



**Rockwell**

**DOCUMENT CHANGE NOTICE**

**DOCUMENT NAME:** KIM-1 USER'S MANUAL

**DOCUMENT NUMBER:** 29650N33

**REVISION:** 0

**NATURE OF CHANGE:** New KIM-1 boards have SST switch in upper left corner of keyboard, and switch direction is changed.

**ISSUE DATE:** October 1977

The pages listed below have been added, deleted or changed since this document was last printed. Please insert the attached material into your copy of the document, replacing the existing pages where applicable. SEE NOTE.

**Change Page**

**Comment**

Title page (rear)	Updates manual to Revision 1
7	Figure 2.1 changed to reflect new SST switch position
9	Description of slide switch position changed
29	Figure 3.6 changed
39	Interval timer reference note changed
46	Reference to SST switch ON position changed
69	Drawing of SST switch at top of Figure 5.6 has been changed
B-1	SST switch position changed

**NOTE TO READER**

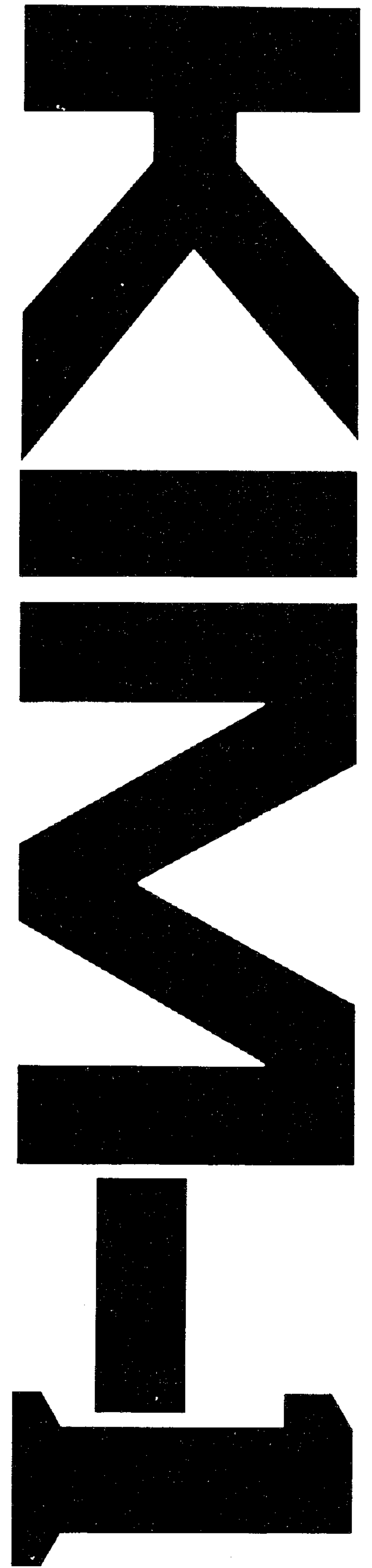
If your KIM-1 board has the SST switch in the upper right corner of the keyboard, DO NOT insert this material into your manual.

Rockwell Microelectronic Devices  
P.O. Box 3669  
Anaheim, CA 92803

# KIM-1

MICROCOMPUTER MODULE

## USER MANUAL



Rockwell International

\$5.00

© 1977 Rockwell International

OCTOBER, 1977  
DOCUMENT No. 29650N33  
REVISION -1-

MICROELECTRONIC DEVICES - REGIONAL SALES OFFICES

**WESTERN REGION, U.S.A.**

3310 Miraloma Avenue  
P.O. Box 3669  
Anaheim, Ca. 92803  
*Phone: (714) 632-3698*

**EASTERN REGION, U.S.A.**

Carolier Office Building  
850-870 U.S. Route 1  
North Brunswick, New Jersey 08902  
*Phone: (201) 246-3630*

**MIDWEST REGION, U.S.A.**

*Contact:*  
John G. Twist Company  
1301 Higgins Road  
Elk Grove Village, Illinois 60007  
*Phone: (312) 593-0200*

**CENTRAL REGION, U.S.A.**

2855 Coolidge Road, Suite 101  
Troy, Michigan 48064  
*Phone: (313) 435-1638*

**FAR EAST**

Rockwell International Overseas Corp.  
Ichiban-cho Central Building  
22-1 Ichiban-cho, Chiyoda-ku  
Tokyo 102, Japan  
*Phone: 265-8808*

**EUROPE**

Rockwell International GmbH  
Microelectronic Device Division  
Fraunhoferstrasse 11  
D-8033 Munchen-Martinsried  
Germany  
*Phone: (089) 859-9575*



**Rockwell International**

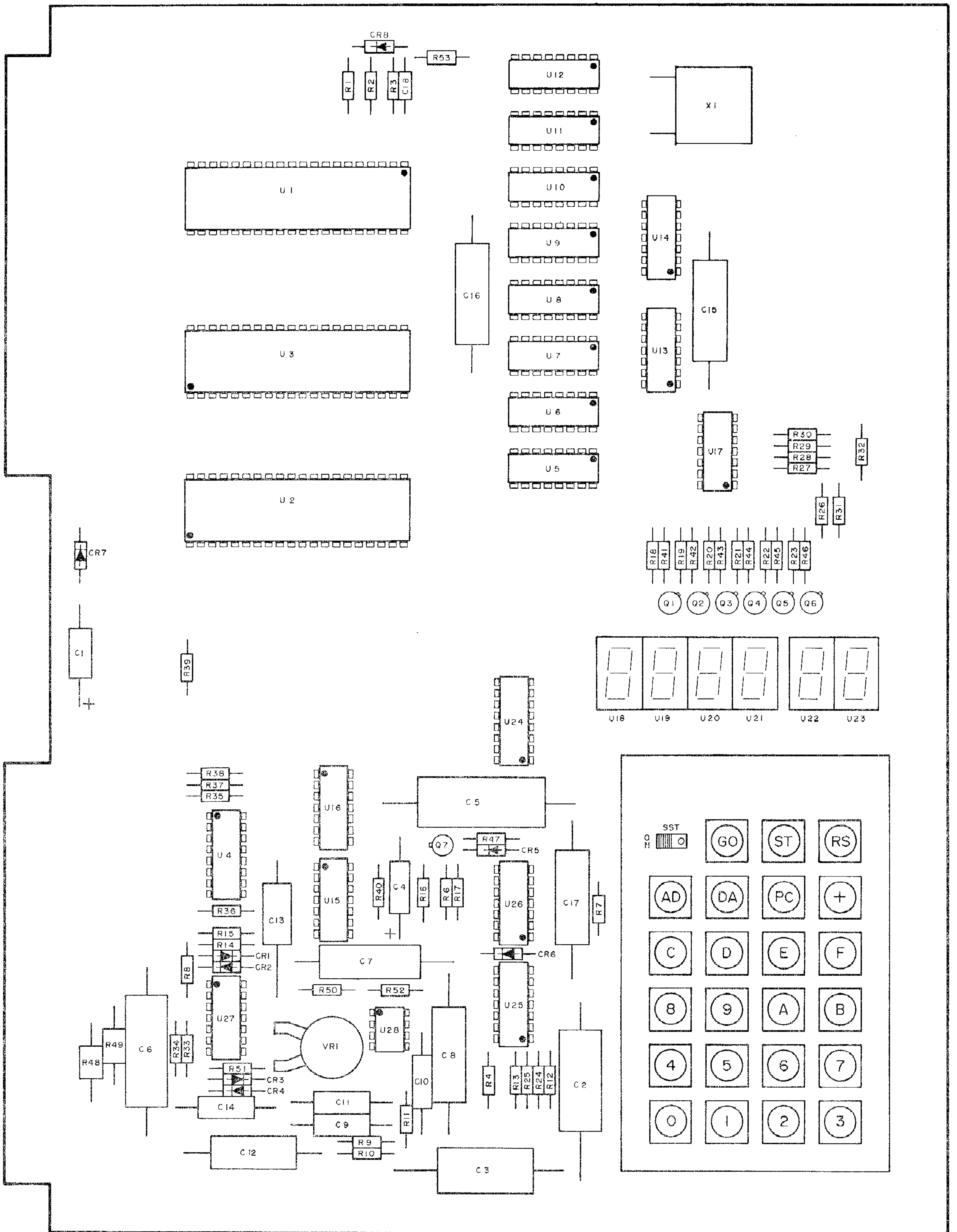
HOME OFFICE SALES/MARKETING (714) 632-3729

FOR ASSISTANCE, CALL OR WRITE THE OFFICE NEAREST YOU.

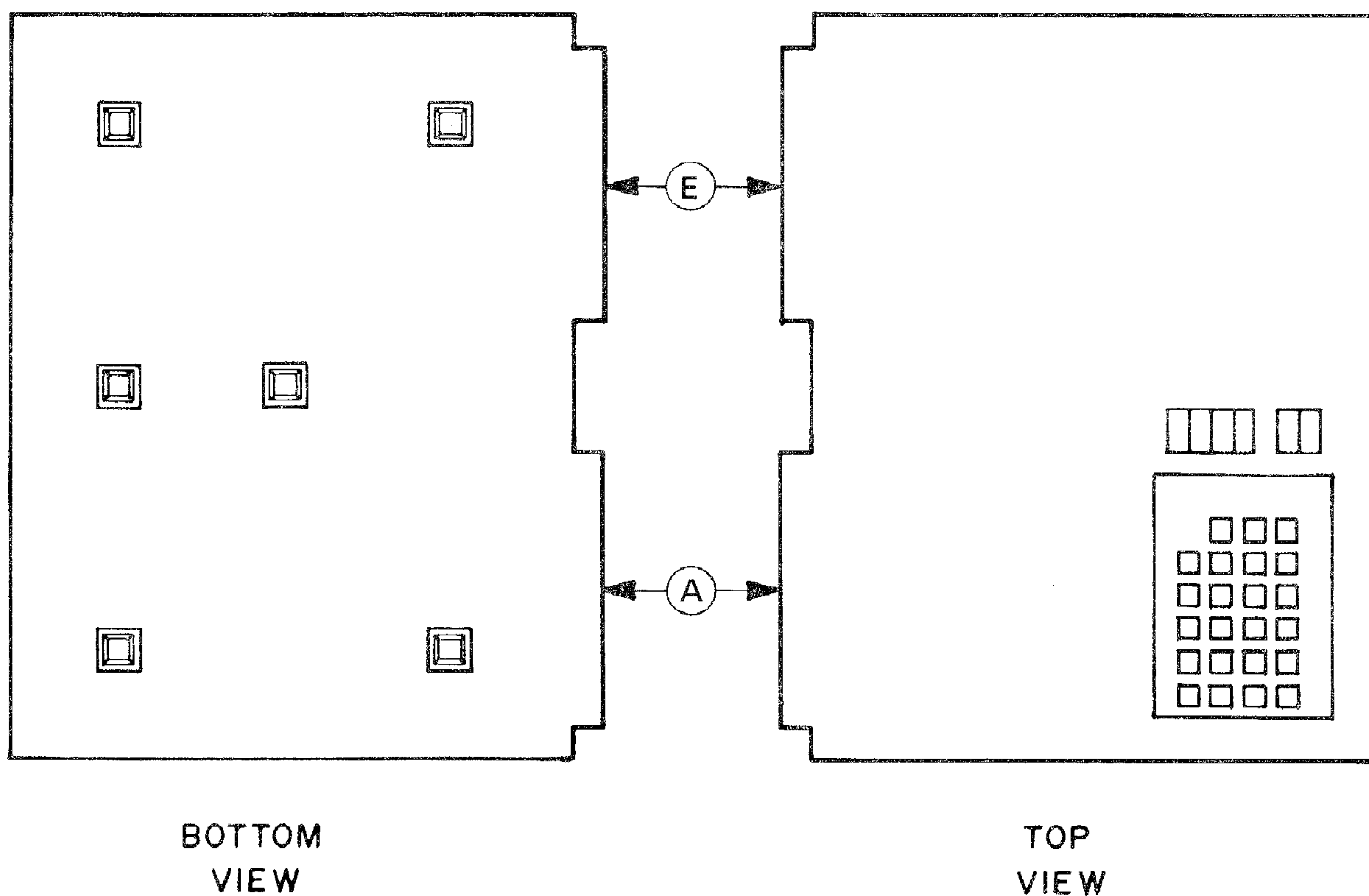
The information in this manual has been reviewed and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. The material in this manual is for informational purposes only and is subject to change without notice.

# APPENDIX B

## KIM-1 PARTS LAYOUT

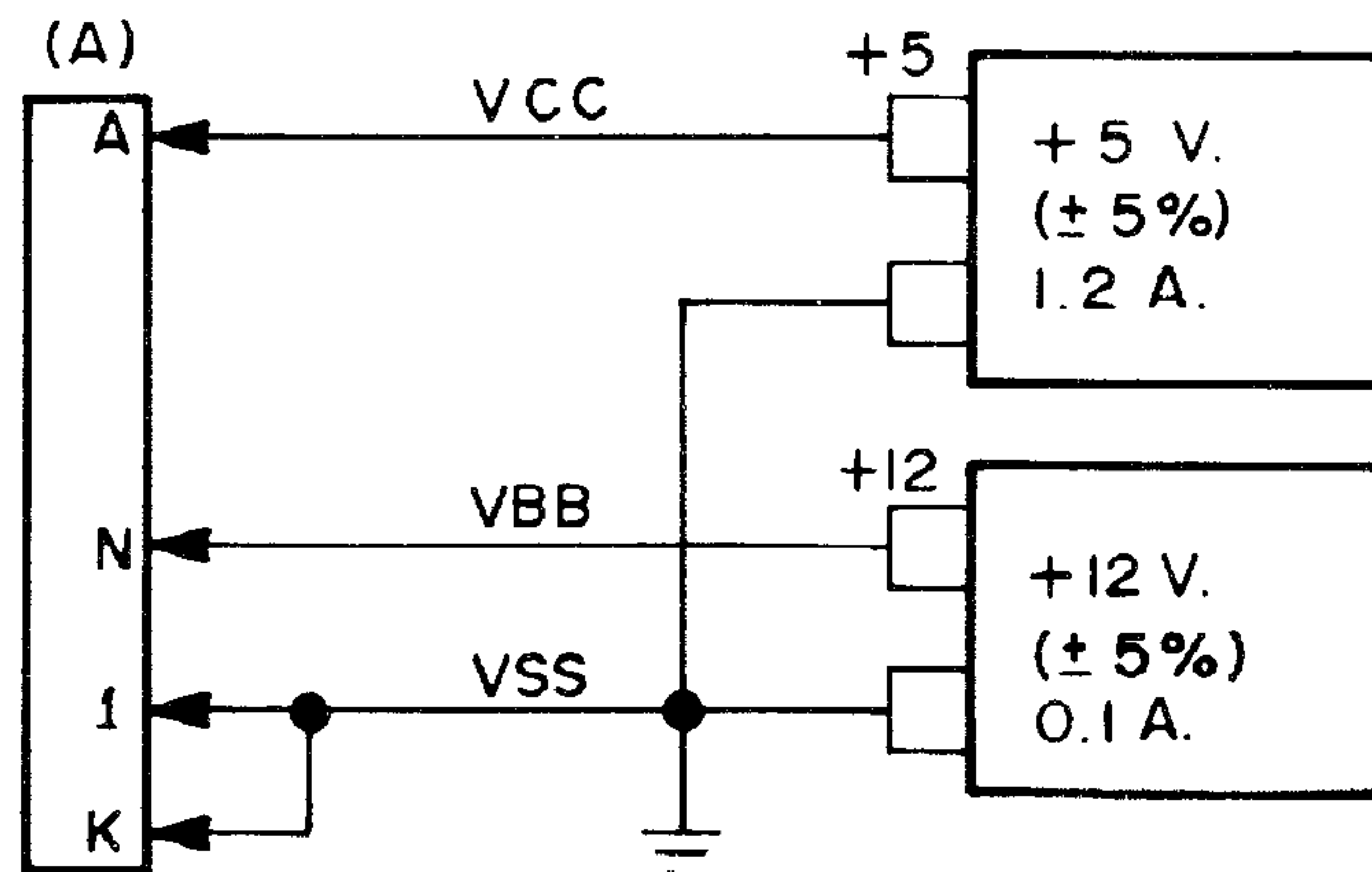


Place the module such that the keyboard is to your lower right and observe that two connector locations extend from the module to your left. The connector area on the lower left is referred to as the Application connector (A). You will note that a 44 pin board edge connector is already installed at this location. The connector area to the upper left is for use by you for future system expansion and is referred to as the Expansion connector (E).



*KIM-1 Module  
FIGURE 2.1*

Remove the (A) connector from the module and connect the pins as shown in the sketch.



*Power Supply Connections*  
*FIGURE 2.2*

Reinstall the (A) connector making certain that the orientation is correct.

- Note 1: The +12 volt power supply is required only if you will be using an audio cassette recorder in your system.
- Note 2: The jumper from pin A-K to Vss (Pin A-1) is essential for system operation. If you expand your system later, this jumper will be removed and we'll tell you what to do to pin A-K.
- Note 3: If you don't have the proper power supplies already available, you may wish to construct the low cost version shown with schematic and parts list in Appendix D. In any event, your power supply must be regulated to insure correct system operation and must be capable of supplying the required current levels indicated in the sketch.

Now, recheck your connections, turn on your power supplies, and depress  (reset). You should see the LED display digits light as your first check that the system is operational. If not, recheck your hookup or refer to Appendix C (In Case of Trouble).

#### 2.4 LETS TRY A SIMPLE PROGRAM

Assuming that you have completed successfully all of the steps thus far, a simple program now can be tried to demonstrate the operation of the system and increase your confidence that everything works properly. We'll be using only the keyboard and display on the module for this example. (In the next two sections we'll worry about the teleprinter and the audio cassette).

For our first example, we will add two 8 bit binary numbers together and display the result. We presume that you are familiar with the hexadecimal representation of numbers and the general rules for binary arithmetic.

First check and be sure that the slide switch in the upper left corner of the keyboard is pushed to the right (SST Mode is OFF). Now proceed with the following key sequence:

<u>Press Keys</u>	<u>See On Display</u>	<u>Step #</u>
<input type="button" value="AD"/>	xxxx xx	1
<input type="button" value="0"/> <input type="button" value="0"/> <input type="button" value="0"/> <input type="button" value="2"/>	0002 xx	2
<input type="button" value="DA"/>	0002 xx	3
<input type="button" value="1"/> <input type="button" value="8"/>	0002 18	4
<input type="button" value="+"/> <input type="button" value="A"/> <input type="button" value="5"/>	0003 A5	5
<input type="button" value="+"/> <input type="button" value="0"/> <input type="button" value="0"/>	0004 00	6
<input type="button" value="+"/> <input type="button" value="6"/> <input type="button" value="5"/>	0005 65	7
<input type="button" value="+"/> <input type="button" value="0"/> <input type="button" value="1"/>	0006 01	8
<input type="button" value="+"/> <input type="button" value="8"/> <input type="button" value="5"/>	0007 85	9
<input type="button" value="+"/> <input type="button" value="F"/> <input type="button" value="A"/>	0008 FA	10
<input type="button" value="+"/> <input type="button" value="A"/> <input type="button" value="9"/>	0009 A9	11
<input type="button" value="+"/> <input type="button" value="0"/> <input type="button" value="0"/>	000A 00	12
<input type="button" value="+"/> <input type="button" value="8"/> <input type="button" value="5"/>	000B 85	13
<input type="button" value="+"/> <input type="button" value="F"/> <input type="button" value="B"/>	000C FB	14
<input type="button" value="+"/> <input type="button" value="4"/> <input type="button" value="C"/>	000D 4C	15
<input type="button" value="+"/> <input type="button" value="4"/> <input type="button" value="F"/>	000E 4F	16
<input type="button" value="+"/> <input type="button" value="1"/> <input type="button" value="C"/>	000F 1C	17

What you have just done is entered a program and stored it in the RAM at locations 0002 through 000F. You should have noticed the purpose of several special keys on your keyboard:

- AD - selects the address entry mode
- DA - selects the data entry mode
- + - increments the address without changing the entry mode
- 0 TO  F - 16 entry keys defining the hex code for address or data entry

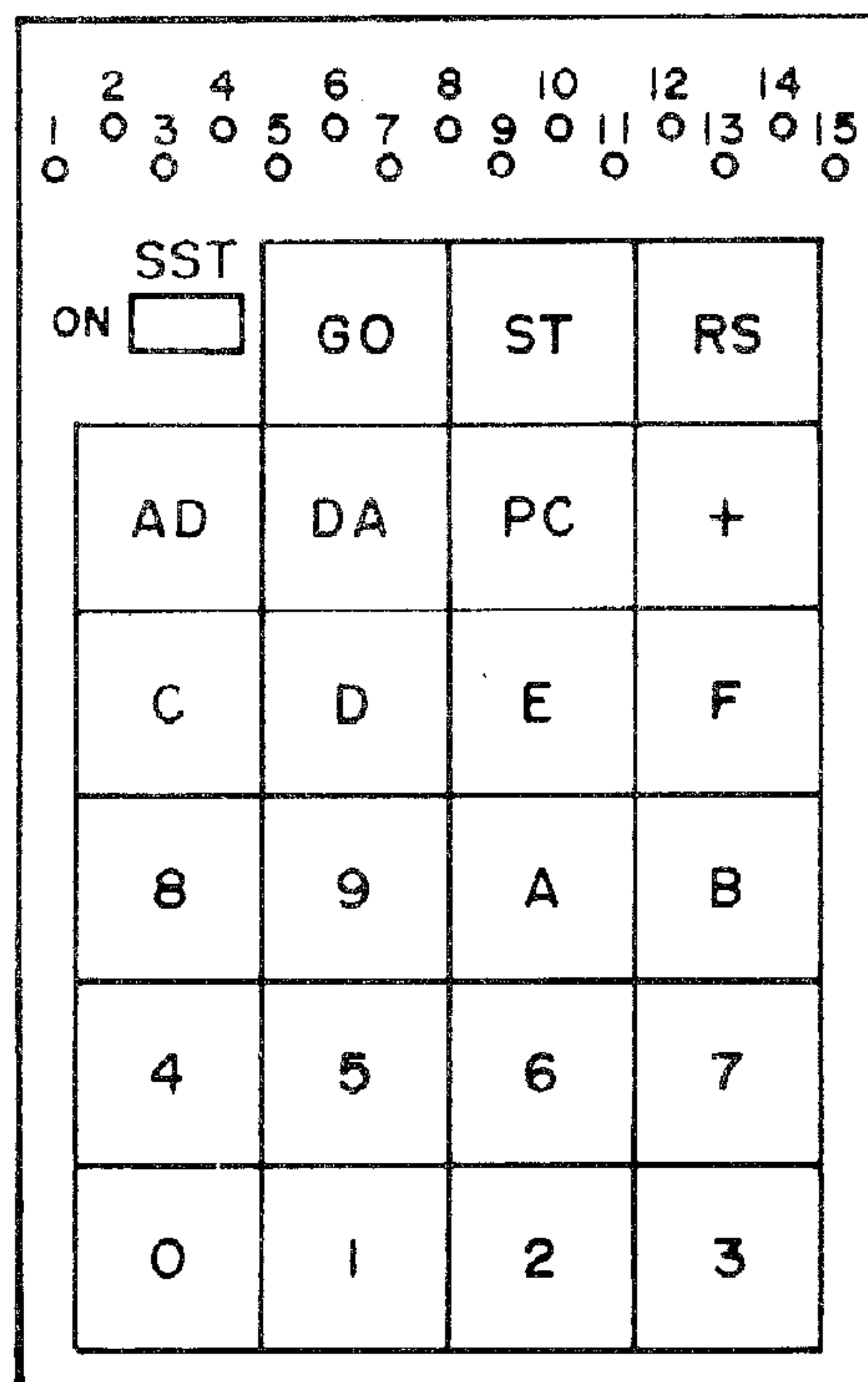
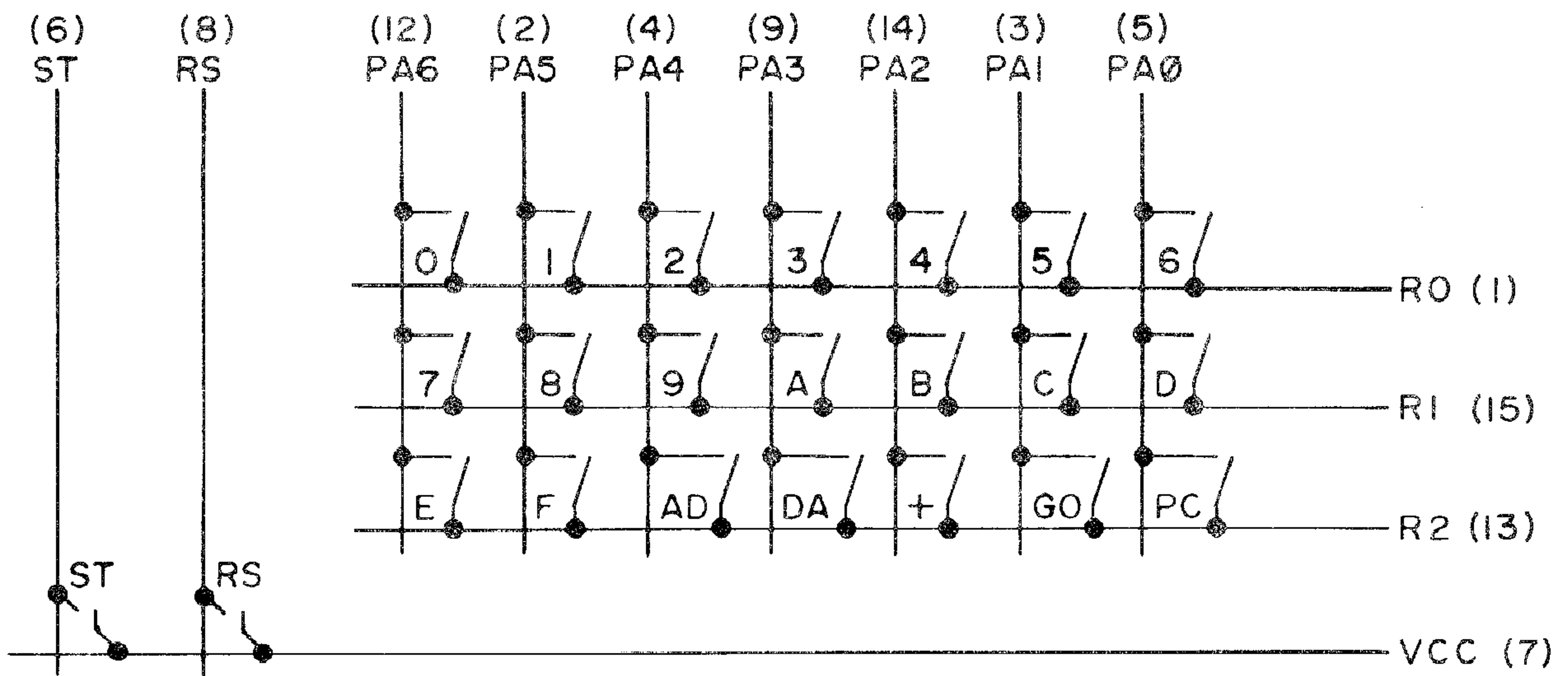
You've noticed as well that your display contains 6 digits. The four on the left are used to display the hex code for an address. The two on the right show the hex code for the data stored at the address shown. Therefore, when you pressed  AD (step 1) and  0  0  0  2 (step 2), you defined the address entry mode, selected the address 0002, and displayed the address 0002 in the four left-most display digits. Incidentally, when we show an "x" in the display chart, we mean that we don't know what will be displayed and we "don't care."

Next you pressed  DA (step 3) followed by  1  8 (step 4). Here, you have defined the data entry mode and entered the value 18 to be stored at your selected address 0002. Of course, the 18 then was displayed in the two right-most digits of your display.

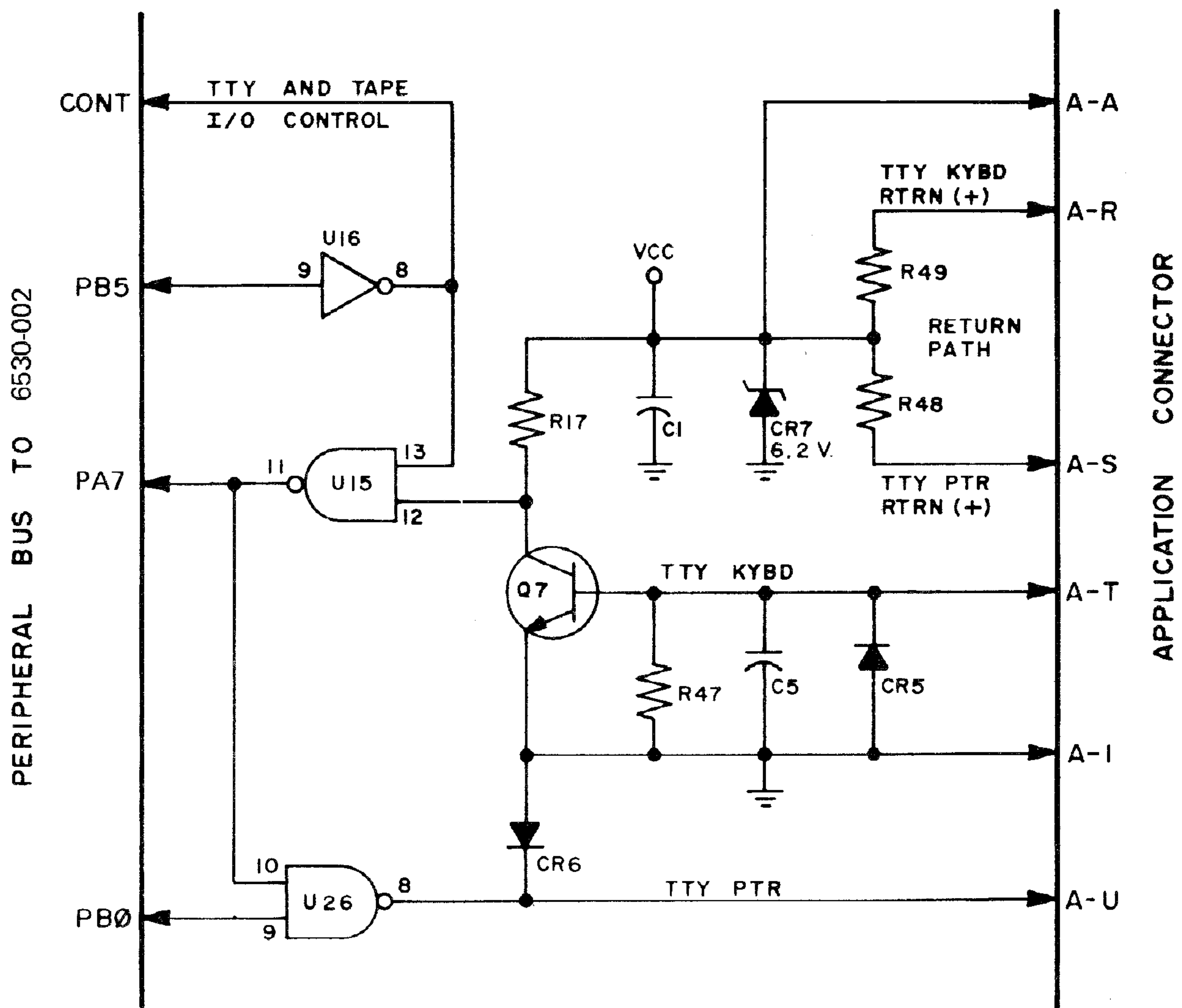
You remained in the data entry mode but began to press  + followed by a two digit number (steps 5 to 17). Note that each depression of the  + key caused the address displayed to increase by one. The hex keys following the  + key continued to enter the data field of the display. This procedure is merely a convenience when a number of successive address locations are to be filled.

If you made any mistakes in pressing the keys, you should have noticed that correcting an error is simply a matter of reentering the data until the correct numbers show on the display.





Keyboard Detail  
FIGURE 3.6



TTY Interface  
FIGURE 3.7

ADDRESS	AREA	LABEL	FUNCTION
00EF	↑ Machine Register Storage Buffer ↓	PCL	Program Counter - Low Order Byte
00F0		PCH	Program Counter - High Order Byte
00F1		P	Status Register
00F2		SP	Stack Pointer
00F3		A	Accumulator
00F4		Y	Y-Index Register
00F5		X	X-Index Register
1700	↑ Application I/O ↓	PAD	6530-003 A Data Register
1701		PADD	6530-003 A Data Direction Register
1702		PBD	6530-003 B Data Register
1703		PBDD	6530-003 B Data Direction Register
1704 ↓ 170F	↑ Interval Timer ↓		6530-003 Interval Timer (See page H-5 of Appendix H, or Section 1.6 of Hardware Manual)
17F5	↑ Audio Tape Load & Dump ↓	SAL	Starting Address - Low Order Byte
17F6		SAH	Starting Address - High Order Byte
17F7		EAL	Ending Address - Low Order Byte
17F8		EAH	Ending Address - High Order Byte
17F9		ID	File Identification Number
17FA	↑ Interrupt Vectors ↓	NMIL	NMI Vector - Low Order Byte
17FB		NMIH	NMI Vector - High Order Byte
17FC		RSTL	RST Vector - Low Order Byte
17FD		RSTH	RST Vector - High Order Byte
17FE		IRQL	IRQ Vector - Low Order Byte
17FF		IRQH	IRQ Vector - High Order Byte
1800	↑ Audio Tape ↓	DUMPT	Start Address - Audio Tape Dump
1873		LOADT	Start Address - Audio Tape Load
1C00	↑ STOP Key + SST ↓		Start Address for NMI using KIM "Save Machine" Routine (Load in 17FA & 17FB)
17F7	↑ Paper Tape Dump (Q) ↓	EAL	Ending Address - Low Order Byte
17F8		EAH	Ending Address - High Order Byte

Special Memory Addresses  
FIGURE 3.13

### 3.3 KIM-1 OPERATING PROGRAMS

Figure 3-14 shows a simplified flow chart of the KIM-1 operating programs. This section provides a brief explanation of these programs to assist you in understanding the various operating modes of the system.

First, you should note that when power is first applied to your KIM-1 module and the RS (reset) key is depressed, control of the system automatically is assumed by the operating program. This is true, as well, for any succeeding depression of the reset key.

For each depression of the reset key, the system is initialized. At this time, stack pointer values are set, the I/O configuration is established, and essential status flags are conditioned. Next the program determines whether the system is to respond to TTY inputs or is to operate with the keyboard and display on the KIM-1 module.

If the TTY mode has been selected, the program halts and awaits a first key depression from the TTY (the RubOut Key). Upon receipt of this key depression, the program automatically performs a bit rate measurement and stores the correct value for use in receiving and decoding succeeding data transfers from the TTY. Note that this bit rate measurement is performed after each depression of the reset key.

The program will proceed immediately to a routine causing the prompting message ("KIM") to be typed on the TTY. Now, the program halts at the loop called "Get Character". As each key is depressed on the TTY, the coded data is accepted and analyzed in the routine called "Execute Key". The various keys depressed will cause the program to branch to the appropriate subroutines required to perform the desired operation. Upon completion of the individual key executions, the program returns to the "Get Key" loop and awaits the next key depression.

#### 4. Recall Program Counter

Whenever the  $\overline{\text{NMI}}$  interrupt pin of the 6502 microprocessor array is activated, the program execution in progress will halt and the internal registers of the 6502 are saved in special memory locations before the control of the system is returned to the operating program. In the KIM-1 system, the NMI interrupt may occur in response to a depression of the **ST** key (stop) or, when operating in the Single Step mode, after each program instruction is executed following the depression of the **GO** key.

The **PC** key allows you automatically to recall the value of the Program Counter at the time an interrupt occurred. You may have performed a variety of operations since the interrupt such as inspecting the contents of various machine registers stored at specific memory locations. However, when you press the **PC** key, the contents of the Program Counter at the time of the interrupt are recalled to the address field of the display. You now may continue program execution from that point by pressing the **GO** key.

#### 5. Execute a Program

Select the starting address of the desired program. Now, press the **GO** key and program execution will commence starting with the address appearing on the display.

#### 6. Terminate a Program

The **ST** key is provided to allow termination of program execution. As mentioned earlier, the **ST** key activates the NMI interrupt input of the 6502 microprocessor array.

Note: The **ST** key will operate correctly only if you store the correct interrupt vector at locations 17FA and 17FB. For most of your work with the KIM-1 system, you should store the address 1C00 in these locations as follows:

<b>AD</b>			
<b>1</b>	<b>7</b>	<b>F</b>	<b>A</b>
<b>DA</b>		<b>0</b>	<b>0</b>
<b>+</b>		<b>1</b>	<b>C</b>

Now, when the  $\overline{\text{NMI}}$  interrupt occurs, the program will return to location 1C00 and will proceed to save all machine registers before returning control to the operating program.

You should remember to define the  $\overline{\text{NMI}}$  vector each time the power to the system has been interrupted. A failure of the system to react to the ST key means you have forgotten to define the  $\overline{\text{NMI}}$  vector.

#### 7. Single Step Program Execution

In the process of debugging a new program, you will find the single step execution mode helpful. To operate in this mode, move the SST slide switch to the ON position (to your left). Now, depress the GO key for each desired execution of a program step. The display will show the address and data for the next instruction to be executed. Note that in the course of stepping through a program, certain addresses will appear to be skipped. A program instruction will occupy one, two, or three bytes of memory depending upon the type of instruction. In single instruction mode, all of the bytes involved in the execution of the instruction are accessed and the program will halt only on the first byte of each successive instruction.

Note: SST mode also makes use of the  $\overline{\text{NMI}}$  interrupt of the 6502 microprocessor array. Again, the  $\overline{\text{NMI}}$  vector must be defined as described in (6) above if the SST mode is to work correctly.

This covers all of the standard operations you may perform from the KIM-1 keyboard. Using combinations of the operations described, you may wish to perform certain specialized tasks as follows:

##### 1. Define the $\overline{\text{IRQ}}$ Vector

You will recall that a separate interrupt input labelled  $\overline{\text{IRQ}}$  is available as an input to the 6502 microprocessor array. If you wish to use this feature, you should enter the address to which the program will jump. The  $\overline{\text{IRQ}}$  vector is stored in locations 17FE and 17FF.

##### 2. Interrogate Machine Status

We have mentioned that after an  $\overline{\text{NMI}}$  interrupt in response to the ST key or during the SST mode, the contents of various machine registers are stored in specific memory locations. If you wish to inspect these locations, their addresses are:

<u>Press Keys</u>	<u>See Displayed</u>	<u>Comments</u>
[AD] [0] [2] [0] [0]	0200 A9	Select first instruction address
ON [SST]	0200 A9	Set SST to ON; All selector switches open
[GO]	0202 8D	Accumulator now loaded with \$01
[GO]	0205 EE	PADD now loaded
[GO]	0208 AD	PA0 now toggled
[GO]	020B 49	Switch values (PA1-PA7) now loaded
[GO]	020D 4A	Accumulator now complemented
[GO]	020E AA	Accumulator now right shifted 1 Bit
[AD] [0] [0] [F] [3]	00F3 xx	Display Accumulator
[+]	00F4 xx	Display Y - INDEX
[+]	00F5 00	Display X - INDEX
[PC]	020E AA	Restore PC (TAX will execute next)
[GO]	020F CA	Accumulator now loaded in X-INDEX
[AD] [0] [0] [F] [3]	00F3 00	Display Accumulator
[+]	00F4 xx	Display Y-INDEX
[+]	00F5 00	Display X-INDEX (A=0→X)
[PC]	020F CA	Restore PC
[GO]	0210 10	DEX now completed
[AD] [0] [0] [F] [5]	00F5 FF	Display X-INDEX (X<0)
[PC]	0210 10	Restore PC
[GO]	0212 30	No branch (Result of DEX <u>not</u> positive)
[GO]	0205 EE	Branch (Result of DEX <u>is</u> negative).

*SST Mode: Sample Operation*  
*FIGURE 5.6*

