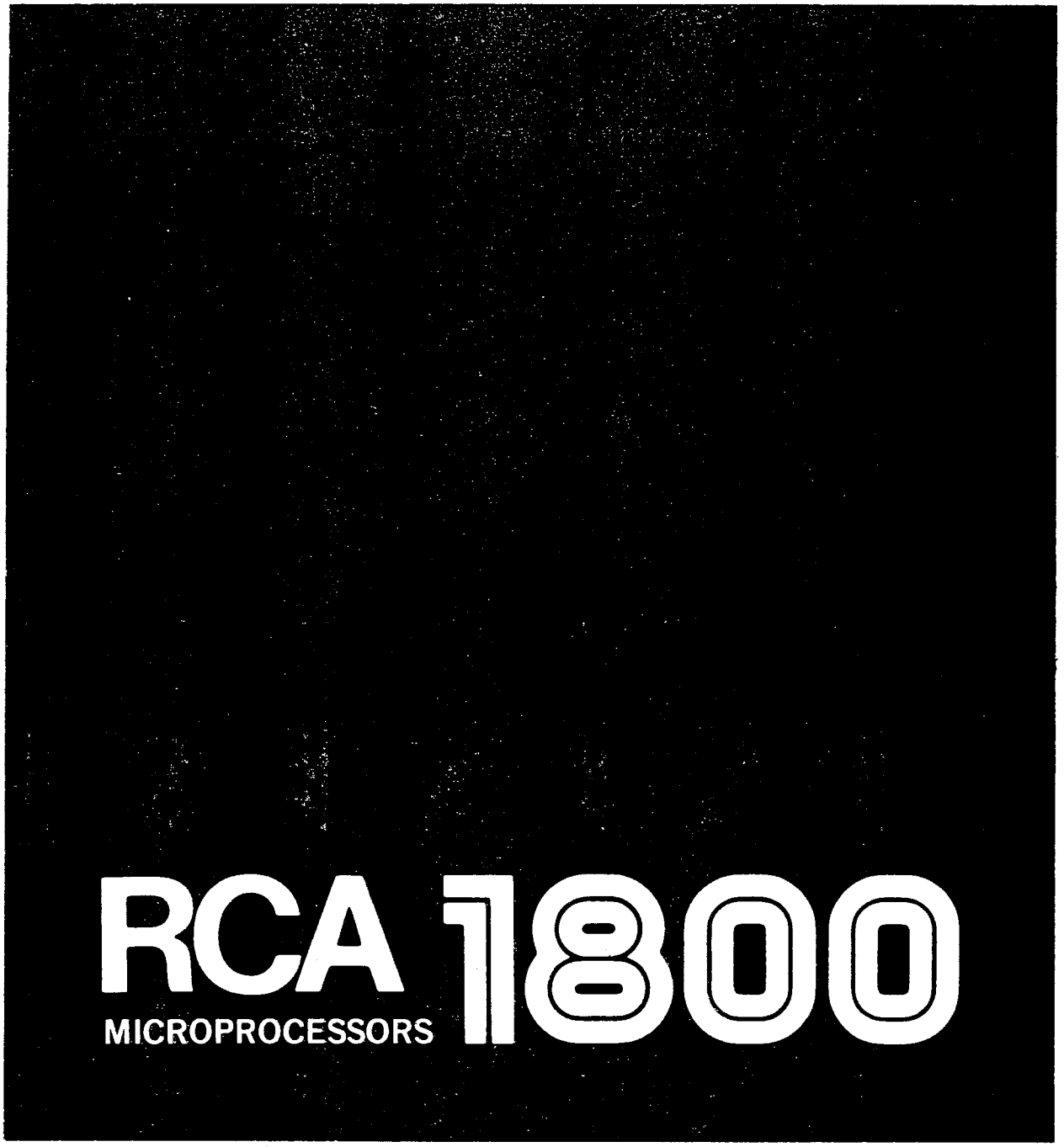


Instruction Manual for RCA COSMAC Microterminal

MPM-212 Suggested Price $2.00

Instruction Manual for
RCA COSMAC Microterminal

RCA | Solid State Division | Somerville, NJ 08876

Printed in USA/1-77

Foreword

The RCA COSMAC Microterminal CDP18S021 is a portable data terminal designed to operate with the CDP18S020 Evaluation Kit or with comparable user-designed RCA 1800 series microprocessor systems. The Microterminal is a low-power, low-cost, small-size, non-hard-copy alternative to conventional teletypewriter or similar terminals. It is particularly effective where portability and/or minimum cost are major system parameters.

This Manual is designed as a guide for users of the Microterminal. It includes a description of the hardware, the software programs available in a supplied ROM, and the operating modes of the device. Installation instructions are included for integrating the Microterminal into the Evaluation Kit.

For additional information on the RCA COSMAC Microprocessor CDP1802 and the Evaluation Kit CDP18S020, the user is referred to the following manuals:

MPM-201A User Manual for the RCA CDP1802 COSMAC Microprocessor

MPM-203 Evaluation Kit Manual for the RCA CDP1802 COSMAC Microprocessor

Table of Contents

Fig. 1 - RCA COSMAC Microterminal CDP18S021.

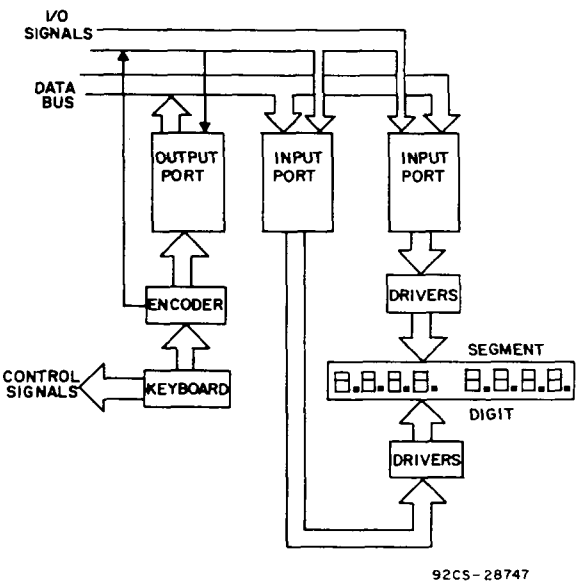


Fig. 2 - Block diagram of COSMAC Microterminal keyboard and display unit.

System Hardware

The RCA COSMAC Microterminal CDP18S021 consists of keyboard and display unit, its connecting cable and mating connector, and a ROM containing a Utility Program (UT5) to run the terminal and various subroutines that user programs can access. Although initially designed for use with the COSMAC Evaluation Kit CDP18S020, the Microterminal can also be used with comparable user-designed hardware support systems. The following descriptive material provides the information necessary to interface the Microterminal to the Evaluation Kit or to an appropriate system.

Keyboard/Display Unit

A photograph of the Microterminal is given in Fig. 1 and a block diagram of the keyboard/display unit in Fig. 2. The Microterminal system is divided into three functional sections: control, display, and keyboard.

The control section contains the requisite hardware for controlling the operation of the microprocessor system. The function keys are as follows:

R	Reset: resets the logic of the Microterminal and microprocessor system. Puts the CDP1802 in Reset state.
RU	Run Utility: starts execution of the Utility Program (UT5), which is at location 8000.

RP	**Run Program: starts program execution at location** 0000 with R0 as program counter.
CONT/STEP	Slide switch to enable continuous or single-step operation of the microprocessor system.
↔	Entry Mode Control: this key toggles between the address entry and data entry modes.
INC	Increment Address: each depression increments the address shown in the display. In the data entry mode, it also causes the data byte shown to be written to the address shown before incrementing the address.
$P	Start Addressed Program: starts program execution at the location shown in the address display.
CA	Clear Address: clears (resets) the address display to 0000.

The R, RU, RP, and CONT/STEP controls perform the same functions as the corresponding switches on the Evaluation Kit. For users designing their own systems, the execution of these control functions can be seen in the Evaluation Kit Manual for the RCA CDP1802 COSMAC Microprocessor, MPM-203.

The keyboard section of the Microterminal contains 16 digit keys (0 through F) that are used to enter hexadecimal numbers into the address or data field. The destination of entered data is controlled by the ↔ mode toggling switch. The hexadecimal digit keys and some of the control function keys are encoded and sent to the microprocessor system on the bidirectional data bus, as shown in the logic diagram of Fig. 3. Logic is provided for scanning and signalling keyboard activity to the microprocessor. The actual scanning, debounce, and decoding algorithms are performed by software routines in the Utility Program.

The display section consists of an eight-digit seven-segment LED display, display drivers, and refresh logic. In either the data entry or address entry mode of operation, the display shows a four-digit address field on the left side and a two-digit data field on the right separated by blank positions. In other subroutine-oriented display modes, all eight-digits are available in two groups of four separated by blank positions. This subject is discussed further in the next section, System Software. Illuminated decimal points in either the address or data field indicate the current operating mode. Software routines in the Utility Program perform digit selection, multiplexing, and hexadecimal-to-seven-segment code conversion.

Keyboard Operation

Depression of any key (except R, RU, or RP) produces a signal on EF3 of the CDP1802 as shown in Fig. 3. This signal starts the Utility Program which then issues an input instruction 6C to bring in the encoded data. This data is compared with a code table to discern which key was depressed. As shown in Fig. 3, the keys are in three groups.

Coding is produced by Exclusive-Or gates which invert certain bits depending on key group. To eliminate key "bounce", a debouncing routine repeatedly scans the input until consecutive valid codes are found. Subroutines to scan, decode, and debounce keys are given in the UT5 listing, Appendix B.

Display Operation

Each LED is composed of seven-segments and a decimal point. The eight LED's are connected to eight-segment drivers which in turn, are driven by an output port (U6). Output instruction 63 is used to latch segment data into this port. A second output port (U5) is used to turn on the selected digit with output instruction 64 used to latch the digit selection information. Multiplexing and segment control are provided by subroutines in UT5.

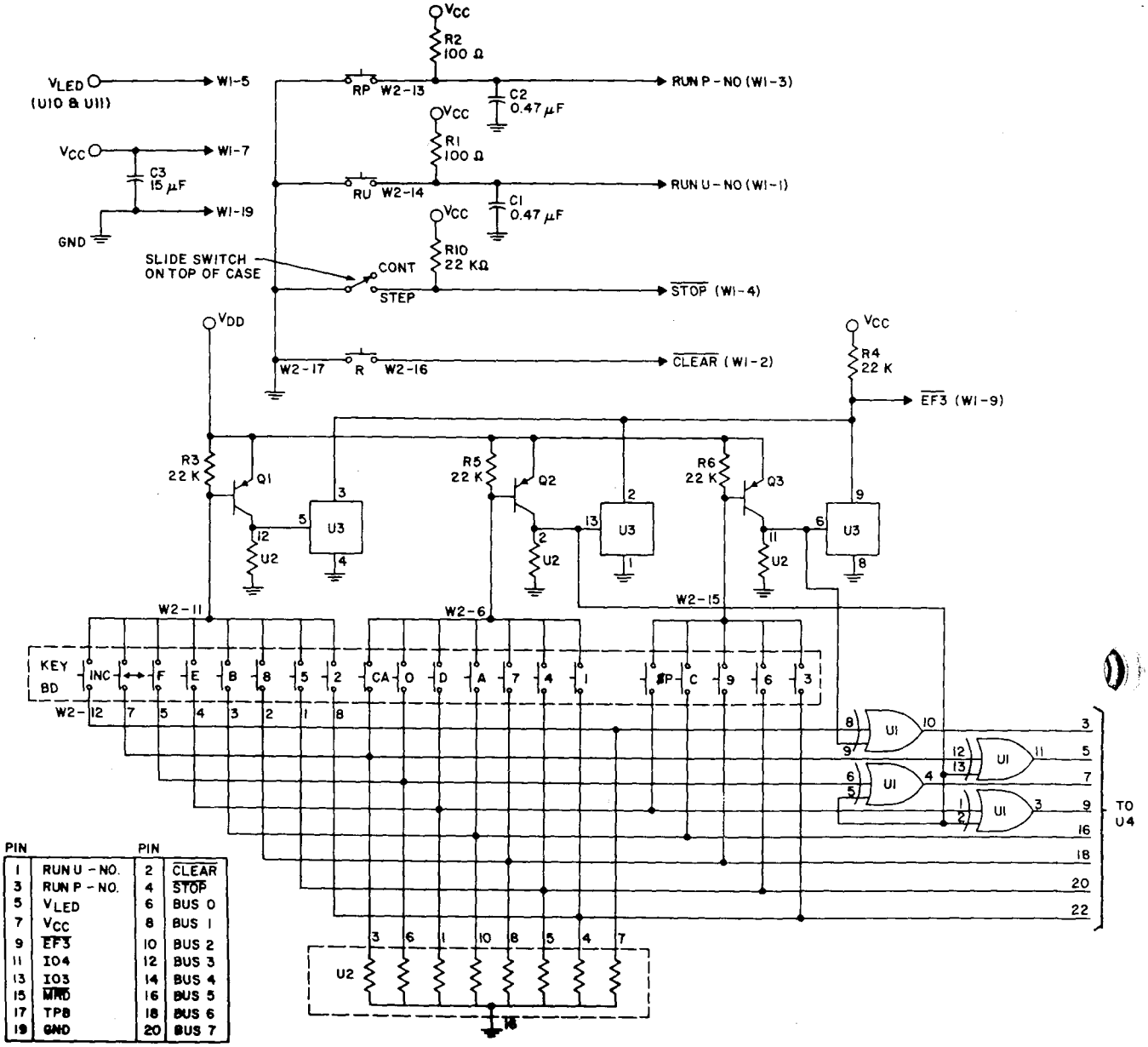
Digit and segment codes are shown in Fig. 4. To display a character, the respective codes for the segments of the character are fetched and the result sent to output port #3. The appropriate digit code is then sent to output port #4 to complete the operation.

A separate 5-volt supply V_{LED} is provided for the LED's and their drivers. The rest of the logic is supplied from V_{CC}, which may range from 5 to 12 volts. It should, of course, be the same voltage and from the same supply as the interfacing signals to the microprocessor system.

Utility ROM

The Utility ROM contains the software routines required to interface the Microterminal to a microprocessor system. These programs are collectively known as UT5. The ROM is a mask-programmed CDP1832D device, branded CDPR522, with a 512 x 8 configuration. UT5 is programmed to occupy memory locations 8000 through 81FF. It requires 32 bytes of RAM starting at location 8C00 for a stack area. For user-designed microprocessor systems, a CDP1824D RAM at location 8C00 will fulfill UT5 stack requirements. Standard COSMAC hardware support systems already have such a RAM.

There are two main sections in UT5: the KEYUT routine which handles the terminal interface, and user-oriented subroutines. These functions are described in the next section, System Software.

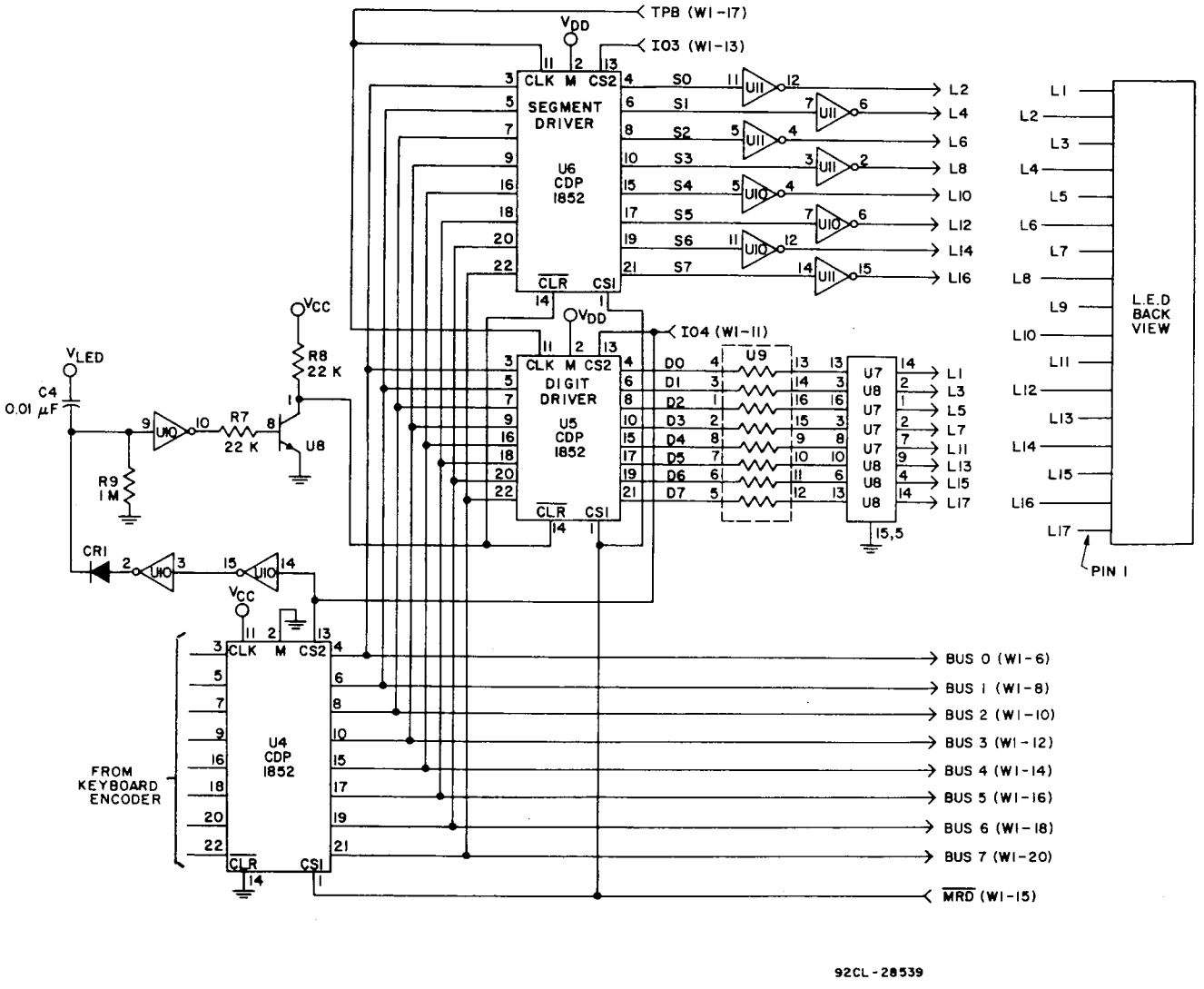


PIN		PIN	
1	RUN U - NO.	2	CLEAR
3	RUN P - NO.	4	STOP
5	V_{LED}	6	BUS 0
7	V_{CC}	8	BUS 1
9	EF3	10	BUS 2
11	IO4	12	BUS 3
13	IO3	14	BUS 4
15	MRD	16	BUS 5
17	TPB	18	BUS 6
19	GND	20	BUS 7

Parts List:

U1 = CD4030AE	U4 = CDP1852D
U2 = Resistor module, 10 kΩ, Beckman Inst.	U5 = CDP1852D
	U6 = CDP1852D
	U7 = CA3081
U3 = CD4066AE	U8 = CA3081

Fig. 3a - Schematic and logic diagram for Microterminal CDP18S021.
(Cont'd. on Page 11.)



Parts List (Cont'd.)

U9 = Resistor module, 470 Ω, Beckman Inst.
U10 = CD4049BE
U11 = CD4049BE
J1 = Header 3428-1002 3M Company
P1 = Connector 3421-3000 3M Company

Fig. 3a - Schematic and logic diagram for Microterminal CDP18S021. (Cont'd. from Page 10.)

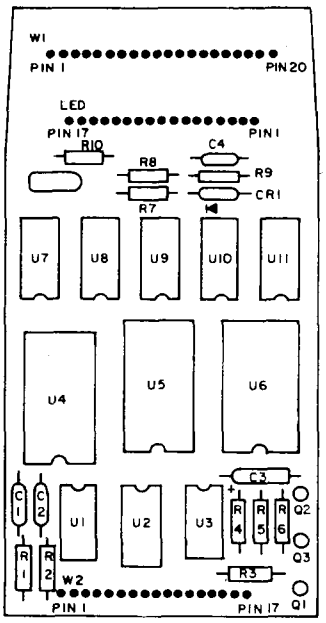
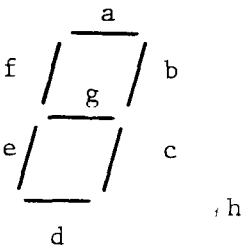


Fig. 3b - Microterminal assembly drawing.

SEGMENT CODE

Segment		Bit	Hexadecimal
a	1101	1111	DF
b	1111	1101	FD
c	0111	1111	7F
d	1111	0111	F7
e	1110	1111	EF
f	1111	1110	FE
g	1111	1011	FB
h (point)	1011	1111	BF



DIGIT CODE

Digit		Bit	Hexadecimal
1	1000	0000	80
2	0100	0000	40
3	0010	0000	20
4	0001	0000	10
5	0000	1000	08
6	0000	0100	04
7	0000	0010	02
8	0000	0001	01

Fig. 4 - Segment and digit codes for seven-segment LED display.

System Software

The Microterminal interface is controlled by routines in its associated ROM. These routines handle keyboard scan, key debounce and decode, display code conversion and multiplexing, standard modes of terminal operation, and display function subroutines addressable by a user program.

Microterminal Operating Modes

There are three basic operations that can be performed with the Microterminal: memory read, memory write, and CPU register readout. These operations are discussed below.

Memory Read

When the Microterminal is in the Address Entry mode, the contents of memory at the location shown in the address field are displayed as a two-digit hexadecimal byte in the data field. When RU is pressed, the terminal starts in the Address Entry mode, as denoted by lighted decimal points in the address field. The user then may enter the desired address by pressing the appropriate hexadecimal keys. Numbers are shifted in from right to left. (If the terminal is already in the Data Entry mode, it is first necessary to press the ↔ button to change modes.)

Example 1: Read memory location 801F.

First, make sure the CONT/STEP switch is in the CONT (leftmost) position. Then,

Enter	Press	Display	Comment
	R	---	Reset the CPU
	RU	0.0.0.0. X X*	Start UT5.
8		0.0.0.8. X X	
0		0.0.8.0 X X	Enter desired address, 801F
1		0.8.0.1 X X	
F		8.0.1.F. 3 2	Contents of 801F now displayed

To read contiguous addresses, press the INC button:

	INC	8.0.2.0. 8 0	Contents of 8020 is 80.
	INC	8.0.2.1. A 6	Contents of 8021 is A6.
	etc.		

*X denotes a don't care or intermediate display.

Memory Write

When the Microterminal is in the Data Entry mode, the byte in the data field is written to the address location indicated when the INC button is pushed. Moreover, the address is incremented by 1 and the next byte displayed in the data field. The Data Entry mode is signified by lighted decimal points in the data field. Hexadecimal numbers are entered in the data field from right to left. If the terminal is in the Address Entry mode, it is necessary to press the ↔ button to change modes.

Example 2: Change the data byte at location 20 to a 11. Assume initial arbitrary state with UT5 already running.

First, make sure the CONT/STEP switch is in the CONT position. Then,

Enter	Press	Display	Comment
	---	X X X X X.X.	Arbitrary initial conditions.
	CA	0 0 0 0 X.X.	Clear address.
	↔	0.0.0.0. X X	Go to Address Entry mode.
2		0.0.0.2. X X	Begin Address Entry.
0		0.0.2.0. X X	Address established.
	↔	0 0 2 0 X.X.	Go to Data Entry Mode.
1		0 0 2 0 X.1.	Begin Data Entry.
1		0 0 2 0 1.1.	Data established, but not written.
	INC	0 0 2 1 X.X.	Data has now been written to memory. The content of memory location 21 is then displayed in data field.

Example 3: Write an F6 into the next location (21).

Enter	Press	Display	Comment
	---	0 0 2 1 X.X.	End of Example 2.
F		0 0 2 1 X.F.	Enter the data byte.
6		0 0 2 1 F.6.	
	INC	0 0 2 2 X.X.	Write the data byte.

To skip a location without altering its contents, either press INC or switch to the memory address mode and go to the desired location.

Starting a User Program

A user program can be started in one of two ways. After R has been pressed, pressing the RP button will cause execution to start from memory location 0000 with P = X = 0. These steps will, of course, stop the Utility Program and cause the display to turn off. To recall the Utility program, the user should press R followed by RU.

To start from any location, the user should enter the starting address via the Address Entry mode and then press $P. In this case, R should not be pressed before $P because UT5 must be running for the $P command to be recognized.

CPU Register Readout

Registers R1 through RF of the CPU can be read out at the point where a user program is halted, as described below. When this feature is used with a "planted" IDLE instruction, it provides the means for implementing an elementary breakpoint for debugging purposes. This discussion assumes the Microterminal is being used with an Evaluation Kit system (CDP18S020).

Assume a breakpoint is required at location XXXX of the user program and at that point CPU registers and certain memory locations are to be examined. First, the user should write an idle instruction (op code 00) into location XXXX before starting the program. The user program will idle when it reaches that location. The user should start his program with RP or $P and when it idles do the following (program execution stops at the idle instruction):

1. Press R to reset the system.
2. Put the CONT/STEP switch in the STEP position.
3. Press RU four times. The CPU is now in the S0 state as indicated by both SC0 and SC1 state code LED's being off (on the Evaluation Kit) and the memory address lines=0002. The address lines indicate the number of the next register to be displayed.
4. Press RU once more. The contents of R2 are now displayed in the address lines LED's of the Evaluation Kit.
5. Press RU once more. The CPU is again in the S0 state as indicated by SC0 and SC1=0. The address lines=0003.
6. Press RU once more and the contents of R3 will be displayed. Continue pressing RU in the same sequence to display each subsequent register. After RF, the final register displayed is R1. Register R0 is not displayed.

Memory locations at the breakpoint can be examined by pressing R, switching to the CONT mode, pressing RU, and examining the memory contents as previously described.

Single-Step Operation

When the Microterminal is used with the Evaluation Kit, the single-step option can be used with the RU button to provide a readout of CPU registers as discussed previously. The RP button and the single-step control allow stepping through a user program and observing system operation on the LED's of the Evaluation Kit. For details of single-step operation, the reader should refer to the Evaluation Kit Manual for the RCA CDP1802 COSMAC Microprocessor, MPM-203, Section II, Design and Operation, subsection 3, CONTROL (page 2-5).

Microterminal Utility Subroutines (UT5)

The Microterminal Utility Program UT5 contains two main sections: a routine called KEYUT which takes care of terminal interfacing and a group of user-oriented subroutines. These programs are discussed next. The names of these subroutines and their absolute addresses are given below:

Subroutine	Address
KEYUT	8000
ENTRY	8108
REGDIS	81A6
LEDD	816C
COUNT	814F
CALL	81E4
RETURN	81F4

KEYUT

In the servicing of Microterminal operations, KEYUT performs two main tasks:

1. Refreshes the display periodically by sending out both segment and digit signals to the two output ports, and
2. Periodically scans for keyboard inputs and executes the required actions.

Fig. 5 shows a flowchart of KEYUT operations. The first few instructions in KEYUT are data fetches via registers R2, R3, . . . RF, R1. These instructions perform no useful function in the Continuous mode of operation. However, they do cause the contents of registers R1 through RF to appear on the memory address bus generating the CPU register readout function.

KEYUT spends most of its time in a loop of refreshing the display and waiting for a key depression. When a key is pressed, the program decodes it and performs the required action. KEYUT waits for the key to be released (refreshing the display in the meantime) before returning to the main loop. KEYUT is wholly a part of the Utility program. It cannot be called by user programs.

ENTRY

As a convenience to the user, the ENTRY subroutine at location 8108 initializes the CPU registers necessary for the COSMAC Standard Call and Return Technique. This technique allows multiple levels of subroutine nesting by use of a stack in RAM. The Call and Return subroutines are included on the ROM (UT5). The reader should refer to the section on Programming Techniques in the User Manual for the RCA CDP1802 COSMAC Microprocessor, MPM-201A, for a detailed description of this technique for general subroutining.

After the ENTRY routine is called via a long branch instruction, control is returned to the user program at location 0005 with

P = 3
X = 2
R2 = 8C1F (top of stack)
R4 = 81E4 (standard CALL routine)
R5 = 81F4 (standard RETURN routine)

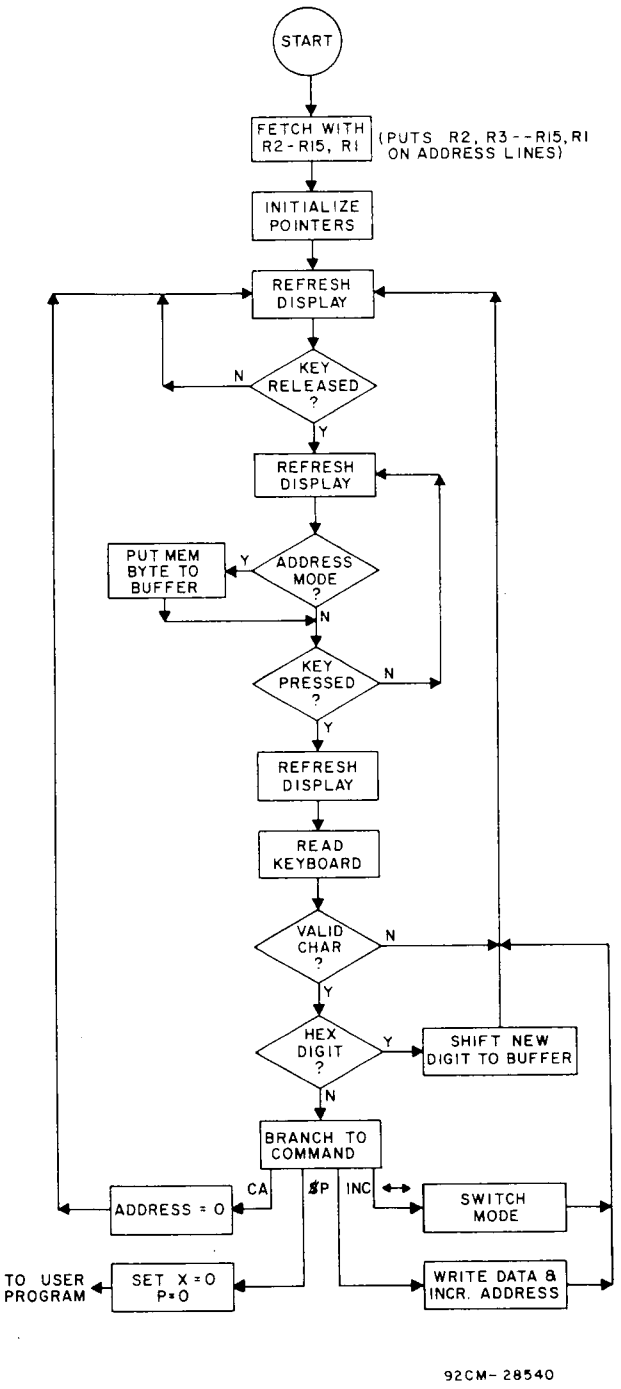


Fig. 5 - Flowchart for Microterminal utility program KEYUT.

Thus, a long branch to ENTRY (op code C08108) as the first three bytes of a user program saves writing the initialization code otherwise required to establish the Standard Call and Return Technique. The program returns control to memory location 0005. Examples 4 and 5 below show how ENTRY can be used for subroutine calls.

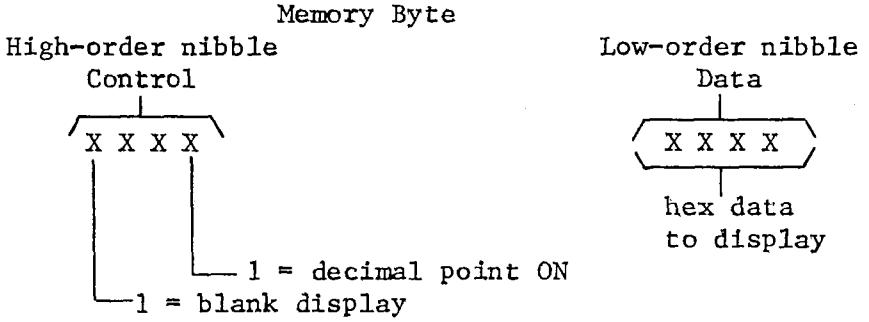
COUNT

The COUNT subroutine at location 814F is an independent program that displays memory sequentially starting from location 0000, incrementing the address approximately once per second. This program provides an automatic read-back of a user program previously entered. To run this program, a user would set the Microterminal address to 814F and press the $P key.

Readout Subroutines - REGDIS and LEDD

Two subroutines are provided to output data to the display via a user program. The subroutine REGDIS at location 81A6 sends the contents of registers RA and RB to the display. Each register is displayed as a four-digit hexadecimal number. RA appears at the left of the display. REGDIS uses the standard call and return technique. To call REGDIS, the following code is used: D481A6. REGDIS should be called frequently to obtain maximum LED brightness.

The other subroutine, called LEDD and located at 816C, is more general purpose. It allows user control of all eight-digits of display, plus their decimal points. LEDD reads out eight consecutive bytes of memory, starting at the location pointed to by RF, interpreting the bits at each location as a control and data character as follows:



LEDD thus allows displays of all eight digits. Unused display positions are specified as blanks in the appropriate memory positions. LEDD leaves the data pointer RF at its initial value when it exits.

Examples of Use of UT5 Subroutines

Example 4: Display registers RA and RB. Increment RA.

Enter	Press	Display	Comment
	R		
	RU	0.0.0.0. X X	Start UT5.
	↔	0 0 0 0 X.X.	Go to Data Entry mode.
C0		0 0 0 0 C.0.	Enter C then 0 to
	INC	0 0 0 1 X.X.	set up a long branch to ENTRY.

Enter	Press	Display		Comment
81		0 0 0 1	8.1.	
	INC	0 0 0 2	X.X.	
08		0 0 0 2	0.8.	
	INC	0 0 0 3	X.X.	Skip to location 0005
	INC	0 0 0 4	X.X.	the return point
	INC	0 0 0 5	X.X.	from ENTRY.
04		0 0 0 5	D.4.	Call to REGDIS.
	INC	0 0 0 6	X.X.	
81		0 0 0 6	8.1.	
	INC	0 0 0 7	X.X.	
A6		0 0 0 7	A.6.	
	INC	0 0 0 8	X.X.	
1A		0 0 0 8	1.A.	Increment RA.
	INC	0 0 0 9	X.X.	
30		0 0 0 9	3.0.	Loop back to 0005.
	INC	0 0 0 A	X.X.	
05		0 0 0 A	0.5.	
	INC	0 0 0 B	X.X.	Program is now loaded. To start it running press
	R	---		
	RP	X X X X	X X X X	
		RA	RB	

Example 5: Display the number 3.14.

Enter	Press	Display		Comment
	R			
	RU	0.0.0.0.	X X	Start UT5.
		0 0 0 0	X.X.	Go to data entry mode.
C0		0 0 0 0	C.0.	Set up long
	INC	0 0 0 1	X.X.	
81		0 0 0 1	8.1.	branch to ENTRY.
	INC	0 0 0 2	X.X.	
08		0 0 0 2	0.8.	
	INC	0 0 0 3	X.X.	Skip to location 0005.
	INC	0 0 0 4	X.X.	
	INC	0 0 0 5	X.X.	
F8		0 0 0 5	F.8.	Point RF to 0020, the display area.
	INC	0 0 0 6	X.X.	
00		0 0 0 6	0.0.	
	INC	0 0 0 7	X.X.	
BF		0 0 0 7	B.F.	
	INC	0 0 0 8	X.X.	
F8		0 0 0 8	F.8.	
	INC	0 0 0 9	X.X.	
20		0 0 0 9	2.0.	
	INC	0 0 0 A	X.X.	
AF		0 0 0 A	A.F.	
	INC	0 0 0 B	X.X.	

Enter	Press	Display		Comment
D4		0 0 0 B	D.4.	Call LEDD.
	INC	0 0 0 C	X.X.	
81		0 0 0 C	8.1.	
	INC	0 0 0 D	X.X.	
6C		0 0 0 D	6.C.	
	INC	0 0 0 E	X.X.	
30		0 0 0 E	3.0.	Branch back to call
	INC	0 0 0 F	X.X.	
0B		0 0 0 F	0.B.	LEDD.
	INC	0 0 1 0	X.X.	

Above is the main program. Next, set up the digits to be displayed starting from location 0020 (where RF is pointing).

	CA	0 0 0 0	X.X.	
	↔	0.0.0.0.	X X	Go to address entry mode.
20		0.0.2.0.	X X	Set address to 20.
	↔	0 0 2 0	X.X.	Go to data entry mode.
80		0 0 2 0	8.0.	First 5 digits
	INC	0 0 2 1	X.X.	
80		0 0 2 1	8.0.	are blanked.
	INC	0 0 2 2	X.X.	
80		0 0 2 2	8.0.	
	INC	0 0 2 3	X.X.	
80		0 0 2 3	8.0.	
	INC	0 0 2 4	X.X.	
80		0 0 2 4	8.0.	
	INC	0 0 2 5	X.X.	
13		0 0 2 5	1.3.	Display 3. when program is run
	INC	0 0 2 6	X.X.	
01		0 0 2 6	0.1.	Display 1
	INC	0 0 2 7	X.X.	
04		0 0 2 7	0.4.	Display 4
	INC	0 0 2 8	X.X.	

The program is now loaded. To start it, press

	R	---	
	RP		3.14

Appendix A -

Instructions for Installing the CDP18S021 Microterminal in the CDP18S020 Evaluation Kit

The CDP18S021 Microterminal and the CDP18S020 Evaluation Kit have been designed to interface with each other. As a result, installation is straightforward and requires only a minimum amount of time. To assist in this installation, a step-by-step procedure is given below. It is recommended that the entire procedure be reviewed before the installation is begun.

Procedure

1. Unpack the shipping container for the CDP18S021 Microterminal. The items provided are:
   a. CDP18S021 Microterminal unit
   b. Mating connector
   c. Utility ROM CDPR522 (a CDP1832D ROM mask-programmed with Utility program UT5)
      NOTE: The CDPR522 ROM should not be removed from its shipping carrier until called for by the instructions.
2. Mount and solder the Microterminal mating connector at J1 on the solder side of the Evaluation Kit PC card. J1 is located next to the User I/O Connector P3. Fig. A-1 gives the assembly detail. The pin assignments for connector J1, the keyboard connector, are given in Fig. A-2.
3. The Microterminal includes control keys which duplicate the RESET (S2), RUN (S3), RUN P (S4), and CONT/STEP (S5) controls on the Evaluation Kit. In order to use the RUN U and RUN P Microterminal controls at a V_{CC} greater than 5 volts without damage to U11, it is necessary to remove the corresponding switches (S3 and S4) on the Evaluation Kit. Several alternatives are possible and are listed below. One of these alternatives must be selected and completed during this step if it is intended to operate the Evaluation Kit at a V_{CC} above 5 volts. Only S3 and S4 need to be removed; the other switches are redundant. The alternatives are:
   a. Remove switch S3 (RUN U) and switch S4 (RUN P) from the Evaluation Kit PC card and discard them. See Fig. A-1. Or,
   b. Disable the RUN U and RUN P functions from the Microterminal by removing wires at pins 1 and 3 from connector J1. See Fig. A-1. Or,

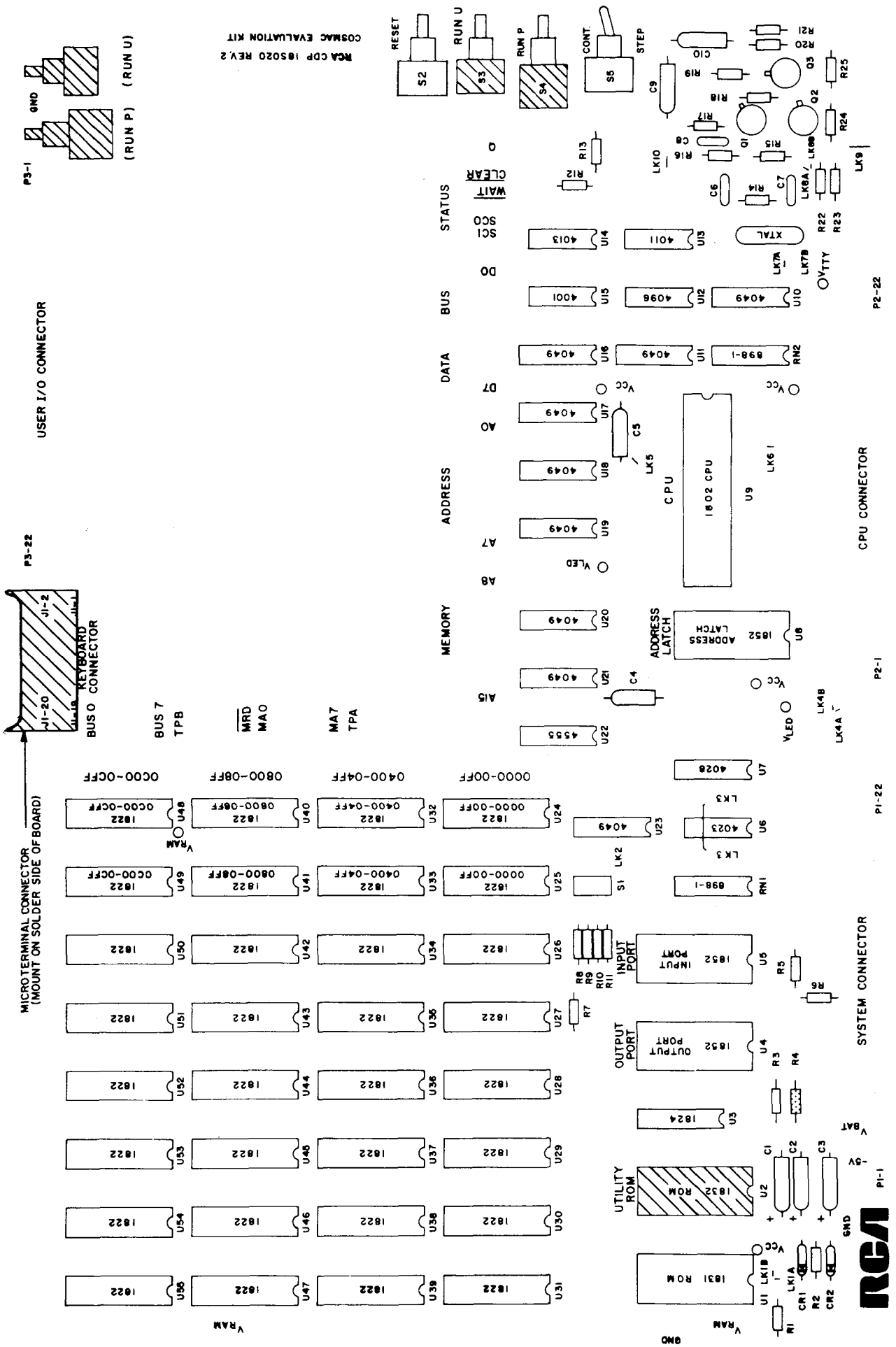


Fig. A-1 - Modification of Evaluation Kit PC card for use with RCA COSMAC Microtermir CDP18S021.

c. Remove switch S3 (RUN U) and switch S4 (RUN P) from the Evaluation Kit PC card and relocate them in positions next to the User I/O connector P3. Jumpers can be inserted between the alternate locations and the original switch positions in order to regain PC card control of the RUN U and RUN P functions. See Fig. A-1.

4. Remove the CDPR512 Utility ROM (CDP1832 - UT4, 24 pins) from the U2 socket location on the Evaluation Kit PC card. Replace it with the CDPR522 Utility ROM (CDP1832 - UT5) supplied with the Microterminal. Carefully observe the pin 1 orientation as indicated in Fig. A-1.

5. Insert the J1 mating connector attached to the end of the ribbon cable coming from the back of the Microterminal into the J1 connector just mounted on the Evaluation Kit PC card. Make sure that the pin 1's, marked by Δ , of both connectors line up.

6. If Link 4B has been installed on the PC card, it must be removed. This link connects the Service Request from the input port U5 to the $\overline{EF3}$ line. Because the Microterminal will be using $\overline{EF3}$, this link plus any user-added connections to that flag line should also be removed. The input port can be connected to another flag line or Link 4A can be installed to connect the Service Request to the INTERRUPT input. For additional information on these connections, refer to the Evaluation Kit Manual, Section II - Design and Operation, Subsection 6 - Input/Output.

BUS 7	BUS 6	BUS 5	BUS 4	BUS 3	BUS 2	BUS 1	BUS 0	$\overline{STOP}$	$\overline{RESET}$
20	18	16	14	12	10	8	6	4	2
19	17	15	13	11	9	7	5	3	1
GND	TPB	$\overline{MRD}$	I/03	I/04	EF3	V_{CC}	V_{LED}	$\overline{RUN\ P}$	$\overline{RUN\ U}$

Fig. A-2 - Pin assignments for keyboard connector J1.

Appendix B -

Utility Program UT5 Listing

```
!M
0000 ;                  0001
0000 ;                  0002            ...
0000 ;                  0003            ORG #8000
8000 7100;              0004 KEYUT:     ;#7100 ...DISABLE INTERRUPT
8002 0203040506070 8;   0005            LDN 2;LDN 3;LDN 4;LDN 5;LDN 6;LDN 7;LDN 8
8009 090A0B0C0D0E0F;    0006            LDN 9;LDN A;LDN B;LDN C;LDN D;LDN E;LDN F
8010 01;                0007            LDN 1
8011 F880B3;            0008            LDI     A.1(OPTION);PHI 3 ...INIT
8014 F8FEA3;            0009            LDI     A.0(OPTION) ;PLO 3 ...SETUP P
8017 C0810E;            0010            LBR INIT...INITIALIZE AND GO
801A ;                  0011            ...
801A ;                  0012            ...
801A ;                  0013            ...SUBROUTINE LED
801A ;                  0014            ...DISPLAYS 9 DIGITS IN DISPLAY
801A ;                  0015            ...OF SEVEN SEGMENTS
801A ;                  0016            ...
801A ;                  0017            ...USES R7,RD,RB
801A ;                  0018            ...
801A ;                  0019            ...
801A ;                  0020            ...
801A ;                  0021            ...
801A F800A7;            0022 LED:       LDI     A.0(DIGITS);PLO 7 ..TO DISPLAY
801D 94BD;              0023            GHI 4;PHI D ...POINT TO TRANSLATION TAB
801F F880AB;            0024            LDI     #80;PLO B ...START DISPLAY
8022 E2;                0025            SEX 2 ...USE R2 AS AUX
8023 2222;              0026 LOOP1:     DEC 2;DEC 2 ...GET READY FOR OUTPUTS
8025 8B52;              0027            GLO B;STR 2 ...PUT IN DIGIT NO.
8027 47FA0FFC36AD;      0028            LDA 7;ANI #0F;ADI A.0(DTAB);PLO D..POINT TO
802D 4DBB;              0029            LDA D;PHI B ...FETCH DISPLAY DIGIT
802F 64;                0030            OUT 4 ...PUT OUT DIGIT
8030 8CFA01;            0031            GLO C;ANI #01 ...GET MODE
8033 323F;              0032            BZ DISD ...DATA MODE
8035 8BFA0F;            0033            GLO B;ANI #0F ...1ST DIGIT?
8038 3A44;              0034            BNZ OUTD ...PROCEED IF NOT
803A 9BFABF;            0035 LDOT:      GHI B;ANI #BF ...ELSE SHOW DOT
803D 3045;              0036            BR OUTD+1 ...PROCEED
803F 8BFAFC;            0037 DISD:      GLO B;ANI #FC ...7TH DIGIT?
8042 323A;              0038            BZ LDOT ...SHOW IT IF SO
8044 9B52;              0039 OUTD:      GHI B;STR 2 ...& SEGMENT DISPLAY
8046 63;                0040            OUT 3 ...SEGMENT MUX
8047 D48147;            0041 DELAY:     SEP 4;;A(DELSUB) ...DELAY
804A E3;                0042            SEX 3 ...PC=AUX
804B 63FF;              0043            OUT 3;;#FF ...TURN OFF SEGMENT
804D E2;                0044            SEX 2 ...R2=AUX
804E 87FF06;            0045 CHEND:     GLO 7;SMI A.0(DIGITS+6) ...DONE?
8051 3B54;              004E            BNF CD ...CONTINUE
8053 D5;                0047            SEP 5 ...EXIT
8054 8BF6AB;            0048 CD:        GLO B;SHR;PLO B ...SHIFT
8057 FAF3;              0049            ANI #F3 ...2 BLANK DIGITS?
8059 3A23;              0050            BNZ LOOP1 ...NO, KEEP GOING!
```

```
805B 304E;          0051          BR CHEND ...SEE IF DONE
805D ;              0052          ...
805D ;              0053          ...
805D ;              0054          ...HKEY SUBROUTINE
805D ;              0055          ...READS 1 HEX DIGIT FROM KEYBOARD
805D ;              0056          ...AND STORES IN RA.0
805D ;              0057          ...
805D D4801A;        0058 HKEY1:   SEP 4;,A(LED) ...SCAN & REFRESH
8060 365D;          0059          B3 HKEY1 ...ELSE SCAN
8062 D4801A;        0060 HKEY2:   SEP 4;,A(LED) ...SCAN & REFRESH
8065 8CFA01;        0061          GLO C;ANI #01 ...MODE=ADDRS?
8068 326D;          0062          BZ HKEY2B ...NO;SKIP
806A D480D3;        0063          SEP 4;,A(UPDRE0) ...UPDATE RE.0
806D 3E62;          0064 HKEY2B:  BN3 HKEY2 ...& SCAN
806F D4801A;        0065          SEP 4;,A(LED) ...DELAY THRU BOUNCE
8072 94BA;          0066          GHI 4;PHI A ...POINT RA
8074 F822AA;        0067          LDI A.0(KTAB);PLO A ...TO TAB
8077 6C;            0068          INP 4 ...READ BYTE
8078 4A;            0069 FCOM:    LDA A ...FETCH FROM TABLE
8079 F3;            0070          XOR ...MATCH WITH INPUT
807A 3283;          0071          BZ FMAT ...FOUND
807C 8AFF36;        0072          GLO A;SMI A.0(KTAB+#14) ...ELSE, END?
807F 335D;          0073          BDF HKEY1 ...YES,ERR, DO OVER
8081 3078;          0074          BR FCOM ...CONTINUE SCAN
8083 8AFF23AA;      0075 FMAT:    GLO A;SMI A.0(KTAB+1);PLO A ...COMPUTE DISP
8087 FAF0;          0076          ANI #F0 ...>10?
8089 3A90;          0077          BNZ SPBRN ...YES, DO BRANCHES
808B D4809E;        0078          SEP 4;,A(SHIFTD) ...PUT IN DIGIT
808E 305D;          0079          BR HKEY1 ...BACK
8090 8AFA0FFC9AAA;  0080 SPBRN:   GLO A;ANI #0F;ADI A.0(BTAB);PLO A..POINT TO
8096 93BA;          0081          GHI 3;PHI A ...SETUP UPPER HALF
8098 4AA3;          0082          LDA A;PLO 3 ...POINT TO ROUTINE ENTRY
809A ;              0083          ...
809A ;              0084          ...
809A ;              0085          ...
809A FA;            0086 BTAB:    ,A.0(XCHN) ...EXCHANGE DATA & ADDR MODE
809B F0;            0087          ,A.0(DOLP) ...$P
809C F6;            0088          ,A.0(INC) ...INCREMENT ADDRS
809D FE;            0089          ,A.0(OPTION) ...EXTRA
809E ;              0090          ...
809E ;              0091          ...
809E ;              0092          ...SHIFT SUBROUTINE
809E ;              0093          ...ONE PLACE RIGHT AND PUT
809E ;              0094          ...UPDATES BUF & REG
809E ;              0095          ...IN NEW DIGIT FROM RA.0
809E ;              0096          ...
809E ;              0097          ....
809E 8CFA01;        0098 SHIFTD:  GLO C;ANI #01 ...MODE=ADDRS?
80A1 32E5;          0099          BZ DAS ...NO
80A3 9FFEFEFEFE52;  0100 ADS:     GHI F;SHL;SHL;SHL;SHL;STR 2
80A9 8FF6F6F6F6;    0101          GLO F;SHR;SHR;SHR;SHR
80AE F1BF;          0102          OR;PHI F ...COMPOSE RF.1
80B0 8FFEFEFEFE52;  0103          GLO F;SHL;SHL;SHL;SHL;STR 2
80B6 8AF1AF;        0104          GLO A;OR;PLO F ...COMPOSE RF.0
80B9 F800A7;        0105 UPDRF:   LDI A.0(DIGITS);PLO 7 ...POINT TO BUF
80BC 9FF6F6F6F657;  0106          GHI F;SHR;SHR;SHR;SHR;STR 7 ...1ST DIGIT
80C2 17;            0107          INC 7 ...BUMP
80C3 9FFA0F57;      0108          GHI F;ANI #0F;STR 7 ...2ND DIGIT
80C7 17;            0109          INC 7 ...BUMP
80C8 8FF6F6F6F657;  0110          GLO F;SHR;SHR;SHR;SHR;STR 7 ...3RD DIGIT
80CE 17;            0111          INC 7 ...BUMP
80CF 8FFA0F57;      0112          GLO F;ANI #0F;STR 7 ...4TH DIGIT
80D3 4FAE;          0113 UPDRE0:  LDA F;PLO E ...FETCH BYTE FROM MEMORY
80D5 2F;            0114          DEC F ...FIX RF
80D6 F804A7;        0115 UPDBY:   LDI A.0(DIGITS+4);PLO 7 ...POINT TO BUF
80D9 8EF6F6F6F657;  0116          GLO E;SHR;SHR;SHR;SHR;STR 7 ...1ST TO BUF
```

```
80DF 17;              0117          INC 7 ...BUMP R7
80E0 8EFA0F57;        0118          GLO E;ANI #0F;STR 7 ...2ND DIGIT TO BUF
80E4 D5;              0119          SEP 5 ...RETURN
80E5 8E;              0120 DAS:     GLO E ...GET DIGIT
80E6 FEFEFEFE52;      0121          SHL;SHL;SHL;SHL;STR 2
80EB 8AF1AE;          0122          GLO A;OR;PLO E ...COMPOSE NEW BYTE
80EE 30D6;            0123          BR UPDBY ...& UPDATE DISPLAY
80F0 ;                0124          ...
80F0 ;                0125          ...
80F0 ;                0126          ...
80F0 ;                0127          ...SPECIAL BRANCH FOR
80F0 ;                0128          ...EXTRA FUNCTIONS
80F0 ;                0129          ...
80F0 ;                0130          ...
80F0 9FB0;            0131 DOLP:    GHI F;PHI 0 ...SETUP R0
80F2 8FA0;            0132          GLO F;PLO 0 ...FOR PC
80F4 E0;              0133          SEX 0
80F5 D0;              0134          SEP 0
80F6 8E5F1F;          0135 INC:     GLO E;STR F;INC F ...WRITE & BUMP
80F9 38;              0136          ;#38 ...SKIP A BYTE
80FA 1C;              0137 XCHN:    INC C ...FLIP MODE
80FB C08102;          0138          LBR CUP
80FE F800BFAF;        0139 OPTION:  LDI #00;PHI F;PLO F ...CLEAR RF
8102 D480B9;          0140 CUP:     SEP 4;.A(UPDRF) ...REFRESH
8105 C0805D;          0141          LBR HKEY1 ...& SCAN KEYS
8108 ;                0142          ...
8108 ;                0143          ...
8108 ;                0144          ...
8108 ;                0145          ...ENTRY POINT FOR STARTING
8108 ;                0146          ...PROGRAM AT LOC 3 WITH P=3
8108 ;                0147          ...R2 = STACK POINTER = #8C1F
8108 ;                0148          ...R4 = CALL SUBROUTINE POINTER
8108 ;                0149          ...R5 = RETURN SUBROUTINE POINTER
8108 ;                0150          ...
8108 ;                0151          ...
8108 F800B3;          0152 ENTRY:   LDI #00;PHI 3
810B F805A3;          0153          LDI #05;PLO 3 ...STARTS AT 0005
810E F881B4B5;        0154 INIT:    LDI A.1(CALL);PHI 4;PHI 5 ...SUBROUTINE
8112 F8E4A4;          0155          LDI A.0(CALL);PLO 4 ...CALL
8115 F8F4A5;          0156          LDI A.0(RETPGM);PLO 5 ...& RETURN
8118 F88CB2B7;        0157          LDI #8C;PHI 2;PHI 7 ...SETUP
811C F81FA2;          0158          LDI #1F;PLO 2 ...POINTER TO STACK
811F AC;              0159          PLO C ...MODE = ADDRS
8120 E2;              0160          SEX 2 ...R2=AUX
8121 D3;              0161          SEP 3 ...GO
8122 ;                0162          ...
8122 ;                0163          ...
8122 0A8E80814E4041;  0164 KTAB:    ;#0A8E80814E40412E20211E10110608040209010C
8129 2E20211E101106;  0164
8130 08040209010C;    0164
8136 447DC1517852425D4050486 2C661C2CAFF
8136 447DC1517852425D4050486;0165 DTAB:    ;#447DC1517852425D4050486 2C661C2CAFF
813D 5D4050486 2C661;  0165
8144 C2CAFF;          0165
8147 ;                0166          ...
8147 ;                0167          ...DELAY SUBROUTINE
8147 ;                0168          ...DELAYS FOR A FIXED TIME
8147 ;                0169          ...
8147 ;                0170          ...
8147 F820AD;          0171 DELSUB:  LDI #20;PLO D
814A 2D;              0172 DL:      DEC D ...COUNT
814B 8D;              0173          GLO D ...DONE?
814C 3A4A;            0174          BNZ DL ...NO, KEEP GOING
814E D5;              0175          SEP 5 ...RETURN
814F ;                0176          ...
```

```
814F ;                  0177          ...
814F ;                  0178          ...
814F ;                  0179          ...COUNTER PROGRAM
814F ;                  0180          ...TO COUNT UP & DISPLAY
814F ;                  0181          ...THE COUNT IN LED DISPLAY
814F ;                  0182          ...MEANT FOR EXERCISE ONLY
814F ;                  0183          ...
814F ;                  0184          ...
814F F881B3;            0185 COUNT:   LDI A.1(INCR);PHI 3 ...SETUP
8152 F858A3;            0186          LDI A.0(INCR);PLO 3 ...POINTERS
8155 C0810E;            0187          LBR INIT ...FOR ROUTINE
8158 F800BFAF;          0188 INCR:    LDI #00;PHI F;PLO F ...ZERO RF
815C 1F;                0189 INCRM:   INC F ...ELSE BUMP RF
815D F880A8;            0190 DELAY1:  LDI #80;PLO 8 ...SET DISPLAY TIME
8160 D480B9;            0191          SEP 4;.A(UPDRF) ...UPDATE DISPLAY
8163 D4801A;            0192 SHOW:    SEP 4;.A(LED) ...& LIGHT UP
8166 28;                0193          DEC 8 ...DELAY
8167 88;                0194          GLO 8 ...TIME
8168 3A63;              0195          BNZ SHOW ...KEEP LIGHTING UP
816A 305C;              0196          BR INCRM ...ADD 1
816C ;                  0197          ...
816C ;                  0198          ...
816C ;                  0199          ...LEDD SUBROUTINE
816C ;                  0200          ...REFRESH DISPLAY
816C ;                  0201          ...AT BUFFER M(R(F)) FOR 8 BYTES
816C ;                  0202          ...USES RD,RE,RF
816C ;                  0203          ...
816C ;                  0204          ...BIT 7 = 1 = BLANK
816C ;                  0205          ...BIT 4 = 1 = POINT
816C ;                  0206          ...BITS 0 -> 3 = HEX DIGIT
816C ;                  0207          ...
816C ;                  0208          ...
816C F880AE;            0209 LEDD:    LDI #80;PLO E ...START WITH LEFTMOST DIGIT
816F F881BD;            0210          LDI A.1(DTAB);PHI D ...SEGMENT TABLE POINTER
8172 0FFA80;            0211 LOOPD:   LDN F;ANI #80 ...TURN OFF?
8175 3A97;              0212          BNZ SKIPD ...YES, SKIP DIGIT
8177 2222;              0213          DEC 2;DEC 2 ...GET 2 FREE RAM BYTES
8179 8E52;              0214          GLO E;STR 2 ...READY FOR DIGIT OUT
817B 0FFA0FFC36AD;      0215          LDN F;ANI #0F;ADI A.0(DTAB);PLO D ..TAB
8181 4DBE;              0216          LDA D;PHI E ...FETCH SEGMENTS
8183 0FFA10;            0217          LDN F;ANI #10 ...DECIMAL?
8186 328C;              0218          BZ DISPD ...NO, SHOW DIGIT ONLY
8188 9EFABFBE;          0219          GHI E;ANI #BF;PHI E ...ELSE ADD POINT
818C 64;                0220 DISPD:   OUT 4 ...SELECT DIGIT
818D 9E52;              0221          GHI E;STR 2 ...PUT IN SEGMENTS
818F 63;                0222          OUT 3 ...TURN ON
8190 D48147;            0223          SEP 4;.A(DELSUB) ...DELAY
8193 E3;                0224          SEX 3 ...PC=AUX
8194 63FF;              0225          OUT 3;.#FF ...TURN OFF SEGMENT
8196 E2;                0226          SEX 2 ...R2=AUX
8197 1F;                0227 SKIPD:   INC F ...BUMP POINTER
8198 8EF6AE;            0228          GLO E;SHR;PLO E ...SHIFT
819B 3A72;              0229          BNZ LOOPD ...CONTINUE
819D 2F2F2F2F;          0230          DEC F;DEC F;DEC F;DEC F ...ELSE
81A1 2F2F2F2F;          0231          DEC F;DEC F;DEC F;DEC F ...FIX POINTER
81A5 D5;                0232          SEP 5 ...AND RETURN
81A6 ;                  0233          ...
81A6 ;                  0234          ...
81A6 ;                  0235          ...
81A6 ;                  0236          ...REGDIS SUBROUTINE
81A6 ;                  0237          ...DISPLAYS HEX CONTENT OF RA AND RB
81A6 ;                  0238          ...IN LED DISPLAY
81A6 ;                  0239          ...
81A6 ;                  0240          ...
81A6 F88CBF;            0241 REGDIS:  LDI A.1(DIGITS);PHI F ...SETUP
81A9 F800AF;            0242          LDI A.0(DIGITS);PLO F ...POINTER
81AC 9AF6F6F6F65F;      0243          GHI A;SHR;SHR;SHR;SHR;STR F       ..DIGIT 1
81B2 1F9AFA0F5F;        0244          INC F;GHI A;ANI #0F;STR F         ..DIGIT 2
```

```
81B7 1F8AF6F6F6F65F;0245          INC F;GLO A;SHR;SHR;SHR;SHR;STR F..DIGIT 3
81BE 1F8AFA0F5F;    0246          INC F;GLO A;ANI #0F;STR F        ..DIGIT 4
81C3 1F9BF6F6F6F65F;0247          INC F;GHI B;SHR;SHR;SHR;SHR;STR F..DIGIT 5
81CA 1F9BFA0F5F;    0248          INC F;GHI B;ANI #0F;STR F        ..DIGIT 6
81CF 1F8BF6F6F6F65F;0249          INC F;GLO B;SHR;SHR;SHR;SHR;STR F..DIGIT 7
81D6 1F8BFA0F5F;    0250          INC F;GLO B;ANI #0F;STR F        ..DIGIT 8
81DB F800AF;        0251          LDI A.0(DIGITS);PLO F ...RESET POINTER
81DE D4816C;        0252          SEP 4;.A(LEDD) ...REFRESH
81E1 D5;            0253          SEP 5 ...RETURN
81E2 ;              0254          ...
81E2 ;              0255          ...
81E2 ;              0256          ...
81E2 ;              0257          ...DUMMY SAVE AREA
81E2 ;              0258          ...MAPPED INTO #8CXX
81E2 ;              0259          ...
81E2 ;              0260          ...
81E2 ;              0261          PAGE
8200 ;              0262          DIGITS=#8C00
8200 ;              0263          STACK=#8C1F
8200 ;              0264          ...
8200 ;              0265          ...MOVE TO BOTTOM
8200 ;              0266          ...
8200 ;              0267          ORG #81E3
81E3 ;              0268          ...
81E3 ;              0269          ...
81E3 ;              0270          ...
81E3 ;              0271          ...CALL SUBROUTINE
81E3 ;              0272          ...CALLS SUBROUTINES IN COSMAC
81E3 ;              0273          ...USES R2,R3,R4,R5,R6
81E3 ;              0274          ...AT EXIT X=2
81E3 ;              0275          ...
81E3 D3;            0276 EXITA:   SEP 3 ...BACK
81E4 E2;            0277 CALL:    SEX 2 ...POINT TO STACK
81E5 96;            0278          GHI 6 ...SAVE R6
81E6 73;            0279          STXD  ...INTO STACK
81E7 86;            0280          GLO 6 ...LOWER HALF
81E8 73;            0281          STXD  ...INTO STACK
81E9 93;            0282          GHI 3 ...PUT R3
81EA B6;            0283          PHI 6...INTO R6
81EB 83;            0284          GLO 3 ...BOTH HALVES
81EC A6;            0285          PLO 6
81ED 46;            0286          LDA 6 ...& PUT SUBROUTINE ADDRESS
81EE B3;            0287          PHI 3 ...INTO R3
81EF 46;            0288          LDA 6
81F0 A3;            0289          PLO 3
81F1 30E3;          0290          BR EXITA ...GO TO SUBROUTINE
81F3 ;              0291          ...
81F3 ;              0292          ...
81F3 ;              0293          ...
81F3 ;              0294          ...RETPGM SUBROUTINE
81F3 ;              0295          ...RETURNS FROM SUBROUTINES IN COSMAC
81F3 ;              0296          ...USES R2,R3,R4,R5
81F3 ;              0297
81F3 ;              0298          ...AT EXIT X=2
81F3 ;              0299          ...
81F3 D3;            0300 EXITR:   SEP 3 ...RETURN TO MAIN PROGRAM
81F4 96;            0301 RETPGM:  GHI 6 ...R6 -> R3
81F5 B3;            0302          PHI 3 ...1ST HALF
81F6 86;            0303          GLO 6
81F7 A3;            0304          PLO 3 ...2ND HALF
81F8 E2;            0305          SEX 2 ...POINT TO STACK
81F9 12;            0306          INC 2 ...RECOVER
81FA 72;            0307          LDXA ...R6
81FB A6;            0308          PLO 6 ...LOWER HALF
81FC F0;            0309          LDX
81FD B6;            0310          PHI 6 ...UPPER HALF
81FE 30F3;          0311          BR EXITR ...BACK
8200 ;              0312          END
```