

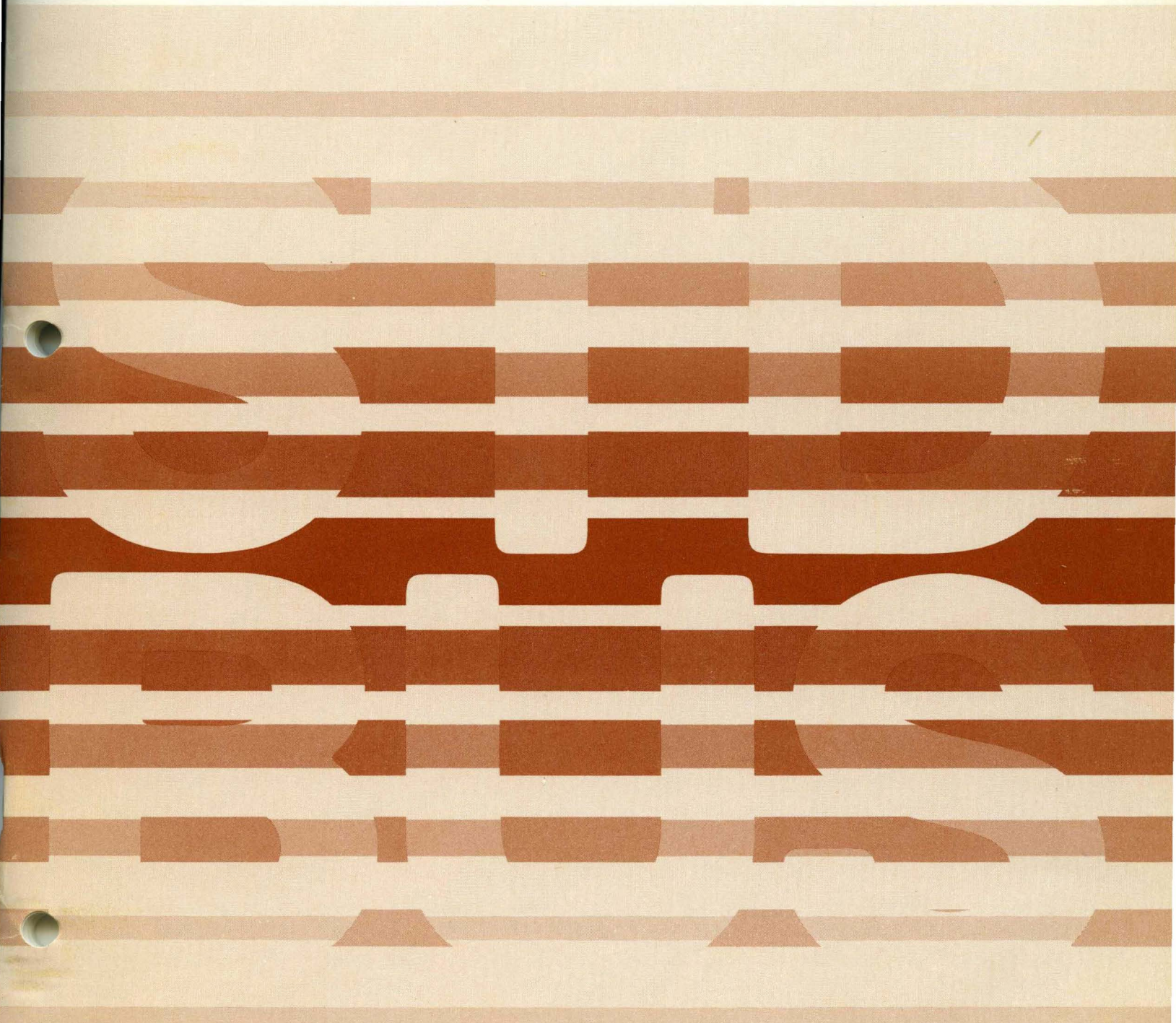


PRO-LOG
CORPORATION

STD 7000

MP4
Keyboard Monitor

USER'S MANUAL



NOTICE

The information in this document is provided for reference only. Pro-Log does not assume any liability arising out of the application or use of the information or products described herein.

This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of Pro-Log, nor the rights of others.

Printed in U.S.A. Copyright© 1981 by Pro-Log Corporation, Monterey, CA 93940. All rights reserved. However, any part of this document **may be reproduced** with Pro-Log Corporation cited as the source.

PROGRAM

MP-4 KEYBOARD MONITOR

THIS PROGRAM IS INTENDED FOR USE WITH THE PRO-LOG 7801 (8085) OR 7803 (Z80) CPU BOARDS AND THE 7303 KEYBOARD/DISPLAY CARD. IT PROVIDES THE USER WITH A MONITOR AND DEBUGGING PROGRAM THAT RESIDES IN A 2716 EPROM, FROM ADDRESS 0000 HEX TO 07FF HEX.

THE PROGRAM ALSO PROVIDES THE USER WITH SEVERAL INDEPENDENTLY ACCESSIBLE PROGRAM MODULES (SUBROUTINES) THAT CAN BE CALLED BY THE USER OR MOVED INTO HIS PROGRAMS. THESE INCLUDE MODULES TO ALLOW HIM TO FULLY EXERCISE THE 7303 KEYBOARD AND ALPHA-NUMERIC DISPLAYS.

IN ADDITION, THE CAPABILITY TO TRANSFER DATA FROM A PRO-LOG PROM PROGRAMMER TO THE USER'S SYSTEM, OR VICE VERSA, IS PROVIDED WHEN UTILIZED IN CONJUNCTION WITH A PIN-114 PARALLEL INTERFACE TO PROM PROGRAMMER.

| PRODUCT CONFIG. DOCUMENT | SYSTEM DESIGNATION | | BASIC MEMORY ADDRESSES IN HEX | | PORT ASSIGNMENT FOR I/O | |
|--------------------------|--|------------------|---|-------------------------------|--------------------------|--------------------|
| | DESIGNATION | CLOCK STATE TIME | ROM 1st/LAST ADR | RAM 1st/LAST ADR | FUNCTION | PORT BIT |
| 105680 | PS-1B (8085): 7801, 7303, MP-4 | 0.32 usec. | 0000-07FF (0800-1FFF)* | 2000-23FF (2400-2FFF)* | Switch S1 Switch S2 | IN D0-6 IN D0-7 |
| 105685 | PS-3B (Z80): 7803, 7303, MP-4 and | 0.40 usec. | *Empty sockets. User may add to CPU by installing ROMs and RAMs. Additional ROM and RAM can be added with 7701 or 7702 memory boards. | | Keyboard Inputs | IN D0-0 thru D0-5 |
| | | Keyboard Enables | | | OUT D0-0 thru D0-3 | |
| | | Display Data | | | OUT D0 all bits | |
| 105675 | PIN 114: 7505, 7140, RC50-6 | | | | Display Control | OUT D1-0 thru D3 |
| | | | | | Programmer Data | IN 52 all bits |
| | | | | | Programmer Data | OUT 50 all bits |
| | | | | | Programmer Handshake in | IN 51-0 thru 51-2 |
| | | | | | Programmer Handshake out | OUT 51-5 thru 51-7 |



TABLE OF CONTENTS

| | PAGE |
|--|------|
| 1. GENERAL | 5 |
| 2. SYSTEM REQUIREMENTS | 5 |
| 3. MONITOR DESCRIPTION | 6 |
| 4. MONITOR OPERATION | 7 |
| 5. MONITOR/PROM PROGRAMMER INTERFACE | 26 |
| 6. MONITOR USE WITH M900B OR M980 PROM PROGRAMMER | 27 |
| 7. MONITOR USE WITH M824 OR M825 MICROPROCESSOR SYSTEM ANALYZER | 27 |
| 8. MONITOR PROGRAM LISTING | |
| A. FORWARD | 29 |
| B. MONITOR CONTROL PROGRAM FLOWCHART | 30 |
| C. MONITOR PROGRAM MEMORY MAP | 33 |
| D. RAM VARIABLE MEMORY MAP | 34 |
| E. PROGRAM LISTING | 35 |
| 9. APPLICATION SOFTWARE MODULE DESCRIPTIONS | 67 |



MP-4 KEYBOARD MONITOR PROGRAM

General

The MP-4 Keyboard Monitor Program is intended to be a program development aid and program debugging tool for the STD BUS user. It is designed to be used with the PRO-LOG 7303 Keyboard/Display Card as the I/O device and a 7801 (8085) or 7803 (Z80) CPU Card. It is fully compatible and useable with the PRO-LOG M900, M900B, M920 and M980 PROM Programmers as well as the M824 and M825 System Analyzers. In addition to providing the user with several useful editing and debugging functions, there are many resident, independently accessible program modules (subroutines) that may be called by the user or moved into his programs. These include routines to fully exercise the 7303 alpha-numeric displays and keyboard, a variable time delay module, and a Hex to ASCII conversion routine.

The Monitor can also double as a diagnostic tool for the engineer or technician. When switches S1 and S2 on the 7303 card are closed, or if the 7303 is not in place, program control is transferred to address 0800 Hex (ROM 1) and all monitor functions are bypassed. If the Monitor PROM is retained in the finished product, and if the user's permanent main program starts at address 0800 Hex, the Monitor program will be transparent until the 7303 is inserted into the rack and either or both switches are not closed. At that time, all Monitor functions will become available to allow debugging in place.

Monitor resident software is provided to allow the user to transfer data from the microprocessor system to the PROM programmer and vice versa. This requires a PRO-LOG PROM Programmer equipped with the 9104, 9114, or 9814 Parallel Interface Option, and a PIN-114 Parallel Interface Option for STD BUS Prototyping Systems.

System Requirements

The Monitor program resides in one Intel 2716 EPROM or equivalent and occupies addresses 0000 through 07FF Hex. Program parameters and stack operations use addresses 2000 through 204F Hex. User programs must start at locations which are not part of these address fields. In addition, all but the first two restart locations are available to the user for vectored interrupt. However, the Non-Maskable locations 0024-0026 Hex (8085) and 0066-0068 Hex (Z80) are utilized in the program to cause the contents of the internal registers to be displayed in the 7303 displays upon the occurrence of a Non-Maskable Interrupt (NMI). This interrupt can be generated coincident with any program address by the M824 or M825 System Analyzers.

Also, restart location 0038 Hex contains an unconditional jump to location 2020 Hex in RAM. The user can program an unconditional jump into locations 2020-2022 to vector his Restart 7 interrupt to any location in memory.

With all RAM socketed on the 7801 or 7803 CPU Card, user programs and data may reside in address locations 0800-1FFF (ROM) or 2050-2FFF (RAM). Additional memory may be added (4000-FFFF) with the addition of PRO-LOG 7701 or 7702 memory boards.

Monitor Description

The Monitor program utilizes the eight alpha-numeric displays and twenty-five keys provided on the 7303 Keyboard/Display Card. The keyboard is divided into a Hex key pad (keys 0-F), a system RESET key, and 8 function keys. Each key has a removable, transparent cap. Labels are provided with which the user may label each of the function keys (keys 10-17) by removing the cap, inserting the label, and replacing the cap. A keyboard layout diagram is shown in Figure 1. to facilitate labeling.

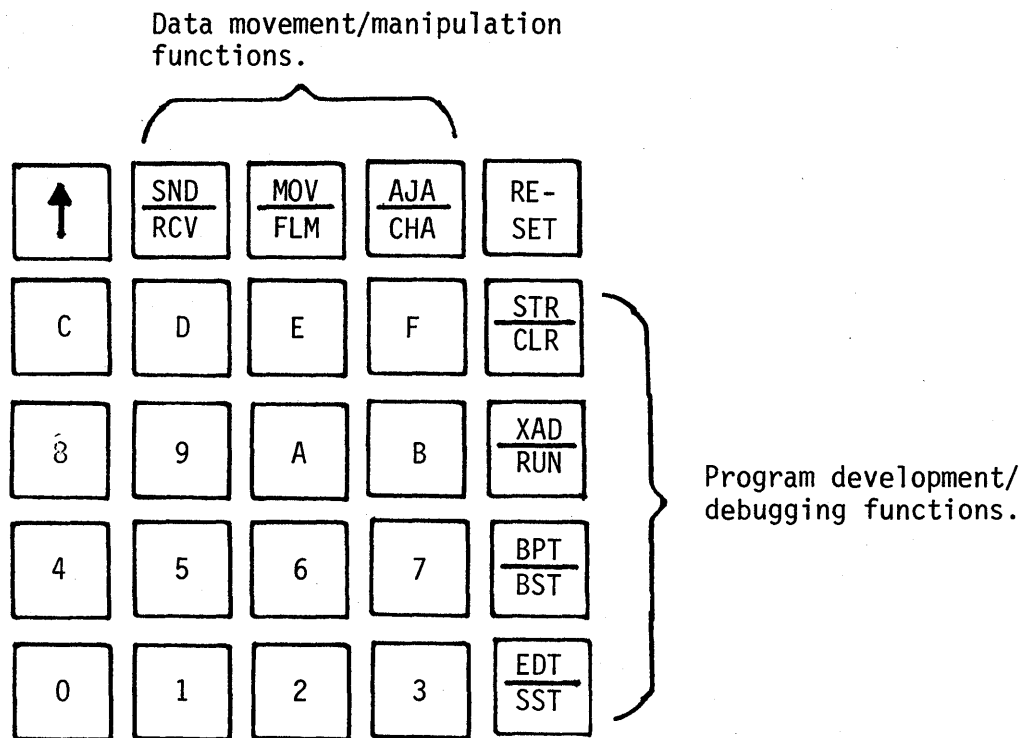


Figure 1. 7303 Keyboard Labeling.

Each of seven keys provides the capability of implementing the functions or commands so labeled, one upper and one lower function per key. To select an upper function, the user first depresses the shift key (↑), then the desired key. To select a lower function the user merely depresses the desired key. Operation is similar to that of many conventional hand-held calculators.

The function keys are grouped by application. The horizontal row of function keys provide several data movement, manipulation and memory conditioning capabilities. The vertical column of function keys provide program development and debugging capabilities.

Upon power-on or pushbutton reset, the prompt word "READY?" is displayed in the 7303 alpha-numeric displays. The subsequent selection of any command will result in the display being blanked, followed by the name of the command for approximately 3/4 second. This verifies to the user that he has selected the desired function or command. Throughout several functions, the prompt word "BUSY" is displayed while the processor is completing the execution of that function. Several other prompt words are utilized by individual routines to enhance user entry or information.

All Monitor routines are exited by depressing the "RESET" key, or upon completion of the function, in which case control is returned to the Monitor and "READY?" is again displayed.

Monitor Operation

To select a specific command, depress the appropriate combination of shift (↑) and function key. The display will blank and then the name of the function will appear centered in the display for 3/4 second. SST (Single Step), BST (Back Step), and CLR (Clear Entry) are not commands.

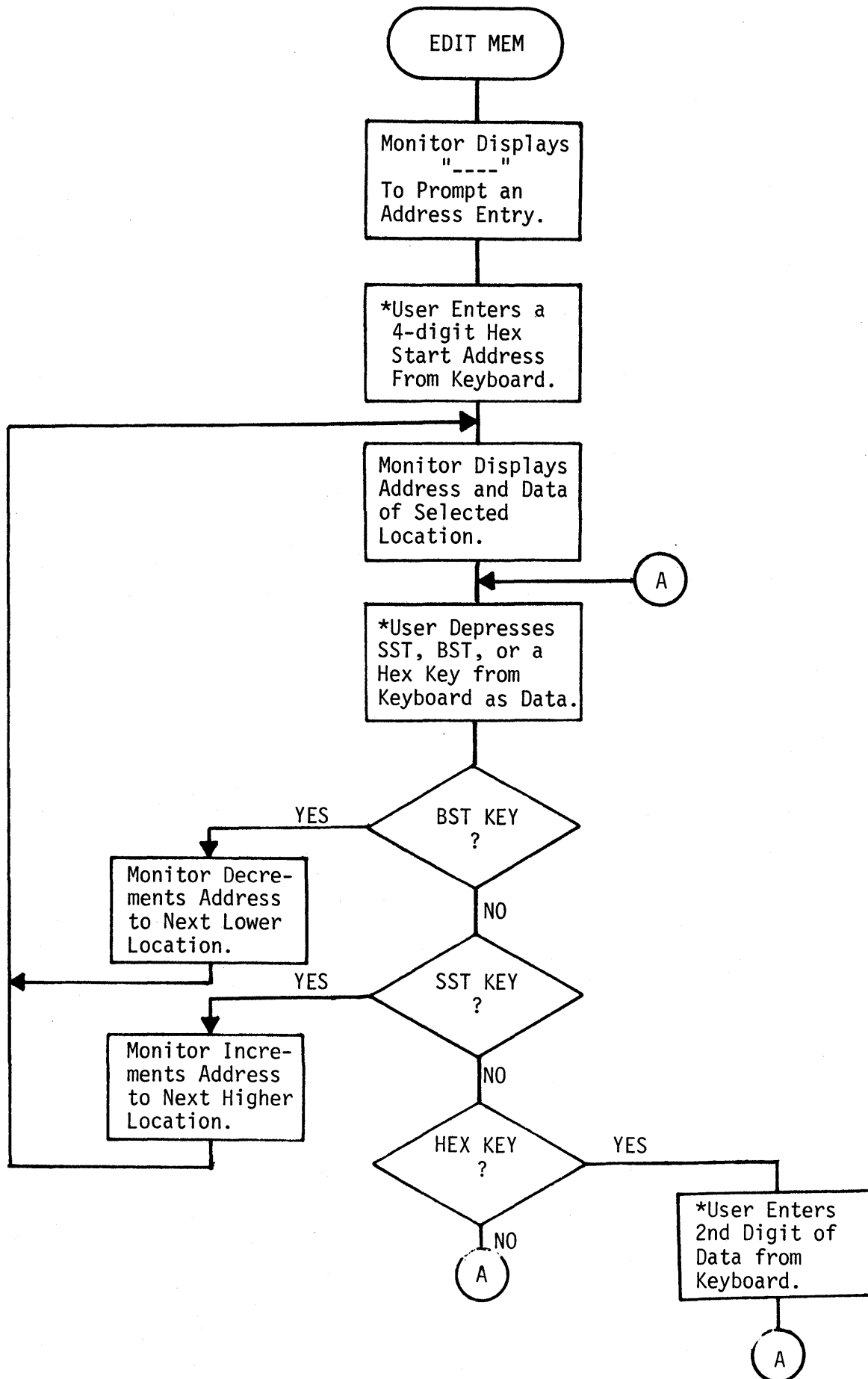
All command functions except STR and RUN require the user to enter one or more 4-digit Hex addresses. When an address entry is required, the prompt "____" will be centered in the display. Enter the address, most significant digit first, from the hex keypad. As each digit is entered, it will be displayed, thereby replacing the prompt characters. After entering all 4 digits, depress SST. The address will disappear and the function will continue execution.

Several commands (EDT, STR, FLM) may require the user to enter 2-digit Hex data. As before, enter the data from the Hex keypad, most significant digit first. The data will appear in the display as it is entered. At the completion of entry, depress SST and the function will continue execution.

Should the user enter an address or data digit in error, it may be corrected by depressing the CLR key. Each time the CLR key is depressed, the next digit to the left will blank starting with the last entry. The digit(s) may then be re-entered. All corrections must be made before SST is depressed.

During the course of the execution of the commands RCV, MOV and FLM, data is written into RAM. If the RAM location fails to accept the data, or ROM is written to, or the address written to does not exist in the system, then the execution of the function will halt and the address of the bad location and its data will be displayed. To abort further execution, depress RESET. To continue execution, depress SST. Execution will continue until completion, or until another bad location is encountered.

A description of each command, its suggested application, and instructions for its utilization is included in the following pages.





EDIT data in memory.

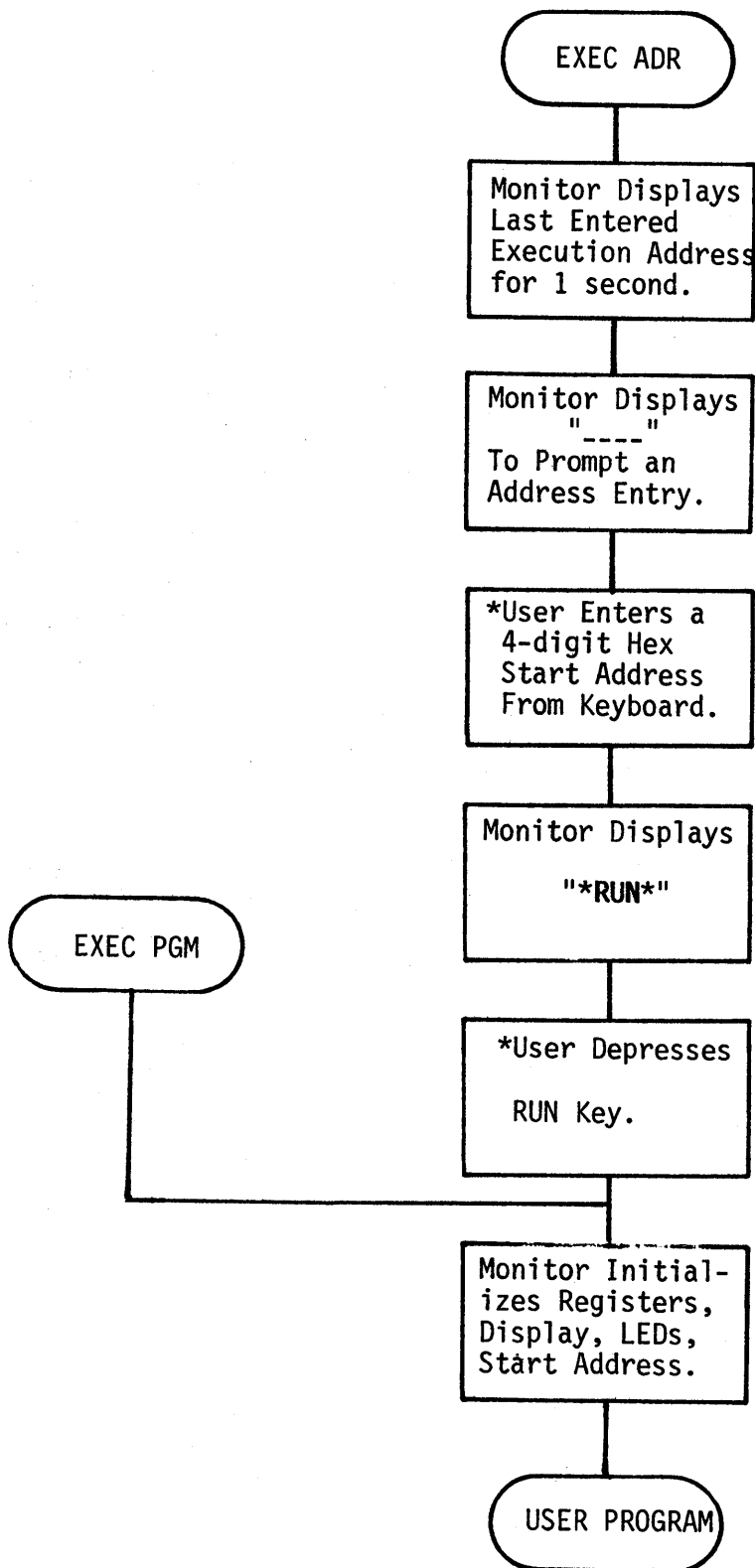
This command allows the user to display a desired address and resident data in the 7303 alpha-numeric displays, and modify the data, if desired. After entering the desired starting address, the address and data will be displayed as 4 Hex digits of address and 2 Hex digits of data separated by 2 spaces (AAAA DD). The user may modify the data by merely depressing 2 Hex keys on the keyboard. The data will be displayed as it is entered, and will replace the original data. SST must be depressed to enter the data into memory, following data entry.

If the user desires not to modify data, he may advance to the next higher location by depressing SST. At that time the address will be incremented and the address and data of the next location will be displayed. The user may also depress BST, thereby decrementing the address to the next lower location and display its address and data.

To exit this command function, depress RESET.

The purpose of this function is to allow the user to enter program and data into RAM for subsequent execution and debugging.

Note: If an "E" appears to the immediate left of the data, following a data entry, it informs the user that the data was not accepted, i.e., the RAM location is bad, or that area of memory is ROM or non-existent in the system. To continue, depress SST to increment to the next location. To abort, depress RESET.





Enter EXECUTION ADDRESS into memory.

This command allows the user to enter a program execution starting address into memory and is intended to allow the user to exercise programs entered into RAM or ROM.

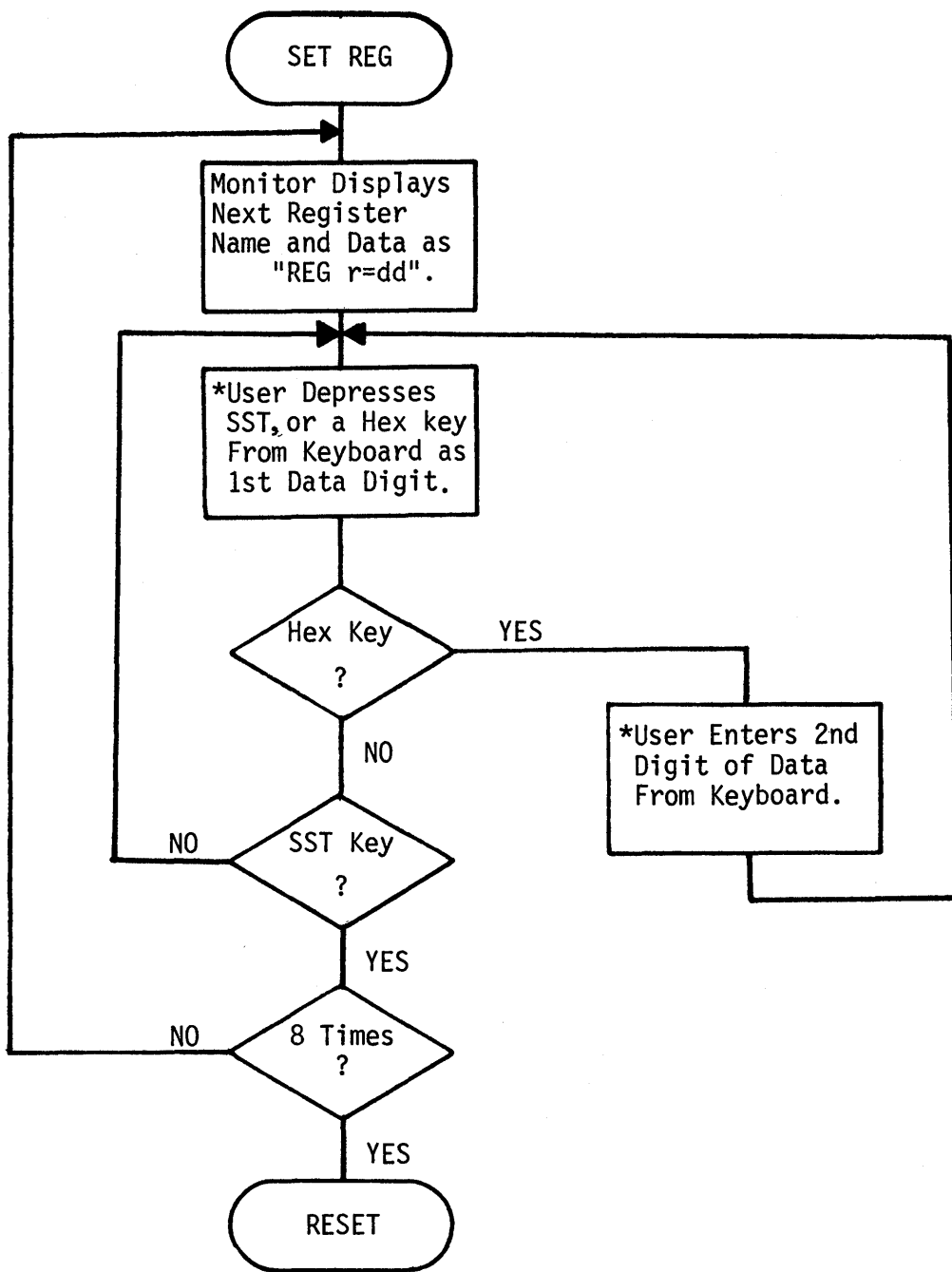
Upon selection of this command, the Monitor will display the last execution address entered in the 7303 alpha-numeric displays for approximately one second. The display will then blank and the prompt "----" will be displayed to indicate to the user that a new 4-digit Hex execution start address must be entered. The user may retain the current address by depressing RESET. Otherwise, the user enters the new address and depresses SST.

After address entry, the display is blanked and then "*RUN*" is displayed. The program may now be executed by depressing RUN. The execution address will be retained for subsequent program executions until it is changed by this command.



RUN the program at the previously entered execution address.

This command will allow the user to execute a program in RAM or ROM, repeatedly. The execution start address is the last address entered via the XAD command. Upon selection of this command, the Monitor blanks the display and LED's, disables the keyboard, loads the registers with the values entered via the last STR command, retrieves the execution address from memory and jumps to that address.





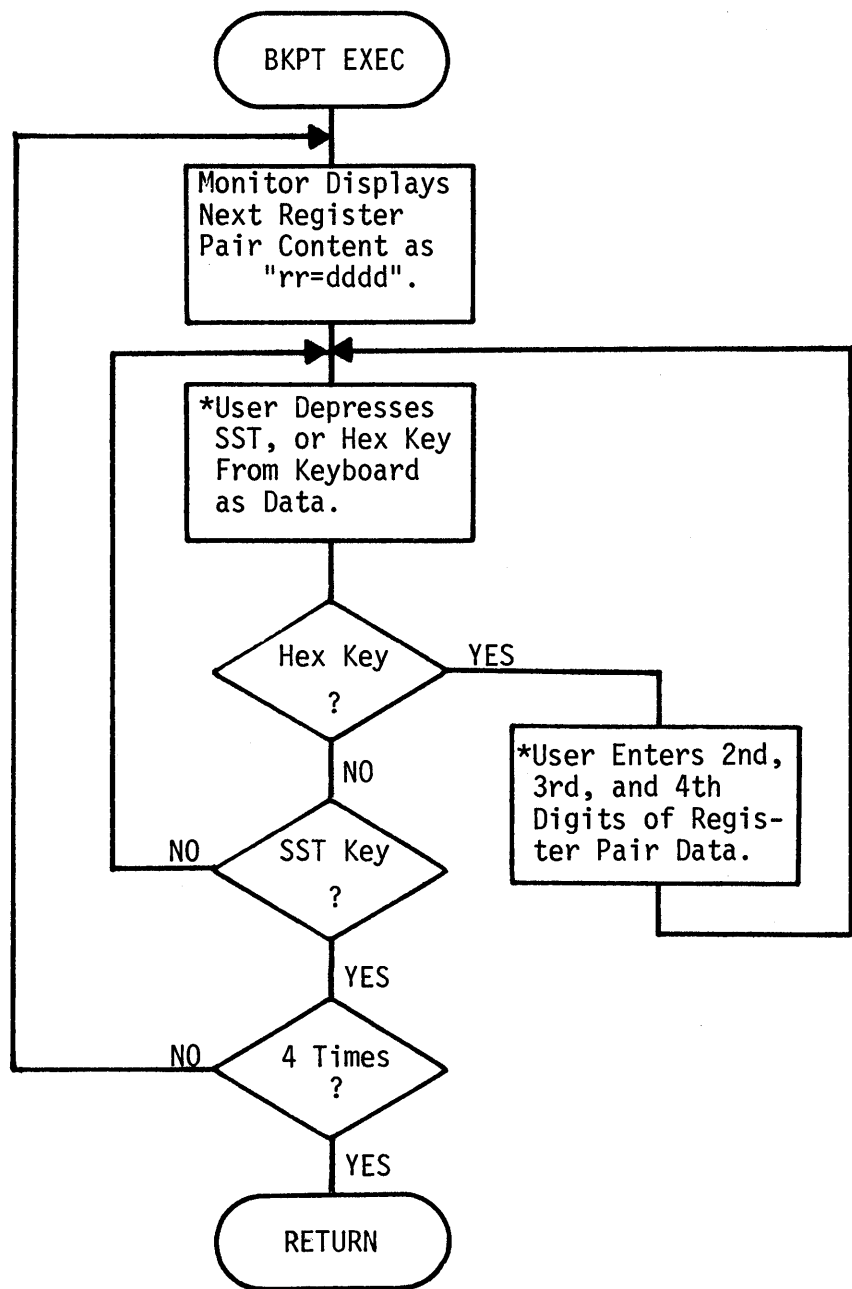
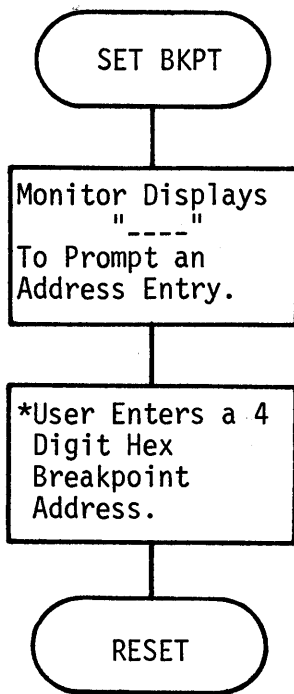
Pre-SET Registers for program execution.

This command allows the user to enter a set of initial register values into memory. Each time the user depresses RUN to commence execution of a program, or a portion of a program, these values will be loaded into the internal registers before jumping to that program. In this way, program execution can be started with the processor in a known state. The entered values will remain the same until changed by subsequent repetition of this command.

Upon selection of this command function, the Monitor will display "REG H=dd", where "dd" is the last 2-digit Hex data entered for the initialization of the H register. To enter new data, depress the desired 2 Hex keys, then depress the SST key. To leave the data unchanged, depress the SST key directly. Each time the SST key is depressed, the next register name and content will be displayed. Depressing the SST key after the contents of the flag register, F, are displayed will return control to the Monitor, and "READY?" will be displayed.

Register name and contents are displayed in the following order:

H, L, D, E, B, C, A, F.





Set Breakpoint at user entered address.

This command allows the user to enter a breakpoint at any desired address in a program that resides in RAM. When the breakpoint is encountered during program execution, the execution will terminate and the internal register contents will be displayed.

Upon selection of this command function, the Monitor will display "----" as an address prompt. Enter a 4-digit Hex address and depress the SST key. The Monitor will replace the original data at the designated address with a Restart 2 (D7 Hex) and save the original data in memory. Upon depression of the SST key, control will be returned to the Monitor and "READY?" will again be displayed. Care must be taken to insure that the breakpoint only be set at an address that contains the first byte of an instruction. If this is not done, the restart instruction will be misinterpreted as operand data.

Breakpoint Execution.

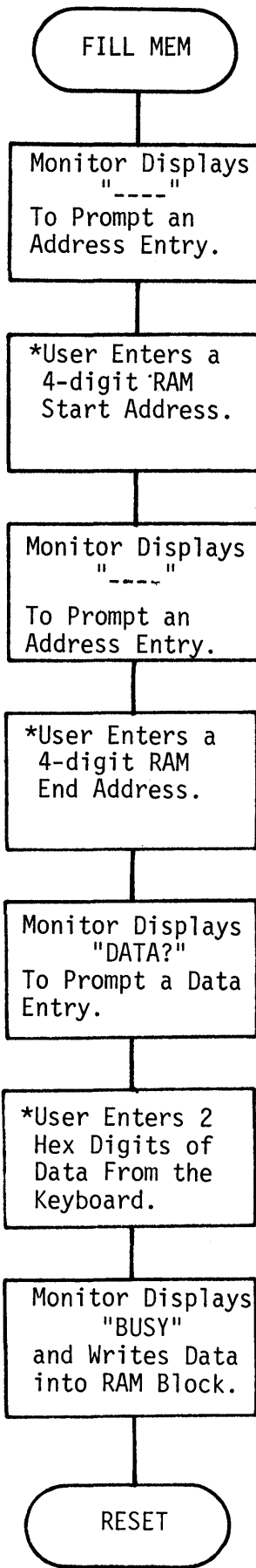
Upon encountering a breakpoint, program execution will terminate and the register contents will be displayed, starting with "HL=dddd", where "dddd" is the 4-digit Hex data contained in the HL register pair. At this point, the user may change the data by entering 4 Hex digits of new data from the keyboard and depressing SST. The data will remain unaltered by directly depressing SST, at which time the next register pair name and data will be displayed in the same manner. Data is displayed in the following order:

"HL=dddd", "DE=dddd", "BC=dddd", and "AF=dddd".

After the AF register pair data is displayed, execution can be resumed by depressing SST. The Monitor will replace the restart instruction with the original data and execution will resume at the breakpoint address. To leave the breakpoint set for subsequent executions, depress RESET instead of SST.

Non-Maskable Interrupt Service.

Upon the occurrence of a non-maskable interrupt, the register pair contents will be displayed and can be altered as described above. However, the Monitor will not replace the breakpoint with original data.





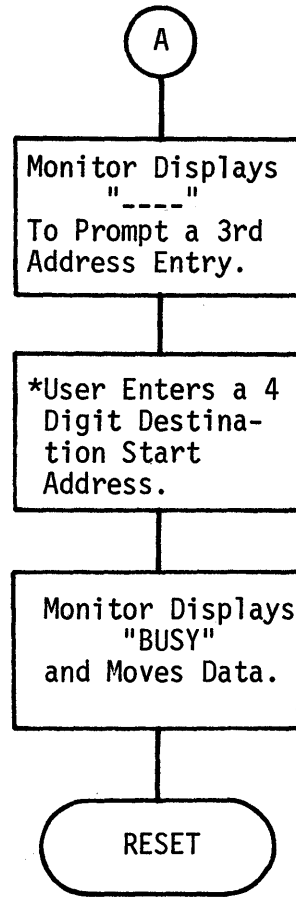
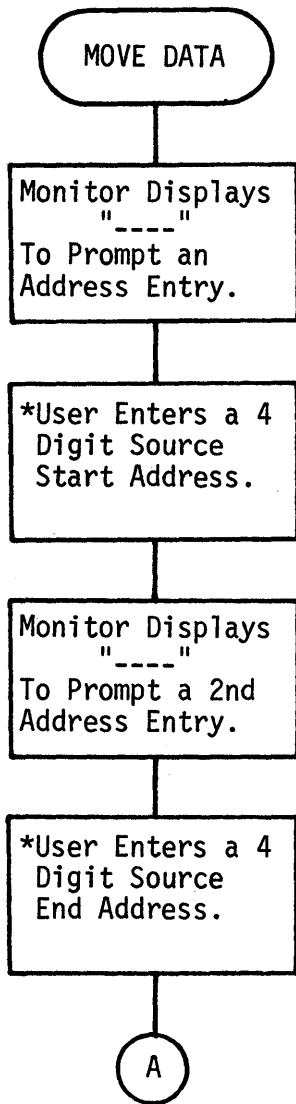
Fill a block of memory with a single data value.

This command allows the user to write the same data value into all RAM locations that lie between the user-entered address limits, inclusively. The data can be any user-entered 2-digit Hex value.

Upon selection of this command function, the Monitor will prompt an address entry by displaying "----". Enter the starting address to be written to. The Monitor will prompt another address entry. Enter the ending address to be written to. The Monitor will now display "DATA?" to prompt a data entry. Enter a 2-digit Hex value from the keyboard. The data will be displayed as it is entered. Upon completion of the data entry, the Monitor will display "BUSY" while it writes the entered data into all locations within the address field. It then exits to RESET.

This command may also be used to test the integrity of a block of RAM. Should any RAM location fail to accept the desired data, execution will halt and the bad location address and data will be displayed. Depressing SST will cause execution of the function to continue (See Monitor Operation).

It is suggested to the user that prior to entering a program into RAM for development or debugging, the memory be filled with the data that corresponds to the unprogrammed state of the PROM (00 or FF) in which it will eventually reside. Then, if the program is subsequently transferred to the PROM programmer via the SND command, those areas of memory not actually part of the program will remain unprogrammed in the PROM.



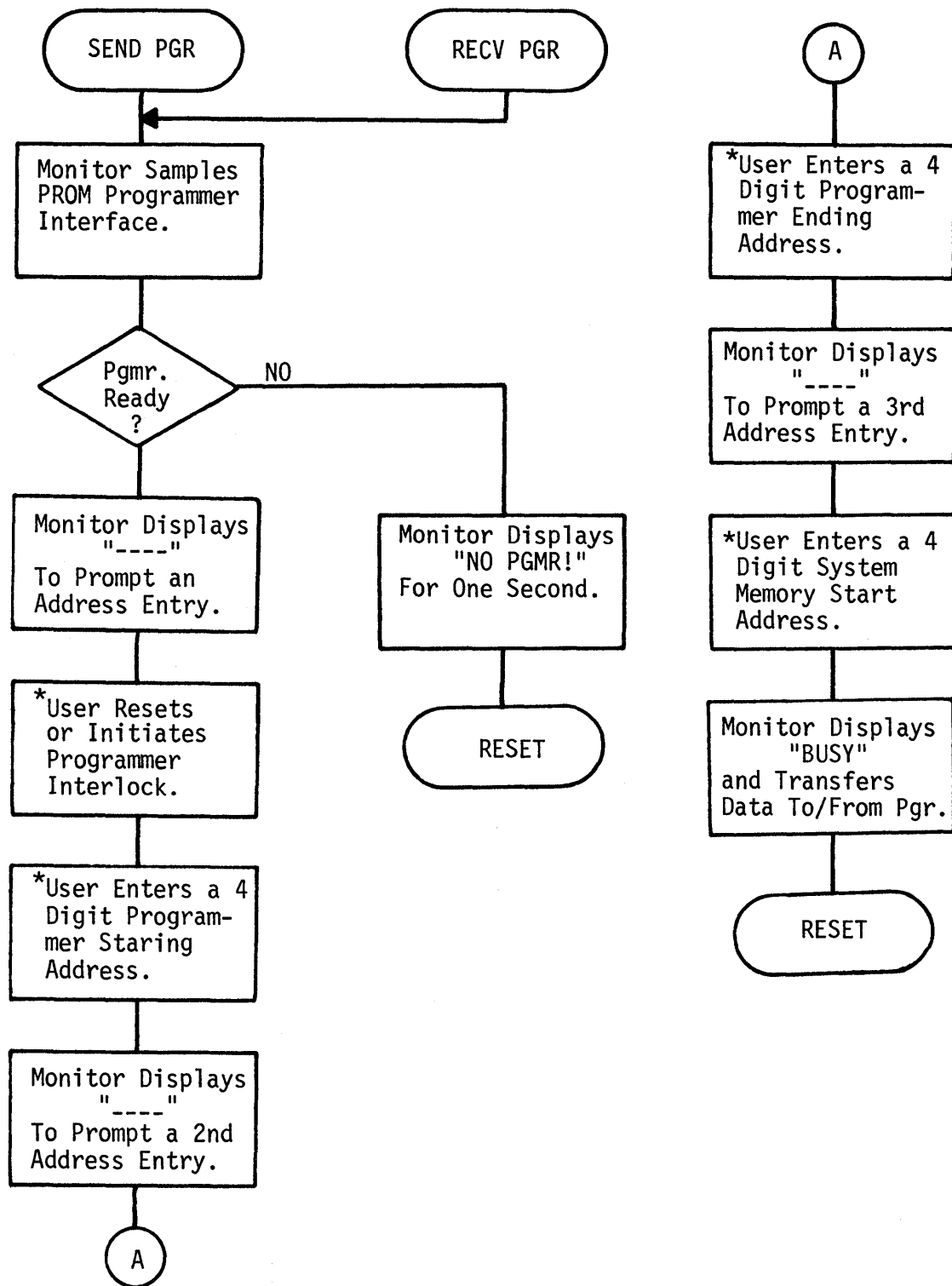


Move a block of data to another location in RAM.

This command allows the user to move any size block of data from ROM or RAM to another area of RAM. Both backward and forward moves are allowed. As the Monitor moves the data, it clears the source area by writing 00 Hex (NOP) into each source location.

Upon selection of this command function, the Monitor displays "----" to prompt an address entry. Enter the 4-digit Hex source starting address. Then the Monitor will prompt a second address entry. Enter the source ending address. The Monitor again prompts a third address entry. Enter the destination starting address. At the completion of the last entry, the Monitor will display "BUSY", and begin moving data to the new locations. If any location fails to accept the data, execution will halt and the bad location address and data will be displayed. To resume execution, depress SST (see "Monitor Operation"). At the completion of the move, the Monitor will exit to RESET and display "READY?".

This command can be used to move any Monitor software module into the user's program. The source start and end addresses, and the destination address are saved in memory for subsequent and repetitive address adjustment operations that utilize the AJA command.



| |
|-----|
| SND |
|-----|

Send a block of data to the PROM programmer.

This command allows the user to transfer a block of memory data from either ROM or RAM to any PRO-LOG PROM Programmer equipped with a Parallel Interface Option. To use this command, the PIN-114 Prototyping System Interface to PROM Programmer must be installed to connect the microprocessor system to the programmer. When using an M900 or M920 programmer, only personality modules for PROMs with a 512 word capacity, or greater, may be used. This restriction does not exist when using the buffer equipped M900B or M980 units.

Upon selection of this command function, the Monitor determines if the PIN-114 is installed, if the programmer is connected, and if the programmer is turned on. If any of these conditions are not true, the Monitor displays "NO PGMR!" for one second, then exits to RESET and displays "READY?".

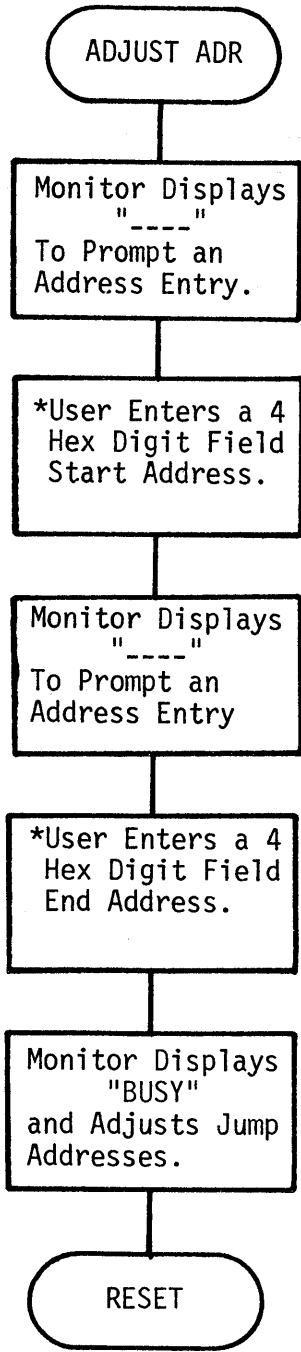
If the programmer is on-line, depress the RESET key on the M900 or M900B programmer to initiate the interlock. If an M920 is used, turn it off, then on again. The M980 should not be RESET - the user must select Parallel Interface Option from the keyboard, initially. All but the M980 must be reset before each subsequent data transfer.

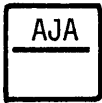
Next, the Monitor will display "____" as an address prompt. Enter a 4-digit Hex programmer buffer or PROM starting address. The Monitor will prompt a subsequent address entry. Enter the programmer buffer or PROM ending address. The Monitor will prompt a third address entry. Enter the microprocessor system starting address. The Monitor will then display "BUSY" and transfer the block of data to the PROM programmer. Upon completion of the transfer, the Monitor will exit to RESET and display "READY?", after dropping the interlock signal.

| |
|-----|
| RCV |
|-----|

Receive a block of data from the PROM programmer and store it in RAM.

This command allows the user to transfer a block of data from the PROM programmer PROM or buffer to system RAM. All restrictions and interface requirements specified for the SND command apply, as does the operating procedure. A further restriction exists in that the data must be transferred to valid RAM locations. Should a RAM location fail to accept the data, or if the RAM capacity of the system is exceeded, or if transfer is to ROM, then execution will stop and the address and data of the bad location will be displayed. Transfer may be resumed by depressing SST (see "Monitor Operation").





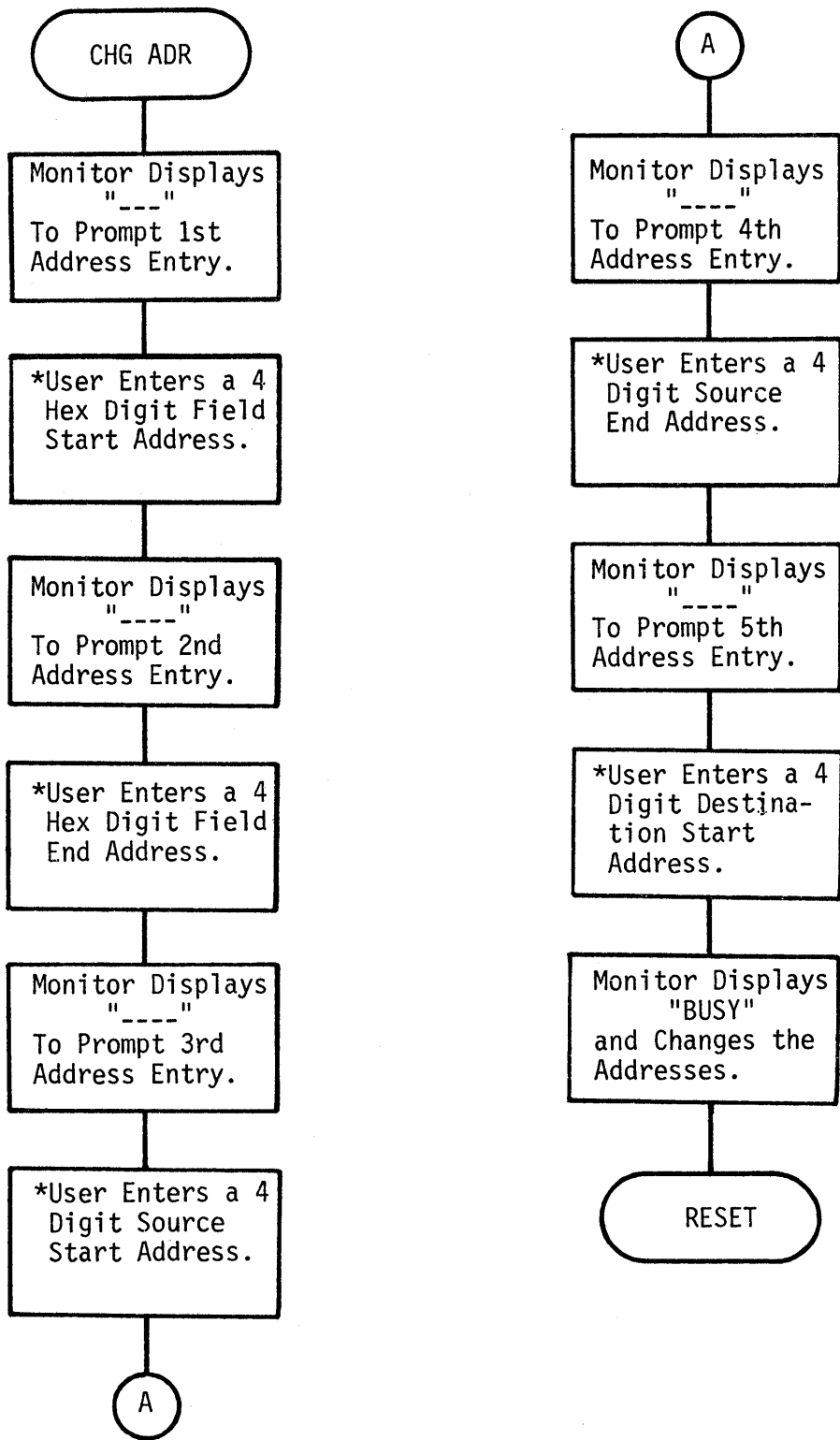
Adjust the addresses contained in 3-byte instructions after a MOVE

Whenever a program or subroutine is moved from one area of memory to another, it will not execute correctly unless the addresses contained in all 3-byte Jump, Load, and Store instructions are changed to reflect the program's new location. This includes addresses contained within the moved program, and addresses in other areas of memory that may refer to the moved program's original location.

This command allows the user to adjust the move-affected addresses within any field of data. It may be used over as many fields as desired, and over any size data field. All adjustments made to addresses are based on the parameters of the last MOV operation. i.e., source start, source end, and destination start addresses. The Monitor determines the displacement (source minus destination), or how far the data was moved. It then searches for 3-byte instructions. The operand (2nd and 3rd bytes) of each instruction found is examined. If the operand value lies between the source start and end addresses, inclusive, the displacement is added or-subtracted from the operand.

Upon selection of this command function, the Monitor displays "----" as an address prompt. Enter the 4 Hex digit starting address of the field of data to be adjusted. The Monitor will then prompt a second address entry. Enter the 4 digit ending address of the data field. The Monitor will then display "BUSY" as it searches for 3-byte instructions and adjust operands. Upon completion, the Monitor exits to RESET and displays "READY?".

NOTE: This command operates only on the subset of instructions common to both 8085 and Z80. If Z80 special instructions are used, care must be taken to insure they are not altered by this routine.





Change all address bytes contained in 3-byte instructions.

This command allows the user to change all page and line addresses contained in all 3-byte instruction operands within any area of RAM, by adding or subtracting an offset to those operands. The offset, or displacement, is computed by the Monitor from the source start and destination start addresses entered by the user. If the user has developed a program in RAM, and desires to relocate it to ROM, he can utilize this command to change the addresses of all Jump, Load and Store instructions that would be affected, then transfer the altered program to the PROM programmer. The converse is also true, in that the user can transfer the contents of a PROM to RAM, change all dependent addresses, and execute the program.

Upon selection of this command function, the Monitor will prompt the entry of five (5) addresses by displaying "----". Enter the following 4 Hex digit addresses in the listed order:

- 1) The starting address of the data whose operands are to be altered.
- 2) The ending address of the data whose operands are to be altered.
- 3) The source starting address, i.e. the starting address of the area in which the program was intended to operate.
- 4) The source end address.
- 5) The destination start address, i.e., the starting address of the area in which the user desires the program to execute.

Upon completion of the last address entry, the Monitor will change all addresses required, while displaying "BUSY". When finished, the Monitor will exit to RESET and display "READY?".

If the user wishes to change addresses in segments, in order to bypass data tables, the first area of data must be changed with the CHAD command, but subsequent areas may be changed with the AJA command. The source start, source end, and destination start address will be retained in memory until the next CHA operation.

*This command operates on those instructions that are common to the 8085 and Z80.
*If Z80 special instructions are used in the program, care must be taken to insure
*they are not altered by this function.

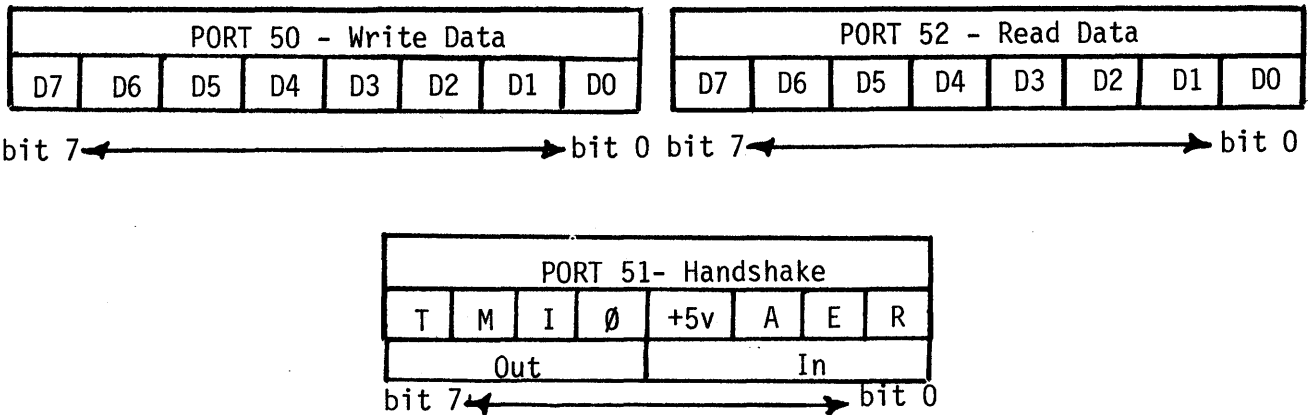
Monitor/PROM Programmer Interface

The Monitor program contains the necessary software to implement a direct parallel data interface with the M900, M900B, M920 and M980 PRO-LOG PROM Programmers. The M900 and M920 must be equipped with a 9104 Parallel Interface Option which is described in detail in Section 8 of the respective operating manuals. The M900B requires the 9114 Parallel Interface Option which is described in Section 11 of the operating manual. The M980 provides the 9814 Parallel Interface as described in Section 19 of the operating Manual.

In addition, the PIN-114 Parallel Interface to PROM Programmer is required. This consists of a 7507 I/O Module Mounting Rack Interface Card, an RC50-6 Ribbon Cable, and the 7140 Parallel Interface Adapter. To effect the interface, the 7507 must be inserted into the STD system rack, the 7140 inserted into the 25-pin "D" connector of the PROM programmer, and the RC50-6 cable connected between them.

Data is transferred to and from the PROM programmer via the SND and RCV commands of the Monitor. The user has the choice of programming PROMs with debugged machine code directly from the system, or transferring ROM based programs/subroutines into the system for subsequent modification/debugging.

The following diagrams describe the port/bit assignments of the PIN-114:



All handshake lines are active when low, i.e., 1 = low, Ø = high.

Definitions:

| | | | | | |
|-------|---|----------------|------|---|---------------------------------|
| to | { | T = Transfer* | from | { | +5v = programmer power |
| pgmr. | | M = Mode* | | | A = Address* |
| | | I = Interlock* | | | E = Error (not used by Monitor) |
| | | Ø = constant | | | R = Response* |

Monitor Use with M900B or M980 PROM Programmer

The most serious drawback to a RAM based development tool is the volatility of the memory. All too often, RAM based programs are altered or destroyed by electrical noise, power fluctuations, or program bugs that overwrite the program.

The M900B and M980 PROM Programmers are equipped with CMOS RAM buffers that provide short to intermediate, power-off, data retention for several minutes to several days. These programmers may serve as a separate program storage medium that is totally immune to the electrical and software bugs that destroy programs.

In order to prevent the total or partial loss of RAM based programs, it is recommended that the user always transfer his program into the programmer buffer before execution, modification or other debugging. In that way, a copy of the last valid program can rapidly be reloaded into RAM on request, even after programmer and system power are lost. It also provides the user with storage for earlier versions of his program, prior to modifications. Should his modification not work, he can easily retrieve the earlier version and try a different solution.

Additionally, if the M900B or M980 is equipped with an appropriate interface option and is interfaced to other systems or I/O devices (CRT, TTY, computers, etc.), the programmer can serve as a temporary holding buffer. Data can be transferred from the interfaced device or system to the programmer, then transferred from the programmer to the Monitor system for debugging or modification.

Monitor Use with M824 or M825 Microprocessor System Analyzer

The user may enhance the execution, debugging and diagnostic capabilities of the Monitor by combining its application with the added hardware diagnostic capabilities of the PRO-LOG M824 (Z80) or M825 (8085) Microprocessor System Analyzer. The analyzer performs the following basic functions:

- a. Tracks the actual instruction sequence as the program executes, and displays desired address, control and data information when it occurs.
- b. Monitors control states and data passing between the processor and the system it controls and allows the user to correlate actions or reactions with program execution.
- c. Provides a multi-qualified trigger to conventional oscilloscope when analog measurements are desirable.

The analyzer allows the user to dynamically monitor his program, single step instruction-by-instruction or machine cycle-by-machine cycle through the program, stop the program execution at any address and machine cycle, capture data and stop the processor after any number of passes through an instruction sequence, and trigger a non-maskable interrupt upon the occurrence of any program address.

The primary debugging capabilities of the Monitor include the ability to execute any program sequence with known register conditions on entry, and the ability to set a RAM based breakpoint, at which time the register contents are displayed and can be altered and execution resumed.

When used in conjunction with the analyzer's ability to stop the processor at any address and after any number of passes through the program sequence, the Monitor will permit the user to execute and exercise only those RAM and ROM based routines in which he is interested. This will allow him to independently exercise external hardware items controlled by the processor system, to diagnose and troubleshoot hardware problems, and to debug the integration of the software and hardware.

Because the Monitor will display the contents of the internal registers upon the occurrence of a non-maskable interrupt, the analyzer's address-generated interrupt can be used to effect a breakpoint in ROM or RAM. The user merely sets the breakpoint address on the analyzer switches and arms the interrupt. When that address occurs on the address bus, the processor will be interrupted and its register contents displayed in the fashion discussed during the explanation of the BPT command.

If the 7303 Keyboard/Display Card is not in place, the non-maskable interrupt routine bypasses the register display routine. Instead the contents of the registers are pushed onto and pulled off of the stack and a return is executed. To see the register contents at any program location, place the analyzer in the STOP mode at location 0000 Hex. Set the desired address on the ADDRESS switches, set the CYCLE COUNT switch to M1, the INSTRUCTION COUNT switch to 10 and ADDRESS + INSTRUCTIONS, and depress the INTERRUPT switch. Now depress CLEAR. The processor will stop at address 057B Hex and display F5. Single stepping from that point, cycle-by-cycle, will allow the user to see the registers being stored on the stack. The data displayed will be as follows:

F5, (A), (F), C5, (B), (C), D5, (D), (E), E5, (H), (L), ...

where (r) means the contents of register r. (For a more in-depth discussion of the analyzer functions and capabilities, see the Users Manual for M824 (M825) System Analyzer.)

MONITOR PROGRAM LISTING

FORWARD

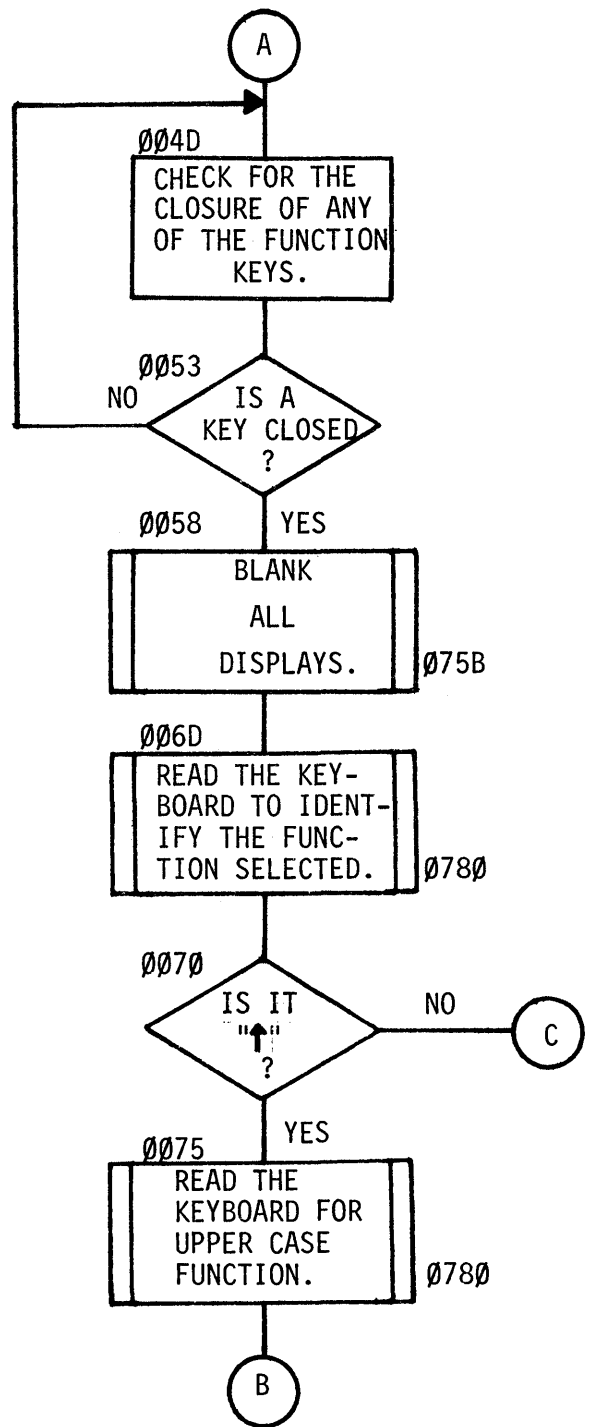
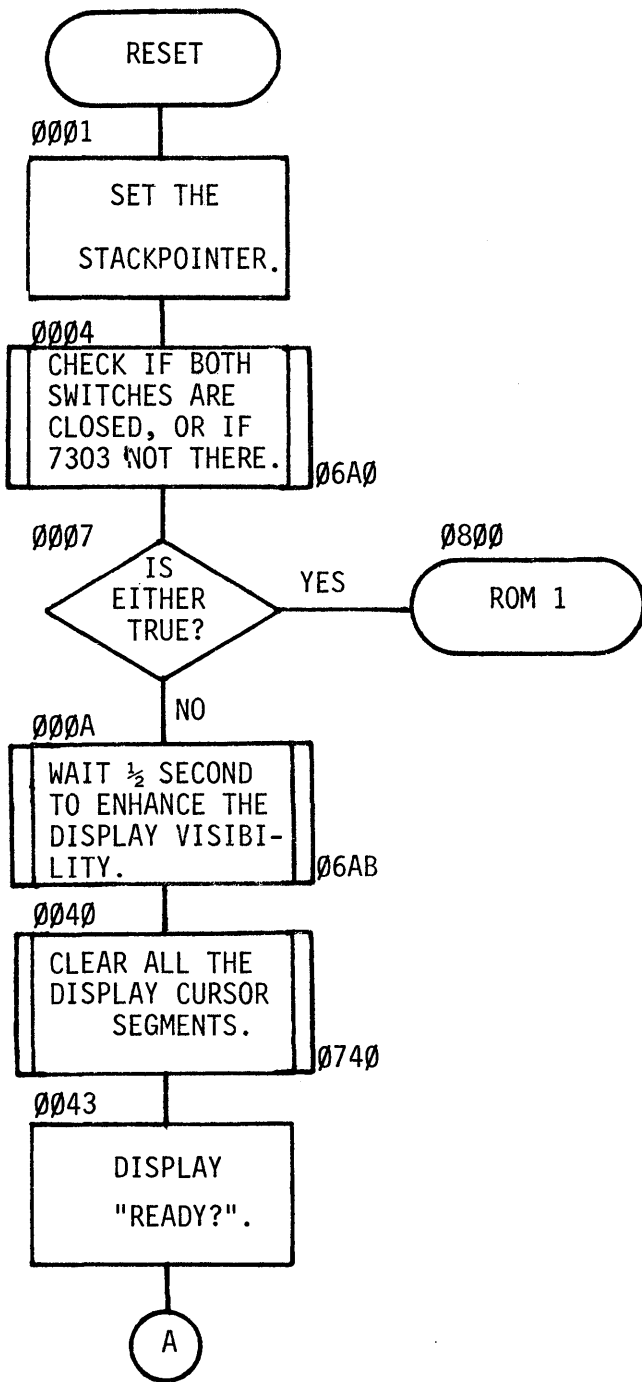
The following pages contain the Monitor program listing, main program flow charts, and pertinent memory maps. The source is written STD mnemonics, and utilizes only those instructions common to both the 8085 and Z80. An explanation of the mnemonics can be found on the PRO-LOG STD 8085 or Z80 PROCESSOR PROGRAMMING AIDS cards, in the USER MANUAL FOR M824 or M825 SYSTEM ANALYZER, and in the PRO-LOG MICROPROCESSOR USER'S GUIDE. The object or machine code listing is fully compatible for use with 8080A, 8085, and Z80 microprocessors. When provided by PRO-LOG, the program is contained in one Intel 2716 EPROM, or equivalent, and resides in locations 0000 to 07FF. In addition, the following conventions are used throughout the program listing:

- a. Labels enclosed by apprentheses () indicate subroutine entry points.
- b. Labels enclosed by brackets [] indicate Monitor command entry points.
- c. CST mnemonic implies location is a table data byte, vice instruction.

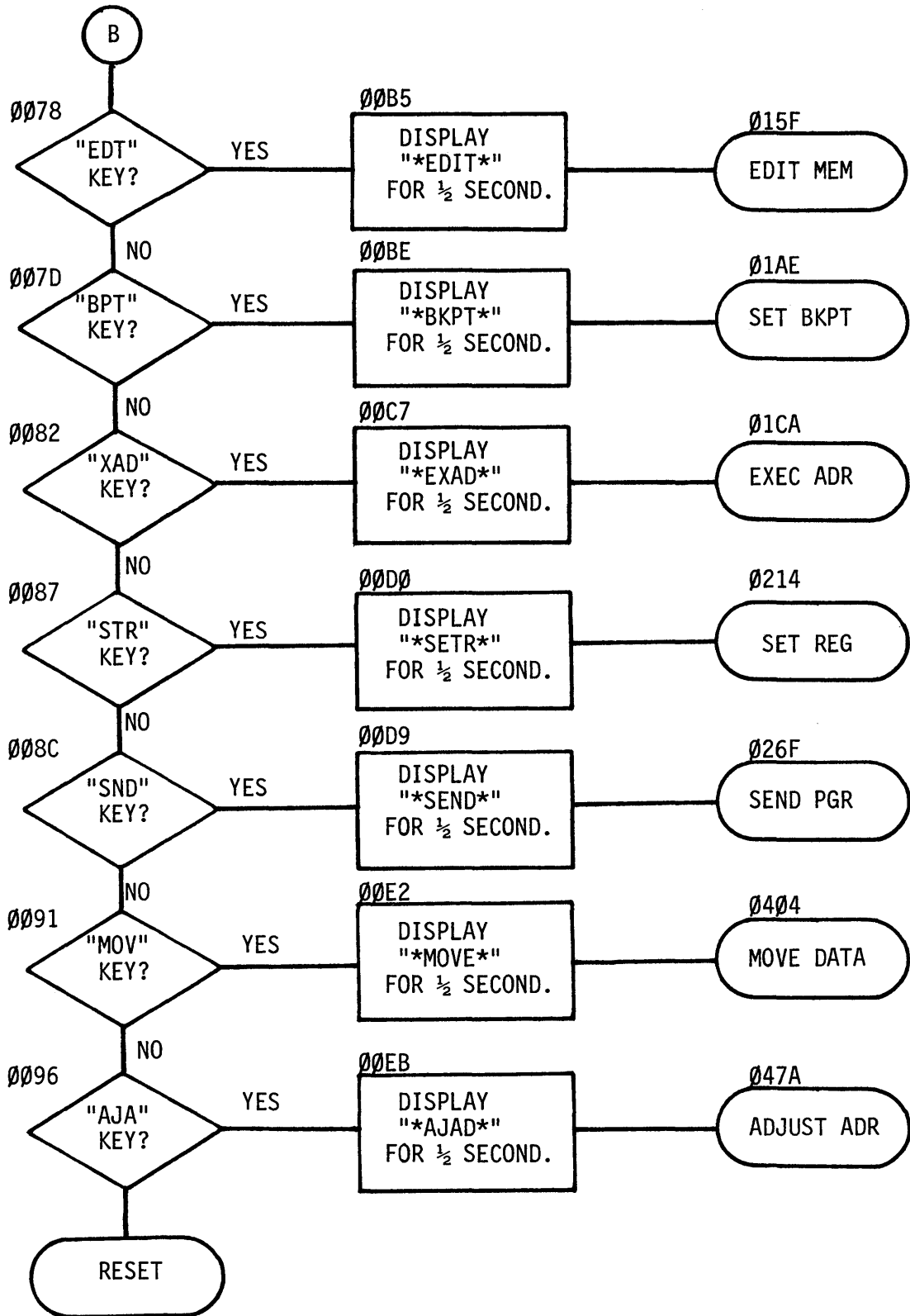
The flowcharts utilize standard ANSI symbols. The program addresses that correspond to each flowchart element are written above the upper left shoulder of each symbol. If a routine that accomplishes a particular flowchart operation resides elsewhere in memory (subroutines, etc.) its address is written to the right of the symbol.

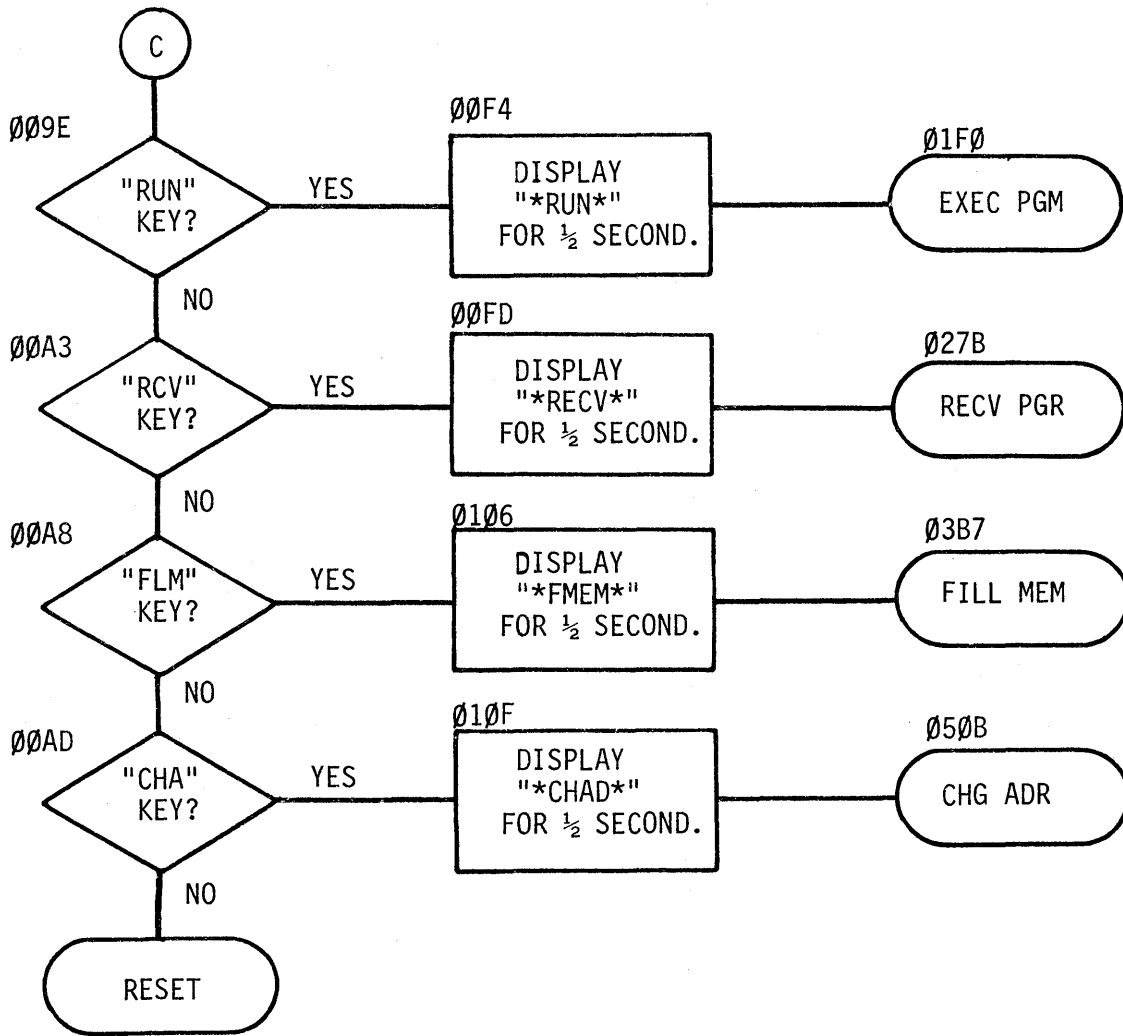
Two memory maps are provided:

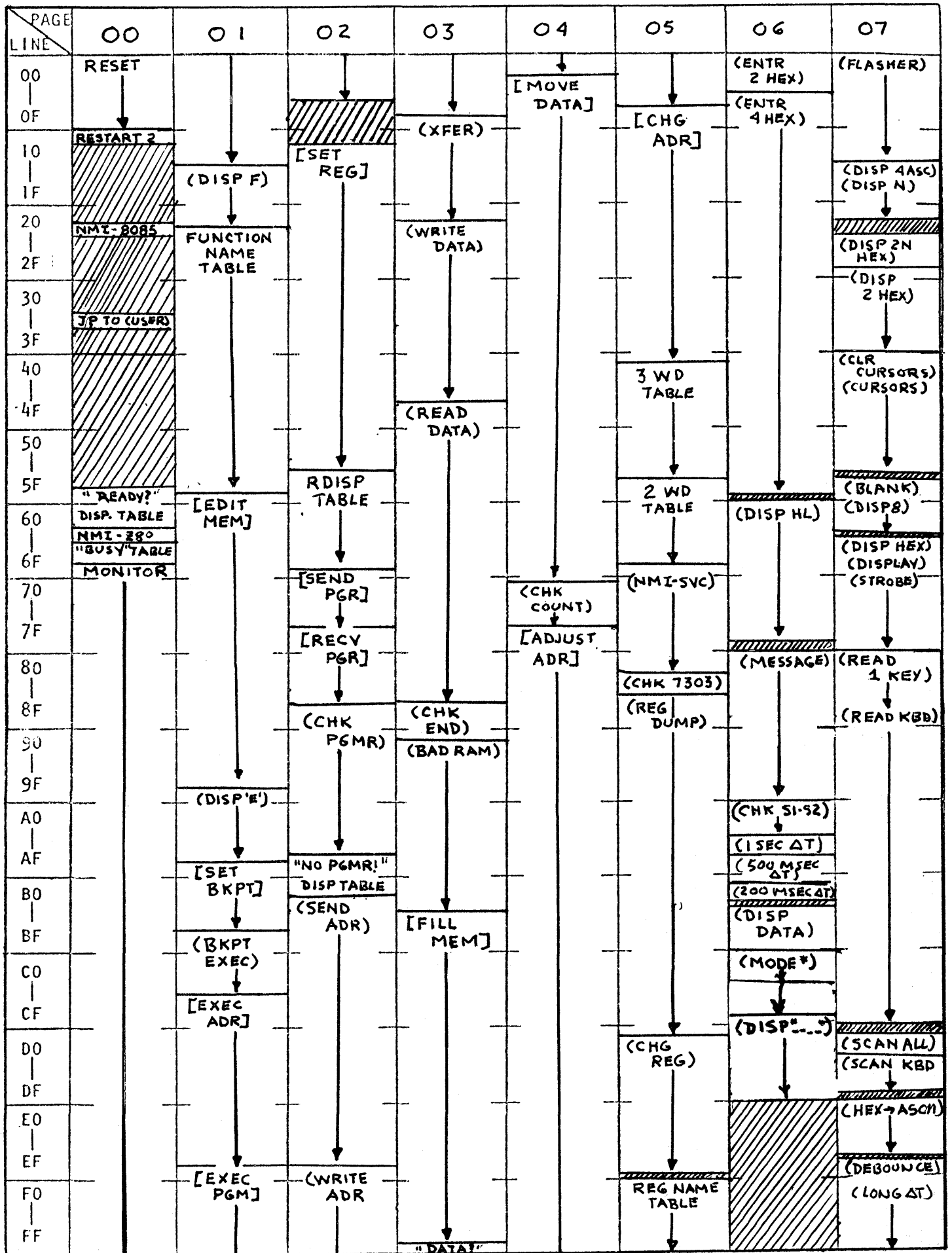
- a. 2K ROM memory map. This map indicates the relative location in ROM of each routine, table, and unused area.
- b. 1 Page memory map. This map indicates the relative location of each RAM variable or program parameter assigned for use by the Monitor. It also shows locations reserved for stack operation. All unassigned RAM space is available for user programs and data.



MONITOR PROGRAM MAIN FLOWCHART







2K MEMORY MAP - MP-4 KEYBOARD MONITOR PGM

| LINE | LABEL | LINE | LABEL | LINE | LABEL | LINE | LABEL |
|------|----------------|------|---------|------|-------|------|-------|
| 00 | (L) RSAVE | 40 | | 80 | | 00 | |
| 1 | (H) | 1 | | 1 | | 1 | |
| 2 | (E) | 2 | | 2 | | 2 | |
| 3 | (D) | 3 | | 3 | | 3 | |
| 4 | (C) | 4 | | 4 | | 4 | |
| 5 | (B) | 5 | | 5 | | 5 | |
| 6 | (F) | 6 | | 6 | | 6 | |
| 7 | (A) | 7 | | 7 | | 7 | |
| 8 | ISAVE | 8 | | 8 | | 8 | |
| 9 | | 9 | | 9 | | 9 | |
| A | | A | | A | | A | |
| B | ENTR BUFR | B | | B | | B | |
| C | | C | | C | | C | |
| D | | D | | D | | D | |
| E | | E | | E | | E | |
| 0F | RUNAD (LINE) | 4F | STACK-1 | 8F | | 0F | |
| 1C | (PAGE) | 50 | | 90 | | 1C | |
| 1 | CUR ADR (LINE) | 1 | | 1 | | 1 | |
| 2 | (PAGE) | 2 | | 2 | | 2 | |
| 3 | END ADR (LINE) | 3 | | 3 | | 3 | |
| 4 | (PAGE) | 4 | | 4 | | 4 | |
| 5 | SST ADR (LINE) | 5 | | 5 | | 5 | |
| 6 | (PAGE) | 6 | | 6 | | 6 | |
| 7 | SEN ADR (LINE) | 7 | | 7 | | 7 | |
| 8 | (PAGE) | 8 | | 8 | | 8 | |
| 9 | DST ADR (LINE) | 9 | | 9 | | 9 | |
| A | (PAGE) | A | | A | | A | |
| B | DISPL (LSB) | B | | B | | B | |
| C | (MSB) | C | | C | | C | |
| D | | D | | D | | D | |
| E | | E | | E | | E | |
| 1F | | 5F | | 9F | | 1F | |
| 20 | (USER) C3 | 60 | | A0 | | 20 | |
| 1 | LL | 1 | | 1 | | 1 | |
| 2 | PP | 2 | | 2 | | 2 | |
| 3 | (F) RDATA | 3 | | 3 | | 3 | |
| 4 | (A) | 4 | | 4 | | 4 | |
| 5 | (C) | 5 | | 5 | | 5 | |
| 6 | (B) | 6 | | 6 | | 6 | |
| 7 | (E) | 7 | | 7 | | 7 | |
| 8 | (D) | 8 | | 8 | | 8 | |
| 9 | (L) | 9 | | 9 | | 9 | |
| A | (H) | A | | A | | A | |
| B | | B | | B | | B | |
| C | | C | | C | | C | |
| D | | D | | D | | D | |
| E | | E | | E | | E | |
| 2F | | 6F | | AF | | 2F | |
| 30 | STACK | 70 | | B0 | | 30 | |
| 1 | | 1 | | 1 | | 1 | |
| 2 | | 2 | | 2 | | 2 | |
| 3 | | 3 | | 3 | | 3 | |
| 4 | | 4 | | 4 | | 4 | |
| 5 | | 5 | | 5 | | 5 | |
| 6 | | 6 | | 6 | | 6 | |
| 7 | | 7 | | 7 | | 7 | |
| 8 | | 8 | | 8 | | 8 | |
| 9 | | 9 | | 9 | | 9 | |
| A | | A | | A | | A | |
| B | | B | | B | | B | |
| C | | C | | C | | C | |
| D | | D | | D | | D | |
| E | | E | | E | | E | |
| 3F | | 7F | | BF | | 3F | |

RAM VARIABLE MAP

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|-------|----------|-------|---------------|---------------------------------|----------|
| PAGE ADR | LINE ADR | INSTR | LABEL | INSTR | MODIFIER | COMMENTS | CHECKSUM |
| 00 | 00 | 00 | RESET | NOP | | | 037A68 |
| | 1 | 31 | | LDPI | SP | INITIALIZE STACK POINTER. | |
| | 2 | 50 | | - | STACK | | |
| | 3 | 20 | | - | | | |
| | 4 | CD | | JS | | ARE S1 AND S2 CLOSED, OR | |
| | 5 | A0 | | - | (CHK S1*S2) | 7303 NOT THERE? | |
| | 6 | 06 | | - | | | |
| | 7 | CA | | JP | Z | IF TRUE, GO TO ROM 1. | |
| | 8 | 00 | | - | PAGE 08 | | |
| | 9 | 08 | | - | | | |
| | A | CD | | JS | | NO. WAIT 500 MSEC'S TO | |
| | B | AB | | - | (500 MSEC ΔT) | ENHANCE DISPLAY VISIBILITY. | |
| | C | 06 | | - | | | |
| | D | C3 | | JP | | GO TO START OF MONITOR PROGRAM. | |
| | E | 40 | | - | START | | |
| | F | 00 | | - | | | |
| 00 | 10 | C3 | RST 2 | JP | | GO EXECUTE BREAKPOINT | |
| | 1 | BD | | - | (BKPT EXEC) | SERVICE ROUTINE. | |
| | 2 | 01 | | - | | | |
| | 3 | | | | | | |
| | 4 | | | | | | |
| | 5 | | | | | | |
| | 6 | | | | | | |
| | 7 | | | | | | |
| | 18 | | RST 3 | | | | |
| | 9 | | | | | | |
| | A | | | | | | |
| | B | | | | | | |
| | C | | | | | | |
| | D | | | | | | |
| | E | | | | | | |
| | F | | | | | | |
| 00 | 20 | | RST 4 | | | | |
| | 1 | | | | | | |
| | 2 | | | | | | |
| | 3 | | | | | | |
| | 24 | C3 | NMI-8085 | JP | | DISPLAY REGISTER CONTENTS ON | |
| | 5 | 6E | | - | (NMI-SVC) | OCCURRENCE OF 8085 NMI. | |
| | 6 | 05 | | - | | | |
| | 7 | | | | | | |
| | 28 | | RST 5 | | | | |
| | 9 | | | | | | |
| | A | | | | | | |
| | B | | | | | | |
| | 2C | | RST 5.5 | | | (8085 ONLY) | |
| | D | | | | | | |
| | E | | | | | | |
| | F | | | | | | |
| 00 | 30 | | RST 6 | | | | |
| | 1 | | | | | | |
| | 2 | | | | | | |
| | 3 | | | | | | |
| | 34 | | RST 6.5 | | | (8085 ONLY) | |
| | 5 | | | | | | |
| | 6 | | | | | | |
| | 7 | | | | | | |
| | 38 | C3 | RST 7 | JP | | EXECUTE INTERRUPT ROUTINE | |
| | 9 | 20 | | - | (USER) | VECTERING JUMP ENTERED BY | |
| | A | 20 | | - | | USER INTO RAM. | |
| | B | | | | | | |
| 8 | 3C | | RST 7.5 | | | (8085 ONLY) | |
| 2 | D | | | | | | |
| 1 | E | | | | | | |
| | F | | | | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|----------|-------|-------------|---|----------------------------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR | MODIFIER | COMMENTS | |
| 00 | 40 | CD | START | JS | | T CLEAR ALL CURSOR SEGMENTS. | |
| | 1 | 40 | | - | (CLR CURS) | ↓ | |
| | 2 | 07 | | - | | | |
| | 3 | 21 | | LDPI | HL | T DISPLAY "READY?" | |
| | 4 | 5E | | - | RDYTBL | | |
| | 5 | 00 | | - | | | |
| | 6 | 0E | | LDCI | | | |
| | 7 | 08 | | - | 08 | | |
| | 8 | 1E | | LDEI | | | |
| | 9 | 07 | | - | 07 | | |
| | A | CD | | JS | | | |
| | B | 1A | | - | (DISP N) | | |
| | C | 07 | | - | | | |
| | D | 3E | | LDAI | | T ENABLE KEYBOARD COLUMNS. | |
| | E | 0F | | - | 0F | | |
| | F | D3 | | OPA | | | |
| 00 | 50 | DO | | - | KEYBOARD | ↓ | |
| | 51 | DB | NO KEY | IPA | | T CHECK FOR CLOSURE OF FUNCTION KEYS. | |
| | 2 | DO | | - | KEYBOARD | | |
| | 3 | E6 | | ANAI | | | |
| | 4 | 30 | | - | 30 | | |
| | 5 | CA | | JP | Z | | |
| | 6 | 51 | | - | NO KEY | | |
| | 7 | 00 | | - | | | |
| | 8 | CD | | JS | | T KEY DEPRESSED; BLANK DISPLAYS. | |
| | 9 | 5B | | - | (BLANK) | | |
| | A | 07 | | - | | | |
| | B | C3 | | JP | | T GO TO MONITOR PROGRAM. | |
| | C | 6D | | - | MONITOR | | |
| | D | 00 | | - | | | |
| | 5E | A0 | RDYTBL | -CST- | | (SP) "READY?" DISPLAY TABLE. | |
| | F | D2 | | | | R | |
| 00 | 60 | C5 | | | | E | |
| | 1 | C1 | | | | A | |
| | 2 | C4 | | | | D | |
| | 3 | D9 | | | | Y | |
| | 4 | BF | | | | ? | |
| | 5 | A0 | | -CST- | | (SP) | |
| | 66 | C3 | NMI-Z80 | JP | | T DISPLAY REGISTER CONTENTS ON OCCURRENCE OF Z80 NMI. | |
| | 7 | 6E | | - | (NMI-SYC) | | |
| | 8 | 05 | | - | | | |
| | 69 | C2 | "BUSY" | -CST- | | B "BUSY" DISPLAY TABLE. | |
| | A | D5 | | | | U | |
| | B | D3 | | | | S | |
| | C | D9 | | -CST- | | Y | |
| | 6D | CD | MONITOR | JS | | T READ KEYBOARD FOR FUNCTION. | |
| | E | 8F | | - | (READ KBD) | | |
| | F | 07 | | - | | | |
| 00 | 70 | FE | | CPAI | | T SHIFT? | |
| | 1 | 14 | | - | "↑" | "↑" = KEY 14. | |
| | 2 | C2 | | JP | NZ | | |
| | 3 | 9E | | - | LWR FUNC | | NO. CHECK LOWER FUNCTIONS. |
| | 4 | 00 | | - | | | |
| | 5 | CD | | JS | | T YES. READ KEYBOARD FOR UPPER FUNCTION. | |
| | 6 | 80 | | - | (READ 1KEY) | | |
| | 7 | 07 | | - | | | |
| | 8 | FE | | CPAI | | | EDIT MEMORY? |
| | 9 | 10 | | - | "EDT" | "EDT" = KEY 10. | |
| 1068 | A | CA | | JP | Z | | |
| 9 | B | B5 | | - | DEDIT | | |
| 8 | C | 00 | | - | | | |
| 28 | D | FE | | CPAI | | | SET BREAKPOINT? |
| 1 | E | 11 | | - | "BPT" | "BPT" = KEY 11. | |
| | F | CA | | JP | Z | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|----------|--------|------------|-----------------------------------|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 00 | 80 | BE | | - | DBKPT | | |
| | 1 | 00 | | - | | | |
| | 2 | FE | | CPAI | | ENTER EXECUTION ADDRESS? | |
| | 3 | 12 | | - | "XAD" | "XAD" = KEY 12. | |
| | 4 | CA | | JP | Z | | |
| | 5 | C7 | | - | DEXAD | | |
| | 6 | 00 | | - | | | |
| | 7 | FE | | CPAI | | SET REGISTERS? | |
| | 8 | 13 | | - | "STR" | "STR" = KEY 13. | |
| | 9 | CA | | JP | Z | | |
| | A | DO | | - | DSETR | | |
| | B | 00 | | - | | | |
| | C | FE | | CPAI | | SEND DATA TO PROM PGR? | |
| | D | 15 | | - | "SND" | "SND" = KEY 15. | |
| | E | CA | | JP | Z | | |
| | F | D9 | | - | DSEND | | |
| 00 | 90 | 00 | | - | | | |
| | 1 | FE | | CPAI | | MOVE BLOCK OF DATA? | |
| | 2 | 16 | | - | "MOV" | "MOV" = KEY 16. | |
| | 3 | CA | | JP | Z | | |
| | 4 | E2 | | - | DMOVE | | |
| | 5 | 00 | | - | | | |
| | 6 | FE | | CPAI | | ADJUST ADDRESSES AFTER MOVE? | |
| | 7 | 17 | | - | "AJA" | "AJA" = KEY 17. | |
| | 8 | CA | | JP | Z | | |
| | 9 | EB | | - | DAJAD | | |
| | A | 00 | | - | | | |
| | B | C3 | | JP | | T WAIT FOR VALID KEY. | |
| | C | 00 | | - | RESET | | |
| | D | 00 | | - | | | |
| | 9E | FE | LWRFUNC | CPAI | | T EXECUTE PROGRAM? | |
| | F | 12 | | - | "RUN" | "RUN" = KEY 12. | |
| 00 | A0 | CA | | JP | Z | | |
| | 1 | F4 | | - | DRUN | | |
| | 2 | 00 | | - | | | |
| | 3 | FE | | CPAI | | RECEIVE DATA FROM PROM PGR? | |
| | 4 | 15 | | - | "RCV" | "RCV" = KEY 15. | |
| | 5 | CA | | JP | Z | | |
| | 6 | FD | | - | DRECV | | |
| | 7 | 00 | | - | | | |
| | 8 | FE | | CPAI | | FILL MEMORY BLOCK W/KNOWN | |
| | 9 | 16 | | - | "FLM" | VALUE? "FLM" = KEY 16. | |
| | A | CA | | JP | Z | | |
| | B | 06 | | - | DEMEM | | |
| | C | 01 | | - | | | |
| | D | FE | | CPAI | | CHANGE PROGRAM ADDRESSES? | |
| | E | 17 | | - | "CHA" | "CHA" = KEY 17. | |
| | F | CA | | JP | Z | | |
| 00 | B0 | 0F | | - | DCHAD | | |
| | 1 | 01 | | - | | | |
| | 2 | C3 | | JP | | T WAIT FOR VALID KEY. | |
| | 3 | 00 | | - | RESET | | |
| | 4 | 00 | | - | | | |
| | B5 | 21 | DEDIT | LDPI | HL | T DISPLAY "EDIT"; GO EDIT MEMORY. | |
| | 6 | 26 | | - | "EDIT" | | |
| | 7 | 01 | | - | | | |
| | 8 | CD | | JS | | | |
| | 9 | 18 | | - | (DISP F) | | |
| | A | 01 | | - | | | |
| | B | C3 | | JP | | | |
| | C | 5F | | - | [EDIT MEM] | | |
| | D | 01 | | - | | | |
| 1 | B E | 21 | DBKPT | LDPI | HL | T DISPLAY "BKPT"; GO SET A | |
| | F | 2B | | - | "BKPT" | BREAKPOINT. | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|----------|--------|--------------|----------|---|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 00 | C 0 | 01 | | - | | | |
| | 1 | CD | | JS | | | |
| | 2 | 18 | | - | (DISP F) | | |
| | 3 | 01 | | - | | | |
| | 4 | C3 | | JP | | | |
| | 5 | AE | | - | [SET BKPT] | | |
| | 6 | 01 | | - | | | |
| | C 7 | 21 | DEXAD | LDPI | HL | T | DISPLAY "EXAD"; GO ENTER EXECUTION ADDRESS. |
| | 8 | 30 | | - | "EXAD" | | |
| | 9 | 01 | | - | | | |
| | A | CD | | JS | | | |
| | B | 18 | | - | (DISP F) | | |
| | C | 01 | | - | | | |
| | D | C3 | | JP | | | |
| | E | CA | | - | [EXEC ADR] | | |
| | F | 01 | | - | | | |
| 00 | D 0 | 21 | DSETR | LDPI | HL | T | DISPLAY "SETR"; GO SET REGISTER VALUES FOR EXECUTION. |
| | 1 | 35 | | - | "SETR" | | |
| | 2 | 01 | | - | | | |
| | 3 | CD | | JS | | | |
| | 4 | 18 | | - | (DISP F) | | |
| | 5 | 01 | | - | | | |
| | 6 | C3 | | JP | | | |
| | 7 | 14 | | - | [SET REG] | | |
| | 8 | 02 | | - | | | |
| | D 9 | 21 | DSEND | LDPI | HL | T | DISPLAY "SEND"; GO SEND DATA TO PROM PROGRAMMER. |
| | A | 3A | | - | "SEND" | | |
| | B | 01 | | - | | | |
| | C | CD | | JS | | | |
| | D | 18 | | - | (DISP F) | | |
| | E | 01 | | - | | | |
| | F | C3 | | JP | | | |
| 00 | E 0 | 6F | | - | [SEND PGR] | | |
| | 1 | 02 | | - | | | |
| | E 2 | 21 | DMOVE | LDPI | HL | T | DISPLAY "MOVE"; GO MOVE A BLOCK OF DATA. |
| | 3 | 3F | | - | "MOVE" | | |
| | 4 | 01 | | - | | | |
| | 5 | CD | | JS | | | |
| | 6 | 18 | | - | (DISP F) | | |
| | 7 | 01 | | - | | | |
| | 8 | C3 | | JP | | | |
| | 9 | 04 | | - | [MOVE DATA] | | |
| | A | 04 | | - | | | |
| | E B | 21 | DAJAD | LDPI | HL | T | DISPLAY "AJAD"; GO ADJUST ADDRESSES. |
| | C | 44 | | - | "AJAD" | | |
| | D | 01 | | - | | | |
| | E | CD | | JS | | | |
| | F | 18 | | - | (DISP F) | | |
| 00 | F 0 | 01 | | - | | | |
| | 1 | C3 | | JP | | | |
| | 2 | 7A | | - | [ADJUST ADR] | | |
| | 3 | 04 | | - | | | |
| | F 4 | 21 | DRUN | LDPI | HL | T | DISPLAY "RUN"; GO EXECUTE PROGRAM. |
| | 5 | 49 | | - | "RUN" | | |
| | 6 | 01 | | - | | | |
| | 7 | CD | | JS | | | |
| | 8 | 18 | | - | (DISP F) | | |
| 1 | 9 | 01 | | - | | | |
| 0 | A | C3 | | JP | | | |
| 8 | B | FO | | - | [EXEC PGM] | | |
| 9 | C | 01 | | - | | | |
| 2 | F D | 21 | DRECV | LDPI | HL | T | DISPLAY "RECV"; GO RECEIVE DATA FROM PROM PROGRAMMER. |
| 1 | E | 4F | | - | "RECV" | | |
| | F | 01 | | - | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|----------|--------|--------------|-----------------------------------|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 01 | 00 | CD | | JS | | | |
| | 1 | 18 | | - | (DISP F) | | |
| | 2 | 01 | | - | | | |
| | 3 | C3 | | JP | | | |
| | 4 | 7B | | - | [RECV PGR] | | |
| | 5 | 02 | | - | | | |
| | 06 | 21 | DFMEM | LDPI | HL | T DISPLAY "FMEM"; GO FILL A BLOCK | |
| | 7 | 54 | | - | "FMEM" | OF MEMORY WITH ENTERED VALUE. | |
| | 8 | 01 | | - | | | |
| | 9 | CD | | JS | | | |
| | A | 18 | | - | (DISP F) | | |
| | B | 01 | | - | | | |
| | C | C3 | | JP | | | |
| | D | B7 | | - | [FILL MEM] | | |
| | E | 03 | | - | | | |
| | 0F | 21 | DCHAD | LDPI | HL | T DISPLAY "CHAD"; GO CHANGE ALL | |
| 01 | 10 | 59 | | - | "CHAD" | JP AND LDPI PAGE AND LINE | |
| | 1 | 01 | | - | | ADDRESSES WITHIN A BLOCK OF | |
| | 2 | CD | | JS | | MEMORY. | |
| | 3 | 18 | | - | (DISP F) | | |
| | 4 | 01 | | - | | | |
| | 5 | C3 | | JP | | | |
| | 6 | 0B | | - | [CHG ADR] | | |
| | 7 | 05 | | - | | | |
| | 18 | 0E | (DISP F) | LDCI | | T DISPLAY FUNCTION NAME FOR | |
| | 9 | 06 | | - | 06 | APPROX. 500 MILLISECONDS. | |
| | A | 1E | | LDEI | | | |
| | B | 06 | | - | 06 | | |
| | C | CD | | JS | | | |
| | D | 1A | | - | (DISP N) | | |
| | E | 07 | | - | | | |
| | F | CD | | JS | | | |
| 01 | 20 | AB | | - | (500 MSECAT) | | |
| | 1 | 06 | | - | | | |
| | 2 | CD | | JS | | | |
| | 3 | 5B | | - | (BLANK) | | |
| | 4 | 07 | | - | | | |
| | 5 | C9 | | RTS | | EXIT. | |
| | 26 | AA | "EDIT" | -CST- | | * FUNCTION NAME DISPLAY TABLE. | |
| | 7 | C5 | | | | E | |
| | 8 | C4 | | | | D | |
| | 9 | C9 | | | | I | |
| | A | D4 | | | | T | |
| | 2B | AA | "BKPT" | | | * B | |
| | C | C2 | | | | K | |
| | D | CB | | | | P | |
| | E | DO | | | | T | |
| | F | D4 | | | | | |
| 01 | 30 | AA | "EXAD" | | | * E | |
| | 1 | C5 | | | | X | |
| | 2 | D8 | | | | A | |
| | 3 | C1 | | | | D | |
| | 4 | C4 | | | | | |
| | 35 | AA | "SETR" | | | * S | |
| | 6 | D3 | | | | E | |
| | 7 | C5 | | | | T | |
| | 8 | D4 | | | | | |
| | 9 | D2 | | | | R | |
| 106821 | 3A | AA | "SEND" | | | * S | |
| | B | D3 | | | | E | |
| | C | C5 | | | | N | |
| | D | CE | | | | D | |
| | E | C4 | | | | | |
| | 3F | AA | "MOVE" | -CST- | | * | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|------------|--------|----------------|----------|--|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 01 | 40 | CD | | -CST- | | M | |
| | 1 | CF | | | | O | |
| | 2 | D6 | | | | V | |
| | 3 | C5 | | | | E | |
| | 44 | AA | "AJAD" | | | * | |
| | 5 | C1 | | | | A | |
| | 6 | CA | | | | J | |
| | 7 | C1 | | | | A | |
| | 8 | C4 | | | | D | |
| | 49 | AA | "RUN" | | | * | |
| | A | D2 | | | | R | |
| | B | D5 | | | | U | |
| | C | CE | | | | N | |
| | D | AA | | | | * | |
| | E | AO | | | | (SP) | |
| | 4F | AA | "RECV" | | | * | |
| 01 | 50 | D2 | | | | R | |
| | 1 | C5 | | | | E | |
| | 2 | C3 | | | | C | |
| | 3 | D6 | | | | V | |
| | 54 | AA | "FMEM" | | | * | |
| | 5 | C6 | | | | F | |
| | 6 | CD | | | | M | |
| | 7 | C5 | | | | E | |
| | 8 | CD | | | | M | |
| | 59 | AA | "CHAD" | | | * | |
| | A | C3 | | | | C | |
| | B | C8 | | | | H | |
| | C | C1 | | | | A | |
| | D | C4 | | | | D | |
| | E | AA | | -CST- | | * | |
| | 5F | 1E | [EDIT MEM] | LDEI | | T | ENTER 4 DIGIT START ADDRESS AND DISPLAY IT. |
| 01 | 60 | 05 | | - | Ø5 | | |
| | 1 | CD | | JS | | | |
| | 2 | 09 | | - | (ENTR 4HEX) | | |
| | 3 | 06 | | - | | | |
| | 64 | CD | NEXT ADR | JS | | T | BLANK DISPLAYS. |
| | 5 | 5B | | - | (BLANK) | | |
| | 6 | 07 | | - | | | |
| | 7 | 1E | | LDEI | | T | DISPLAY ADDRESS AND DATA; INCREMENT TO NEXT ADDRESS. |
| | 8 | 04 | | - | Ø4 | | |
| | 9 | CD | | JS | | | |
| | A | 60 | | - | (DISP HL) | | |
| | B | 06 | | - | | | |
| | C | 1E | | LDEI | | | |
| | D | 00 | | - | ØØ | | |
| | E | CD | | JS | | | |
| | F | 2F | | - | (DISP 2 HEX) | | |
| 01 | 70 | 07 | | - | | | |
| | 71 | CD | INVALID | JS | | T | READ KEYBOARD. |
| | 2 | 80 | | - | (READ 1 KEY) | | |
| | 3 | 07 | | - | | | |
| | 4 | FE | | CPAI | | T | IF HEX KEY, ENTER NEW DATA. |
| | 5 | 10 | | - | "SST" = KEY 10 | | |
| | 6 | DA | | JP | C1 | | |
| | 7 | 86 | | - | CHNG DATA | | |
| | 8 | 01 | | - | | | |
| | 9 | CA | | JP | Z | T | IF "SST" KEY, LIST FORWARD THRU MEMORY. |
| 106821 | A | 64 | | - | NEXT ADR | | |
| | B | 01 | | - | | | |
| | C | FE | | CPAI | | T | IF NOT "BST" KEY, WAIT FOR VALID ENTRY. |
| | D | 11 | | - | "BST" = KEY 11 | | |
| | E | C2 | | JP | NZ | | |
| | F | 71 | | - | INVALID | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|-------------|--------|---------------|---|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 01 | 80 | 01 | | - | | | |
| | 1 | 2B | | DCP | HL | IF "BST" KEY, LIST BACKWARD THRU MEMORY. | |
| | 2 | 2B | | DCP | HL | | |
| | 3 | C3 | | JP | | | |
| | 4 | 64 | | - | NEXT ADR | | |
| | 5 | 01 | | - | | | |
| | 86 | 2B | CHNG DATA | DCP | HL | DECREMENT TO CURRENT ADDRESS. ENTER AND DISPLAY 2 HEX DIGITS OF DATA. | |
| | 7 | 01 | | LDPI | BC | | |
| | 8 | 03 | | - | Ø3 | | |
| | 9 | 03 | | - | Ø3 | | |
| | A | E5 | | PSP | HL | | |
| | B | 1E | | LDEI | | | |
| | C | 01 | | - | Ø1 | | |
| | D | 21 | | LDPI | HL | | |
| | E | 0D | | - | ENTR BUFR+2 | | |
| | F | 20 | | - | | | |
| 01 | 90 | CD | | JS | | | |
| | 1 | 17 | | - | (ENTR 4HEX)+4 | | |
| | 2 | 06 | | - | | | |
| | 3 | 7D | | LDA | L | STORE NEW DATA AT CURRENT ADDRESS. | |
| | 4 | E1 | | PLP | HL | | |
| | 5 | 77 | | STAN | (HL) | IF NOT ACCEPTED, DISPLAY AN ERROR "E" NEXT TO DATA. | |
| | 6 | BE | | CPAN | (HL) | | |
| | 7 | C4 | | JS | NZ | | |
| | 8 | 9E | | - | (DISP "E") | | |
| | 9 | 01 | | - | | | |
| | A | 23 | | ICP | HL | INCREMENT TO NEXT ADDRESS. | |
| | B | C3 | | JP | | | |
| | C | 64 | | - | NEXT ADR | | |
| | D | 01 | | - | | | |
| | 9E | 1E | (DISP "E") | LDEI | | DISPLAY AN "E" IN 3RD DISPLAY FROM RIGHT. | |
| | F | 02 | | - | Ø2 | | |
| 01 | A0 | 16 | | LDDI | | | |
| | 1 | C5 | | - | "E" | | |
| | 2 | CD | | JS | | | |
| | 3 | 6D | | - | (DISPLAY) | | |
| | 4 | 07 | | - | | | |
| | A5 | CD | WAIT | JS | | WAIT FOR "SST" KEY. | |
| | 6 | 80 | | - | (READ 1 KEY) | | |
| | 7 | 07 | | - | | | |
| | 8 | FE | | CPAI | | | |
| | 9 | 10 | | - | "SST"=KEY10 | | |
| | A | C2 | | JP | NZ | | |
| | B | A5 | | - | WAIT | | |
| | C | 01 | | - | | | |
| | D | C9 | | RTS | | EXIT. | |
| | A E | 1E | [SET BKPT] | LDEI | | ENTER AND DISPLAY 4 DIGIT BREAKPOINT ADDRESS. | |
| | F | 05 | | - | Ø5 | | |
| 01 | B0 | CD | | JS | | | |
| | 1 | 09 | | - | (ENTR 4HEX) | | |
| | 2 | 06 | | - | | | |
| | 3 | 7E | | LDAN | (HL) | GET ORIGINAL INSTRUCTION BYTE AND SAVE IT. | |
| | 4 | 32 | | STAD | | | |
| | 5 | 08 | | - | I SAVE | | |
| | 6 | 20 | | - | | | |
| | 7 | 3E | | LDAI | | REPLACE IT WITH BREAKPOINT INSTRUCTION. | |
| | 8 | D7 | | - | D7 | | |
| | 9 | 77 | | STAN | (HL) | | |
| 1068 | A | C3 | | JP | | | |
| | B | 00 | | - | RESET | | |
| | C | 00 | | - | | | |
| 128 | B D | CD | (BKPT EXEC) | JS | | DISPLAY AND, IF DESIRED, CHANGE REGISTER CONTENTS. | |
| | E | 89 | | - | (REG DUMP) | | |
| | F | 05 | | - | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|------------|--------|----------------|----------|--------------------------------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 01 | C0 | E3 | | XCPT | HL | ↓ | DECREMENT RETURN ADDRESS. |
| | 1 | 2B | | DCP | HL | ↓ | |
| | 2 | F5 | | PSP | AF | ↓ | RESTORE ORIGINAL INSTRUCTION |
| | 3 | 3A | | LDAD | | ↓ | BYTE. |
| | 4 | 08 | | - | ISAVE | | |
| | 5 | 20 | | - | | | |
| | 6 | 77 | | STAN | (HL) | | |
| | 7 | F1 | | PLP | AF | ↓ | |
| | 8 | E3 | | XCPT | HL | | CONTINUE PROGRAM EXECUTION. |
| | 9 | C9 | | RTS | | EXIT. | |
| C | A | 2A | [EXEC ADR] | LDPD | HL | ↓ | DISPLAY LAST EXECUTION |
| B | | 0F | | - | RUNAD | | START ADDRESS, FOR 1 SECOND. |
| C | | 20 | | - | | | |
| D | | 1E | | LDEI | | | |
| E | | 02 | | - | Ø2 | | |
| F | | CD | | JS | | | |
| 01 | D0 | 60 | | - | (DISP HL) | | |
| | 1 | 06 | | - | | | |
| | 2 | CD | | JS | | | |
| | 3 | A8 | | - | (1 SEC AT) | | |
| | 4 | 06 | | - | | ↓ | |
| | 5 | CD | | JS | | ↓ | ENTER AND DISPLAY NEW |
| | 6 | 09 | | - | (ENTR 4HEX) | | 4-DIGIT START ADDRESS. |
| | 7 | 06 | | - | | | |
| | 8 | 22 | | STPD | HL | | |
| | 9 | 0F | | - | RUNAD | | |
| A | | 20 | | - | | ↓ | |
| B | | 21 | | LDPI | HL | ↓ | DISPLAY "*RUN*" |
| C | | 49 | | - | *RUN* | | |
| D | | 01 | | - | | | |
| E | | 0E | | LDCI | | | |
| F | | 06 | | - | Ø6 | | |
| 01 | E0 | 1E | | LDEI | | | |
| | 1 | 06 | | - | Ø6 | | |
| | 2 | CD | | JS | | | |
| | 3 | 1A | | - | (DISP N) | | |
| | 4 | 07 | | - | | ↓ | |
| E | 5 | CD | WAIT RUN | JS | | ↓ | WAIT FOR "RUN" KEY. |
| | 6 | 80 | | - | (READ 1 KEY) | | |
| | 7 | 07 | | - | | | |
| | 8 | FE | | CPAI | | | |
| | 9 | 12 | | - | "RUN" = KEY 12 | | |
| A | | C2 | | JP | NZ | | |
| B | | E5 | | - | WAIT RUN | | |
| C | | 01 | | - | | ↓ | |
| D | | CD | | JS | | ↓ | BLANK DISPLAYS. |
| E | | 5B | | - | (BLANK) | | |
| F | | 07 | | - | | ↓ | |
| 01 | F0 | 2A | [EXEC PGM] | LDPD | HL | ↓ | GET CURRENT EXECUTION START |
| | 1 | 0F | | - | RUNAD | | ADDRESS. |
| | 2 | 20 | | - | | | |
| | 3 | E5 | | PSP | HL | ↓ | SAVE IT ON STACK. |
| | 4 | 3E | | LDAI | | ↓ | BLANK LED'S; DISABLE KEYBOARD. |
| | 5 | 00 | | - | ØØ | | |
| | 6 | D3 | | OPA | | | |
| | 7 | D0 | | - | DØ | ↓ | |
| | 8 | 2A | | LDPD | HL | ↓ | LOAD REGISTERS WITH INITIAL |
| | 9 | 23 | | - | RDATA | | DATA. |
| 106821 | A | 20 | | - | | | |
| | B | E5 | | PSP | HL | | |
| | C | F1 | | PLP | AF | | |
| | D | 2A | | LDPD | HL | | |
| | E | 25 | | - | RDATA+2 | | |
| | F | 20 | | - | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|------------|--------|--------------|--|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 02 | 00 | E5 | | PSP | HL | | |
| | 1 | C1 | | PLP | BC | | |
| | 2 | 2A | | LDPD | HL | | |
| | 3 | 27 | | - | RDATA+4 | | |
| | 4 | 20 | | - | | | |
| | 5 | EB | | XCP | HL, DE | | |
| | 6 | 2A | | LDPD | HL | | |
| | 7 | 29 | | - | RDATA+6 | | |
| | 8 | 20 | | - | | | |
| | 9 | C9 | | RTS | | GET START ADDRESS; GO EXECUTE. | |
| | A | | | | | | |
| | B | | | | | | |
| | C | | | | | | |
| | D | | | | | | |
| | E | | | | | | |
| | F | | | | | | |
| | 01 | | | | | | |
| | 1 | | | | | | |
| | 2 | | | | | | |
| | 3 | | | | | | |
| | 14 | 0E | [SET REG] | LDCI | | DISPLAY "REG" IN LEFT 3 DISPLAYS. | |
| | 5 | 03 | | - | 03 | | |
| | 6 | 21 | | LDPI | HL | | |
| | 7 | 5C | | - | RDISP TABL | | |
| | 8 | 02 | | - | | | |
| | 9 | 1E | | LDEI | | | |
| | A | 07 | | - | 07 | | |
| | B | CD | | JS | | | |
| | C | 1A | | - | (DISP N) | | |
| | D | 07 | | - | | | |
| | E | E5 | | PSP | HL | | |
| | F | 21 | | LDPI | HL | | |
| 02 | 20 | 2C | | - | RDATA+9 | | |
| | 1 | 20 | | - | | | |
| | 2 | 06 | | LDBI | | SET REGISTER COUNT = 8. | |
| | 3 | 08 | | - | 08 | | |
| | 24 | E3 | NEXT RDATA | XCPT | HL | DISPLAY REGISTER NAME AND DATA. | |
| | 5 | 0E | | LDCI | | | |
| | 6 | 02 | | - | 02 | | |
| | 7 | 1E | | LDEI | | | |
| | 8 | 03 | | - | 03 | | |
| | 9 | CD | | JS | | INCREMENT TO NEXT REGISTER NAME. | |
| | A | 1A | | - | (DISP N) | | |
| | B | 07 | | - | | | |
| | C | E3 | | XCPT | HL | | |
| | D | 2B | | DCP | HL | DECREMENT TO NEXT ADDRESS THAT CONTAINS REGISTER DATA. | |
| | E | 2B | | DCP | HL | | |
| | F | 1E | | LDEI | | | |
| 02 | 30 | 00 | | - | 00 | | |
| | 1 | CD | | JS | | | |
| | 2 | 2F | | - | (DISP 2 HEX) | | |
| | 3 | 07 | | - | | | |
| | 34 | CD | NOT "SST" | JS | | READ KEYBOARD. | |
| | 5 | 80 | | - | (READ 1 KEY) | | |
| | 6 | 07 | | - | | | |
| | 7 | FE | | CPAI | | IF HEX KEY, GO CHANGE DATA. | |
| | 8 | 10 | | - | "SST"=KEY 10 | | |
| 106821 | 9 | DA | | JP | C1 | | |
| | A | 46 | | - | CHNG RDATA | | |
| | B | 02 | | - | | | |
| | C | C2 | | JP | NZ | OTHERWISE WAIT FOR "SST" KEY | |
| | D | 34 | | - | NOT "SST" | | |
| | E | 02 | | - | | | |
| | 3F | 05 | NXT RCOUNT | DCB | | DO 8 TIMES. | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|------------|--------|---------------|----------|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 02 | 4 0 | C2 | | JP | NZ | | |
| | 1 | 24 | | - | NEXT RDATA | | |
| | 2 | 02 | | - | | | |
| | 3 | C3 | | JP | | | |
| | 4 | 00 | | - | RESET | | |
| | 5 | 00 | | - | | | |
| | 4 6 | 2B | CHNG RDATA | DCP | HL | | |
| | 7 | C5 | | PSP | BC | | |
| | 8 | 01 | | LDPI | BC | | |
| | 9 | 03 | | - | 03 | | |
| | A | 03 | | - | 03 | | |
| | B | E5 | | PSP | HL | | |
| | C | 1E | | LDEI | | | |
| | D | 01 | | - | 01 | | |
| | E | 21 | | LDPI | HL | | |
| | F | 0D | | - | ENTR BUFR+2 | | |
| 02 | 5 0 | 20 | | - | | | |
| | 1 | CD | | JS | | | |
| | 2 | 17 | | - | (ENTR 4HEX)+4 | | |
| | 3 | 06 | | - | | | |
| | 4 | 7D | | LDA | L | | |
| | 5 | E1 | | PLP | HL | | |
| | 6 | 77 | | STAN | (HL) | | |
| | 7 | 23 | | ICP | HL | | |
| | 8 | C1 | | PLP | BC | | |
| | 9 | C3 | | JP | | | |
| | A | 3F | | - | NXT RCOUNT | | |
| | B | 02 | | - | | | |
| | 5 C | D2 | RDISP TABL | -CST- | | | |
| | D | C5 | | | | | |
| | E | C7 | | | | | |
| | F | C8 | | | | | |
| 02 | 6 0 | BD | | | | | |
| | 1 | CC | | | | | |
| | 2 | BD | | | | | |
| | 3 | C4 | | | | | |
| | 4 | BD | | | | | |
| | 5 | C5 | | | | | |
| | 6 | BD | | | | | |
| | 7 | C2 | | | | | |
| | 8 | BD | | | | | |
| | 9 | C3 | | | | | |
| | A | BD | | | | | |
| | B | C1 | | | | | |
| | C | BD | | | | | |
| | D | C6 | | | | | |
| | E | BD | | | | | |
| | 6 F | CD | [SEND PGR] | JS | | | |
| 02 | 7 0 | 8C | | - | (CHK PGMR) | | |
| | 1 | 02 | | - | | | |
| | 2 | CD | | JS | | | |
| | 3 | B5 | | - | (SEND ADR) | | |
| | 4 | 02 | | - | | | |
| | 5 | CD | | JS | | | |
| | 6 | 23 | | - | (WRITE DATA) | | |
| | 7 | 03 | | - | | | |
| | 8 | C3 | | JP | | | |
| | 9 | 84 | | - | UNLOCK PGR | | |
| 106 | A | 02 | | - | | | |
| 82 | 7 B | CD | [RECV PGR] | JS | | | |
| | C | 8C | | - | (CHK PGMR) | | |
| | D | 02 | | - | | | |
| | E | CD | | JS | | | |
| | F | B5 | | - | (SEND ADR) | | |

| HEXADecimal | | | Mnemonic | | | TITLE | DATE |
|-------------|----------|--------|------------|--------|-------------|-------|--|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | | |
| 02 | 8 0 | 02 | | - | | ↓ | |
| | 1 | CD | | JS | | ↓ | RECEIVE DATA FIELD FROM PGMR. |
| | 2 | 4B | | - | (READ DATA) | ↓ | |
| | 3 | 03 | | - | | ↓ | |
| | 8 4 | AF | UNLOCK PGR | CLAC | | ↓ | DEACTIVATE INTERLOCK. |
| | 5 | D3 | | OPA | | ↓ | |
| | 6 | 51 | | - | 51 | ↓ | |
| | 7 | D3 | | OPA | | ↓ | |
| | 8 | 50 | | - | 50 | ↓ | |
| | 9 | C3 | | JP | | ↓ | |
| | A | 00 | | - | RESET | ↓ | |
| | B | 00 | | - | | ↓ | |
| | 8 C | DB | (CHK PGMR) | IPA | | ↓ | IS PROGRAMMER THERE? |
| | D | 51 | | - | 51 | ↓ | |
| | E | FE | | CPAI | | ↓ | |
| | F | FF | | - | FF | ↓ | |
| 02 | 9 0 | CA | | JP | Z | ↓ | |
| | 1 | 9D | | - | NO PGMR | ↓ | |
| | 2 | 02 | | - | | ↓ | |
| | 3 | E6 | | ANAI | | ↓ | IS IT TURNED ON? |
| | 4 | 08 | | - | 08 | ↓ | |
| | 5 | CA | | JP | Z | ↓ | |
| | 6 | 9D | | - | NO PGMR | ↓ | |
| | 7 | 02 | | - | | ↓ | |
| | 8 | 3E | | LDAI | | ↓ | SET INTERLOCK* LOW. |
| | 9 | 20 | | - | 20 | ↓ | |
| | A | D3 | | OPA | | ↓ | |
| | B | 51 | | - | 51 | ↓ | |
| | C | C9 | | RTS | | ↓ | EXIT. |
| | 9 D | 0E | NO PGMR | LDCI | | ↓ | DISPLAY "NO PGMR!" |
| | E | 08 | | - | 08 | ↓ | |
| | F | 1E | | LDEI | | ↓ | |
| 02 | A 0 | 07 | | - | 07 | ↓ | |
| | 1 | 21 | | LDPI | HL | ↓ | |
| | 2 | AD | | - | "NO PGMR!" | ↓ | |
| | 3 | 02 | | - | | ↓ | |
| | 4 | CD | | JS | | ↓ | |
| | 5 | 1A | | - | (DISP N) | ↓ | |
| | 6 | 07 | | - | | ↓ | |
| | 7 | CD | | JS | | ↓ | DELAY 1 SECOND TO ENHANCE VISIBILITY. |
| | 8 | A8 | | - | (1 SEC ΔT) | ↓ | |
| | 9 | 06 | | - | | ↓ | |
| | A | C3 | | JP | | ↓ | |
| | B | 00 | | - | RESET | ↓ | |
| | C | 00 | | - | | ↓ | |
| | A D | CE | "NO PGMR!" | -CST- | | ↓ | "NO PGMR!" DISPLAY TABLE. |
| | E | CF | | | | ↓ | |
| | F | A0 | | | | ↓ | (SP) |
| 02 | B 0 | D0 | | | | ↓ | P |
| | 1 | C7 | | | | ↓ | G |
| | 2 | CD | | | | ↓ | M |
| | 3 | D2 | | | | ↓ | R |
| | 4 | A1 | | -CST- | | ↓ | ! |
| | B 5 | 1E | (SEND ADR) | LDEI | | ↓ | ENTER AND DISPLAY 4-DIGIT PGMR. START ADDRESS. |
| | 6 | 05 | | - | 05 | ↓ | |
| | 7 | CD | | JS | | ↓ | |
| | 8 | 09 | | - | (ENTR 4HEX) | ↓ | |
| | 9 | 06 | | - | | ↓ | |
| 1 0 5 | A | 22 | | STPD | HL | ↓ | |
| | B | 11 | | - | CUR ADR | ↓ | |
| 8 | C | 20 | | - | | ↓ | |
| 2 | D | CD | | JS | | ↓ | BLANK DISPLAY. |
| | E | 5B | | - | (BLANK) | ↓ | |
| | F | 07 | | - | | ↓ | |

| HEXADECEMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|-------------|--------|--------------|----------|--|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 02 | C 0 | 1E | | LDEI | | T | ENTER AND DISPLAY 4-DIGIT PGMR. END ADDRESS. |
| | 1 | 05 | | - | Ø5 | | |
| | 2 | CD | | JS | | | |
| | 3 | 09 | | - | (ENTR 4HEX) | | |
| | 4 | 06 | | - | | | |
| | 5 | 22 | | STPD | HL | | |
| | 6 | 13 | | - | END ADR | | |
| | 7 | 20 | | - | | | |
| | 8 | CD | | JS | | | |
| | 9 | 5B | | - | (BLANK) | | |
| | A | 07 | | - | | | |
| | B | 1E | | LDEI | | T | ENTER AND DISPLAY 4-DIGIT µP SYSTEM START ADDRESS. |
| | C | 05 | | - | Ø5 | | |
| | D | CD | | JS | | | |
| | E | 09 | | - | (ENTR 4HEX) | | |
| | F | 06 | | - | | | |
| 02 | D 0 | 22 | | STPD | HL | | |
| | 1 | 15 | | - | SST ADR | | |
| | 2 | 20 | | - | | | |
| | D 3 | DB | ADR LINE HI | IPA | | T | WAIT FOR ADDRESS* LINE TO GO LOW. |
| | 4 | 51 | | - | 51 | | |
| | 5 | FE | | CPAI | | | |
| | 6 | 2C | | - | 2C | | |
| | 7 | C2 | | JP | NZ | | |
| | 8 | D3 | | - | ADR LINE HI | | |
| | 9 | 02 | | - | | | |
| | A | 21 | | LDPI | HL | | |
| | B | 12 | | - | CUR ADR+1 | | |
| | C | 20 | | - | | | |
| | D | CD | | JS | | | |
| | E | EE | | - | (WRITE ADR) | | |
| | F | 02 | | - | | | |
| 02 | E 0 | 21 | | LDPI | HL | T | TRANSFER FROM FIELD ADDRESSES TO PROGRAMMER. |
| | 1 | 14 | | - | END ADR+1 | | |
| | 2 | 20 | | - | | | |
| | 3 | CD | | JS | | | |
| | 4 | EE | | - | (WRITE ADR) | | |
| | 5 | 02 | | - | | | |
| | E 6 | DB | ADR LINE LO | IPA | | | |
| | 7 | 51 | | - | 51 | | |
| | 8 | FE | | CPAI | | | |
| | 9 | 6C | | - | 6C | | |
| | A | CA | | JP | Z | | |
| | B | E6 | | - | ADR LINE LO | | |
| | C | 02 | | - | | | |
| | D | C9 | | RTS | | | |
| | E E | CD | (WRITE ADR) | JS | | T | EXIT. |
| | F | CO | | - | (MODE*) | | |
| 02 | F 0 | 06 | | - | | T | SET MODE* LINE LOW; SET LOOP COUNT = 2 = (C); CHECK FOR M980. |
| | 1 | CA | | JP | Z | | |
| | 2 | FE | | - | 2ND HALF +10 | | |
| | 3 | 02 | | - | | T | IF M980 NOT PRESENT, SEND ONE DIGIT OF PAGE ADDRESS. |
| | F 4 | 7E | 2ND HALF | LDAN | (HL) | | |
| | 5 | 0F | | RRA | | | |
| | 6 | 0F | | RRA | | | |
| | 7 | 0F | | RRA | | | |
| | 8 | 0F | | RRA | | | |
| | 9 | D3 | | OPA | | | |
| | A | 50 | | - | 5Ø | | |
| 1080 | B | CD | | JS | | T | WRITE 2-DIGIT PAGE ADDRESS, THEN 2-DIGIT LINE ADDRESS TO PROGRAMMER. |
| | C | 0E | | - | (XFER) | | |
| | D | 03 | | - | | | |
| 1282 | F E | 7E | +10 | LDAN | (HL) | T | |
| | F | D3 | | OPA | | | |

| HEXADecimal | | | Mnemonic | | | TITLE | DATE |
|-------------|----------|--------|--------------|--------|--------------|-------|--|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | | |
| 03 | 00 | 50 | | - | 50 | | |
| | 1 | CD | | JS | | | |
| | 2 | 0E | | - | (XFER) | | |
| | 3 | 03 | | - | | | |
| | 4 | 2B | | DCP | HL | | |
| | 5 | 0D | | DCC | | | |
| | 6 | C2 | | JP | NZ | | |
| | 7 | F4 | | - | 2ND HALF | | |
| | 8 | 02 | | - | | | |
| | 9 | C9 | | RTS | | EXIT. | |
| | A | | | | | | |
| | B | | | | | | |
| | C | | | | | | |
| | D | | | | | | |
| | 0E | DB | (XFER) | IPA | | T | RESPONSE* HIGH? |
| | F | 51 | | - | 51 | | |
| 03 | 10 | 1F | | RRAC | | | |
| | 1 | DA | | JP | CI | | |
| | 2 | 0E | | - | (XFER) | | |
| | 3 | 03 | | - | | | |
| | 4 | 3E | | LDAI | | T | YES. SET TRANSFER* LOW. |
| | 5 | E0 | | - | E0 | | |
| | 6 | D3 | | OPA | | | |
| | 7 | 51 | | - | 51 | | |
| | 18 | DB | RESP HI | IPA | | T | RESPONSE* LOW? |
| | 9 | 51 | | - | 51 | | |
| | A | 1F | | RRAC | | | |
| | B | D2 | | JP | CO | | |
| | C | 18 | | - | RESP HI | | |
| | D | 03 | | - | | | |
| | E | 3E | | LDAI | | T | SET TRANSFER* HIGH. |
| | F | 60 | | - | 60 | | |
| 03 | 20 | D3 | | OPA | | | |
| | 1 | 51 | | - | 51 | | |
| | 2 | C9 | | RTS | | EXIT. | |
| | 23 | 1E | (WRITE DATA) | LDEI | | T | DISPLAY "BUSY" |
| | 4 | 05 | | - | 05 | | |
| | 5 | 21 | | LDPI | HL | | |
| | 6 | 69 | | - | "BUSY" | | |
| | 7 | 00 | | - | | | |
| | 8 | CD | | JS | | | |
| | 9 | 18 | | - | (DISP 4 ASC) | | |
| | A | 07 | | - | | | |
| | B | 2A | | LDPD | HL | T | GET ADDRESSES SET UP FOR DATA TRANSFER. |
| | C | 13 | | - | END ADR | | |
| | D | 20 | | - | | | |
| | E | EB | | XCP | HL, DE | | |
| | F | 2A | | LDPD | HL | | |
| 03 | 30 | 11 | | - | CUR ADR | | |
| | 1 | 20 | | - | | | |
| | 2 | E5 | | PSP | HL | | |
| | 3 | 2A | | LDPD | HL | | |
| | 4 | 15 | | - | SST ADR | | |
| | 5 | 20 | | - | | | |
| | 36 | 7E | WRITE NXT | LDAN | (HL) | T | GET NEXT DATA BYTE, TRANSFER IT TO PROGRAMMER. |
| | 7 | D3 | | OPA | | | |
| | 8 | 50 | | - | 50 | | |
| | 9 | CD | | JS | | | |
| 106 | A | 0E | | - | (XFER) | | |
| 8 | B | 03 | | - | | | |
| 2 | C | E3 | | XCPT | HL | | |
| 1 | D | CD | | JS | | T | LAST DATA TRANSFERRED? |
| | E | 8C | | - | (CHK END) | | |
| | F | 03 | | - | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|-------------|--------|-------------|----------|---------------------------------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 03 | 40 | CA | | JP | Z | | |
| | 1 | 49 | | - | END WRITE | | |
| | 2 | 03 | | - | | | |
| | 3 | 23 | | ICP | HL | | |
| | 4 | E3 | | XCPT | HL | | NO. INCREMENT ADDRESSES. |
| | 5 | 23 | | ICP | HL | | |
| | 6 | C3 | | JP | | | WRITE NEXT DATA. |
| | 7 | 36 | | - | WRITE NXT | | |
| | 8 | 03 | | - | | | |
| | 49 | E1 | END WRITE | PLP | HL | | |
| | A | C9 | | RTS | | | EXIT. |
| | 4B | 1E | (READ DATA) | LDEI | | | DISPLAY "BUSY". |
| | C | 05 | | - | 05 | | |
| | D | 21 | | LDPI | HL | | |
| | E | 69 | | - | "BUSY" | | |
| | F | 00 | | - | | | |
| 03 | 50 | CD | | JS | | | |
| | 1 | 18 | | - | (DISP 4ASC) | | |
| | 2 | 07 | | - | | | |
| | 3 | 2A | | LDPD | HL | | GET ADDRESSES SET UP |
| | 4 | 13 | | - | END ADR | | FOR DATA TRANSFER. |
| | 5 | 20 | | - | | | |
| | 6 | EB | | XCP | HL, DE | | |
| | 7 | 2A | | LDPD | HL | | |
| | 8 | 11 | | - | CUR ADR | | |
| | 9 | 20 | | - | | | |
| | A | E5 | | PSP | HL | | |
| | B | 2A | | LDPD | HL | | |
| | C | 15 | | - | SST ADR | | |
| | D | 20 | | - | | | |
| | E | 3E | | LDAI | | | SET MODE* LINE HIGH. |
| | F | 20 | | - | 20 | | |
| 03 | 60 | D3 | | OPA | | | |
| | 1 | 51 | | - | 51 | | |
| | 62 | DB | READ NXT | IPA | | | RESPONSE* HIGH? |
| | 3 | 51 | | - | 51 | | |
| | 4 | 1F | | RRAC | | | |
| | 5 | DA | | JP | CI | | |
| | 6 | 62 | | - | READ NXT | | |
| | 7 | 03 | | - | | | |
| | 8 | 3E | | LDAI | | | YES. SET TRANSFER* LOW. |
| | 9 | A0 | | - | A0 | | |
| | A | D3 | | OPA | | | |
| | B | 51 | | - | 51 | | |
| | 6C | DB | RSP HIGH | IPA | | | RESPONSE* LOW? |
| | D | 51 | | - | 51 | | |
| | E | 1F | | RRAC | | | |
| | F | D2 | | JP | CO | | |
| 03 | 70 | 6C | | - | RSP HIGH | | |
| | 1 | 03 | | - | | | |
| | 2 | DB | | IPA | | | YES. READ DATA BYTE FROM |
| | 3 | 52 | | - | 52 | | PROGRAMMER. |
| | 4 | 77 | | STAN | (HL) | | STORE IT IN RAM. |
| | 5 | BE | | CPAN | (HL) | | IF NOT ACCEPTED BY RAM, DISPLAY |
| | 6 | C4 | | JS | NZ | | ADDRESS AND DATA AT BAD |
| | 7 | 92 | | - | (BAD RAM) | | LOCATION. |
| | 8 | 03 | | - | | | |
| | 9 | 3E | | LDAI | | | SET TRANSFER* HIGH. |
| | A | 20 | | - | 20 | | |
| | B | D3 | | OPA | | | |
| | C | 51 | | - | 51 | | |
| | D | E3 | | XCPT | HL | | CUR PRGMR ADR = END PRGMR ADR? |
| | E | CD | | JS | | | |
| | F | 8C | | - | (CHK END) | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|------------|--------|--------------|----------|---|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 03 | 8 0 | 03 | | - | | ↓ | |
| | 1 | CA | | JP | Z | ↓ | IF YES EXIT. |
| | 2 | 8A | | - | END READ | ↓ | |
| | 3 | 03 | | - | | ↓ | |
| | 4 | 23 | | ICP | HL | ↓ | NO. INCREMENT CUR ADR'S. |
| | 5 | E3 | | XCPT | HL | ↓ | |
| | 6 | 23 | | ICP | HL | ↓ | |
| | 7 | C3 | | JP | | ↓ | READ NEXT DATA. |
| | 8 | 62 | | - | READ NXT | ↓ | |
| | 9 | 03 | | - | | ↓ | |
| | 8 A | E1 | END READ | PLP | HL | ↓ | |
| | B | C9 | | RTS | | ↓ | EXIT. |
| | 8 C | 7B | (CHK END) | LDA | E | ↓ | SUBTRACT: END ADR - CUR ADR (DE - HL) |
| | D | BD | | CPA | L | ↓ | |
| | E | C0 | | RTS | NZ | ↓ | EXIT. CUR ≠ END → Z=0 |
| | F | 7A | | LDA | D | ↓ | |
| 03 | 9 0 | BC | | CPA | H | ↓ | |
| | 1 | C9 | | RTS | | ↓ | EXIT. IF CUR=END → Z=0 |
| | 9 2 | D5 | (BAD RAM) | PSP | DE | ↓ | DISPLAY ADDRESS OF BAD RAM LOCATION. |
| | 3 | CD | | JS | | ↓ | |
| | 4 | 5B | | - | (BLANK) | ↓ | |
| | 5 | 07 | | - | | ↓ | |
| | 6 | 1E | | LDEI | | ↓ | |
| | 7 | 04 | | - | 04 | ↓ | |
| | 8 | CD | | JS | | ↓ | |
| | 9 | 60 | | - | (DISP HL) | ↓ | |
| | A | 06 | | - | | ↓ | |
| | B | 1E | | LDEI | | ↓ | DISPLAY DATA. |
| | C | 00 | | - | 00 | ↓ | |
| | D | E5 | | PSP | HL | ↓ | |
| | E | CD | | JS | | ↓ | |
| | F | 2F | | - | (DISP 2 HEX) | ↓ | |
| 03 | A 0 | 07 | | - | | ↓ | |
| | A 1 | CD | WAIT-SST | JS | | ↓ | WAIT FOR "SST" KEY TO BE DEPRESSED. |
| | 2 | 80 | | - | (READ 1 KEY) | ↓ | |
| | 3 | 07 | | - | | ↓ | |
| | 4 | FE | | CPAI | | ↓ | |
| | 5 | 10 | | - | "SST"=KEY10 | ↓ | |
| | 6 | C2 | | JP | NZ | ↓ | |
| | 7 | A1 | | - | WAIT-SST | ↓ | |
| | 8 | 03 | | - | | ↓ | |
| | 9 | CD | | JS | | ↓ | DISPLAY "BUSY" AGAIN. |
| | A | 5B | | - | (BLANK) | ↓ | |
| | B | 07 | | - | | ↓ | |
| | C | 1E | | LDEI | | ↓ | |
| | D | 05 | | - | 05 | ↓ | |
| | E | 21 | | LDPI | HL | ↓ | |
| | F | 69 | | - | "BUSY" | ↓ | |
| 03 | B 0 | 00 | | - | | ↓ | |
| | 1 | CD | | JS | | ↓ | |
| | 2 | 18 | | - | (DISP 4ASC) | ↓ | |
| | 3 | 07 | | - | | ↓ | |
| | 4 | E1 | | PLP | HL | ↓ | |
| | 5 | D1 | | PLP | DE | ↓ | |
| | 6 | C9 | | RTS | | ↓ | EXIT. |
| | B 7 | 1E | [FILL MEM] | LDEI | | ↓ | ENTER 4-DIGIT START ADDRESS AND DISPLAY IT. |
| | 8 | 05 | | - | 05 | ↓ | |
| | 9 | CD | | JS | | ↓ | |
| 1 0 6 | A | 09 | | - | (ENTR 4HEX) | ↓ | |
| | B | 06 | | - | | ↓ | |
| 2 8 | C | 22 | | STPD | HL | ↓ | |
| 1 1 | D | 11 | | - | CUR ADR | ↓ | |
| | E | 20 | | - | | ↓ | |
| | F | CD | | JS | | ↓ | BLANK DISPLAY. |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|-----------|--------|-------------|----------|---|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 03 | C 0 | 5B | | - | (BLANK) | | |
| | 1 | 07 | | - | | | |
| | 2 | 1E | | LDEI | | | |
| | 3 | 05 | | - | 05 | | ENTER 4-DIGIT END ADDRESS AND DISPLAY. |
| | 4 | CD | | JS | | | |
| | 5 | 09 | | - | (ENTR 4HEX) | | |
| | 6 | 06 | | - | | | |
| | 7 | 22 | | STPD | HL | | |
| | 8 | 13 | | - | END ADR | | |
| | 9 | 20 | | - | | | |
| | A | CD | | JS | | | BLANK DISPLAY. |
| | B | 5B | | - | (BLANK) | | |
| | C | 07 | | - | | | |
| | D | 1E | | LDEI | | | DISPLAY "DATA?" |
| | E | 07 | | - | 07 | | |
| | F | 0E | | LDCI | | | |
| 03 | D 0 | 05 | | - | 05 | | |
| | 1 | 21 | | LDPI | HL | | |
| | 2 | FF | | - | "DATA?" | | |
| | 3 | 03 | | - | | | |
| | 4 | CD | | JS | | | |
| | 5 | 1A | | - | (DISP N) | | |
| | 6 | 07 | | - | | | |
| | 7 | 1E | | LDEI | | | ENTER AND DISPLAY 2-DIGIT HEX VALUE. |
| | 8 | 01 | | - | 01 | | |
| | 9 | CD | | JS | | | |
| | A | 00 | | - | (ENTR 2HEX) | | |
| | B | 06 | | - | | | |
| | C | 45 | | LDB | L | | SAVE IT. |
| | D | CD | | JS | | | BLANK DISPLAY. |
| | E | 5B | | - | (BLANK) | | |
| | F | 07 | | - | | | |
| 03 | E 0 | 1E | | LDEI | | | DISPLAY "BUSY" |
| | 1 | 05 | | - | 05 | | |
| | 2 | 21 | | LDPI | HL | | |
| | 3 | 69 | | - | "BUSY" | | |
| | 4 | 00 | | - | | | |
| | 5 | CD | | JS | | | |
| | 6 | 18 | | - | (DISP 4ASC) | | |
| | 7 | 07 | | - | | | |
| | 8 | 2A | | LDPD | HL | | SET UP ADDRESSES TO FILL MEMORY. |
| | 9 | 13 | | - | END ADR | | |
| | A | 20 | | - | | | |
| | B | EB | | XCP | HL,DE | | |
| | C | 2A | | LDPD | HL | | |
| | D | 11 | | - | CUR ADR | | |
| | E | 20 | | - | | | |
| | E F | 78 | NXT WRITE | LDA | B | | WRITE USER ENTERED VALUE INTO RAM. |
| 03 | F 0 | 77 | | STAN | (HL) | | |
| | 1 | BE | | CPAN | (HL) | | IF NOT ACCEPTED, DISPLAY ADDRESS AND DATA OF BAD MEMORY LOCATION. |
| | 2 | C4 | | JS | NZ | | |
| | 3 | 92 | | - | (BAD RAM) | | |
| | 4 | 03 | | - | | | |
| | 5 | CD | | JS | | | CUR ADR = END ADR? |
| | 6 | 8C | | - | (CHK END) | | |
| | 7 | 03 | | - | | | |
| | 8 | CA | | JP | Z | | IF SO, EXIT. |
| | 9 | 00 | | - | RESET | | |
| 106821 | A | 00 | | - | | | |
| | B | 23 | | ICP | HL | | NO. INCREMENT ADDRESS AND FILL NEXT LOCATION. |
| | C | C3 | | JP | | | |
| | D | EF | | - | NXT WRITE | | |
| | E | 03 | | - | | | |
| | F | C4 | "DATA?" | -CST- | | | "DATA?" DISPLAY TABLE. |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|-------------|--------|-------------|----------|------------------------------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 04 | 00 | CI | | -CST- | | A | |
| | 1 | D4 | | | | T | |
| | 2 | CI | | | | A | |
| | 3 | BF | | -CST- | | ? | |
| | 4 | IE | [MOVE DATA] | LDEI | | T | ENTER 4-DIGIT SOURCE |
| | 5 | 05 | | - | Ø5 | | START ADDRESS, AND |
| | 6 | CD | | JS | | | DISPLAY. (SST) |
| | 7 | 09 | | - | (ENTR 4HEX) | | |
| | 8 | 06 | | - | | | |
| | 9 | 22 | | STPD | HL | | |
| | A | 15 | | - | SST ADR | | |
| | B | 20 | | - | | | |
| | C | CD | | JS | | T | BLANK DISPLAY. |
| | D | 5B | | - | (BLANK) | | |
| | E | 07 | | - | | | |
| | F | IE | | LDEI | | T | ENTER AND DISPLAY 4-DIGIT |
| 04 | 10 | 05 | | - | Ø5 | | SOURCE END ADDRESS. (SEN) |
| | 1 | CD | | JS | | | |
| | 2 | 09 | | - | (ENTR 4HEX) | | |
| | 3 | 06 | | - | | | |
| | 4 | 22 | | STPD | HL | | |
| | 5 | 17 | | - | SEN ADR | | |
| | 6 | 20 | | - | | | |
| | 7 | CD | | JS | | T | BLANK DISPLAY. |
| | 8 | 5B | | - | (BLANK) | | |
| | 9 | 07 | | - | | | |
| | A | IE | | LDEI | | T | ENTER AND DISPLAY 4-DIGIT |
| | B | 05 | | - | Ø5 | | DESTINATION START ADDRESS. |
| | C | CD | | JS | | | (DST) |
| | D | 09 | | - | (ENTR 4HEX) | | |
| | E | 06 | | - | | | |
| | F | 22 | | STPD | HL | | |
| 04 | 20 | 19 | | - | DST ADR | | |
| | 1 | 20 | | - | | | |
| | 2 | 21 | | LDPI | HL | T | DISPLAY "BUSY". |
| | 3 | 69 | | - | "BUSY" | | |
| | 4 | 00 | | - | | | |
| | 5 | IE | | LDEI | | | |
| | 6 | 05 | | - | Ø5 | | |
| | 7 | CD | | JS | | | |
| | 8 | 18 | | - | (DISP 4ASC) | | |
| | 9 | 07 | | - | | | |
| | A | 2A | | LDPD | HL | T | PUT SST ADR IN DE. |
| | B | 15 | | - | SST ADR | | |
| | C | 20 | | - | | | |
| | D | EB | | XCP | HL, DE | | |
| | E | 2A | | LDPD | HL | T | PUT SEN ADR IN HL. |
| | F | 17 | | - | SEN ADR | | |
| 04 | 30 | 20 | | - | | | |
| | 1 | 7D | | LDA | L | T | DETERMINE DATA BLOCK SIZE-1. |
| | 2 | 93 | | SUA | E | | |
| | 3 | 4F | | LDC | A | | |
| | 4 | 7C | | LDA | H | | |
| | 5 | 9A | | SCA | D | | |
| | 6 | 47 | | LDB | A | | RESULT IN BC AS BYTE COUNT. |
| | 7 | 2A | | LDPD | HL | T | PUT DST ADR IN HL. |
| | 8 | 19 | | - | DST ADR | | |
| | 9 | 20 | | - | | | |
| - | A | 7D | | LDA | L | T | DETERMINE MOVE DIRECTION. |
| 0 | B | 93 | | SUA | E | | |
| 8 | C | 7C | | LDA | H | | |
| 2 | D | 9A | | SCA | D | | |
| 1 | E | D2 | | JP | CO | T | SST ADR > DST ADR ⇒ BACKWARD |
| | F | 57 | | - | FORWARD | | MOVE, LOW ORDER BYTES FIRST? |

| HEXADECEIMAL | | | MNEMONIC | | | TITLE | DATE |
|--------------|-------------|--------|-------------|--------|--------------|----------|-----------------------------------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 04 | 40 | 04 | | - | | ↓ | |
| | 41 | CD | MOV BAK | JS | | ↓ | YES. DECREMENT BYTE COUNT. |
| | 2 | 71 | | - | (CHK COUNT) | ↓ | |
| | 3 | 04 | | - | | ↓ | |
| | 4 | CA | | JP | Z | ↓ | IF BYTE COUNT = 0, EXIT. |
| | 5 | 00 | | - | RESET | ↓ | |
| | 6 | 00 | | - | | ↓ | |
| | 7 | 1A | | LDAN | (DE) | ↓ | MOVE DATA TO DESTINATION |
| | 8 | 77 | | STAN | (HL) | ↓ | |
| | 9 | BE | | CPAN | (HL) | ↓ | IF NOT ACCEPTED, DISPLAY |
| | A | C5 | | PSP | BC | ↓ | ADDRESS AND DATA OF BAD |
| | B | C4 | | JS | NZ | ↓ | MEMORY LOCATION. |
| | C | 92 | | - | (BAD RAM) | ↓ | |
| | D | 03 | | - | | ↓ | |
| | E | C1 | | PLP | BC | ↓ | |
| | F | 3E | | LDAI | | ↓ | SET SOURCE = 00 (NOP) |
| 04 | 50 | 00 | | - | 00 | ↓ | |
| | 1 | 12 | | STAN | (DE) | ↓ | |
| | 2 | 13 | | ICP | DE | ↓ | GO MOVE NEXT BYTE. |
| | 3 | 23 | | ICP | HL | ↓ | |
| | 4 | C3 | | JP | | ↓ | |
| | 5 | 41 | | - | MOV BAK | ↓ | |
| | 6 | 04 | | - | | ↓ | |
| | 57 | 09 | FORWARD | ADP | HL, BC | ↓ | SET UP FORWARD MOVE, |
| | 8 | EB | | XCP | HL, DE | ↓ | HIGHER ORDER BYTES FIRST. |
| | 9 | 09 | | ADP | HL, BC | ↓ | |
| | A | EB | | XCP | HL, DE | ↓ | |
| | 5B | CD | MOV FWD | JS | | ↓ | DECREMENT BYTE COUNT. |
| | C | 71 | | - | (CHK COUNT) | ↓ | |
| | D | 04 | | - | | ↓ | |
| | E | CA | | JP | Z | ↓ | IF BYTE COUNT = 0, EXIT. |
| | F | 00 | | - | RESET | ↓ | |
| 04 | 60 | 00 | | - | | ↓ | |
| | 1 | 1A | | LDAN | (DE) | ↓ | MOVE DATA TO DESTINATION. |
| | 2 | 77 | | STAN | (HL) | ↓ | |
| | 3 | BE | | CPAN | (HL) | ↓ | IF NOT ACCEPTED, DISPLAY |
| | 4 | C5 | | PSP | BC | ↓ | ADDRESS AND DATA OF BAD |
| | 5 | C4 | | JS | NZ | ↓ | MEMORY LOCATION. |
| | 6 | 92 | | - | (BAD RAM) | ↓ | |
| | 7 | 03 | | - | | ↓ | |
| | 8 | C1 | | PLP | BC | ↓ | |
| | 9 | 3E | | LDAI | | ↓ | SET SOURCE = 00 (NOP) |
| | A | 00 | | - | 00 | ↓ | |
| | B | 12 | | STAN | (DE) | ↓ | |
| | C | 1B | | DCP | DE | ↓ | GO MOVE NEXT BYTE. |
| | D | 2B | | DCP | HL | ↓ | |
| | E | C3 | | JP | | ↓ | |
| | F | 5B | | - | MOV FWD | ↓ | |
| 04 | 70 | 04 | | - | | ↓ | |
| | 71 | 0B | (CHK COUNT) | DCP | BC | ↓ | (BC) = BYTE COUNT - 2. |
| | 2 | 3E | | LDAI | | ↓ | |
| | 3 | FE | | - | FE | ↓ | |
| | 4 | B9 | | CPA | C | ↓ | |
| | 5 | C0 | | RTS | NZ | ↓ | IF BYTE COUNT ≠ 0, EXIT WITH Z=0. |
| | 6 | 3E | | LDAI | | ↓ | |
| | 7 | FF | | - | FF | ↓ | |
| | 8 | B8 | | CPA | B | ↓ | IF BYTE COUNT = 0, Z=1. |
| - | 9 | C9 | | RTS | | ↓ | EXIT. |
| 0 | 7A | 1E | [AJUSTADR] | LDEI | | ↓ | ENTER AND DISPLAY 4-DIGIT |
| 6 | B | 05 | | - | 05 | ↓ | START ADDRESS OF FIELD OVER |
| 8 | C | CD | | JS | | ↓ | WHICH OPERAND ADDRESSES |
| 2 | D | 09 | | - | (ENTR 4 HEX) | ↓ | ARE TO BE ADJUSTED. |
| 1 | E | 06 | | - | | ↓ | |
| | F | 22 | | STPD | | ↓ | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|-----------|--------|-------------|----------|----------------------------------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 04 | 8 0 | 11 | | - | CUR ADR | | |
| | 1 | 20 | | - | | | |
| | 2 | CD | | JS | | | |
| | 3 | 5B | | - | (BLANK) | | BLANK DISPLAY. |
| | 4 | 07 | | - | | | |
| | 5 | 1E | | LDEI | | | ENTER AND DISPLAY 4-DIGIT |
| | 6 | 05 | | - | Ø5 | | FIELD END ADDRESS. |
| | 7 | CD | | JS | | | |
| | 8 | 09 | | - | (ENTR 4HEX) | | |
| | 9 | 06 | | - | | | |
| | A | 22 | | STPD | HL | | |
| | B | 13 | | - | END ADR | | |
| | C | 20 | | - | | | |
| | 8 D | 21 | CONT. CHG | LDPI | HL | | DISPLAY "BUSY". |
| | E | 69 | | - | "BUSY" | | |
| | F | 00 | | - | | | |
| 04 | 9 0 | 1E | | LDEI | | | |
| | 1 | 05 | | - | Ø5 | | |
| | 2 | CD | | JS | | | |
| | 3 | 18 | | - | (DISP 4ASC) | | |
| | 4 | 07 | | - | | | |
| | 5 | 2A | | LDPD | HL | | GET DESTINATION START ADDRESS |
| | 6 | 19 | | - | DST ADR | | FROM LAST MOVE. (DST) |
| | 7 | 20 | | - | | | |
| | 8 | E5 | | PSP | HL | | |
| | 9 | 2A | | LDPD | HL | | GET SOURCE START ADDRESS |
| | A | 15 | | - | SST ADR | | FROM LAST MOVE. (SST) |
| | B | 20 | | - | | | |
| | C | 7C | | LDA | H | | SUBTRACT: (DST-SST) |
| | D | 2F | | CMA | | | |
| | E | 67 | | LDH | A | | |
| | F | 7D | | LDA | L | | |
| 04 | A 0 | 2F | | CMA | | | |
| | 1 | 6F | | LDL | A | | |
| | 2 | 23 | | ICP | HL | | |
| | 3 | D1 | | PLP | DE | | |
| | 4 | 19 | | ADP | HL,DE | | |
| | 5 | 22 | | STPD | HL | | SAVE IT AS DISPLACEMENT. (DISPL) |
| | 6 | 1B | | - | DISPL | | |
| | 7 | 20 | | - | | | |
| | 8 | 2A | | LDPD | HL | | GET ADDRESS OF FIRST OPERAND |
| | 9 | 11 | | - | CUR ADR | | TO BE FIXED. (CUR) |
| | A | 20 | | - | | | |
| | B | 2B | | DCP | HL | | |
| | A C | 23 | ADJ LOOP | ICP | HL | | (CUR BECOMES NEXT FIXED) |
| | D | EB | | XCP | HL,DE | | |
| | E | 2A | | LDPD | HL | | GET ADDRESS OF LAST OPERAND |
| | F | 13 | | - | END ADR | | TO BE FIXED. (END) |
| 04 | B 0 | 20 | | - | | | |
| | 1 | EB | | XCP | HL,DE | | |
| | 2 | 7B | | LDA | E | | SUBTRACT: (END-CUR) |
| | 3 | 95 | | SUA | L | | |
| | 4 | 7A | | LDA | D | | |
| | 5 | 9C | | SCA | H | | |
| | 6 | DA | | JP | CI | | IF CUR > END, EXIT. |
| | 7 | 00 | | - | RESET | | |
| | 8 | 00 | | - | | | |
| | 9 | 06 | | LDBI | | | SET UP 3-WORD TABLE START |
| | A | 1A | | - | 1A | | ADDRESS. |
| | B | 11 | | LDPI | DE | | |
| | C | 42 | | - | 3 WD TABLE | | |
| | D | 05 | | - | | | |
| 1 0 6 8 2 1 | B E | 1A | CHK 3 WD | LDAN | (DE) | | IS DATA IN 3-WORD TABLE? |
| | F | BE | | CPAN | (HL) | | |

| HEXADECEIMAL | | | MNEMONIC | | | TITLE | DATE |
|--------------|----------|--------|----------|--------|------------|----------------------------------|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 04 | C 0 | CA | | JP | Z | T YES. CHECK OPERAND. | |
| | 1 | DE | | - | 3 WORD | | |
| | 2 | 04 | | - | | | |
| | 3 | 05 | | DCB | | | |
| | 4 | 13 | | ICP | DE | | |
| | 5 | C2 | | JP | NZ | | |
| | 6 | BE | | - | CHK 3 WD | | |
| | 7 | 04 | | - | | | |
| | 8 | 06 | | LDBI | | T NO. SET UP 2-WORD TABLE | |
| | 9 | 12 | | - | 12 | START ADDRESS. | |
| | A | 11 | | LDPI | DE | | |
| | B | 5C | | - | 2 WD TABLE | | |
| | C | 05 | | - | | | |
| | C D | 1A | CHK 2 WD | LDAN | (DE) | T IS DATA IN 2-WORD TABLE? | |
| | E | BE | | CPAN | (HL) | | |
| | F | CA | | JP | Z | YES. SKIP OVER OPERAND. | |
| 04 | D 0 | DA | | - | SKIP | | |
| | 1 | 04 | | - | | | |
| | 2 | 05 | | DCB | | | |
| | 3 | 13 | | ICP | DE | | |
| | 4 | C2 | | JP | NZ | | |
| | 5 | CD | | - | CHK 2 WD | | |
| | 6 | 04 | | - | | | |
| | 7 | C3 | | JP | | T NO. GO TO NEXT WORD. | |
| | 8 | AC | | - | ADJ LOOP | | |
| | 9 | 04 | | - | | | |
| | D A | 23 | SKIP | ICP | HL | T SKIP OPERAND; GO TO NEXT WORD. | |
| | B | C3 | | JP | | | |
| | C | AC | | - | ADJ LOOP | | |
| | D | 04 | | - | | | |
| | D E | E5 | 3 WORD | PSP | HL | T GET SOURCE END ADDRESS | |
| | F | 2A | | LDPD | HL | FROM LAST MOVE. (SEN) | |
| 04 | E 0 | 17 | | - | SEN ADR | | |
| | 1 | 20 | | - | | | |
| | 2 | EB | | XCP | HL, DE | | |
| | 3 | 2A | | LDPD | HL | T GET SOURCE START ADDRESS | |
| | 4 | 15 | | - | SST ADR | FROM LAST MOVE. (SST) | |
| | 5 | 20 | | - | | | |
| | 6 | 44 | | LDB | H | | |
| | 7 | 4D | | LDC | L | | |
| | 8 | E1 | | PLP | HL | | |
| | 9 | 23 | | ICP | HL | | |
| | A | 7B | | LDA | E | T SUBTRACT: (OPERAND - SEN) | |
| | B | 96 | | SUAN | (HL) | | |
| | C | 23 | | ICP | HL | | |
| | D | 7A | | LDA | D | | |
| | E | 9E | | SCAN | (HL) | | |
| | F | DA | | JP | CI | IF OPERAND > SEN, IGNORE IT. | |
| 04 | F 0 | AC | | - | ADJ LOOP | | |
| | 1 | 04 | | - | | | |
| | 2 | 2B | | DCP | HL | T SUBTRACT: (OPERAND - SST) | |
| | 3 | 7E | | LDAN | (HL) | | |
| | 4 | 91 | | SUA | C | | |
| | 5 | 23 | | ICP | HL | | |
| | 6 | 7E | | LDAN | (HL) | | |
| | 7 | 98 | | SCA | B | | |
| | 8 | DA | | JP | CI | IF OPERAND < SST, IGNORE IT. | |
| 1 | 0 | AC | | - | ADJ LOOP | | |
| 1 | 0 | A | 04 | - | | | |
| 1 | 0 | B | 2B | DCP | HL | T SST < OPERAND < SEN: | |
| 1 | 0 | C | EB | XCP | HL, DE | | |
| 1 | 0 | D | 2A | LDPD | HL | ADD DISPLACEMENT TO OPERAND. | |
| 1 | 0 | E | 1B | - | DISPL | | |
| 1 | 0 | F | 20 | - | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|-----------|--------|-------------|----------|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 05 | 00 | EB | | XCP | HL, DE | | |
| | 1 | 7E | | LDAN | (HL) | | |
| | 2 | 83 | | ADA | E | | |
| | 3 | 77 | | STAN | (HL) | | |
| | 4 | 23 | | ICP | HL | | |
| | 5 | 7E | | LDAN | (HL) | | |
| | 6 | 8A | | ACA | D | | |
| | 7 | 77 | | STAN | (HL) | | |
| | 8 | C3 | | JP | | | |
| | 9 | AC | | - | ADJ LOOP | | |
| | A | 04 | | - | | | |
| | 0B | 1E | [CHG ADR] | LDEI | | | |
| | C | 05 | | - | Ø5 | | |
| | D | CD | | JS | | | |
| | E | 09 | | - | (ENTR 4HEX) | | |
| | F | 06 | | - | | | |
| 05 | 10 | 22 | | STPD | HL | | |
| | 1 | 11 | | - | CUR ADR | | |
| | 2 | 20 | | - | | | |
| | 3 | CD | | JS | | | |
| | 4 | 5B | | - | (BLANK) | | |
| | 5 | 07 | | - | | | |
| | 6 | 1E | | LDEI | | | |
| | 7 | 05 | | - | Ø5 | | |
| | 8 | CD | | JS | | | |
| | 9 | 09 | | - | (ENTR 4HEX) | | |
| | A | 06 | | - | | | |
| | B | 22 | | STPD | HL | | |
| | C | 13 | | - | END ADR | | |
| | D | 20 | | - | | | |
| | E | CD | | JS | | | |
| | F | 5B | | - | (BLANK) | | |
| 05 | 20 | 07 | | - | | | |
| | 1 | 1E | | LDEI | | | |
| | 2 | 05 | | - | Ø5 | | |
| | 3 | CD | | JS | | | |
| | 4 | 09 | | - | (ENTR 4HEX) | | |
| | 5 | 06 | | - | | | |
| | 6 | 22 | | STPD | HL | | |
| | 7 | 15 | | - | SST ADR | | |
| | 8 | 20 | | - | | | |
| | 9 | CD | | JS | | | |
| | A | 5B | | - | (BLANK) | | |
| | B | 07 | | - | | | |
| | C | 1E | | LDEI | | | |
| | D | 05 | | - | Ø5 | | |
| | E | CD | | JS | | | |
| | F | 09 | | - | (ENTR 4HEX) | | |
| 05 | 30 | 06 | | - | | | |
| | 1 | 22 | | STPD | HL | | |
| | 2 | 17 | | - | SEN ADR | | |
| | 3 | 20 | | - | | | |
| | 4 | CD | | JS | | | |
| | 5 | 5B | | - | (BLANK) | | |
| | 6 | 07 | | - | | | |
| | 7 | 1E | | LDEI | | | |
| | 8 | 05 | | - | Ø5 | | |
| | 9 | CD | | JS | | | |
| | A | 09 | | - | (ENTR 4HEX) | | |
| | B | 06 | | - | | | |
| | C | 22 | | STPD | HL | | |
| | D | 19 | | - | DST ADR | | |
| | E | 20 | | - | | | |
| | F | C3 | | JP | | | |

↓

STORE NEW OPERAND.
 GO LOOK FOR MORE 3-WORD OPERATIONS.

↓

ENTER AND DISPLAY 4-DIGIT FIELD START ADDRESS.

↓

BLANK DISPLAY.

↓

ENTER AND DISPLAY 4-DIGIT FIELD END ADDRESS.

↓

BLANK DISPLAY.

↓

ENTER AND DISPLAY 4-DIGIT SOURCE START ADDRESS.

↓

BLANK DISPLAY.

↓

ENTER AND DISPLAY 4-DIGIT SOURCE END ADDRESS.

↓

BLANK DISPLAY.

↓

ENTER AND DISPLAY 4-DIGIT DESTINATION START ADDRESS.

↓

GO CHANGE ADDRESS OPERANDS.

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|------------|--------|------------|-----------|-----------------------------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 05 | 40 | 8D | | - | CONT. CHG | | |
| | 1 | 04 | | - | | | |
| | 42 | C3 | 3 WD TABLE | -CST- | | JP UN | TABLE OF 3-BYTE OPERATIONS. |
| | 3 | C2 | | | | " Z0 | |
| | 4 | CA | | | | " Z1 | |
| | 5 | D2 | | | | " CO | |
| | 6 | DA | | | | " C1 | |
| | 7 | E2 | | | | " PO | |
| | 8 | EA | | | | " P1 | |
| | 9 | F2 | | | | " SO | |
| | A | FA | | | | " S1 | |
| | B | CD | | | | JS UN | |
| | C | C4 | | | | Z0 | |
| | D | CC | | | | Z1 | |
| | E | D4 | | | | CO | |
| | F | DC | | | | C1 | |
| 05 | 50 | E4 | | | | PO | |
| | 1 | EC | | | | P1 | |
| | 2 | F4 | | | | SO | |
| | 3 | FC | | | | S1 | |
| | 4 | 01 | | | | LDPI BC | |
| | 5 | 11 | | | | " DE | |
| | 6 | 21 | | | | " HL | |
| | 7 | 31 | | | | " SP | |
| | 8 | 22 | | | | STPD HL | |
| | 9 | 2A | | | | LDPD HL | |
| | A | 32 | | | | STAD | |
| | B | 3A | | | | LDAD | |
| | 5C | 3E | 2 WD TABLE | | | LDAI | TABLE OF 2-BYTE OPERATIONS. |
| | D | 06 | | | | LDBI | |
| | E | 0E | | | | LDCI | |
| | F | 16 | | | | LDDI | |
| 05 | 60 | 1E | | | | LDEI | |
| | 1 | 26 | | | | LDHI | |
| | 2 | 2E | | | | LDLI | |
| | 3 | 36 | | | | LDMI (HL) | |
| | 4 | C6 | | | | ADAI | |
| | 5 | CE | | | | ACAI | |
| | 6 | D6 | | | | SUAI | |
| | 7 | DE | | | | SCAI | |
| | 8 | E6 | | | | ANAI | |
| | 9 | EE | | | | XRAI | |
| | A | F6 | | | | ORAI | |
| | B | FE | | | | CPAI | |
| | C | D3 | | | | OPA | |
| | D | DB | | -CST- | | IPA | |
| | 6E | F5 | (NMI-SVC) | PSP | AF | T | IS 7303 CARD IN THE RACK? |
| | F | CD | | JS | | | |
| 05 | 70 | 84 | | - | (CHK 7303) | | |
| | 1 | 05 | | - | | | |
| | 2 | CA | | JP | Z | | |
| | 3 | 7A | | - | NO 7303 | | |
| | 4 | 05 | | - | | | |
| | 5 | F1 | | PLP | AF | | |
| | 6 | CD | | JS | | T | DISPLAY REGISTER CONTENTS |
| | 7 | 89 | | - | (REG DUMP) | | IN 7303 DISPLAYS. |
| | 8 | 05 | | - | | | |
| | 9 | C9 | | RTS | | | EXIT. |
| 106821 | 7A | F1 | NO 7303 | PLP | AF | T | PUSH AND PULL REGISTERS |
| | B | F5 | | PSP | AF | | TO DISPLAY THEIR CONTENTS |
| | C | C5 | | PSP | BC | | ON ANALYZER. |
| | D | D5 | | PSP | DE | | |
| | E | E5 | | PSP | HL | | |
| | F | E1 | | PLP | HL | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|---------------|-----------|--------|----------------|----------------------------------|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 05 | 8 0 | D1 | | PLP | DE | | |
| | 1 | C1 | | PLP | BC | | |
| | 2 | F1 | | PLP | AF | | |
| | 3 | C9 | | RTS | | EXIT. | |
| | 8 4 | DB (CHK 7303) | IPA | | | T READ 7303 INPUT PORT. | |
| | 5 | DO | | - | DO | | |
| | 6 | FE | | CPAI | | T IF DATA = FE, 7303 NOT THERE, | |
| | 7 | FF | | - | FF | T AND Z=1. | |
| | 8 | C9 | | RTS | | EXIT. | |
| | 8 9 | 22 (REG DUMP) | STPD | HL | | T STORE REGISTER AND FLAG | |
| | A | 00 | | - | RSAVE | T CONTENTS FOR DISPLAY. | |
| | B | 20 | | - | | | |
| | C | EB | | XCP | HL,DE | | |
| | D | 22 | | STPD | HL | | |
| | E | 02 | | - | RSAVE+2 | | |
| | F | 20 | | - | | | |
| 05 | 9 0 | C5 | | PSP | BC | | |
| | 1 | E1 | | PLP | HL | | |
| | 2 | 22 | | STPD | HL | | |
| | 3 | 04 | | - | RSAVE+4 | | |
| | 4 | 20 | | - | | | |
| | 5 | F5 | | PSP | AF | | |
| | 6 | E1 | | PLP | HL | | |
| | 7 | 22 | | STPD | HL | | |
| | 8 | 06 | | - | RSAVE+6 | | |
| | 9 | 20 | | - | | | |
| | A | 21 | | LDPI | HL | T SET UP REG. PAIR NAME TABLE | |
| | B | F0 | | - | REG NAME | T START ADDRESS. | |
| | C | 05 | | - | | | |
| | D | E5 | | PSP | HL | | |
| | E | 21 | | LDPI | HL | T SET UP REG. CONTENTS | |
| | F | 00 | | - | RSAVE | T START ADDRESS. | |
| 05 | A 0 | 20 | | - | | | |
| | 1 | 06 | | LDRI | | T SET PAIR COUNT = 4 | |
| | 2 | 04 | | - | 04 | | |
| | A 3 | 1E | NXT PAIR | LDEI | | T DISPLAY CONTENTS OF NEXT PAIR. | |
| | 4 | 00 | | - | 00 | | |
| | 5 | CD | | JS | | | |
| | 6 | B8 | | - | (DISP DATA) | | |
| | 7 | 06 | | - | | | |
| | 8 | E3 | | XCPT | HL | | |
| | 9 | CD | | JS | | T DISPLAY PAIR NAME | |
| | A | 18 | | - | (DISP 4ASC) | | |
| | B | 07 | | - | | | |
| | C | E3 | | XCPT | HL | | |
| | A D | CD | "SST" KEY | JS | | T READ KEYBOARD. | |
| | E | 80 | | - | (READ 1 KEY) | | |
| | F | 07 | | - | | | |
| 05 | B 0 | FE | | CPAI | | T IS KEY HEX (0-F)? | |
| | 1 | 10 | | - | "SST" = KEY 10 | | |
| | 2 | DC | | JS | C1 | YES. GO CHANGE REG. CONTENTS. | |
| | 3 | D2 | | - | (CHG REG) | | |
| | 4 | 05 | | - | | | |
| | 5 | C2 | | JP | NZ | T NO. IF NOT KEY "SST", WAIT | |
| | 6 | AD | | - | "SST" KEY | T FOR IT. | |
| | 7 | 05 | | - | | | |
| | 8 | 05 | | DCB | | T DO FOUR PAIRS (HL,DE,BC,AF). | |
| 106890 | 9 | C2 | | JP | NZ | | |
| | A | A3 | | - | NXT PAIR | | |
| | B | 05 | | - | | | |
| 128821 | C | E1 | | PLP | HL | | |
| | D | CD | | JS | | T BLANK DISPLAY. | |
| | E | 5B | | - | (BLANK) | | |
| | F | 07 | | - | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|-----------|--------|----------------|--|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 05 | C0 | 2A | | LDPD | HL | T RESTORE CURRENT REGISTER CONTENTS AND FLAGS. | |
| | 1 | 06 | | - | RSAVE+6 | | |
| | 2 | 20 | | - | | | |
| | 3 | E5 | | PSP | HL | | |
| | 4 | F1 | | PLP | AF | | |
| | 5 | 2A | | LDPD | HL | | |
| | 6 | 04 | | - | RSAVE+4 | | |
| | 7 | 20 | | - | | | |
| | 8 | E5 | | PSP | HL | | |
| | 9 | C1 | | PLP | BC | | |
| | A | 2A | | LDPD | HL | | |
| | B | 02 | | - | RSAVE+2 | | |
| | C | 20 | | - | | | |
| | D | EB | | XCP | HL,DE | | |
| | E | 2A | | LDPD | HL | | |
| | F | 00 | | - | RSAVE | | |
| 05 | D0 | 20 | | - | | ↓ | |
| | 1 | C9 | | RTS | | EXIT. | |
| | D2 | C5 | (CHG REG) | PSP | BC | T ENTER AND DISPLAY 4-DIGIT HEX VALUE. | |
| | 3 | D5 | | PSP | DE | | |
| | 4 | 2B | | DCP | HL | | |
| | 5 | 2B | | DCP | HL | | |
| | 6 | E5 | | PSP | HL | | |
| | 7 | 01 | | LDPI | BC | | |
| | 8 | 05 | | - | 05 | | |
| | 9 | 05 | | - | 05 | | |
| | A | 1E | | LDEI | | | |
| | B | 03 | | - | 03 | | |
| | C | 21 | | LDPI | HL | | |
| | D | 0B | | - | ENTR BUFR | | |
| | E | 20 | | - | | | |
| | F | CD | | JS | | | |
| 05 | E0 | 17 | | - | (ENTR 4HEX)+14 | | |
| | 1 | 06 | | - | | | |
| | 2 | EB | | XCP | HL,DE | | |
| | 3 | E1 | | PLP | HL | ↓ | |
| | 4 | 73 | | STEN | (HL) | T STORE VALUE AS REGISTER CONTENTS. | |
| | 5 | 23 | | ICP | HL | | |
| | 6 | 72 | | STDN | (HL) | ↓ | |
| | 7 | 23 | | ICP | HL | | |
| | 8 | D1 | | PLP | DE | | |
| | 9 | C1 | | PLP | BC | | |
| | A | 3E | | LDAI | | T SET Z=1. | |
| | B | 10 | | - | 10 | | |
| | C | FE | | CPAI | | | |
| | D | 10 | | - | 10 | ↓ | |
| | E | C9 | | RTS | | EXIT. | |
| | F | | | | | | |
| 05 | F0 | C8 | REG NAME | -CST- | | H REGISTER PAIR NAME TABLE. | |
| | 1 | CC | | | | L | |
| | 2 | BD | | | | = | |
| | 3 | A0 | | | | (SP) | |
| | 4 | C4 | | | | D | |
| | 5 | C5 | | | | E | |
| | 6 | BD | | | | = | |
| | 7 | A0 | | | | (SP) | |
| | 8 | C2 | | | | B | |
| 106821 | 9 | C3 | | | | C | |
| | A | BD | | | | = | |
| | B | A0 | | | | (SP) | |
| | C | C1 | | | | A | |
| | D | C6 | | | | F | |
| | E | BD | | | | = | |
| | F | A0 | | | | (SP) | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|-------------|--------|----------------|---|---------------------------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 06 | 00 | 01 | (ENTR 2HEX) | LDPI | BC | T SET ENTRY COUNT = 2 (+1) | |
| | 1 | 03 | | - | Ø3 | ↓ | |
| | 2 | 03 | | - | Ø3 | ↓ | |
| | 3 | 21 | | LDPI | HL | T SET DATA ENTRY BUFFER START ADDRESS. | |
| | 4 | 0D | | - | ENTR BUFR+2 | ↓ | |
| | 5 | 20 | | - | | ↓ | |
| | 6 | C3 | | JP | | ↓ | |
| | 7 | 0F | | - | NXT ENTRY | ↓ | |
| | 8 | 06 | | - | | ↓ | |
| | 09 | CD | (ENTR 4HEX) | JS | | T SET ENTRY COUNT = 4 (+1); | |
| | A | CD | | - | (DISP "----") | ↓ | DISPLAY "----" AS PROMPT. |
| | B | 06 | | - | | ↓ | |
| | C | 21 | | LDPI | HL | T SET DATA ENTRY BUFFER START ADDRESS. | |
| | D | 0B | | - | ENTR BUFR | ↓ | |
| | E | 20 | | - | | ↓ | |
| | 0F | CD | NXT ENTRY | JS | | T READ KEYBOARD FOR ENTRY. | |
| 06 | 10 | 80 | | - | (READ 1 KEY) | ↓ | |
| | 1 | 07 | | - | | ↓ | |
| | 2 | FE | | CPAI | | T NON-HEX ENTRY? | |
| | 3 | 10 | | - | "SST" = KEY 10 | ↓ | |
| | 4 | D2 | | JP | CO | ↓ | |
| | 5 | 2B | | - | NOT HEX | ↓ | |
| | 6 | 06 | | - | | ↓ | |
| | 17 | 0D | +14 | DCC | | T NO. ENTRY BUFFER FULL? | |
| | 8 | CA | | JP | Z | ↓ | |
| | 9 | 27 | | - | BUFR FULL | ↓ | |
| | A | 06 | | - | | ↓ | |
| | B | 77 | | STAN | (HL) | T NO. STORE ENTRY AND DISPLAY. | |
| | C | 57 | | LDD | A | ↓ | |
| | D | D5 | | PSP | DE | ↓ | |
| | E | CD | | JS | | ↓ | |
| | F | 68 | | - | (DISP HEX) | ↓ | |
| 06 | 20 | 07 | | - | | ↓ | |
| | 1 | D1 | | PLP | DE | ↓ | |
| | 2 | 1D | | DCE | | ↓ | |
| | 3 | 23 | | ICP | HL | ↓ | |
| | 4 | C3 | | JP | | ↓ | |
| | 5 | 0F | | - | NXT ENTRY | ↓ | |
| | 6 | 06 | | - | | ↓ | |
| | 27 | 0C | BUFR FULL | ICC | | T BUFFER FULL; LOOK FOR CORRECTION OR ACCEPTANCE. | |
| | 8 | C3 | | JP | | ↓ | |
| | 9 | 0F | | - | NXT ENTRY | ↓ | |
| | A | 06 | | - | | ↓ | |
| | 2B | CA | NOT HEX | JP | Z | T IS KEY "SST"? | |
| | C | 43 | | - | ACCEPT ENTR | ↓ | |
| | D | 06 | | - | | ↓ | |
| | E | FE | | CPAI | | T NO. IS KEY "CLR"? | |
| | F | 13 | | - | "CLR" = KEY 13 | ↓ | |
| | 0 | C2 | | JP | NZ | ↓ | |
| | 1 | 0F | | - | NXT ENTRY | ↓ | |
| | 2 | 06 | | - | | ↓ | |
| | 3 | 79 | | LDA | C | T YES. IS THIS VERY FIRST ENTRY? | |
| | 4 | B8 | | CPA | B | ↓ | |
| | 5 | CA | | JP | Z | ↓ | |
| | 6 | 0F | | - | NXT ENTRY | ↓ | |
| | 7 | 06 | | - | | ↓ | |
| | 8 | 1C | | ICE | | T NO. BLANK LAST ENTRY DISPLAYED; | |
| | 9 | 2B | | DGP | HL | POINT TO PREVIOUS BUFFER | |
| | A | 0C | | ICC | | LOCATION FOR CORRECTED ENTRY. | |
| | B | 16 | | LDDI | | ↓ | |
| | C | A0 | | - | AØ | ↓ | |
| | D | CD | | JS | | ↓ | |
| | E | 6D | | - | (DISPLAY) | ↓ | |
| | F | 07 | | - | | ↓ | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|-------------|--------|------------|---|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 06 | 4 0 | C3 | | JP | | | |
| | 1 | OF | | - | NXT ENTRY | | |
| | 2 | 06 | | - | | | |
| | 4 3 | 79 | ACCEPT ENTR | LDA | C | T BUFFER FULL? | |
| | 4 | FE | | CPAI | | | |
| | 5 | 01 | | - | Ø1 | | |
| | 6 | C2 | | JP | NZ | | |
| | 7 | OF | | - | NXT ENTRY | | |
| | 8 | 06 | | - | | | |
| | 9 | 21 | | LDPI | HL | T YES. SET ENTRY BUFFER | |
| | A | 0B | | - | ENTR BUFR | ADDRESS TO FIRST ENTRY. | |
| | B | 20 | | - | | | |
| | C | 7E | | LDAN | (HL) | T COMBINE 1ST 2 ENTRIES, | |
| | D | 07 | | RLA | | PUT IN REG. D. | |
| | E | 07 | | RLA | | | |
| | F | 07 | | RLA | | | |
| | 0 | 07 | | RLA | | | |
| | 1 | 23 | | ICP | HL | | |
| | 2 | B6 | | ORAN | (HL) | | |
| | 3 | 57 | | LDD | A | | |
| | 4 | 23 | | ICP | HL | T COMBINE 2ND 2 ENTRIES, | |
| | 5 | 7E | | LDAN | (HL) | PUT IN REG. E. | |
| | 6 | 07 | | RLA | | | |
| | 7 | 07 | | RLA | | | |
| | 8 | 07 | | RLA | | | |
| | 9 | 07 | | RLA | | | |
| | A | 23 | | ICP | HL | | |
| | B | B6 | | ORAN | (HL) | | |
| | C | 5F | | LDE | A | | |
| | D | EB | | XCP | HL, DE | ENTRIES IN HL REG. PAIR. | |
| | E | C9 | | RTS | | EXIT. | |
| | F | | | | | | |
| 06 | 6 0 | 55 | (DISP HL) | LDD | L | T DISPLAY LSD OF REG. L IN | |
| | 1 | CD | | JS | | DISPLAY DESIGNATED BY (E). | |
| | 2 | 68 | | - | (DISP HEX) | | |
| | 3 | 07 | | - | | | |
| | 4 | 1C | | ICE | | | |
| | 5 | 7D | | LDA | L | T DISPLAY MSD OF REG. L IN | |
| | 6 | 0F | | RRA | | NEXT DISPLAY TO LEFT. | |
| | 7 | 0F | | RRA | | | |
| | 8 | 0F | | RRA | | | |
| | 9 | 0F | | RRA | | | |
| | A | 57 | | LDD | A | | |
| | B | CD | | JS | | | |
| | C | 68 | | - | (DISP HEX) | | |
| | D | 07 | | - | | | |
| | E | 1C | | ICE | | | |
| | F | 54 | | LDD | H | T DISPLAY LSD OF REG. H IN | |
| 06 | 7 0 | CD | | JS | | 2ND DISPLAY TO LEFT. | |
| | 1 | 68 | | - | (DISP HEX) | | |
| | 2 | 07 | | - | | | |
| | 3 | 1C | | ICE | | | |
| | 4 | 7C | | LDA | H | T DISPLAY MSD OF REG. H IN | |
| | 5 | 0F | | RRA | | 3RD DISPLAY TO LEFT. | |
| | 6 | 0F | | RRA | | | |
| | 7 | 0F | | RRA | | | |
| | 8 | 0F | | RRA | | | |
| 1 | 9 | 57 | | LDD | A | | |
| 06 | A | CD | | JS | | | |
| 8 | B | 68 | | - | (DISP HEX) | | |
| 2 | C | 07 | | - | | | |
| 8 | D | C9 | | RTS | | EXIT. (E) _{EXIT} = (E) _{ENTRY} + 3. | |
| | E | | | | | | |
| | F | | | | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|--------------|--------|---------------|---|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 06 | 8 0 | 1E | (MESSAGE) | LDEI | | T DISPLAY MESSAGE FROM LEFT TO RIGHT, ADDING 1 MORE CHARACTER TO RIGHT SIDE OF DISPLAY EACH 200 MSECS. | |
| | 1 | 08 | | - | 08 | | |
| | 8 2 | 56 | 8 CHAR'S | LDDN | (HL) | | |
| | 3 | 1D | | DCE | | | |
| | 4 | F5 | | PSP | AF | | |
| | 5 | CD | | JS | | | |
| | 6 | 6D | | - | (DISPLAY) | | |
| | 7 | 07 | | - | | | |
| | 8 | F1 | | PLP | AF | | |
| | 9 | 23 | | ICP | HL | | |
| | A | C2 | | JP | NZ | | |
| | B | 82 | | - | 8 CHAR'S | | |
| | C | 06 | | - | | | |
| | D | CD | | JS | | | |
| | E | B1 | | - | (200 MSEC ΔT) | | |
| | F | 06 | | - | | | |
| 06 | 9 0 | 7E | | LDAN | (HL) | | |
| | 1 | FE | | CPAI | | | |
| | 2 | FF | | - | FF | FF = END. | |
| | 3 | C8 | | RTS | Z | EXIT IF END OF MESSAGE. | |
| | 4 | 7D | | LDA | L | | |
| | 5 | D6 | | SUAI | | | |
| | 6 | 07 | | - | 07 | | |
| | 7 | 6F | | LDL | A | | |
| | 8 | 7C | | LDA | H | | |
| | 9 | DE | | SCAI | | | |
| | A | 00 | | - | 00 | | |
| | B | 67 | | LDH | A | | |
| | C | C3 | | JP | | | |
| | D | 80 | | - | (MESSAGE) | | |
| | E | 06 | | - | | | |
| | F | | | | | | |
| 06 | A 0 | DB | (CHK S1'S2) | IPA | | T READ 7303 INPUT PORT | |
| | 1 | DO | | - | D0 | | |
| | 2 | FE | | CPAI | | T IS 7307 THERE? | |
| | 3 | FF | | - | FF | | |
| | 4 | C8 | | RTS | Z | NO. EXIT WITH Z=1. | |
| | 5 | FE | | CPAI | | T YES. S1, S2 BOTH CLOSED? | |
| | 6 | C0 | | - | C0 | | |
| | 7 | C9 | | RTS | | EXIT. YES: Z=1; NO: Z=0 | |
| | A 8 | CD | (1 SEC ΔT) | JS | | T DELAY ~ 1 SECOND. | |
| | 9 | AB | | - | (500MSECΔT) | | |
| | A | 06 | | - | | | |
| | A B | 01 | (500 MSECΔT) | LDPI | BC | DELAY 500 MILLISECONDS. | |
| | C | 4F | | - | FEAF | | |
| | D | FE | | - | | | |
| | E | C3 | | JP | | | |
| | F | F7 | | - | (LONG ΔT) | | |
| 06 | B 0 | 07 | | - | | | |
| | B 1 | 01 | (200 MSECΔT) | LDPI | BC | T DELAY 200 MILLISECONDS. | |
| | 2 | B8 | | - | 65B8 | | |
| | 3 | 65 | | - | | | |
| | 4 | C3 | | JP | | | |
| | 5 | F7 | | - | (LONG ΔT) | | |
| | 6 | 07 | | - | | | |
| | 7 | | | | | | |
| | B 8 | 0E | (DISP DATA) | LDCI | | T DISPLAY 2 MEMORY LOCATIONS | |
| | 9 | 02 | | - | 02 | HEX DATA IN POSITIONS | |
| | A | CD | | JS | | (E) ENTRY TO (E)+3 | |
| | B | 27 | | - | (DISP 2NHEX) | | |
| | C | 07 | | - | | | |
| | D | 1E | | LDEI | | | |
| | E | 07 | | - | 07 | | |
| | F | C9 | | RTS | | EXIT. | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|-------|---------------|--------|-----------|--|------|
| PAGE ADR | LINE ADR | INSTR | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 06 | C 0 | 3E | (MODE*) | LDAI | | T SET MODE* LINE LOW. | |
| | 1 | 60 | | - | 00 | ↓ | |
| | 2 | D3 | | OPA | | | |
| | 3 | 51 | | - | 51 | ↓ | |
| | 4 | 0E | | LDCI | | T SET ADDRESS BYTE COUNT = 2. | |
| | 5 | 02 | | - | 02 | ↓ | |
| | 6 | DB | | IPA | | T SET M980 FLAG | |
| | 7 | 52 | | - | 52 | ↓ | |
| | 8 | A7 | | ANA | A | | |
| | 9 | C8 | | RTS | Z | EXIT. IF (A) ≠ 0, THEN Z = 0 ⇒ M980 PRESENT, | |
| | A | FE | | CPAI | | OR 9118 PRESENT. TEST FOR M900B | |
| | B | FF | | - | FF | WITH 9118 ⇒ Z = 1. | |
| | C | C9 | | RTS | | EXIT. IF Z = 0, M980. | |
| | C D | D5 | (DISP "----") | PSP | DE | T DISPLAY 4 UNDERLINES (----) IN | |
| | E | 11 | | LDPI | DE | POSITIONS 2-5 AS A PROMPT. | |
| | F | 02 | | - | 02 | | |
| 06 | D 0 | DF | | - | DF = "-" | | |
| | 1 | 0E | | LDCI | | ↓ | |
| | 2 | 04 | | - | 04 | | |
| | D 3 | CD | NEXT "-" | JS | | | |
| | 4 | 6D | | - | (DISPLAY) | | |
| | 5 | 07 | | - | | | |
| | 6 | 1C | | ICE | | | |
| | 7 | 0D | | DCC | | | |
| | 8 | C2 | | JP | NZ | | |
| | 9 | D3 | | - | NEXT "-" | | |
| | A | 06 | | - | | ↓ | |
| | B | 01 | | LDPI | BC | T SET ENTRY COUNT = 4(+). | |
| | C | 05 | | - | 05 | ↓ | |
| | D | 05 | | - | 05 | | |
| | E | D1 | | PLP | DE | | |
| | F | C9 | | RTS | | EXIT. | |
| 06 | E 0 | | | | | | |
| | 1 | | | | | | |
| | 2 | | | | | | |
| | 3 | | | | | | |
| | 4 | | | | | | |
| | 5 | | | | | | |
| | 6 | | | | | | |
| | 7 | | | | | | |
| | 8 | | | | | | |
| | 9 | | | | | | |
| | A | | | | | | |
| | B | | | | | | |
| | C | | | | | | |
| | D | | | | | | |
| | E | | | | | | |
| | F | | | | | | |
| 06 | F 0 | | | | | | |
| | 1 | | | | | | |
| | 2 | | | | | | |
| | 3 | | | | | | |
| | 4 | | | | | | |
| | 5 | | | | | | |
| | 6 | | | | | | |
| | 7 | | | | | | |
| | 8 | | | | | | |
| 106821 | 9 | | | | | | |
| | A | | | | | | |
| | B | | | | | | |
| | C | | | | | | |
| | D | | | | | | |
| | E | | | | | | |
| | F | | | | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|--------------|--------|---------------|----------|--|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 07 | 0 | C5 | (FLASHER) | PSP | BC | T | DISPLAY CURSOR PATTERN |
| | 1 | D5 | | PSP | DE | | FROM REG. D. |
| | 2 | CD | | JS | | | |
| | 3 | 42 | | - | (CURSORS) | | |
| | 4 | 07 | | - | | | |
| | 5 | 3E | | LDAI | | T | TURN ON ALL LED'S. |
| | 6 | FF | | - | FF | | |
| | 7 | D3 | | OPA | | | |
| | 8 | D0 | | - | D0 | | |
| | 9 | CD | | JS | | T | DELAY ~ 200 MSEC'S. |
| | A | B1 | | - | (200 MSEC AT) | | |
| | B | 06 | | - | | | |
| | C | CD | | JS | | T | TURN OFF ALL CURSORS |
| | D | 40 | | - | (CLR CURSORS) | | AND LED'S. |
| | E | 07 | | - | | | |
| | F | CD | | JS | | T | DELAY ~ 500 MSEC'S. |
| 07 | 1 | 0B | | - | (500 MSEC AT) | | |
| | 1 | 06 | | - | | | |
| | 2 | D1 | | PLP | DE | | |
| | 3 | C1 | | PLP | BC | | |
| | 4 | C9 | | RTS | | | EXIT. |
| | 5 | | | | | | |
| | 6 | | | | | | |
| | 7 | | | | | | |
| | 8 | 0E | (DISP 4ASC) | LDCI | | T | DISPLAY 4 CHARACTERS FROM TABLE. |
| | 9 | 04 | | - | 04 | | (*ASCII CHAR'S) |
| | A | 56 | (DISP N) | LDDN | (HL) | T | DISPLAY N CHAR'S FROM TABLE. |
| | B | CD | | JS | | | N=(C). |
| | C | 6D | | - | (DISPLAY) | | LEFTMOST DISPLAY POSITION = (E)ENTRY |
| | D | 07 | | - | | | LEFTMOST DISPLAY CHAR. = (HL)ENTRY |
| | E | 1D | | DCE | | | |
| | F | 23 | | ICP | HL | | |
| 07 | 2 | 0D | | DCC | | | |
| | 1 | C2 | | JP | NZ | | |
| | 2 | 1A | | - | (DISP N) | | |
| | 3 | 07 | | - | | | |
| | 4 | C9 | | RTS | | | EXIT WITH (E)EXIT=(E)ENTRY N; (HL)EXIT=(HL)ENTRY+N |
| | 5 | | | | | | |
| | 6 | | | | | | |
| | 7 | CD | (DISP 2NHEX) | JS | | T | DISPLAY N 2HEX VALUES FROM |
| | 8 | 2F | | - | (DISP 2HEX) | | MEMORY IN 2N DISPLAYS FROM |
| | 9 | 07 | | - | | | RIGHT TO LEFT. |
| | A | 0D | | DCC | | | N=(C). |
| | B | C2 | | JP | NZ | | |
| | C | 27 | | - | (DISP 2 NHEX) | | |
| | D | 07 | | - | | | |
| | E | C9 | | RTS | | | EXIT. (HL)EXIT=(HL)+N; (E)EXIT=(E)+2N |
| | F | 56 | (DISP 2HEX) | LDDN | (HL) | T | DISPLAY 2 HEX CHARACTERS IN |
| 07 | 3 | CD | | JS | | | MEMORY LOCATION (HL), IN |
| | 1 | 68 | | - | (DISP HEX) | | DISPLAY POSITIONS (E), (E)+1 |
| | 2 | 07 | | - | | | |
| | 3 | 1C | | ICE | | | |
| | 4 | 7E | | LDAN | (HL) | | |
| | 5 | 0F | | RRA | | | |
| | 6 | 0F | | RRA | | | |
| | 7 | 0F | | RRA | | | |
| | 8 | 0F | | RRA | | | |
| | 9 | 57 | | LDD | A | | |
| 1 | A | CD | | JS | | | |
| 0 | B | 68 | | - | (DISP HEX) | | |
| 8 | C | 07 | | - | | | |
| 2 | D | 1C | | ICE | | | (E)EXIT=(E)+2 |
| 1 | E | 23 | | ICP | HL | | (HL)EXIT=(HL)+1 |
| | F | C9 | | RTS | | | EXIT. |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|----------|--------|---------------|--------|-------------------|--|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 07 | 4 0 | 16 | (CLR CURSORS) | LDDI | - | T ENTER WITH REGISTER D PRE-LOADED | |
| | 1 | 00 | | - | ∅∅ | ↓ WITH CURSOR MASK: 0=OFF; 1=ON. | |
| | 4 2 | 3E | (CURSORS) | LDAI | - | T | |
| | 3 | 0F | | - | POSITIONS 3,2,1,0 | ↓ REGISTER D BITS 0-7=DISPLAY | |
| | 4 | A2 | | ANA | D | ↓ POSITIONS 0-7. SET MASK FOR BITS 0-3. | |
| | 5 | D3 | | OPA | - | T SET $\overline{C_U}$ LOW; OUTPUT MASK. | |
| | 6 | D0 | | - | D∅ | ↓ (A)=($\overline{C_U}$, 0,0,0,X,X,X,X) X=CURS. BIT. | |
| | 7 | 3E | | LDAI | - | T STROBE RIGHT SIDE OF 7303 | |
| | 8 | 00 | | - | RIGHT | ↓ DISPLAY. | |
| | 9 | CD | | JS | - | | |
| | A | 73 | | - | (STROBE) | | |
| | B | 07 | | - | - | ↓ | |
| | C | 7A | | LDA | D | T SET MASK FOR BITS 4-7. | |
| | D | 0F | | RRA | - | | |
| | E | 0F | | RRA | - | | |
| | F | 0F | | RRA | - | | |
| 07 | 5 0 | 0F | | RRA | - | | |
| | 1 | E6 | | ANAI | - | | |
| | 2 | 0F | | - | ∅F | ↓ | |
| | 3 | D3 | | OPA | - | T SET $\overline{C_U}$ LOW; OUTPUT MASK. | |
| | 4 | D0 | | - | D∅ | ↓ | |
| | 5 | 3E | | LDAI | - | T STROBE LEFT SIDE OF DISPLAY. | |
| | 6 | 04 | | - | LEFT | | |
| | 7 | C3 | | JP | - | | |
| | 8 | 73 | | - | (STROBE) | ↓ | |
| | 9 | 07 | | - | - | | |
| | A | | | | | | |
| | 5 B | 16 | (BLANK) | LDDI | - | T OUTPUT BLANKS TO ALL DISPLAYS. | |
| | C | A0 | | - | SPACE | ↓ (ASCII SPACE = BLANK) | |
| | 5 D | 1E | (DISP 8) | LDEI | - | T DISPLAY ASCII CHAR. IN (D) IN | |
| | E | F8 | | - | F8 | ↓ ALL 8 DISPLAYS, FROM LEFT TO RIGHT. | |
| | 5 F | CD | NEXT DISP | JS | - | | |
| 07 | 6 0 | 6D | | - | (DISPLAY) | | |
| | 1 | 07 | | - | - | ↓ | |
| | 2 | 1C | | ICE | - | T (E) = ∅∅ ⇒ LAST DISPLAY? | |
| | 3 | C2 | | JP | NZ | | |
| | 4 | 5F | | - | NEXT DISP | ↓ NO. DO NEXT DISPLAY. | |
| | 5 | 07 | | - | - | | |
| | 6 | C9 | | RTS | - | YES. EXIT. | |
| | 7 | | | | | | |
| | 6 8 | 7A | (DISP HEX) | LDA | D | T DISPLAY LEAST SIGNIFICANT HEX | |
| | 9 | CD | | JS | - | ↓ DIGIT IN REGISTER D. | |
| | A | E0 | | - | (HEX → ASCII) | | |
| | B | 07 | | - | - | | |
| | C | 57 | | LDD | A | ↓ | |
| | 6 D | 7A | (DISPLAY) | LDA | D | T OUTPUT ASCII CHARACTER TO 7303 | |
| | E | F6 | | ORAI | - | ↓ DISPLAY DATA PORT. | |
| | F | 80 | | - | PARITY BIT | | |
| 07 | 7 0 | D3 | | OPA | - | | |
| | 1 | D0 | | - | D∅ | ↓ | |
| | 2 | 7B | | LDA | E | T | |
| | 7 3 | E6 | (STROBE) | ANAI | - | OUTPUT DISPLAY POSITION ADDRESS | |
| | 4 | F7 | | - | WRITE BIT | 0-7 IN BITS 0-2 TO CONTROL PORT; | |
| | 5 | D3 | | OPA | - | SET WRITE LINE HIGH (BIT 3)=∅. | |
| | 6 | D1 | | - | D1 | ↓ (A)=(X,X,X,X,WR,A2,A1,A0) | |
| | 7 | F6 | | ORAI | - | T SET WRITE LINE LOW = 1. | |
| | 8 | 08 | | - | WRITE BIT | | |
| 1 0 6 | 9 | D3 | | OPA | - | ↓ | |
| 8 2 | A | D1 | | - | D1 | T SET WRITE LINE HIGH = ∅. | |
| | B | EE | | XRAI | - | | |
| | C | 08 | | - | WRITE BIT | | |
| | D | D3 | | OPA | - | ↓ | |
| | E | D1 | | - | D1 | T NOTE: X=DONT CARE; A2 = $\overline{C_E}$. | |
| | F | C9 | | RTS | - | EXIT. | |



| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|--------------|--------|--------------|--------------------------------|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 07 | 80 | CD | (READ 1 KEY) | JS | | T CHECK FOR ANY KEY CLOSED. | |
| | 1 | D6 | | - | (SCAN KBD) | | |
| | 2 | 07 | | - | | | |
| | 3 | C2 | | JP | NZ | T IF SO, WAIT UNTIL OPEN. | |
| | 4 | 80 | | - | (READ 1 KEY) | | |
| | 5 | 07 | | - | | | |
| | 6 | CD | | JS | | T DEBOUNCE OPENING. | |
| | 7 | EE | | - | (DEBOUNCE) | | |
| | 8 | 07 | | - | | | |
| | 9 | CD | | JS | | | |
| | A | D6 | | - | (SCAN KBD) | | |
| | B | 07 | | - | | | |
| | C | C2 | | JP | NZ | | |
| | D | 80 | | - | (READ 1 KEY) | | |
| | E | 07 | | - | | | |
| | 8F | D5 | (READ KBD) | PSP | DE | T SCAN KEYBOARD BY ENABLING | |
| 07 | 90 | C5 | | PSP | BC | ONE COLUMN AT A TIME AND | |
| | 1 | 1C | | LDDI | | LOOKING FOR A KEY CLOSURE. | |
| | 2 | 88 | | - | 88 | | |
| | 93 | 7A | NXT COLM | LDA | D | | |
| | 4 | 07 | | RLA | | | |
| | 5 | 57 | | LDD | A | | |
| | 6 | E6 | | ANAI | | | |
| | 7 | 0F | | - | 0F | | |
| | 8 | D3 | | OPA | | | |
| | 9 | D0 | | - | D0 | | |
| | A | DB | | IPA | | | |
| | B | D0 | | - | D0 | | |
| | C | E6 | | ANAI | | | |
| | D | 3F | | - | 3F | | |
| | E | EA | | JP | PE | T IF NO KEY OR MULTIPLE KEYS | |
| | F | 93 | | - | NXT COLM | CLOSED. MOVE TO NEXT COLUMN. | |
| 07 | A0 | 07 | | - | | T | |
| | 1 | 5F | | LDE | A | T KEY CLOSED; DEBOUNCE IT. | |
| | 2 | CD | | JS | | | |
| | 3 | EE | | - | (DEBOUNCE) | | |
| | 4 | 07 | | - | | | |
| | 5 | DB | | IPA | | | |
| | 6 | D0 | | - | D0 | | |
| | 7 | E6 | | ANAI | | | |
| | 8 | 3F | | - | 3F | | |
| | 9 | AB | | XRA | E | | |
| | A | C2 | | JP | NZ | | |
| | B | 93 | | - | NXT COLM | | |
| | C | 07 | | - | | | |
| | D | 01 | | LDPI | BC | T | |
| | E | 00 | | - | 00 | ENCODE KEY BY FINDING WHICH | |
| | F | 00 | | - | 00 | ROW, j, AND WHICH COLUMN, K, | |
| | | | | | | KEY IS LOCATED. (j=0-5, K=0-3) | |
| 07 | B0 | 7A | | LDA | D | | |
| | B1 | 1F | COLM COUNT | RRAC | | T | |
| | 2 | DA | | JP | CI | CONVERT COLUMN OUTPUT TO A | |
| | 3 | B9 | | - | ROW COUNT | COUNT, K. | |
| | 4 | 07 | | - | | | |
| | 5 | 04 | | ICB | | | |
| | 6 | C3 | | JP | | | |
| | 7 | B1 | | - | COLM COUNT | | |
| | 8 | 07 | | - | | | |
| 1068 | B9 | 7B | ROW COUNT | LDA | E | T | |
| 0682 | BA | 1F | +1 | RRAC | | CONVERT ROW INPUT TO A | |
| | B | 0C | | ICC | | COUNT, j: | |
| | C | DA | | JP | CI | | |
| | D | C2 | | - | ENCODE | | |
| | E | 07 | | - | | | |
| | F | C3 | | JP | | | |

| HEXADECIMAL | | | MNEMONIC | | | TITLE | DATE |
|-------------|-------------|--------|---------------|--------|-------------|--|------|
| PAGE ADR | LINE ADR | INSTR. | LABEL | INSTR. | MODIFIER | COMMENTS | |
| 07 | C 0 | BA | | - | ROW COUNT+1 | | |
| | 1 | 07 | | - | | | |
| | C 2 | 0D | ENCODE | DCC | | | |
| | 3 | 78 | | LDA | B | PUT K INTO ACCUMULATOR. | |
| | 4 | CA | | JP | Z | | |
| | 5 | CD | | - | EXIT | | |
| | 6 | 07 | | - | | | |
| | C 7 | C6 | ADD ROWS | ADAI | | IF $j \neq 0$, ADD $j \times M$ TO K. ($M = 04 = \#$ OF COLUMNS) | |
| | 8 | 04 | | - | 04 | | |
| | 9 | 0D | | DCC | | | |
| | A | C2 | | JP | NZ | | |
| | B | C7 | | - | ADD ROWS | | |
| | C | 07 | | - | | | |
| | C D | C1 | EXIT | PLP | BC | EXIT WITH ENCODED KEY VALUE | |
| | E | D1 | | PLP | DE | $X = j \times M + K$ IN REG. A. X CORRESPONDS | |
| | F | C9 | | RTS | | TO KEY'S ACTUAL POSITION IN MATRIX. | |
| 07 | D 0 | | | | | | |
| | D 1 | DB | (SCAN ALL) | IPA | | READ 7303 INPUT PORT. | |
| | 2 | DO | | - | D0 | | |
| | 3 | E6 | | ANAI | | ARE S1 AND/OR S2 CLOSED. | |
| | 4 | C0 | | - | C0 | | |
| | 5 | C0 | | RTS | NZ | YES. EXIT WITH $Z = 0$. | |
| | D 6 | 3E | (SCAN KBD) | LDAI | | NO. ENABLE ALL KEYBOARD COLUMNS. | |
| | 7 | 0F | | - | 0F | | |
| | 8 | D3 | | OPA | | | |
| | 9 | DO | | - | D0 | | |
| | A | DB | | IPA | | READ ROW INPUTS. | |
| | B | DO | | - | D0 | | |
| | C | E6 | | ANAI | | MASK OFF TOGGLE SWITCH INPUTS. | |
| | D | 3F | | - | 3F | | |
| | E | C9 | | RTS | | EXIT. IF KEY IS CLOSED, $Z = 0$. | |
| | F | | | | | | |
| 07 | E 0 | E6 | (HEX → ASCII) | ANAI | | CONVERT LSD IN REG. A TO ASCII. | |
| | 1 | 0F | | - | 0F | | |
| | 2 | FE | | CPAI | | IS (A) > 9? | |
| | 3 | 0A | | - | 0A | | |
| | 4 | DA | | JP | C1 | | |
| | 5 | EA | | - | 0 → 9 | | |
| | 6 | 07 | | - | | | |
| | 7 | C6 | | ADAI | | YES. CONVERT TO C1-C6. | |
| | 8 | B7 | | - | B7 | | |
| | 9 | C9 | | RTS | | EXIT. | |
| | E A | F6 | 0-9 | ORAI | | NO. CONVERT TO B0-B9. | |
| | B | B0 | | - | B0 | | |
| | C | C9 | | RTS | | EXIT. | |
| | D | | | | | | |
| | E E | C5 | (DEBOUNCE) | PSP | BC | DEBOUNCE DELAY ~ 20 MSEC'S. | |
| | F | 01 | | LDPI | BC | | |
| 07 | F 0 | 2B | | - | 0A2B | | |
| | 1 | 0A | | - | | | |
| | 2 | CD | | JS | | | |
| | 3 | F7 | | - | (LONG ΔT) | | |
| | 4 | 07 | | - | | | |
| | 5 | C1 | | PLP | BC | | |
| | 6 | C9 | | RTS | | EXIT. | |
| | F 7 | F5 | (LONG ΔT) | PSP | AF | DELAY $\Delta T = 24 N_{BC} t + A t$ | |
| | F 8 | 0B | DELAY | DCP | BC | | |
| 1 | 9 | 78 | | LDA | B | $N_{BC} = (BC)_{10}; t = 1$ time state | |
| 0 | A | B1 | | ORA | C | $A = 29(8085)$ or $31(280)$ | |
| 6 | B | C2 | | JP | NZ | | |
| 8 | C | F8 | | - | DELAY | | |
| 2 | D | 07 | | - | | $\therefore N_{BC} = \frac{\Delta T - A t}{24 t}$. N_{BC} must be in Hex. | |
| 1 | E | F1 | | PLP | AF | | |
| | F | C9 | | RTS | | EXIT. | |

APPLICATION SOFTWARE

MODULE DESCRIPTIONS

The enclosed software module descriptions are intended to provide the user with all the information necessary to access, utilize or move the routines into his program. The modules described are totally independent of requirements of the Monitor program, and are intended to allow the user to fully exercise the capabilities of the 7303 Keyboard/Display Card. There are several other modules available to the user that are not described herein, because they were written to the specifications and tailored to the needs of the Monitor program. The user is encouraged to study the listings carefully, should he find it desirable to utilize those modules.

MODULE NAME: (MESSAGE) _____

ENTRY ADDRESS: 0680 _____

FUNCTIONAL DESCRIPTION:

This module allows the user to display a message in the alpha-numeric displays of the 7303 Keyboard/Display Card. The message can be any length and is displayed in a billboard fashion, i.e., the message moves across the display from right to left. A new character appears at the right each 200 milliseconds, and an old one disappears at the left. The message must be stored in memory, and can be any length. The first character of the message should be at the low address in the designated memory space and the end of the message at the high address. The message is stored as a string of 8-bit ASCII characters with parity bit either 1 or 0. The message body should be preceded by a string of 8 ASCII spaces and also followed by a string of 8 spaces. The last character of the message must be an FF Hex.

ENTRY REQUIREMENTS:

Enter this module with the address of the first ASCII space that precedes the message in the HL register pair.

MEMORY SPACE OCCUPIED: 0680 to 069E

MEMORY REQUIREMENTS: ROM: 31 BYTES RAM: _____ BYTES

REGISTERS ALTERED BY THIS ROUTINE: All registers are altered.

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(200 MSEC ΔT), (DISPLAY)

ADDITIONAL COMMENTS:

(1 SEC ΔT); (500 MSEC ΔT);
MODULE NAME: (200 MSEC ΔT)

ENTRY ADDRESS: $\emptyset 6A8$; $\emptyset 6AB$;
 $\emptyset 6B1$

FUNCTIONAL DESCRIPTION:

These modules provide the user with delays of 1 second, 500 milliseconds, and 200 milliseconds duration. The timing is approximate and based on the clock rates provided on PRO-LOG's 7801 8085 CPU Card. For users of the 7803 Z80 CPU Card, the delays are approximately 1.25 seconds, 625 milliseconds, and 250 milliseconds duration.

ENTRY REQUIREMENTS: None.

MEMORY SPACE OCCUPIED: $\emptyset 6AB$ to $\emptyset 6B6$

MEMORY REQUIREMENTS: ROM: 15 BYTES RAM: _____ BYTES

REGISTERS ALTERED BY THIS ROUTINE: B, C

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(LONG ΔT)

ADDITIONAL COMMENTS:

MODULE NAME: (FLASHER) _____

ENTRY ADDRESS: 0700 _____

FUNCTIONAL DESCRIPTION:

This module causes the user's desired cursor pattern to be displayed over the last displayed data or message for approximately 200 milliseconds (250 msec. for Z80). All cursor segments are then turned off for approximately 500 milliseconds (625 msec. for Z80). The last displayed data or message is not changed. When the cursors are on, all LED's are also turned on, then turned off when the cursors are removed. The module can be used to flag the attention of an operator to the displayed message with a blinking pattern, by accessing this module in a repetitive loop.

ENTRY REQUIREMENTS:

Enter this module with the desired cursor pattern as an 8-bit value in register D. Each bit (0-7) corresponds to a display position (0-7). If the bit is 1, the cursor segment is turned on; if the bit is 0, the cursor is turned off.

MEMORY SPACE OCCUPIED: 0700 to 0714

MEMORY REQUIREMENTS: ROM: 21 BYTES RAM: _____ BYTES

REGISTERS ALTERED BY THIS ROUTINE: A,F

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(CURSORS), (CLR CURSORS), (200 MSEC ΔT), (500 MSEC ΔT)

ADDITIONAL COMMENTS:

MODULE NAME: (DISP 4ASC); (DISP N)

ENTRY ADDRESS: Ø718; Ø71A

FUNCTIONAL DESCRIPTION:

This multiple-entry module allows the user to display 4 or N (1 to 8) ASCII characters in the alpha-numeric displays on the 7303 Keyboard/Display Card. The ASCII data to be displayed must reside in sequential memory locations, the leftmost character at the low address and the rightmost at the high address.

ENTRY REQUIREMENTS:

Enter this module with the character count (Ø1 to Ø8) in register C (DISP N only), the leftmost display position number in the lower three bits of register E, and the low order address of the display data table in the HL register pair.

MEMORY SPACE OCCUPIED: Ø718 to Ø724

MEMORY REQUIREMENTS: ROM: 13 BYTES RAM: BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F, C, D, E, H, L

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(DISPLAY)

ADDITIONAL COMMENTS:

MODULE NAME: (DISP 2N HEX)

ENTRY ADDRESS: 0727

FUNCTIONAL DESCRIPTION:

This module allows the user to display the data contained in N (1 to 4) memory locations as 2-digit Hex numbers in the displays of the 7303 Keyboard/Display Card. The data stored at the low order address is displayed on the right, and data stored at the highest address is displayed in the leftmost desired positions.

ENTRY REQUIREMENTS:

Enter this module with the number of memory locations to be displayed in register C, the rightmost desired display position number in the lower 3 bits of register E, and the low order address of the data in the HL register pair.

MEMORY SPACE OCCUPIED: 0727 to 072E

MEMORY REQUIREMENTS: ROM: 8 BYTES RAM: BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F, C, D, E, H, L

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(DISP 2HEX)

ADDITIONAL COMMENTS:

MODULE NAME: (DISP 2HEX) _____

ENTRY ADDRESS: 072F _____

FUNCTIONAL DESCRIPTION:

This module allows the user to display the contents of any desired memory location in the alpha-numeric displays of the 7303 Keyboard/Display Card. The data is displayed as a 2 Hex digit number in 2 display positions.

ENTRY REQUIREMENTS:

Enter this module with the address of the data to be displayed in the HL register pair, and the desired rightmost display position number in the lower 3 bits of register E.

MEMORY SPACE OCCUPIED: 072F to 073F

MEMORY REQUIREMENTS: ROM: 17 BYTES RAM: _____ BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F, D, E, H, L

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(DISP HEX)

ADDITIONAL COMMENTS:

MODULE NAME: (CLR CURSORS): (CURSORS)

ENTRY ADDRESS: 0740; 0742

FUNCTIONAL DESCRIPTION:

This multiple-entry module allows the user to display different combinations of cursors in the alpha-numeric displays of the 7303 Keyboard/Display Card. The cursor character is a display character with all 16 segments turned on. It can be displayed over existing displayed data, then be turned off without altering the previously displayed data. Entry at the label (CLR CURSORS) turns off all cursors displayed. Entry at the label (CURSORS) causes any desired pattern of cursors on or off to be displayed in the 8 display positions. The pattern is contained in register D as an 8-bit binary number. Each bit position (0 to 7) corresponds directly with the 8 display positions (0 to 7). A bit set to 1 turns a cursor on; a bit set to 0 turns a cursor off.

ENTRY REQUIREMENTS:

Enter (CURSORS) with the desired cursor pattern in register D: 1 = on; 0 = off.

Enter (CLR CURSORS) directly.

MEMORY SPACE OCCUPIED: 0740 to 0759

MEMORY REQUIREMENTS: ROM: 26 BYTES RAM: BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(STROBE)

ADDITIONAL COMMENTS:

For further information about the cursor control, see the 7303 USER'S MANUAL.

MODULE NAME: (BLANK); (DISP 8)

ENTRY ADDRESS: Ø75B; Ø75D

FUNCTIONAL DESCRIPTION:

This multiple-entry module allows the user to display the same character in all 8 displays of the 7303 Keyboard/Display Card.

Entry at the label (BLANK) causes an ASCII space to be displayed, thereby blanking all 8 displays.

Entry at the label (DISP 8) causes the ASCII character contained in register D to be displayed in all 8 displays.

ENTRY REQUIREMENTS:

Enter (BLANK) directly. Enter (DISP 8) with the desired ASCII data to be displayed in register D.

MEMORY SPACE OCCUPIED: Ø75B to Ø766

MEMORY REQUIREMENTS: ROM: 12 BYTES RAM: BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F, E

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(DISPLAY)

ADDITIONAL COMMENTS:

For further information regarding the alpha-numeric display control, see the 7303 USER'S MANUAL.

MODULE NAME: (DISP HEX); (DISPLAY); (STROBE) ENTRY ADDRESS: 0768; 076D; 0773

FUNCTIONAL DESCRIPTION:

This multiple-entry module is the primary display/control routine for the 7303 alpha-numeric displays.

Entry at the label (DISP HEX) causes the least significant 4 bits of Hex data in register D to be displayed in the display position whose number is contained in the lower three bits of register E.

Entry at the label (DISPLAY) causes the ASCII data in register D to be displayed in the display position whose number is contained in the lower three bits of register E. The ASCII data can be of any parity.

Entry at the label (STROBE) causes the write line one the selected display chip to be strobed from high to low to high.

ENTRY REQUIREMENTS:

Enter (DISP HEX) with the Hex data to be displayed in the lower 4 bits of register D.
Enter (DISPLAY) with the ASCII data to be displayed in register D.
Enter both with the display position number in the lower 3 bits of register E.
Enter (STROBE) with bit 2 of register E set to 0 to strobe the right 4 displays, etc.

MEMORY SPACE OCCUPIED: 0768 to 077F

MEMORY REQUIREMENTS: ROM: 24 BYTES RAM: BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F, D (DISP HEX only)

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(HEX → ASCII)

ADDITIONAL COMMENTS:

For a more detailed description of the operation and control of the 7303 alpha-numeric displays, see the 7303 USER'S MANUAL.

MODULE NAME: (READ 1 KEY); (READ KBD)

ENTRY ADDRESS: 0780; 078F

FUNCTIONAL DESCRIPTION:

The 7303 keypad is arranged as a 4 X 6 matrix, with the key faces and positions numbered in Hex from 00 to 17. This multiple-entry module detects, debounces and encodes a single key closure. Multiple key closures are ignored. Upon exit, the accumulator, register A, contains the encoded value of 00 to 17 that corresponds to the positional value of the depressed key. After entry into this module, the keyboard will continue to be scanned for a key closure until one occurs, at which time it will be processed and the routine exited.

Entry at the label (READ 1 KEY) will cause only a single key closure to be processed, i.e., the key must be opened or released before another closure can be processed.

Entry at the label (READ KBD) will cause the routine to continuously process any key closure.

ENTRY REQUIREMENTS: None.

MEMORY SPACE OCCUPIED: 0780 to 07CF

MEMORY REQUIREMENTS: ROM: 80 BYTES RAM: BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(SCAN KBD): (DEBOUNCE)

ADDITIONAL COMMENTS:

See the 7303 USERS MANUAL for more detail regarding the operation and control of the keyboard.

MODULE NAME: (SCAN ALL); (SCAN KBD)

ENTRY ADDRESS: 07D1, 07D6

FUNCTIONAL DESCRIPTION:

This multiple-entry module detects if any toggle switches or keys are closed. Upon exiting the routine, the zero flag reflects the keyboard/switch status. If Z = 0, a key or switch is closed.

Entry at the label (SCAN ALL) detects the status of switches, then keys.

Entry at the label (SCAN KBD) detects the status of the keys only.

ENTRY REQUIREMENTS: None.

MEMORY SPACE OCCUPIED: 07D1 to 07DE

MEMORY REQUIREMENTS: ROM: 14 BYTES RAM: BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F

PROGRAM MODULES UTILIZED BY THIS ROUTINE: None.

ADDITIONAL COMMENTS:

See the 7303 USER'S MANUAL for a more detailed description of the operation and control of the toggle switches and keypad.

MODULE NAME: (HEX → ASCII) _____

ENTRY ADDRESS: 07E0 _____

FUNCTIONAL DESCRIPTION:

This module converts the lower 4 bits of register A to an 8-bit ASCII equivalent, with parity bit set. Upon exit, the converted value is in register A.

ENTRY REQUIREMENTS:

Enter this module with the Hex value to be converted in the lower 4 bits of register A.

MEMORY SPACE OCCUPIED: 07E0 to 07EC

MEMORY REQUIREMENTS: ROM: 13 BYTES RAM: _____ BYTES

REGISTERS ALTERED BY THIS ROUTINE: A, F

PROGRAM MODULES UTILIZED BY THIS ROUTINE: None.

ADDITIONAL COMMENTS:

MODULE NAME: (DEBOUNCE)

ENTRY ADDRESS: 07EE

FUNCTIONAL DESCRIPTION:

This module implements a 20 millisecond (25 msec. for Z80) time delay for the debouncing of switches and keys, although it can be used for other purposes.

ENTRY REQUIREMENTS: None.

MEMORY SPACE OCCUPIED: 07EE to 07F6

MEMORY REQUIREMENTS: ROM: 9 BYTES RAM: BYTES

REGISTERS ALTERED BY THIS ROUTINE: None.

PROGRAM MODULES UTILIZED BY THIS ROUTINE:

(LONG ΔT)

ADDITIONAL COMMENTS:

The delay time of this module is based on the clock rates used with the PRO-LOG 7801 8085 CPU Card. Users of the 7803 Z80 CPU Card can expect a 25% increase in the delay time, due to the lower clock rate.

MODULE NAME: _____ (LONG ΔT)

ENTRY ADDRESS: _____ 07F7

FUNCTIONAL DESCRIPTION:

This module provides the user with a variable time delay routine. The delay is implemented by downcounting the contents of the BC register pair (16 bits) to zero. The delay time can be found from the relationship:

$$\Delta T = 24 N_{BC} t + A t$$

where t is the state time of the processor, N_{BC} is the decimal count in BC, and A is the number of state times executed that are not in the delay loop itself. Therefore, the count for any time delay can be found from the relationship:

$$N_{BC} = \frac{\Delta T - A t}{24 t} \quad (N_{BC} \text{ must be in Hex})$$

ENTRY REQUIREMENTS:

Enter this module with the time delay count in the BC register pair as a 16-bit binary value.

MEMORY SPACE OCCUPIED: 07F7 to 07FF

MEMORY REQUIREMENTS: ROM: 9 BYTES RAM: _____ BYTES

REGISTERS ALTERED BY THIS ROUTINE: B, C

PROGRAM MODULES UTILIZED BY THIS ROUTINE: None.

ADDITIONAL COMMENTS:

For the 8085, $A = 29$. For the Z80, $A = 31$. The longest time delay that can be handled by this routine is approximately 500 milliseconds, depending on the processor clock rate.





USER'S MANUAL



**2411 Garden Road
Monterey, California 93940
Telephone: (408) 372-4593
TWX: 910-360-7082**

Telex: 171879