. A single operational amplifier can be used for this purpose. The amplifier serves the dual purpose of isolation and voltage level adjustment if the 5 Volt range is too small or too large. For simplicity, a buffer amplifier will not be used in this experiment.

In addition, proper functioning of the ladder network depends on all logic "1" voltage levels at the input being the same value. If the signals VØ, V1, and V2 are taken from the TTL outputs of a latch or some other data transfer circuit, this condition will not be met. However, since this experiment is for the purpose only of illustrating the basic principle of digital to analog conversion, no effort will be made to compensate for the loading effects of the ladder network.

At the end of this experiment, we will combine DAC and ADC techniques. We will use a BASIC program to produce a seven step ramp which will be sent to VREF of the 339A comparator (see Experiment 22). A voltage to be measured will be connected to VIN of the comparator. A schematic diagram of this situation is shown below.



The BASIC program which produces the ramp will test the output of the comparator after each ramp step. When the ramp rises above the value of VIN, the program will print a message which states that the voltage is in a particular zone. Early in the PROCEDURE section of this experiment, you will have determined the voltage range of each zone.

PROCEDURE

Be certain that the power supply to the CA-24 board is turned off. If Experiment 22 has just been worked through, leave the 339A IC chip on the breadboard area, but remove all wires and other components.

Form the resistor ladder network by inserting four 2kΩ resistors (red - black - red - gold) and two 1kΩ resistors (brown - black - red - gold) into terminal strip F as shown in Fig. 51. Use two medium jumper wires to connect terminals A1 and A2 to power bus strip E (A1 to the rear line, A2 to the front line). Insert two short jumper wires such that a voltmeter can measure VOUT. In Fig. 51 these are shown connected to the front line of power strip E and to the rear left corner of the ladder network. Finally, use three medium jumper wires to connect each of the three front 2kΩ resistors to the power strip E. Fig. 51 shows the connection such that 5 V will go to the front right 2kΩ resistor (VØ on the schematic shown earlier) while Ø V goes to each of the other two (V1 and V2).

When you are certain that all connections have been made correctly, turn on the power supply. Note the reading on the voltmeter and record it in the last column (voltage step 1) of the table below.

| Output Voltage Step | Input Levels (Volts) | | | VOUT (Volts) |
|---|---|---|---|---|
| | V0 | V1 | V2 | |
| 0 | 0 | 0 | 0 | |
| 1 | 5 | 0 | 0 | |
| 2 | 0 | 5 | 0 | |
| 3 | 5 | 5 | 0 | |
| 4 | 0 | 0 | 5 | |
| 5 | 5 | 0 | 5 | |
| 6 | 0 | 5 | 5 | |
| 7 | 5 | 5 | 5 | |

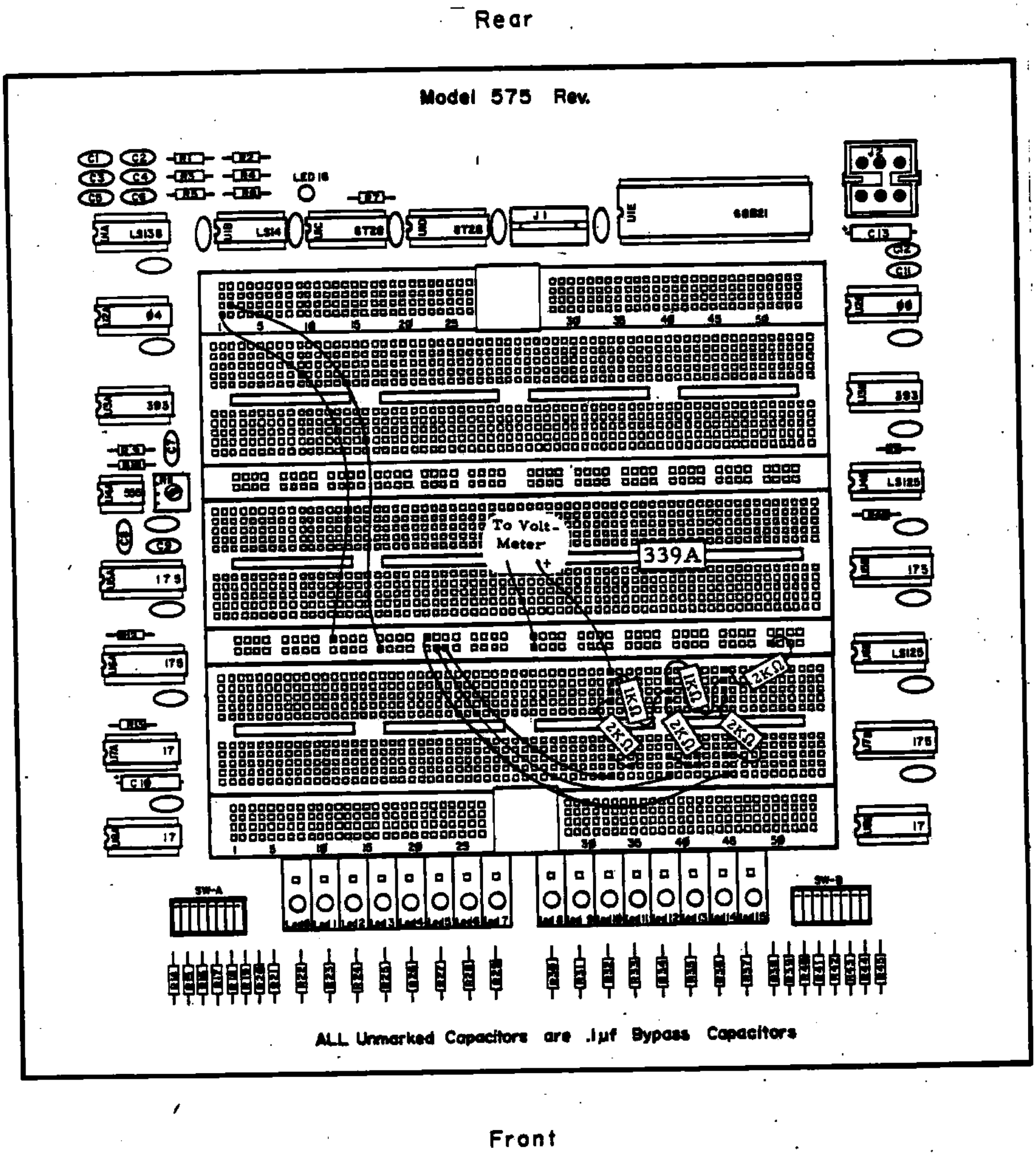


Figure 51
Connections for a Three Input Resistor Ladder Network

Change the wires on power strip E to produce the input voltage level combinations for each output voltage step in the table above. In each case, record the value of VOUT as measured with the voltmeter.

Note that each VOUT value is approximately an integral multiple of the step 1 value.

We are now ready to produce VØ, V1, and V2 with the computer. Connect the VØ signal wire to G17, the V1 wire to G18, and the V2 wire to G19 as shown in Fig. 52. Boot up your computer, enter UNLOCK, then enter NEW. It will be helpful to enter the short program shown below.

```
1Ø INPUT"ENTER NUMBER";N
2Ø POKE 5Ø952,N
3Ø GOTO 1Ø
```

Run the program and enter Ø. The voltmeter should read Ø. Enter 1. The meter should read a value somewhat less than the step 1 value found earlier. Record the value in the table below, second line, last column.

| Output Voltage Step | Value of N Used in Poke 50952, N | VOUT (Volts) |
|---|---|---|
| 0 | 0 | |
| 1 | 1 | |
| 2 | 2 | |
| 3 | 3 | |
| 4 | 4 | |
| 5 | 5 | |
| 6 | 6 | |
| 7 | 7 | |

Enter a new value for N and record the reading of the voltmeter on the appropriate line of the last column in the table above. Continue until the table has been filled in.

Note that the increasing values in the VOUT column are less close to integer multiples of the step 1 value.

Finally, we will set up a circuit which will send a computer generated ramp to VREF of the 339A comparator. The schematic diagram for the circuit is shown at the top of page 154.

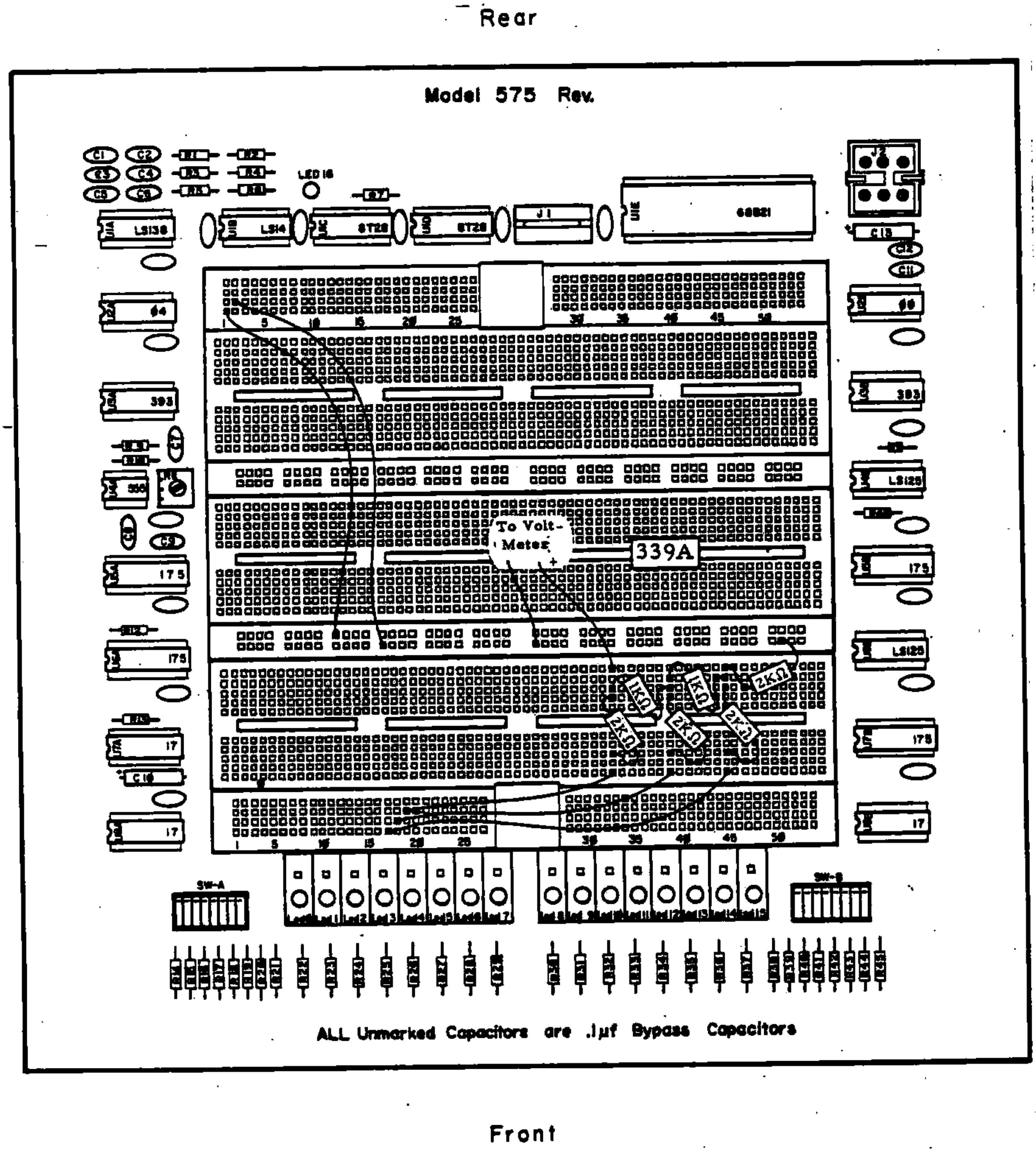
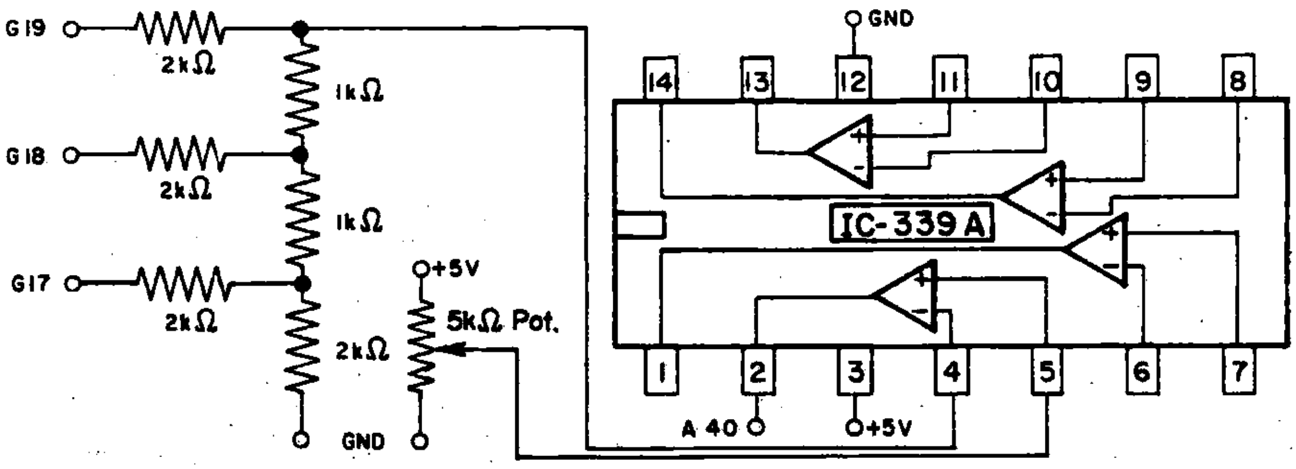


Figure 52
Connections for Latch Generated Levels for Input to the Resistor Ladder Network



In this circuit, the 5kΩ potentiometer produces the VIN voltage to be measured, and the output of the comparator is sent to port A of the PIA (PAØ at A4Ø).

Turn off the power supply to the CA-24 board, then disconnect the voltmeter from the short jumper wires. The GND wire which went to the voltmeter should be connected to pin 12 of the 339A chip, and the signal wire (VOUT for the resistor network) should be connected to pin 4. These connections are shown in Fig. 53. Use one additional jumper wire to connect +5 V to pin 3 of the chip. A medium jumper wire can be used to connect pin 2 to terminal A4Ø. Finally, position the 5 kΩ potentiometer on strip D as shown in Fig. 53. Use two short jumper wires to connect the left and right terminals to +5 V and Ø V (terminal strip E) respectively, and one short jumper wire to connect the middle potentiometer terminal to pin 5 of the 339A IC chip.

Turn on the power supply to the CA-24 board. Enter NEW, then type in the program shown below.

```
10 REM DAC RAMP VOLTAGE SORT PROGRAM
20 POKE 50949,0:POKE 50948,0:POKE 50949,4:REM INPUT, PIA PORT A
30 IT = 10
40 FOR I = 0 TO 7
50 POKE 50952,I:REM SET VREF RAMP LEVEL
60 T = PEEK(50948) AND 1:REM LOOK AT LEVEL AT PA0
70 IF T = 0 THEN 90:REM STOP LOOKING WHEN PA0 GOES LOW
80 NEXT I
90 IF IT = I THEN 40
100 PRINT"VOLTAGE IS IN ZONE";I
110 IT = I
120 GOTO 40
```

The FOR-NEXT loop produces the ramp. You will note, however, that the loop is exited (the ramp is terminated) as soon as the ramp rises above the voltage level produced by the potentiometer. Line 9Ø tests whether the potentiometer voltage has changed since the last test. No new message is displayed until the voltage level changes.

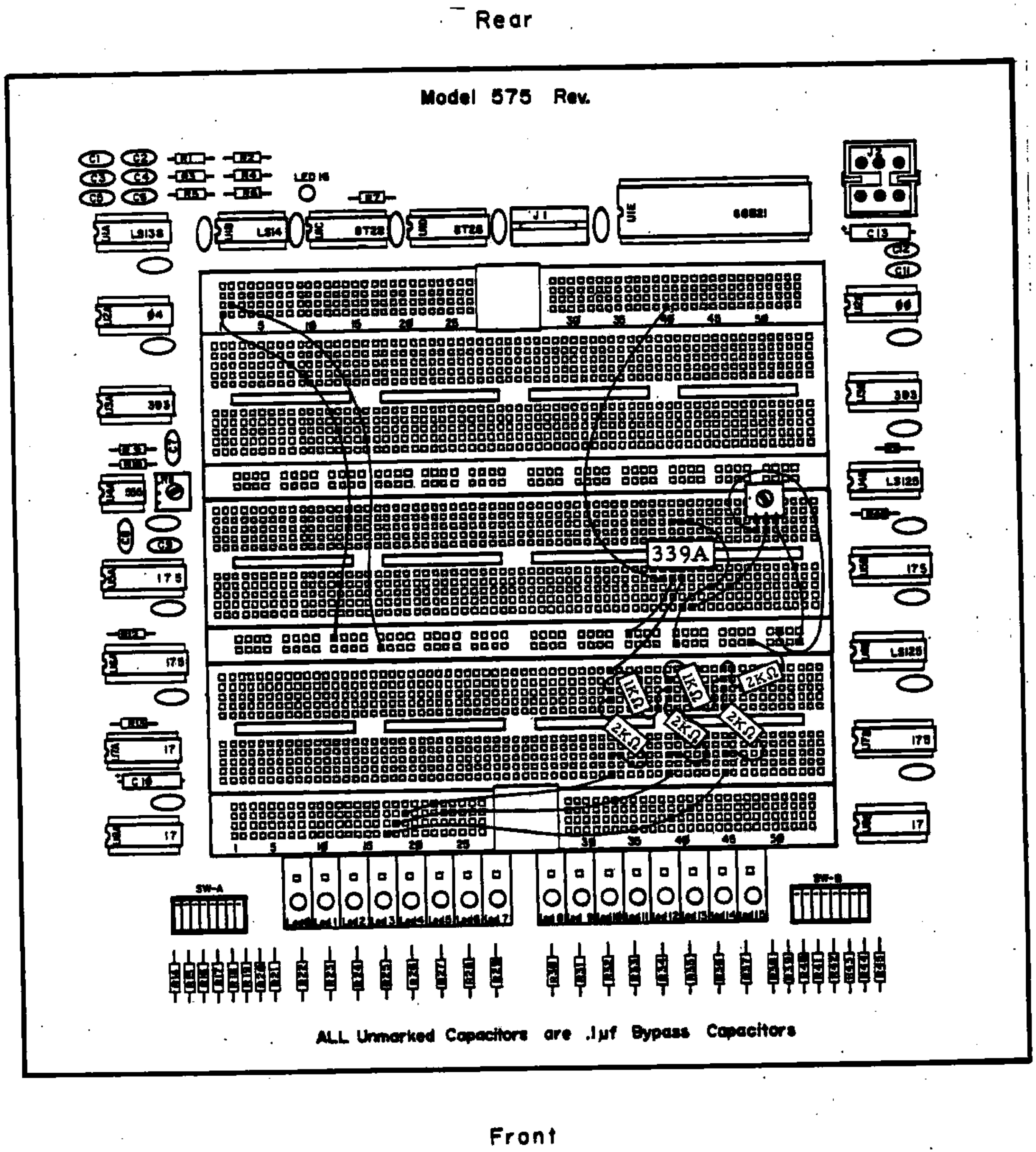


Figure 53
Connections for Combined Digital to Analog Conversion
and Analog to Digital Conversion

When you are certain the program has been entered correctly, set the potentiometer at about its mid-point and RUN the program. Slowly rotate the potentiometer counterclockwise and clockwise. Note that a "VOLTAGE IS IN ZONE 8" message is given for the upper (clockwise) portion of the potentiometer range. This, of course, is because the potentiometer range is from Ø to 5 Volts while the ramp range is from Ø to about 3 Volts.

Better resolution could be achieved by adding one or more segments to the resistor network, and a wider than 3 Volt range could be measured by using an operational amplifier with negative gain between the voltage to be sensed and pin 5 of the 339A.