# PANEL MONITOR XCON-8

# TABLE OF CONTENTS

INTRODUCTION
THEORY OF OPERATION
Power Up and Master Clear 1-5
Clock Interrupts
XCON-8 Modes/Using RST and RTM 1-6
H8 Displays 1-7
H8 Keypad 1-9
DISPLAYING AND ALTERING MEMORY LOCATIONS
Specifying a Memory Address 1-10
Altering a Memory Location 1-12
Stepping Through Memory 1-13
DISPLAYING AND ALTERING REGISTERS
Specifying a Register for Display 1-14
Altering the Contents of a Selected Register 1-15
Stepping Through the Registers 1-15
PROGRAM EXECUTION CONTROL
Initiating Program Execution
Breakpointing 1-16
Single Instruction Operation
Interrupting a Program During Execution 1-17

LOAD/DUMP ROUTINES	
Loading From Tape	1-18
Dumping to Tape	
Copying a Tape	
Tape Errors	
I/O FACILITIES	
Inputting From a Port	1-21
Outputting to a Port	1-21
Addressing Port Pairs	
ADVANCED CONTROL	
16-Bit Tick Counter (TICCNT)	1-22
Using the Keypad	
Display Usage	
Using Interrupts	
FLOPPY BOOT	
FLOFFI BOOT	23
SWITCH SW1 1	-27
MEMORY MAP	27
ROM Code Listing	28
INDEX 1-	106

# INTRODUCTION

This Manual describes the functions and operations of the Heath H8 Panel Monitor Program, XCON-8, which resides permanently in a ROM on the H8 CPU board. XCON-8 provides a sophisticated front panel display and keyboard emulation as well as handling master clear and interrupt operations. Some of the major features of XCON-8 are:

- Memory contents display and alteration.
- Register contents display and alteration.
- Program execution control (both breakpoint and single instruction operation).
- Self-contained bootstraps for program loading and dumping.
- Port input and output routines.

In addition to the above features, XCON-8 can be instructed (by means of a flag byte contained in the H8 RAM) to bypass some or all of its normal functions so the sophisticated user can augment or totally replace them.

Communication with the Panel Monitor is accomplished through three devices: the keypad, the 7-segment displays, and the audio alert. The user enters commands and values through the 16-key keypad, and XCON-8 responds visually through the front panel displays. In addition to the front panel displays, XCON-8 provides the keypad entry and function feedback to the built-in speaker. Appropriate signals (short, medium, and long beeps) indicate that commands and data are accepted or rejected.

# THEORY OF OPERATION

This section will supplement the information contained in the "Operation" and "Circuit Description" sections of your H8 Operation Manual. In order to fully understand how XCON-8 operates, you must be familiar with the H8 front panel and CPU. A thorough knowledge of the 8080 instruction set and its architecture is also essential.

# **Power Up and Master Clear**

XCON-8 initializes the H8 whenever you power-up or master clear (RST). You initiate the power-up operation by turning on the rear panel Power switch. You can master clear by simultaneously depressing both the lower right-hand (RSTØ) and lower left-hand (Ø) keys of the H8 front panel keypad. Both power-up and RST cause a level zero (highest priority) interrupt and result in a long beep from the audio alert.

During initialization, XCON-8 enters a routine which determines the high limit of continuous RAM. Once the high limit of available RAM is determined, the H8 stack pointer (SP) is set to this value. XCON-8 then determines if the RAM starts at  $\emptyset$ , and copies itself from ROM into that RAM space. Control is passed to the front panel command loop. Using this feature, you can immediately determine the total amount of continuous memory above 8K by displaying the stack pointer value.

# **Clock Interrupts**

The Clock Interrupt is a crucial element in the operation of the H8 front panel system. This level one interrupt is generated by the front panel hardware every 2,000  $\mu$ S. XCON-8 uses this interrupt to check for some keyboard commands, to check for user program breakpoints, and to refresh the front panel displays.

XCON-8 performs these functions using a series of subroutines which are executed as necessary when indicated by the interrupts. For this reason, all user programs must maintain a valid stack (at high memory) containing at least 80 free bytes at all times. If this stack space is not available and XCON-8 is running (it can be disabled; see the Advanced Control Section), unpredictable software damage can occur in your program. In the same manner, if your program should execute a DI (Disable Interrupt) instruction, no front panel services including the RTM (Return To Monitor) function are available until an EI (Enable Interrupt) instruction is executed or until a master clear (RST/ $\beta$ ) is performed.

# XCON-8 Modes /Using RST and RTM

XCON-8 is always in either the monitor mode or the user mode. In the monitor mode no user program is executing, XCON-8 loops reading the keypad and refreshing the displays. All commands entered via the keypad are valid; however, the RTM command is meaningless.

When your program is being executed, XCON-8 is in the user mode and the MON LED on the front panel is extinguished. Only two keyboard commands are valid in this mode: RST (master clear) and RTM (Return To Monitor). NOTE: Both of these commands are dual key commands. No single key command is recognized, so a user program may have free use of the entire keypad.

You can return XCON-8 to the monitor mode by using the RTM command (simultaneously press the  $\emptyset$  and the # keys). This command stops program execution at the end of the current instruction, stores the current value of each register, and returns XCON-8 to the monitor mode. You can then continue your program by pressing the GO key. The RST command (simultaneously press the  $\emptyset$  and the / keys) performs the master clear operation described earlier and does not save any register values.

Normally, when a user program is running, XCON-8 is also running. Thus, if XCON-8 is displaying the contents of the HL register pair and the user program is started, it continues to display the contents of this register pair as the program is run. If the user program changes the contents of the HL pair, the change is immediately reflected in the front panel displays. In a similar manner, if a memory location is displayed when a user program is started, it is displayed during the time the user program is run. If the user program changes the contents of the display memory location, the front panel display changes.

Since XCON-8 does not recognize keypad commands in the user mode, the RTM command must be used before the memory location or register being displayed is changed to a new location or a different register. Once you select the new location or different register, you can resume program execution by pressing GO.

NOTE: XCON-8 requires about 10% of the H8 CPU's resources to process the display interrupts. Programs which are compute-bound may be slowed down by simultaneous operation of XCON-8. In this situation, you may wish to turn off the clock interrupts to improve execution time. See "Using Interrupts" on Page 1-24.

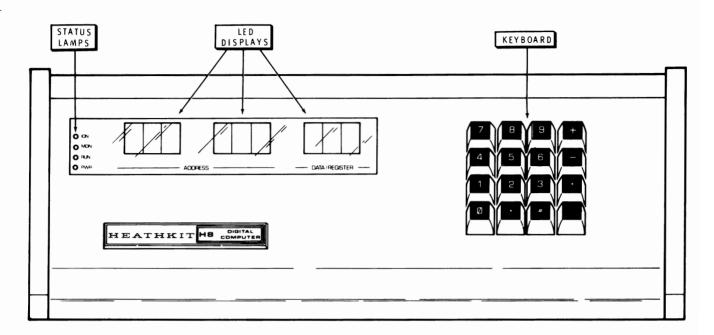


Figure 1-1

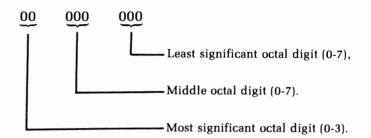
# **H8 Displays**

You must understand the H8 front panel presentation in order to use XCON-8. The display is made up of 9 digits, in three groups of three digits each. See Figure 1-1. Each group of three digits displays one byte (eight bits) of information. This information may be the contents of a designated register or memory location, or it may be the address of a memory location itself. The register names are also displayed.

All binary numbers are converted to octal format for display on the H8 front panel. The following table shows binary to octal conversion.

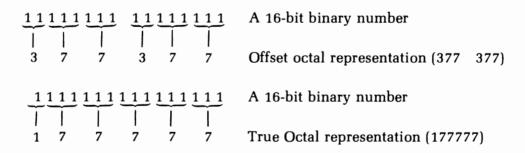
BINARY NUMBER	OCTAL NUMBER
	П
000	Ц
001	1
010	2
011	3
100	4
101	5
110	5
111	

Each byte is displayed as two-and-one-half octal digits. The octal numbers lie in the range of 000 to 377 for binary numbers in the range 000000000 to 111111111, as shown below.



NOTE: As there are only eight bits in a byte, the most significant octal digit only represents two bits and is therefore displayed as 0 to 3. If the user should inadvertently enter the octal digits 4 to 7 into the most significant digit, the most significant bit is lost. Losing this bit converts 4 through 7 into the digits 0 through 3 respectively.

Also note that 16-bit numbers, such as memory addresses and certain register contents, are still displayed as two eight-bit numbers. Therefore, the H8 front panel representation of the number is made up of **two** groups of three octal numbers in the range of 000 to 377. This representation of 16-bit binary numbers is known as **offset octal**, and is used consistantly throughout all H8 displays of 16-bit numbers. Offset octal must not be confused with octal. For example:



The lower example shows true octal representation of a 16-bit binary number. This is **not** used by the H8 front panel displays or any H8 software. Occasionally you will see offset octal numbers printed with a decimal point separating the upper and lower bytes. For example:

377.377

Hi Byte Lo Byte

# **H8** Keypad

The H8 Keypad consists of 16 keys, as shown in Figure 1-1 on Page 1-7. When the keypad is operating under the control of XCON-8, it exhibits a number of unique properties.

- Each keystroke is verified by a short beep from the audio alert.
- Octal digits are entered using the keys 0 through 7.
- Holding a key down continuously repeats the key's function.
- The + key increments memory port or register locations.
- The key decrements memory port or register locations.
- The \* key cancels previous keypad entries.
- The ALTER key causes XCON-8 to enter the alter mode.
- The MEM key causes XCON-8 to enter the display memory mode.
- The REG key causes XCON-8 to enter the register mode.

Many of the keys on the keypad have multiple functions, depending on the XCON-8 mode being used. In the register mode, for example, the numeric keys (1-6) call the register indicated in the upper left-hand corner of the key. When the XCON-8 is in neither the register nor the memory mode, the keys perform the functions indicated in the lower right-hand corner of the key.

The # and / keys have additional special functions, as indicated earlier. When the / key is pressed simultaneously with the Ø key, the RST (master clear) sequence is initiated. When the # sign key is pressed simultaneously with the Ø key, the RTM (Return To Monitor) function is initiated, the user program is stopped, and XCON-8 regains control.

Each key is covered in greater detail as the various function are discussed.

# DISPLAYING AND ALTERING MEMORY LOCATIONS

One of the major features of XCON-8 is its ability to examine the contents of any H8 memory location and to modify the contents of that memory location if it is RAM.

When the H8 is first powered up, XCON-8 is in the display memory mode. This mode is indicated by all digits displaying octal numbers and no decimal points being on.

# **Specifying a Memory Address**

If you wish to display or alter the contents of a memory location, you must first place XCON-8 in the memory address mode and then enter the desired memory address. Place XCON-8 in the memory address mode (if not already there) by pressing the MEM (Memory) key. Specify the address to be displayed or altered by entering the 6-digit address (offset octal).

When you press the MEM key, all the decimal points will light. This indicates that the address may now be entered. Once the full 6-digit address is entered, the decimal points turn off, indicating that address entry is completed. After all 6 digits are entered, the address is displayed in the left-most six displays, and the contents of the addressed memory location are displayed in the right-hand 3 digits.

NOTE: As you press each key, including the MEM key, a short beep indicates successful entry. As each group of three octal digits is successfully entered, a medium beep is sounded. The sequence by which you specify a memory address is shown in Figure 1-2.

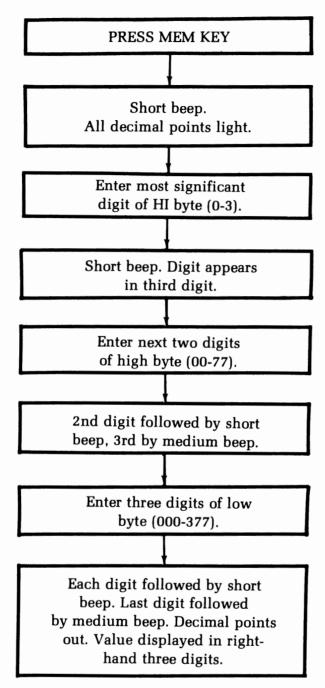


Figure 1-2
Entering a memory address through XCON-8.

NOTE: If you press a non-octal digit key as one of the six address digits, an error is flagged (a long beep). Once this error is flagged, the XCON-8 considers the address complete and extinguishes the decimal points. The entire sequence must be repeated.

# **Altering a Memory Location**

Before you can alter a memory location, you must first display the contents of the memory location by specifying the memory address as described in the preceding paragraphs. After you specify the memory address, press the ALTER key. This will cause XCON-8 to enter the memory alter mode.

When XCON-8 enters the memory alter mode, a single decimal point rotates from right to left through all 9 digits. You can now alter the contents of the displayed location by entering the new octal value (three digits on the keypad). When the three digits have been entered, acoustical verification (a short beep) is given and the memory address is incremented. You can then alter this new location by entering three more digits or pressing one of the following keys, causing the monitor to perform the indicated function:

<b>KEY</b>	<b>FUNCTION</b>
+	Increment the address.
_	Decrement the address.
MEM	Specify a new memory address (leave the memory
	alter mode).
REG	Specify a register for display (leave the memory
	alter mode).
ALTER	Exit from the alter mode (into the display mode).

NOTE: XCON-8 automatically increments the memory address as each entry (3 octal digits) is complete. Therefore, you may load a program in sequential locations very rapidly. Each location is modified by simply entering the three octal digits.

The following example reviews each step as the H8 is turned on; the memory address mode is entered; and the location 040 123 is addressed, altered to 345, checked, and closed.

	DISPLAY	, –	<u>COMMENTS</u>
ххх	x x x	x x x	Random memory display at power up (X=random number.)
X . X . X .	X . X . X .	X . X . X .	MEM key pressed. (In memory address mode, a short beep.)
X.X.0.	X . X . X .	X . X . X .	0 key pressed. (Short beep.)
X.0.4.	X . X . X .	$X \cdot X \cdot X$ .	4 key pressed. (Short beep.)
0.4.0.	X . X . X .	X . X . X .	0 key pressed. (Medium beep.) Contents of location 040 XXX displayed.)
0.4.0.	X.X.1.	X . X . X .	1 key pressed. (Short beep. Contents of 040 XX1 displayed.)
0.4.0.	X.1.2.	X . X . X .	2 key pressed. (Short beep. Contents of 040 X12 displayed.)
040	1 2 3	ххх	3 key pressed. (Medium beep. Contents of desired location 040 123 displayed, decimal points out.)
0.4.0	1.2.3	X . X . X	ALTER key pressed. (Short beep. Decimal points rotate.)
0.4.0.	1.2.3.	X.X.3.	3 key pressed. (Short beep. Decimal points <b>rotate</b> .)
0.4.0.	1.2.3.	X.3.4.	4 key pressed. (Short beep. Decimal points <b>rotate</b> .)
0.4.0.	1.2.4.	X . X . X .	5 key pressed. (Medium beep. Address increments one location. Decimal points <b>rotate</b> .)
0.4.0	1.2.3	3.4.5	-key pressed. (Short beep. Address decrements one location. Decimal points <b>rotate</b> .)
0 4 0	1 2 3	3 4 5	ALTER key pressed. (Short beep. Decimal points go out.)

# **Stepping Through Memory**

When XCON-8 is either in the display memory or alter memory modes, the + and - keys increment and decrement the memory address. Each time you press the key, XCON-8 increments (or decrements) the memory address one location. If you hold the key down, the auto-repeat function of XCON-8 causes the memory address to increment or decrement repeatedly (approximately one location every second).

### DISPLAYING AND ALTERING REGISTERS

XCON-8 can display and alter the contents of the 8080 CPU registers, just as it displays and alters the contents of H8 memory locations. Although the process is quite similar, a few special features should be noted.

# **Specifying a Register for Display**

Press the REG key to specify that a register is to be displayed. After you press the REG key, press a second key (SP through PC, see the Table below) to specify the desired register or register pair.

When the REG key is pressed, six decimal points light, indicating that you must now select a register. NOTE: Simply pressing the REG key causes a register name to appear in the right-hand digits. However, you must select a register using the Register Select key before a register is definitely selected and its true contents are displayed. Once a register is selected, the decimal points are extinguished.

The contents of the selected register pair are displayed in the six left-most displays. The register name (or names) are displayed in the two right-most digits of the right-hand three displays. The registers are selected and displayed in accordance with the following table:

<b>KEY</b>	LEFT 3 DIGITS	MIDDLE 3 DIGITS	RIGHT PAIR	COMMENTS
SP (1) AF (2)	000 to 377 000 to 377	000 to 377 000 to 377	5 P A F	Stack pointer AF Register pair
BC (3)	000 to 377	000 to 377	ЬС	BC Register pair
DE (4)	000 to 377	000 to 377	dЕ	DE Register pair
HL (5)	000 to 377	000 to 377	HL	HL Register pair
PC (6)	000 to 377	000 to 377	Pc	Program counter

NOTE: The contents of any single eight-bit register may lie in the range of 000 to 377 octal. The stack pointer (SP) and the program counter (PC) are 16-bit registers and are displayed as two sets of three octal numbers. Each 3-digit grouping corresponds to one byte (8 bit number). When a register pair is displayed, the left three digits correspond to the left register and the middle three digits correspond to the right register. For example:



# Altering the Contents of a Selected Register

To alter the contents of a register (or register pair), you must first specify it as described in the preceding paragraphs. After you select the register or register pair, press the ALTER key. This will cause the six left-hand decimal points to rotate right to left, indicating that you may enter 6 digits to alter the contents of the indicated register or register pair.

Alternately, you may press one of the following command keys.

<u>KEY</u>	FUNCTION
+	Changes the register pair being displayed.
_	Changes the register pair being displayed.
MEM	Specify a new memory address (leave the alter register mode).
REG	Specify a new register for display (leave the alter register mode).
ALTER	Exit the register alter mode.

NOTE: Stack pointer register (SP) is not a direct display of the real stack pointer register, but simply a copy of the real stack pointer register and is used for display purposes only. The stack pointer cannot be altered from the front panel. To alter the stack pointer register, an SPHL (SPHL = 371) instruction must be written into memory. The desired new stack pointer value is then placed in the HL register pair. XCON-8 single instruction mode is used to execute the SPHL swap instructions, loading the stack pointer with the contents loaded in the HL register pair.

# **Stepping Through the Registers**

Use + and - keys to change the register pair being displayed. For example, if the DE register pair is being displayed, pressing the + key causes the next sequential register pair to be displayed (the HL pair). In the same manner, pressing the - key causes the register to decrement to the preceding pair. For example, if the DE pair is being displayed, pressing the - key displays the BC register pair. NOTE: Holding down either the + key or the - key causes the display to continuously increment or decrement through all the six registers/register pairs.

# PROGRAM EXECUTION CONTROL

XCON-8 supports three basic program execution control facilities:

- Beginning or starting execution.
- Breakpointing.
- Single instruction.

Each of these execution controls permits the programmer to execute the desired portions of a program and examine its effects. He may execute the entire program, or a small group of instructions, or a single program instruction.

# **Initiating Program Execution**

To begin the execution of a program residing in H8 memory, place the address of the first instruction to be executed in the PC (program counter). Use the methods described in "Displaying and Altering Registers" (Page 1-14). Once the address of this first instruction is placed in the program counter, press the GO key and program execution will begin. NOTE: Unless the program disables the front panel, the display continues to be actively updated, although the front panel commands are no longer active (except for RST and RTM). If the program counter is displayed when you press the GO key, XCON-8 continuously monitors the program counter.

# **Breakpointing**

Breakpointing permits the programmer to execute small portions of a program and then return to XCON-8. Breakpointing is especially useful when a program is being "debugged." Small portions of the program may be executed and their results observed. If there is an error, it may be corrected before an entire program is involved.

When the H8 executes a program and encounters a halt instruction, it re-enters XCON-8 and sounds the alarm. All of the registers are preserved and the program counter points to the address following the address of the halt instruction. Thus, you can breakpoint a program from the front panel by inserting halt instructions (HLT = 166) at the desired points throughout the program. When a particular

section of the program is tested and the breakpoint feature is no longer required, you can change the halt to a "no operation" (NOP = 000). Once the halts are changed to NOPs, execution of the NOP simply passes control to the next successive instruction. Program execution for breakpointing uses the GO key as previously described.

NOTE: If you temporarily replace an existing instruction with a halt, you must restore the instruction before resuming program execution. The contents of the program counter point to the address **following** the halt. Therefore, if the instruction which replaced the halt is to be executed, when the program continues, the contents of the program counter must be decremented one location before execution is resumed.

# **Single Instruction Operation**

Any user program may be operated in the single instruction mode. This procedure is identical to the GO command, except that the SI key is pressed rather than the GO key. When the SI key is pressed, a single **instruction** (not a single machine cycle) is executed and then control is returned to XCON-8. Single instruction operation is available for careful inspection of program results and for executing special programs, such as swapping the HL register pair with the stack pointer as discussed in "Altering the Contents of a Selected Register" (Page 1-15).

# **Interrupting a Program During Execution**

You can interrupt a running program (with all registers preserved at the point of interruption) by pressing RTM & Ø. You can then examine and/or alter the contents of various memory locations and all the registers as required. Resume execution of the program at the next sequential instruction by simply pressing the GO key. NOTE: Although all registers and memory locations are preserved when RTM & Ø are pressed, it is very difficult to stop a program at an exact location. Therefore, use the breakpoint feature if you want to stop the program at an exact location.

# LOAD/DUMP ROUTINES

XCON-8 contains a routine that lets you load and dump memory contents from or to a tape. This feature is especially important, as most computers require one of two successive "boot strap" routines to be hand-loaded before a desired program can be loaded into the main memory. All these "boot strap" routines are contained within the XCON-8 ROM, and use sophisticated error checking techniques. Thus, a program can be loaded or dumped by simply pressing a single key.

# **Loading From Tape**

To load from a tape, ready the reader device with the tape to be loaded prior to executing the load command. Place XCON-8 in the display memory mode and press the LOAD key. Once the LOAD key is pressed, XCON-8 starts the tape transport and scans the tape for the first file record.

No change will be seen on the front panel displays until XCON-8 finds the first file. When the first file record is located, XCON-8 checks it to see if it is the first (or only) record in a sequence, and the record is a memory dump record. If it is not a memory dump record, a number two error is flagged (see "Tape Errors" on Page 1-20).

Once a correct record is found, loading proceeds. The loading procedure places the entry point address of the program being loaded in the H8 program counter. The H8 memory is then loaded. The displays continuously show the address being loaded and the data being loaded at these addresses. When the load is complete, XCON-8 sounds a long beep and displays the final memory address. If the load is faulty, a number one error is displayed and the audio alarm continuously beeps. (See "Tape Errors," Page 1-20.)

NOTE: You may abort a partial load by using the CANCEL key. Naturally, the load image resulting from this action is incorrect, and should not be executed.

# **Dumping to Tape**

Before dumping a memory image onto tape, the following three dump parameters are required:

- The entry point address (the program starting address).
- The dump starting address.
- The dump ending address.

Set the desired entry point address by placing this value in the program counter (PC). This value will be placed in the program counter whenever you load the program so execution will begin at this address when you press the GO key.

Place the dump starting address into the first two H8 RAM cells. These are: 040 000 (offset octal) and 040 001 (offset octal). NOTE: The low order byte of the address should be placed into location 040 000 and the high order byte of the starting address should be placed into location 040 001.

Enter the dump ending address as a memory address using the # (MEM) key. Then ready the tape transport and press the DUMP key. As the tape dump takes place, the number of bytes left to be dumped and the contents of the memory location being dumped are displayed on the front panel. You can abort a dump by using the CANCEL key. If the CANCEL key is used, an incomplete dump image is left on the tape. This cannot be loaded at a future date. NOTE: A successful load automatically sets up the following three dump parameters:

- A. The program starting locations are stored in locations 040 000 and 040 001.
- B. The program ending location is displayed.
- C. The program counter contains the program entry point.

Figure 1-3A shows the steps of a typical dump sequence and Figure 1-3B shows the steps of a typical load sequence.

- Set PC to 040 100; (040 100 = entry address).
- 2. Set 040 000 to 100 (100 = low byte of dump start).
- 3. Set 040 001 to 040 (040 = high byte of dump start).
- Enter memory address 052 340 (052 340 = end address of dump).
- 5. Be sure tape is ready.
- 6. Press DUMP.

Figure 1-3A
The H8 memory image dump.

- Be sure tape is ready.
- Press LOAD.

Figure 1-3B
The H8 memory image load.

# Copying a Tape

The beginning and final address of the load image are placed at the appropriate points. Thus, to copy a tape, simply load the tape as described in "Loading From Tape" (Page 1-18). Then ready the dump tape drive and press the DUMP key. A dump then takes place, including entry point, initial address, and final address.

In a similar manner, to load, alter, and then dump, enter only the ending address. The other parameters are unchanged from the load if locations 040 000, 040 001 or the program counter have not been modified during the altering procedure.

# **Tape Errors**

XCON-8 detects two types of tape errors: record errors and checksum errors. In either case, when an error is detected, the tape transport is halted. The error number is then displayed in the center three digits (001 for a checksum error, 002 for a record error) and the alarm is repeatedly sounded. To halt the alarm and return to the command mode, press the CANCEL key.

### RECORD ERRORS

The following are typical causes of record errors.

- Attempting to load a file which is not a memory image. For example, loading an editor text file or a BASIC program file.
- Attempting to start a load in the middle of a load image. Therefore missing the initialization information at the start of the file.
- A tape error which causes a portion of the load image to be missed so the next record read is not in the proper sequence.

### **CHECKSUM ERRORS**

A checksum error is flagged when the CRC (Cyclical Redundancy Check) checksum following a record does not match the CRC calculated by PAM-8. This error means that the record is either incorrectly recorded or the load is faulty. In either case, the load should be attempted again. If successive loads result in repeated failures, the original tape must be suspected as faulty.

# I/O FACILITIES

XCON-8 supports two commands that allow you to perform input and output functions on H8 I/O ports. These front panel instructions permit simple manipulation of the H8 I/O ports without your having to write extensive routines to perform these functions.

# **Inputting From a Port**

To input from a port, press the # key. Then enter three zero digits and the three-digit address (octal) of the desired port. NOTE: The front panel should now display 000 AAA, where AAA is the port address and 000 is meaningless. Press the IN key to read the port, the value is displayed in the three left-most digits of the front panel display.

# **Outputting to a Port**

To output to a specified port, press the # key. Then enter the value to be supplied to the port in the three left-most displays. The port address is entered into the middle three displays. The display is of the form VVV AAA, where V stands for value, and A for address. Pressing the OUT key causes the value to be outputted to the indicated port.

# **Addressing Port Pairs**

Frequently, ports are assigned in pairs, where one of the two port addresses is the control and status register and the other port is the data port. Address port pairs by using the + and - key to change ports. Once the initial port has been defined, the + key increments the port address to a new higher numbered port, and the - key is used to decrement to a lower numbered port.

# ADVANCED CONTROL

One of the advanced features of XCON-8 is its provisions allowing sophisticated users to augment or replace XCON-8's functions. Augmenting or replacing XCON-8 functions is usually done in conjunction with assembly language programs. Sometimes it is possible to implement these features by using the POKE and PEEK commands in BASIC.

# **16-Bit Tick Counter (TICCNT)**

XCON-8 maintains a 16-bit (2 byte) tick counter known as TICCNT. The value of this counter is incremented each time a clock interrupt is processed. As an interrupt occurs once every 2 mS, the counter is incremented once every 2 mS. As long as clock interrupts are not disabled, this value can be used by any program to compute elapsed time. The tick counter may be set to any desired value, but it should not be frequently reset, as this interferes with the front panel refresh cycle. The contents of the tick counter are contained in memory locations 040 033 (the least significant byte) and 040 034 (the most significant byte).

# Using the Keypad

When your program is running, XCON-8 does not recognize any single key command. Thus, all single key patterns are available for your program. To read keypad patterns, you can use one of two routines. First, you may take an input from port IP. PAD; or second, your program may use XCON-8 RCK (read Console Keypad) routine. The input port IP. PAD is permanently assigned to port location 360. Inputting a binary number from this port detects which of the 16 keys are depressed.

The RCK routine provides keypad decoding, keypad debounce routines, autorepeat routines, and acoustical feedback.

NOTE: If you use two key combinations, each key must reside in a separate bank. The first bank includes keys 0-7 and the second bank includes keys 8-#. RCK cannot decode two key combinations.

# **Display Usage**

When a user program is running, XCON-8 normally displays the contents of the selected register or memory location. However, you may disable this process and display any arbitrary segment pattern, or completely disable the display to provide greater computational through-put. The display usage is primarily controlled by setting various bits in the .MFLAG memory cell. This memory cell is found at location 040 010.

### MANUAL UPDATING

By setting the UO.DDU bit in the .MFLAG memory location, you can instruct XCON-8 to continue refreshing the front panel displays and to disable updating. When this is done, XCON-8 continues to refresh the LED's from a 9-byte block of RAM cells found at locations 040 013 thorugh 040 023. When the UO.DDU bit is set in .MFLAG, the contents of these bytes are not altered in any manner by XCON-8.

You can use this technique to display numbers, letters, or arbitrary bar patterns on the front panel displays. For instance, your program may alter the display by inserting any value into FPLEDS. The front panel LED segments will display a decimal integer if you use the octal to 7-segment pattern (DODA) display.

### MANUAL DISPLAY REFRESHING

By setting the UO.NFR (User Option.No Front Panel Refresh) bit in the .MFLAG memory cell, you can instruct XCON-8 to stop refreshing the front panel displays. Setting the UO.NFR bit does not disable the clock interrupts; therefore, the tick counter (TICCNT) is still incremented. But XCON-8 does not refresh the displays from the information contained in the FPLEDS bytes.

NOTE: If you desire, you may write a program to refresh the front panel LED displays. Usually this is done using the clock interrupts. If you undertake an independent front panel refresh program, take extreme care to avoid burning the displays due to excessive refreshing. The total power dissipated in the LEDs is determined by the refresh cycle, and too frequent refreshing will result in excessive display heating.

# **Using Interrupts**

All H8 interrupts cause control to be transferred into the low 64 bytes of memory. XCON-8 occupies this memory space so all interrupts are first processed by XCON-8. Except for level zero interrupts, which are used as master clears, you can supply an interrupt processing routine for each of the seven additional interrupts. The following sections explain the use of each of these interrupts.

### I/O INTERRUPTS

Interrupts numbered 3 through 7 are I/O interrupts. XCON-8 does not process these interrupts in any way. When a level 3 through level 7 interrupt is received, XCON-8 immediately transfers to the user interrupt vectors contained in memory locations 040 037 through 040 064. Each location must contain a jump instruction pointing to the appropriate program location which processes these interrupts.

NOTE: If any of these interrupts occur, you must supply a processing routine for them. This routine must be complete including both entry and exit processing. When you use H8 interrupts, you must use only the available vector which is 6 to insure compatibility with future H8 products. You may also use 2 if you will not be using BUG-8.

### **CLOCK INTERRUPTS**

The level one interrupts are generated by the front panel hardware every 2 mS. XCON-8 normally processes these interrupts. However, by setting a processing vector in UIVEC and setting the UO.INT bit in the .MFLAG cell, XCON-8 enters the users routine each time a clock interrupt is generated.

### SINGLE INSTRUCTION AND BREAKPOINT INTERRUPTS

Level two interrupts are generated by the single instruction hardware contained on the CPU card. When a single instruction is requested, the result of the interrupt is processed by XCON-8. If the single instruction interrupt was generated by XCON-8 in response to a Monitor Mode Single Instruction register condition, XCON-8 processes it. Otherwise, XCON-8 jumps to the user level two interrupt vector (UIVEC). Since the level two interrupt does not affect XCON-8, a level two restart instruction can be used as a breakpoint instruction by the user programs.

# **FLOPPY BOOT**

XCON-8 contains the code necessary to boot-up an operating system from a floppy disk. Two forms of "Boot" let you select the device (H17 or H47) and drive number (0-2 or 0-3). "Boot Primary" refers to the device that you will use most often. "Boot Secondary" provides you with a convenient way to boot from your alternate device, if you have one.

### **BOOT PRIMARY**

The primary boot device is selected by switch SW1 sections 4, 1, and 0 on the extended configuration board. This switch is preset for H17 primary device. You may change the switch sections to select H47 primary device.

<b>DISPLAY</b>	<b>ACTION</b>	<b>COMMENTS</b>
Pro TH TH	Press "1"	Boot H17 primary
	or	
P-1	Press "1"	Boot H47 primary
BOOT SECONDARY		
SEC TH TIT	Press "2"	Boot H17 secondary
	or	
SEC TH TYN	Press "2"	Boot H47 secondary

You may use the "CANCEL" key to abort the boot command and return to the monitor.

### AUTO BOOT

If Switch SW1 section 7 is set to 1, the floppy disk will boot from the primary device automatically at power-up and master clear.

NOTE: We do not recommend auto-booting with a diskette in the drive and the door closed at power-up. Damage could occur to the diskette if you attempt to do so. Rather, power-up the H8 and H17 (H47), insert the diskette, and close the door within 15 seconds. Rebooting with Auto-Boot is the prime reason for its implementation. Software may accomplish this by executing an RST  $\emptyset$ .

### BOOT FROM DRIVE OTHER THAN DRIVE &

Primary and secondary Boot are both designed to access drive  $\emptyset$  on either the H17 or the H47. However, if you have not selected Auto Boot, you may boot from H47 drive 1 or 2 or H17 drive 1, 2, or 3 by following this procedure:

- 1. Use XCON-8 "Altering the Contents of a Selected Register" procedures to set register A to the drive number that you want to boot from.
- 2. If you are booting from a primary alternate drive, simply press "GO."
- 3. If you are booting from a secondary alternate drive, set register PC to 007 367 (the secondary drive address) and press "GO."

NOTE: Register PC is already set for the primary drive address (007 364) at power-up and master clear.

### **ERRORS**

The front panel will display \( \begin{align\*} | \begin{a

- 1. The boot device does not respond within 15 seconds.
- 2. Switch SW1 is set to an undefined setting.
- 3. A disk error occurs.

NOTE: The "boot Err" message will only remain on the display a few seconds. XCON-8 will then return to the panel monitor mode.

# **SWITCH SW1**

The sections of SW1 (on the HA8-8 Extended Configuration Board) have been defined as follows:

SWITCH SECTION	DESCRIPTION
7 6 5 4 3 2 1 0	<del></del>
$X\ X\ X\ X\ X\ X\ X\ 0\ 0$	Port 174/177 = H17
X X X X X X 0 1	Port $174/177 = H47$
X	Port 170/173 = unused
$X\ X\ X\ X\ X\ 0\ 1\ X\ X$	Port $170/173 = H47$
X X X 0 X X X X	Boot primary from port 174/177
$X\ X\ X\ 1\ X\ X\ X\ X$	Boot primary from port 170/173
0 X X X X X X X	Normal
1 X X X X X X X	Auto-Boot

Note that switches 5 and 6 are reserved.

# **MEMORY MAP**

The lower 4K of memory is used as follows:

```
PAM-8
Modified

1K

extensions to PAM-8
supporting extended configuration

2K

H17 ROM
Image
(assembled to reside at 030.000)

4K
```

+ 44		
•		(STACK+O) = RETURN ADDRESS (TO PAM/B)
56 *		· (STACKPTR+14)
+ 85		
<b>+</b> 65		
* 09		(STACK+10) = (HL)
62 +		INTERPOSER'S ROUTINE SHOULD RETURN TO PAM/8 VIA
<b>*</b> £9		HOUT ENABLING INTERRUPTS.
* *9	7	SINGLE STEP. SINGLE STEP INTERRUPTS GENERATED
<b>*</b> 99		BY PAM/8 ARE PROCESSED BY PAM/8.
± 29		ANY SINGLE STEP INTERRUPT RECEIVED WHEN IN
+ + 00		USER MUDE CAUSES A JURY INTRUCEM +UIVEC++5.
* 02		(STACK+0) = (STACKPTR+12)
<b>* 17</b>		(STACK+2) = (AF)
72 *		(STACK+4) = (BC)
* (2)		•
75 *		(SIACK+10) = (PC)
<b>*</b> 9L		THE USER'S ROUTINE SHOULD HANDLE IT'S OWN RETURN
* 77 * 78 *		NTERRUPT.
* 62	1111	TOURSET CONTOURS A SECTION OF THE PROPERTY OF
* 18	THE	FULLUMING INTERFORIS ARE VECTORED DIRECTLY THROUGH *UIVEC*. USER ROTTINE MIST HAVE SETIOD A LIME IN \$117FOCH REFORE ANY
<b>*</b> 28 <b>*</b>	0F THE	SE INTERRUPTS MAY OCCUR.
n	3	I/O 3. CAUSES A DIRECT JUMP THROUGH *UIVEC*+6
# # CO	4	1/0 4. CAUSES A DIRECT JUMP THROUGH #UTVEC#+9
# <u>20</u>		
* # 58	5	I/O 5. CAUSES A DIRECT JUMP THROUGH #UIVEC*+12
** 06	9	I/O 6. CAUSES A DIRECT JUMP THROUGH #UIVEC#+15
92 *	7	I/O 7. CAUSES A DIRECT JUMP THROUGH #UIVEC#+18

97 *** 10 PORTS  98	
99 IP-PAD EQU 3600 PAU INPUT PORT 100 DP-CTL EQU 3600 CONTROL OUTPUT PORT 100 DP-CTL EQU 3600 CONTROL OUTPUT PORT 100 DP-CTL EQU 3600 SEGGEN SELECT OUTPUT PORT 100 OP-SEC EQU 3710 TAPE CONTROL IN 100 OP-TPD EQU 3710 TAPE CONTROL OUT 100 OP-TPD EQU 3710 TAPE CONTROL OUT 100 OP-TPD EQU 3700 TAPE OATA OUT 100 OP-CTL2 EQU 3700 TAPE OATA OUT 100 OP-CTL2 EQU 3620 CONTROL OUT 111 A.STM EQU 0260 STWC CHARACTER SECONDARY CONTROL OF TILE A.STM EQU 0260 STWC CHARACTER STANDED TO THE CONTROL OUT 111 CB-CTL EQU 0000100000 STWC CHARACTER STANDED TO THE CONTROL OUT 100 OUT 1	
100 OP-CIT ENU 3600 CONTROL OUTPUT PORT 102 OP-SEG ENU 3600 01GIT SELECT OUTPUT PORT 102 OP-SEG ENU 3610 SEGMENT SELECT OUTPUT PORT 103 OP-SEG ENU 3610 SEGMENT SELECT OUTPUT PORT 104 OP-FFC ENU 3710 TAPE CONTROL OUT 105 IP-FFC ENU 3700 TAPE CONTROL OUT 105 OP-FFC ENU 3700 TAPE DATA OUT 106 OP-FFC ENU 3700 TAPE DATA OUT 107 IP-CONTROL	
102	
103   19-17C   EQU   3710   TAPE CONTROL IN     104   09-17C   EQU   3700   TAPE DATA OUT     105   09-17D   EQU   3700   TAPE DATA OUT     107   19-20N   EQU   3620   TAPE DATA OUT     108   09-27L2   EQU   3620   TAPE DATA OUT     110   **	TPUR
105 1P. PC EQU 3719 TAPE CONTROL OU 105 1P. PD EQU 3709 TAPE DATA IN 106 DO PT PD EQU 3709 TAPE DATA IN 106 DO PT PD EQU 3709 TAPE DATA IN 107 1P. CON EQU 3529 CONTROL BOT TAPE DATA IN 118 A.SYM EQU 0269 SYNC CHARACTER 113 A.SYM EQU 0269 SYNC CHARACTER 113 A.SYM EQU 0269 SYNC CHARACTER 113 A.SYM EQU 0260 SYNC CHARACTER 114 CB.SYM EQU 0260000 HONTROL BITS. 116 CB.SYM EQU 02600000 SINGLE STEP INT 118 CB.HTL EQU 026000000 CLOCK INFRRUPT 120 CB.SYM EQU 120000000 SYPEAKER ENABLE 123 CB.CLI EQU 0260000000 SYPEAKER ENABLE 124 CB.SYM EQU 0260000000 SYPEAKER ENABLE 125 CB.S.SYM EQU 02600000000000000000000000000000000000	
106 OP-TPD EQU 370Q TAPE DATA IN 106 OP-TPD EQU 370Q TAPE DATA OUT 107 IP-CON EQU 362Q Configure Port I 108 OP-CTL2 EQU 362Q Secondary Contribute EQU 026Q SYNC CHARACTER I 112 A.STN EQU 026Q SYNC CHARACTER I 113 A.STN EQU 026Q STX CHARACTER I 114 CB.ST EQU 000100000 STX CHARACTER I 115 CB.ST EQU 000100000 STX CHARACTER I 116 CB.ST EQU 0001000000 CLOCK INTERUPT I 119 CB.ST EQU 010000000 CLOCK INTERUPT I 119 CB.ST EQU 010000000 SPEAKER ENABLE I 119 CB.ST EQU 010000000 SPEAKER ENABLE I 120 CB.ST EQU 0100000000 SPEAKER ENABLE I 124 CB.ST EQU 000000100 SINGL-STAP EQU 126 CB.ST EQU 000000100 SINGL-STAP ENABLE I 126 CB.ST EQU 010000000 SINGL-STAP ENABLE I 127 CB.ST EQU 010000000 SINGL-STAP ENABLE I 127 CB.ST EQU 010000000 SINGL-STAP ENABLE I 127 CB.ST EQU 010000000 SINGL-ST ENCER ENTER ENTER ENTER ENTER ENTER ENTER EQU 010000000 SINGL-ST ENTER ENT	
107 IP-CON EQU 3700 TAPE DATA OUT 108 OP-CTLZ EQU 3620 CONTIGURE PORT 108 OP-CTLZ EQU 3620 CONTIGURE PORT 110 ** ASCII CHARACTERS. 111 A.STX EQU 0260 SYNC CHARACTER 113 A.STX EQU 0260 SYNC CHARACTER 114 CB-ST EQU 020100008 SINGLE STEP INT 115 CB-ST EQU 020100008 CLOCK INTERUPT 119 CB-CLI EQU 010000008 CLOCK INTERUPT 119 CB-CLI EQU 010000008 SPEAKER ENABLE 112 CB-SPK EQU 1000000008 SPEAKER ENABLE 112 CB-SPK EQU 1000000008 SINGLE STEP INT 113 CB-SPK EQU 0100000008 SPEAKER INTERUPT 115 CB-SPK EQU 000000018 SINGLE STEP ENA 116 CB-CLI EQU 000000018 SINGLE STEP ENA 117 CB-SSI EQU 000000018 SINGLE STEP ENA 118 CB-CLI EQU 000000018 SINGLE STEP ENA 119 CB-CLI EQU 000000018 SINGLE STEP ENA 112 CB-CLI EQU 000000018 SINGLE STEP ENA 112 CB-CLI EQU 000000008 SINGLE EQU 0000000008 SINGLE STEP ENA 112 CB-CLI EQU 0000000008 SINGLE STEP ENA 113 CB-CLI EQU 0000000008 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 00000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 00000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 000000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 000000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 000000000000 SINGLE STEP ENA 115 CB-CLI EQU 0000000000 SINGLE STEP ENA 115 CB-CLI EQU 00000000000 SINGLE STEP ENA 115 CB-CLI EQU 000000000000 SINGLE STEP ENA 115 CB-CLI EQU 00000000000 SINGLE STEP ENA 115 CB-CLI EQU 00000000000000000000000000000000000	
106   17-CUN   CONTINUE   Secondary Control   108   07-CTL2   CONTROL   Secondary Control   110   CONTROL   CONTRO	
110	Por
110	
112	
113 A.STX EQU 002Q STX CHARACTER 115 ++ FRONT PANEL HARDWARE CONTROL BITS. 116 CB.SSI EQU 00010000B SINGLE STEP INT 119 CB.SSI EQU 01000000B CLOCK INTERRUPT 120 CB.SPK EQU 10000000B SPEAKER ENABLE 122 ++ Secondary Control Bytes 124 CB2.SSI EQU 00000000B SINGle-Step Ena 125 CB2.CLI EQU 00000000B SINGle-Step Ena 126 CB2.DKG EQU 00000000B SINGle-Step Ena 127 CB2.SID EQU 001000000B SIGG-O Enable 127 CB2.SID EQU 001000000B SIGG-O Enable	
115   ** FRONT PANEL HARDWARE CONTROL BITS.   116   CB.SSI   EQU   000100008   SINGLE STEP INT   118   CB.HTL   EQU   001000008   CLOCK INTERRUPT   120   CB.SPK   EQU   100000008   SPEAKER   ENBELE   122   ** Secondary Control Bytes   123   CB2.SSI   EQU   0000000108   SINGLE-Step Ena   124   CB2.SSI   EQU   0010000018   URG-O   Enable   127   CB2.SID   EQU   010000008   Side-1   Select   127   CB2.SID   EQU   010000008   Side-1   Select	
116 117 118 C8-551 EQU 000010008 SINGLE STEP INT 119 C8-67L EQU 001000008 GLOCK INTERRUPT 120 C8-5PK EQU 100000008 SPEAKER ENABLE 122 ** Secondary Control Bytes 123 C82-551 EQU 0000000108 SINGle-Step Ena 124 C82-515 EQU 0010000008 SINGle-Step Ena 125 C82-CLI EQU 0010000008 SINGle-Step Ena 126 C82-DKG EQU 0010000008 SIGG-O Enable 127 C82-SID EQU 010000008	
117 C8.5SI EQU 000100008 SINGLE STEP INT 18 C8.MTL EQU 001000008 HONITOR LIGHT 19 C8.5CL EQU 010000008 SPEÄKER ENÄRHUPT 120 C8.5PK EQU 100000008 SPEÄKER ENÄRHUPT 123 C82.SSI EQU 0000000108 SINGle-Step Ena 126 C82.CLI EQU 001000000 CIOCK Interrupt 126 C82.CLI EQU 001000000 URG-O Enable 127 C82.SID EQU 010000008 SIGG-L Select	
118 C8-CII EQU 001000008 HOMITUR LIGHT 119 C8-CLI EQU 010000008 CLOCK INTERRUPT 120 C8-SPK EQU 100000008 SPEAKER ENABLE 122 ** Secondary Control Bytes 123 124 C82-SI EQU 000000018 Single-Step Ena 125 C82-CLI EQU 000000108 Clock Interrupt 126 C82-OKG EQU 0010000008 URG-O Enable 127 C82-SID EQU 010000008 Side-1 Select	RUPT
120 C8.5PK EQU 10000000B SPEAKER ENABLE 122 ** Secondary Control Bytes 123 124 C82.5SI EQU 00000001B Single-Step Ena 125 C82.CLI EQU 00000010B Clock Interrupt 126 C82.0KG EQU 00100000B URG-O Enable 127 C82.SID EQU 01000000B Side-1 Select	7
122 ** Secondary Control Bytes 123 124 CB2.SSI EQU 00000001B Single-Step Ena 125 CB2.CLI EQU 00000010B Clock Interupt 126 CB2.DKG EQU 00100000B URG-0 Enable 127 CB2.SID EQU 01000000B Side-1 Select	
124 C82.SSI EQU 000000018 Single-Step Ena 125 C82.CLI EQU 000000108 Clock Interrupt 126 C82.DRG EQU 0010000008 URG-O Enable 127 C82.SID EQU 010000008 Side-1 Select	/Ram8Go 2/
125 CB2.CL1 EQU 00000010B C 10ck Interrupt 126 CB2.OKG EQU 00100000B URG-0 Enable 127 CB2.SID EQU 01000000B Side-1 Select	
127 C82.SID EQU 01000008 Side	nable
129 ** DISPLAY MODE FLAGS (IN +DSPMOD+)	
131 DM.MR EQU 0	
132 DM.MW EQU 1 MEMORY W	
133 DM.RR EQU 2 REGISTER READ	

13   CALITAN EDU   000000113   Port 1740 Device—Type Mask   139   CALITAN EDU   0000000114   Port 1740 Device—Type Mask   139   CALITAN EDU   000010008   Print 1740 Califor	ype Mask ype Mask 1 => Pri
140 CM. PRI EQU 000010000 Port 1700 Device—Type Hisky 140 CM. PRI EQU 000100000 Henory Test/Mormal 142 CM. Afric EQU 0001000000 Henory Test/Mormal 143 CM. Afric EQU 001000000 Henory Test/Mormal 144 CM. Afric EQU 001000000 Henory Test/Mormal 145 CM. Afric EQU 0010000000 Henory Test/Mormal 145 CM. Afric EQU 001000000 Herory Test/Mormal 145 CM. Afric EQU 001001111 Herory Test/Mormal 145 CM. Afric EQU 00100111 Herory Test/Mormal 145 CM. Afric EQU 0010111 Herory Test/Mormal 145 CM. Afric EQU 0010111 Herory Test/Mormal 145 CM. Afric EQU 001011 Herory Test/Mormal 145 CM. Afric EQU 00101 Herory Test/Mormal 145 CM. Afric EQU 0000000	ype Mask 1 => Primary == 17 600; 1 => 19200
141 CN. NEW EQU   001000008   New Corrections   143 CN. NAME   EQU   10000008   National Rates   1 -> Auto-Boot   143 CN. NAME   EQU   10000008   National Rates   1 -> Auto-Boot   145 CND. HIT EQU   008   No Disk Installed   Valid on   146 CND. HIT EQU   008   No Disk Installed   Valid on   147 CND. HIT EQU   008   No Disk Installed   Valid on   147 CND. HIT EQU   008   No Disk Installed   Valid on   147 CND. HIT EQU   008   No Disk Installed   Valid on   147 CND. HIT EQU   008   No Disk Installed   Valid on   147 CND. HIT EQU   008   No Disk Installed   Valid on   157 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   VALID   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   008   NO DISK Installed   Valid on   158 KND. HIT EQU   VALID   158 KND. HIT EQU   V	600; 1 => 19200
142 Ch.4AU EQU 01000000B Baud Rate: 0 => 9600; 1 => 1920     144 Ch.4BU EQU 01000000B Auto-Boot: 1 => Auto-Boot     145 Ch.ABU EQU 00B	600; 1 =)
143 CNABD EUU 100000008 ALCO-Boot   144 CNABD EUU 100000008   145 CNO.H17 EUU 008   147 CNO.H17 EUU 100   100 CONSTANTS LENGTH VALID ON 100 CONSTANTS LENGTH SOUTH EUU 100 000 000 100 CONSTANTS LENGTH SOUTH EUU 100 000 000 100 CONSTANTS LENGTH SOUTH EUU 100 000 000 CONSTANTS LENGTH SOUTH EUU 100 000 000 EU CONSTANTS LENGTH SOUTH EUU 100 000 000 EU CONSTANTS LENGTH EUU 100 000 000 EU CONSTANTS EUU 100 000 000 EU CONSTANTS EUU 100 000 000 EU CONSTANTS EUU 100 000 EU CONSTANTS EUU 100 000 EU CONSTANTS EUU 100 00 EU CONSTANTS EUU 100 EU CONSTANTS EUU 100 00 EU CONSTANTS EUU 100 EU CONST	444
145 CNO.HI7 EQU	AUT0-80
140	21 20 21 10
147	Valid only
149 **   Boot Constants (Hi7 Rom Dependant)   151   Allo.UNI EQU   41061A   Disk Constants RDH Source   152   BOOTA   EQU   1300   Disk Constants Length   153   BOOTA   EQU   1300   Disk Constants Length   154   ETOT REAL   DISK CONSTANTS LENGTH   155   R.SOP.   EQU   34031A   Hi7 Clock Vector   155   R.SOP.   EQU   34031A   Hi7 Clock Vector   155   R.SOP.   EQU   34031A   Hi7 Clock Vector   155   R.SOP.   EQU   00000001B   DISK CONTINUE	
149 **	
151   Alo.uni Edu   41061A   Boot Device Unit Number   152   800TA   Equ   37132A   Uisk Constants RDM Source   154   800TA   Equ   100	/Ram8Go 2/
152 800TA	
153 BUDTAL	ourc
155 ERPIGNT EQU 30073A Common ADM Code 156 RASDP. EQU 34031A H17 Clock Vector 156 RAGGEK EQU 34031A H17 Clock Vector 157 RAGGER EQU 000000018 H17 Clock Vector 158 ** Segment Definitions H17 Clock Vector 150 ** Segment Definitions H17 Clock Ve	ج
156 ROMCLK EQU 34073A Common ADM Code  156 ROMCLK EQU 34031A H17 Clock Vector  159 ** Segment Definitions 161 S1 Equ 000000018 163 S3 Equ 000000008 164 S4 Equ 000100008 165 S5 Equ 000100008 165 S5 Equ 000100008 165 S6 Equ 000100008 167 S7 Equ 10000008 167 S7 Equ 0001000008 167 S7 Equ 0001011118 170 K.PLUS Equ 101011118 171 K.PLUS Equ 001011118 172 K.MINU Equ 001011118 173 K.STAR Equ 010111118	S
156   RDMCLK   EQU   34031A   H17 Clock Vector     158   **   Segment Definitions     159   160   Su   EQU   000000018     161   S1   EQU   000000008     162   S2   EQU   000010008     163   S3   EQU   001000008     164   S4   EQU   001000008     165   S5   EQU   001000008     165   S5   EQU   001000008     165   S5   EQU   100000008     167   S4   MINU   EQU   101011118   **     172   K.MINU   EQU   101011118   **     174   K.DIVO   EQU   010011118   **     175   K.MINU   EQU   010011118   **     176   K.MINU   EQU   010011118   **     177   K.MINU   EQU   010011118   **     178   K.MINU   EQU   0100111118   **     179   K.MINU   EQU   0100111118   **     170   K.MINU   EQU   0100111118   **     171   K.MINU   EQU   0100111118   **     172   K.MINU   EQU   0100111118   **     173   K.MINU   EQU   0100111118   **     174   K.MINU   EQU   0100111118   **     175   K.MINU   EQU   0100111118   **     176   K.MINU   EQU   0100111118   **     177   K.MINU   EQU   0100111118   **     178   K.MINU   EQU   0100111118   **     179   K.MINU   EQU   0100111118   **     170   K.MINU   EQU   0100111118   **     171   K.MINU   EQU   0100111118   **     172   K.MINU   EQU   0100111118   **     174   K.MINU   EQU   0100111118   **     175   K.MINU   EQU   0100111118   **     176   K.MINU   EQU   0100111118   **     177   K.MINU   EQU   0100111118   **     178   K.MINU   EQU   0100111118   **     179   K.MINU   EQU   0100111118   **     170   K.MINU   EQU   01001	:
158 ** Segment Definitions 159 160 S0 EQU 000000018 160 S1 EQU 000001008 163 S2 EQU 000010008 164 S4 EQU 000010008 165 S5 EQU 0010000008 165 S6 EQU 010000008 167 S7 EQU 100000008 167 S7 EQU 1000010118 178 K.PLUS EQU 100011118 177 K.PLUS EQU 100011118 177 K.PLUS EQU 011011118 177 K.STAR EQU 011011118 177 K.STAR EQU 011011118	
159 S0 EQU 00000018 161 S1 EQU 000000108 163 S3 EQU 000010008 164 S4 EQU 000010008 165 S5 EQU 001000008 165 S6 EQU 001000008 167 S7 EQU 100000008 167 S7 EQU 100011118 170 K-PLUS EQU 100011118 171 K-STAR EQU 0010011118 173 K-STAR EQU 010011118	/Ram8Go 2/
160 50 EQU 00000018 163 51 EQU 00001008 163 53 EQU 000010008 164 54 EQU 000100008 165 55 EQU 001000008 165 56 EQU 0010000008 167 57 EQU 100000008 167 57 EQU 101011118 171 K.PLUS EQU 101011118 172 K.MINU EQU 101011118 173 K.STAR EQU 011011118 174 K.OIVD EQU 011011118	
162 51 EQU 000000108 163 53 EQU 000010008 164 54 EQU 000100008 165 55 EQU 001000008 165 56 EQU 001000008 167 57 EQU 100000008 171 K.PLUS EQU 101011118 172 K.STAR EQU 010011118 174 K.SIAR EQU 010011118	
163 53 EQU 000010008 164 54 EQU 000100008 165 55 EQU 001000008 165 56 EQU 001000008 167 57 EQU 10000008 171 K.PLUS EQU 101011118 172 K.STAR EQU 011011118 174 K.SIAR EQU 011011118 175 K.SIAR EQU 011011118	
164 54 EQU 000100008 165 55 EQU 001000008 166 56 EQU 001000008 167 57 EQU 100000008 169 ** Key Definitions 171 K.PLUS EQU 101011118 173 K.STAR EQU 010011118 174 K.OIVD EQU 010011118 175 K.STAR EQU 010011118	
165 55 EQU 001000008 166 56 EQU 010000008 167 57 EQU 100000008 169 ** Key Definitions 171 K.PLUS EQU 101011118 172 K.MINU EQU 101011118 173 K.STAR EQU 010011118 174 K.DIVD EQU 010011118	
166 \$6 EQU 01000000B  167 \$7 EQU 10000000B  169 ** Key Definitions  171 K.PLUS EQU 10101111B  172 K.MINU EQU 10101111B  173 K.STAR EQU 01101111B  174 K.DIVD EQU 01101111B	
169 ** Key Definitions 170 K.PLUS EQU 101011118 171 K.PLUS EQU 100011118 173 K.STAR EQU 011011118 174 K.OIVD EQU 011011118	
169 ** Key Definitions 170   170   170   171   171   172   172   173   174   174   174   174   175   1	
170 K.PLUS EQU 101011118 + 172 K.MINU EQU 100011118 - 173 K.STAR EQU 011011118 + 174 K.DIVD EQU 010011118 / 175 K.MINU EQU 0100111118 / 175 K.MINU EQU 010011118 / 175 K.MINU EQU 0100111118 / 175 K.MINU EQU 010011118 / 175 K.MINU EQU 01001118 / 175 K.MINU EQU 01001118 / 175 K.MINU EQU 0100118 / 175 K.MINU EQU 010018 / 175 K.MINU EQU 010018 / 175 K.MINU EQU 010018 / 1	, c
171 K.PLUS EQU 101011118 172 K.MINU EQU 100011118 173 K.STAR EQU 011011118 174 K.DIVD EQU 010011118	/Y company
172 K.MINU EQU 100011118 173 K.STAR EQU 011011118 174 K.DIVD EQU 010011118	
174 K.DIVO EQU	
174 K.DIVD EJU	
175 4 1110	
176 K.DUI EQU OGGOLLILB 177 XTEXT TAPE	

1000,000   110		X621	*	TAPE EQ	EQUIVALENCES.			
184	000,000	180X	~	600		RECORD	F - MEMORY	
183 KT-15	000	182x	د∵∞	Fou	•	RECORD	F - RASIC	
184 X 17-48	000	183X	•	600	1 60		- COMPRE	
1855 R. BO S S RECORD TYPE - BASIC PAGO. AND DATA  187	000.004	184X	. 024	EOU	*		- NEW BASIC PR	
186	000.000	185X	œ	600	2		- BASIC DATA	
1885 ** BLOCK SILE FOR INTER-PRODUCT COMMUNICATION.  1905 BKKSIZ EDU 512  1905 BKKSIZ EDU 512  1907 ** IO PORT VALUES.  1907 TS. DUT EDU 3709  1907 TS. DUT EDU 3709  1907 TS. DUT EDU 3719  1907 TS. DUT EDU 3709  1907 TS. DUT EDU 10010108  1907 TS. DUT EDUT 10010108  1907 TS. DUT 10010108  1907 TS. DUT 10010108  1907 T	900 • 000	186X	· œ	EQU	•	RECORD	- BASIC PROG.	
168X ** BLOCK SIEF FOR INTER-FRUDUCT COMMUNICATION.  169X BLKSIZ EQU 512  190X BLKSIZ EQU 5700  190X TS.OUT EQU 3700  190X TS.OUT EQU 11010108  201 HI-HT EQU 110101018  202 HI-HT EQU 110101018  203 HI-HT EQU 110101018  204 HI-HT EQU 110101018  205 HI-HT EQU 110101018  206 HI-LA EQU 110101018  207 HI-HT EQU 110001018  208 HI-LA EQU 110001018  209 HI-LA EQU 1100010018  200 HI-LA EQU 11000100018  210 ** THESE BITS ARE SET IN CELL "NELG.  210 ** THESE BITS ARE SET IN CELL "NELG.  211 UG-RIF EQU 1000000018  212 UG-RIF EQU 000000018  ALLON PRINTE INTERUPT PROCESSING  213 UG-RIF EQU 000000018  ALLON PRINTE INTERUPT PROCESSING  214 UG-RIF EQU 000000018  ALLON PRINTE INTERUPT PROCESSING  215 UG-RIF EQU 0000000018  ALLON PRINTE INTERUPT PROCESSING		187×					:	
1992   1997		1887	:		FUK INIEK-P	מחחחת	3	
193X ** 10 PORT VALUES.  193X 10.01 EQU 3700 TAPE DATA DIT 193X 10.01 EQU 3700 TAPE STATUS DIT 193X 15.001 EQU 3710 TAPE STATUS DIT 193X 15.001 EQU 3710 TAPE STATUS DIT 203 MI.HT EQU 10.010.018 MALT 203 MI.HT EQU 10.010.018 MALT 204 MI.HT EQU 10.010.018 MALT 205 MI.HT EQU 10.010.018 MALT 206 MI.LTI EQU 10.010.018 MALT 207 MI.HT EQU 10.00.018 MALT 208 MI.LTI EQU 10.00.018 MALT 209 MI.HT EQU 00.010.018 MALT 210 WARR EQU EQU 10.00.00.00  211 WARR EQU EQU 10.00.00.00  212 WARR EQU EQU 00.00.00.018 MALT PROCESSING 213 WARR EQU EQU 00.00.00.018 214 WARR EQU EQU 00.00.00.018 215 WARR EQU EQU 00.00.00.018 216 WARR EQU EQU 00.00.00.018 217 WARR EQU EQU 00.00.00.018 218 WARR EQU EQU 00.00.00.018 219 WARR EQU EQU 00.00.00.018 210 WARR EQU EQU 00.00.00.018 211 WARR EQU EQU 00.00.00.018 212 WARR EQU EQU 00.00.00.018 213 WARR EQU EQU 00.00.00.018 214 WARR EQU EQU 00.00.00.018 215 WARR EQU EQU 00.00.00.018 216 WARR EQU EQU 00.00.00.018 217 WARR EQU EQU 00.00.00.018 218 WARR EQU EQU 00.00.00.018 219 WARR EQU EQU 00.00.00.018 210 WARR EQU EQU 00.00.00.018 211 WARR EQU EQU 00.00.00.018 212 WARR EQU EQU 00.00.00.018 213 WARR EQU EQU 00.00.00.018 214 WARR EQU EQU EQU 00.00.00.018 215 WARR EQU EQU EQU 00.00.00.018 216 WARR EQU EQU EQU 00.00.00.018 217 WARR EQU EQU EQU 00.00.00.018 218 WARR EQU EQU EQU 00.00.00.018 219 WARR EQU EQU EQU 00.00.00.018 210 WARR EQU EQU 00.00.00.018 210 WARR EQU EQU EQU 00.00.00.018 211 WARR EQU EQU EQU 00.00.00.018 212 WARR EQU EQU 00.00.00.018 213 WARR EQU EQU 00.00.00.018 214 WARR EQU EQU 00.00.00.00.018 215 WARR EQU EQU 00.00.00.018 216 WARR EQU EQU EQU 00.00.00.018 217 WARR EQU EQU EQU 00.00.00.018 218 WARR EQU EQU EQU 00.00.00.018 218 WARR EQU EQU EQU EQU 00.00.00.018 218 WARR EQU EQU EQU 00.00.00.018 218 WARR EQU	002.000	190x	: 88	EOU	512			
1922 *** 10 PORT VALUES.  1932 *** 10 PORT VALUES.  1932 10 10 EQU 3700 TAPE DATA UNI  1952 12 TO UNI EQU 3710 TAPE STATUS UNI  1952 12 MACHINE INSTRUCTIONS.  200 MITHE EQU 11001018 AFTURA  201 MITHE EQU 11001018 LATI  202 MITHE EQU 11001018 LATI  203 MITHE EQU 11001018 LATI  204 MITHE EQU 11001018 LATI  205 MITHE EQU 11001018 LATI  206 MITHER EQU 11001018 LATI  207 MITHER EQU 11001018 LATI  208 MITHER EQU 11001018 LATI  209 MITHER EQU 11001010 LATI  200 MITHER EQU 11001010 LATI  200 MITHER EQU 11001010 LATI  201 MITHER EQU 11001010 LATI  202 MITHER EQU 11001010 LATI  203 MITHER EQU 11001010 LATI  204 MITHER EQU 11001010 LATI  205 MITHER EQU 11001010 LATI  207 MITHER EQU 11001010 LATI  208 MITHER EQU 100000018 LATI  210 MITHER EQU 100000018 ALLION PAINTE INTERRUPT PROCESSING  210 MITHER EQU 100000018 ALLION PAINTE INTERPROPER PAINTERPROPER PAINTE INTERPROPER PAINTERPROPER PAINTERPROP		191X						
1934 TO.1M EQU. 3700 TAPE DATA IN 1954 TO.0M EQU. 3700 TAPE DATA QUI 1954 TY.1M EQU. 3700 TAPE STATUS QUI 1954 TY.1M EQU. 3710 TAPE STATUS QUI 200 MI.HT EQU. 110010018 HALT 201 MI.HT EQU. 11001018 HALT 202 MI.HE EQU. 11001018 HALT 203 MI.HE EQU. 11001018 HALT 204 MI.HT EQU. 11001018 ANI 205 MI.HT EQU. 001100108 ANI 206 MI.LM EQU. 001100108 ANI 206 MI.LM EQU. 110010018 LXI 0 207 MI.AMI EQU. 110010018 LXI 0 208 MI.LXIO EQU. 000100018 LXI 0 210 WWW. EQU. 000100018 ANI 211 WO.HLY EQU. 100000008 OISABLE HALT PROCESSING 212 WWW. EQU. 000000018 ALLOW PRIVATE INTERRUPT PROCESSING 213 WA.EXTEXT HÖSEQU.		192X	*	IO PORT	VALUE			
199x 15.01 EQU 3700 TAPE DATA IN  195x 17.01 EQU 3710 TAPE STATUS IN  195x 17.001 EQU 3710 TAPE STATUS OUT  195x 17.001 EQU 3710 TAPE STATUS OUT  200 MI.HET EQU 11001018 AETUN  201 MI.HET EQU 11001018 UNP UT  202 MI.HET EQU 11001018 UNP UT  203 MI.HET EQU 11001018 UNP UT  204 MI.HET EQU 11001018 UNP UT  205 MI.HET EQU 110001018 UNP UT  206 MI.HET EQU 110001018 UNP UT  207 MI.HET EQU 110001018 UNP UT  208 MI.HET EQU 110001018 UNP UT  210 WH UNP EQU 110000000 UND REFERSE UN CELL. WELLO.  210 WH UNP EQU 00000000 UND REFERSE UND UNP UT  211 WH UND EQU 00000000 UND REFERSE UND UNP UT  212 WH UNP EQU 00000000 UND REFERSE UND		193X					:	
1995 *** MACHINE INSTRUCTIONS:  1997 *** MACHINE INSTRUCTIONS:  200	000.370	194X	-	E O O	3700	TAPE DA	TI VI	
199X 15.01 E0U 371Q TAPE STATUS IN  199X 15.01 E0U 371Q TAPE STATUS OUT  200 MI.HLT E0U 110010018 AETURN 200 MI.HLT E0U 11001018 AETURN 200 MI.HLT E0U 11001018 AETURN 200 MI.HLT E0U 11001018 AETURN 200 MI.HLT E0U 110011018 ANI 201 MI.HLT E0U 111001108 ANI 202 MI.HLT E0U 111001108 ANI 203 MI.HLT E0U 111001108 ANI 204 MI.LXID E0U 100000018 LAI 210 ** USER OPTION BITS. 211 * THESE BITS ARE SET IN CELL "MELKG. 212 UO.NFR E0U 10000000 ALLON PRIVATE INTERRUTT PROCESSING 213 UO.NFR E0U 000000018 ALLON PRIVATE INTERRUTT PROCESSING 214 UO.NFR E0U 000000018 ALLON PRIVATE INTERRUTT PROCESSING 219 ** XTEXT HOSEQU	000.370	195x	-:	EQU	3700	TAPE DA	TA OUT	
199 ** MACHINE INSTRUCTIONS.  200 MI.HIT EQU 01110108 HALT 201 MI.HIT EQU 01110118 HELT 202 MI.ER EQU 11000118 JIPUT 203 MI.ER EQU 110000118 JIPUT 204 MI.ANT EQU 110000118 JIPUT 205 MI.ANT EQU 0001100109 LXI 206 MI.ANT EQU 000100018 LXI 207 MI.ANT EQU 000100018 LXI 208 MI.ANT EQU 000100018 LXI 211 ** USER OFTION 8ITS. 212 WILL EQU 100000008 DISABLE HALT PROCESSING 213 WILL EQU 100000018 ALLON PRIVATE INTERRUPT PROCESSING 214 UG.ANT EQU 000000018 ALLON PRIVATE INTERRUPT PROCESSING 215 WILL EQU 000000018 ALLON PRIVATE INTERRUPT PROCESSING 217 WG.CLK EQU 000000018 ALLON PRIVATE INTERRUPT PROCESSING	000.371	196X	_ '	E 0.0	3710	TAPE S	ATUS IN	
200 HINHE EQU 01110108 HILL 200 HINHE EQU 011101108 HILL 200 HINHE EQU 011101108 HILL 200 HINHE EQU 011010118 INPUT 200 HINHE EQU 110010118 INPUT 200 HINHE EQU 110010118 INPUT 200 HINHE EQU 110010118 OUTPUT 200 HINHE EQU 01101018 ALD 200 HILL 200 HINHE EQU 01101018 ALD 200 HILL 200 HINHE EQU 01101018 ALD 200 HILL 200 HINHE EQU 01101018 ALD 200 HINHE EQU 217 UGL EQU 000000018 ALD 200 PRIVATE INTERRUPT PROCESSING 219 UGL 200 000000018 ALD 200 PRIVATE INTERRUPT PROCESSING 219 UGL 200 000000018 ALD 200 PRIVATE INTERRUPT PROCESSING 219 UGL 200 000000018 ALD 200 PRIVATE INTERRUPT PROCESSING 219 UGL 200	000.371	X261	2	000	3710			
199 *** MACHINE INSTRUCTIONS.   200 NI-HLT EQU 011101108								
201 MI.HLT EOU 011010108 AETURN 202 MI.RET EOU 110010018 AETURN 203 MI.RET EOU 11001018 LUMP 203 MI.LN EOU 11010018 LUMP 204 MI.LN EOU 11010018 LUMP 205 MI.LN EOU 000100018 LNI COLL MELAC 206 MI.LN EOU 000100018 LNI CELL MELAC 207 MI.ANI EOU 11000108 LNI CELL MELAC 210 ** THESE BITS ARE SET IN CELL MELAC 211 * THESE BITS ARE SET IN CELL MELAC 212 UO.HET EOU 100000008 DISABLE HALT PROCESSING 214 UO.HET EOU 0000000108 ALLON PRIVATE INTERRUPT PROCESSING 215 UO.GLK EOU 0000000108 ALLON PRIVATE INTERRUPT PROCESSING 217 UO.GLK EOU 0000000108 ALLON PRIVATE INTERRUPT PROCESSING 219 XTEXT HÖSEQU		199	*	MACHINE	Z			
202 HI-RET EQU 11001018 AETURN 204 HI-JH EQU 11011018 JUND 205 HI-JH EQU 11011018 JUND 206 HI-LND EQU 101101018 AU 207 HI-LND EQU 101101018 AU 207 HI-LND EQU 101101018 AU 208 HI-LXID EQU 000100018 LXI D 211 ** THESE BITS ARE SET IN CELL "HEAG. 212 THESE BITS ARE SET IN CELL "HEAG. 213 UO.NF. EQU 00000008 DISABLE DISPLAY UPDATE 215 UO.NF. EQU 00000018 ALLOW PRIVATE INTERRUPT PROCESSING 217 UO.CH. EQU 000000018 ALLOW PRIVATE INTERRUPT PROCESSING	144	200		1101	801101110	-		
203 MI.IN EQU 11010118 INPUT 204 MI.JHP EQU 110000118 JUMP 205 MI.DA EQU 110100108 LNA 206 MI.LDA EQU 00111010108 LNA 206 MI.LDA EQU 0111010108 LNI 207 MI.LDA EQU 001010108 LNI 210 +* USER GPTIDN BITS. 210 +* USER GPTIDN BITS. 211 +* UG.HT EQU 000000018 LNI PROCESSING 212 UG.NER EQU 000000018 ALLON PRIVATE INTERRUPT PROCESSING 217 UG.CLK EQU 0000000018 ALLON PRIVATE INTERRUPT PROCESSING 219 XTEXT HÖSEQU	000 311	202	•. •	201	110010011	SETIEN		
204 H.JHP EQU 110000118 Jump 205 H.LDUT EQU 110100118 JUMPUT 206 H.LDUT EQU 111001108 LDA 207 H.AMI EQU 0110109 LDA 208 H.LXID EQU 000100018 LXI D 211 +	000 333	203	2	Fou	110110118	INPUT		
205 MI OUT EQU 110100118 OUTPUT 206 MI.ANA EQU 10101108 LUA 207 MI.ANA EQU 10101108 LXI D  210 ** USER OPTION BITS. 211 * THESE BITS ARE SET IN CELL .MFLAG. 212 * THESE BITS ARE SET IN CELL .MFLAG. 213 UO.MFR EQU 100000008 015ABLE HALT PROCESSING 214 UO.MFR EQU 00000018 015ABLE DISFLOT PANEL 215 UO.MGOU EQU 000000018 ALLOW PALVATE INTERRUPT PROCESSING 217 UO.CLK EQU 000000018 ALLOW PALVATE INTERRUPT PROCESSING 219 XTEXT HOSEQU	000 303	204	AI. JAP	EQU	110000118	0877	/Ra	
206 MI-LDA EQU 001110108 LUA 207 MI-ANI EQU 111001108 ANI 208 MI-LXID EQU 000100018 LXI D 210 +* USER OPTIDN BITS. 211 + 212 + THESE BITS ARE SET IN CELL "MFLAG. 213 - THESE BITS ARE SET IN CELL "MFLAG. 214 UO.HT EQU 100000008 01SABLE HALT PROCESSING 215 UO.NFR EQU 000000108 01SABLE DISPLAY UPDATE 216 UO.600 EQU 000000108 ALLON PRIVATE INTERRUPT 217 UO.CLK EQU 000000018 ALLON PRIVATE INTERRUPT 219 XTEXT HOSEQU	000.323	205	MI.00UT	EQU	110100118	OU TP UT		
207 MIANI EQU 11100110B ANI 208 MI-LXID EQU 00010001B LXI D 210 ** USER OPTION BITS. 211 * THESE BITS ARE SET IN CELL .MFLAG. 213 21 * THESE BITS ARE SET IN CELL .MFLAG. 214 UO.MFR EQU 0000000B DISABLE HALT PROCESSING 215 UO.MFR EQU 00000010B DISABLE DISPLAY UPDATE 215 UO.MFR EQU 00000001B ALLOM PRIVATE INTERRUPT 217 UO.CLK EQU 00000001B ALLOM PRIVATE INTERRUPT 219 XTEXT HÖSEQU	000.072	506	MI-LDA	EQU	00111100	<b>4</b> 07		
206 MILKID EQU 000100018 LXI D 210 ++ USER OPTION BITS. 211 + THESE BITS ARE SET IN CELL "MELAG. 213 - THESE BITS ARE SET IN CELL "MELAG. 214 UG.HLT EQU 10000000B 01SABLE HALT PROCESSING 215 UG.HLT EQU 00000010B 01SABLE DISPLAY UPOATE 216 UG.GOW EQU 00000001B ALLOM PRIVATE INTERRUPT 217 UG.CLK EQU 00000001B ALLOM PRIVATE INTERRUPT 219 XTEXT HDSEQU	000.346	202	HI.ANI	EOU	1110011108			
210 ** USER OPTION BITS. 211 * THESE BITS ARE SET IN CELL .MFLAG. 212 * THESE BITS ARE SET IN CELL .MFLAG. 213 * THESE BITS ARE SET IN CELL .MFLAG. 214 UO.HT EQU 0000000	000.021	208	·LXI		000100018			
210 ** USER DPTION BITS. 211 * THESE BITS ARE SET IN CELL .MFLAG. 213 * THESE BITS ARE SET IN CELL .MFLAG. 214 UO.HLT EQU 1000000B DISABLE HALT PROCESSING 215 UO.MFR EQU 00000010B NO REFESH OF FRONT PANE 216 UO.GOU EQU 00000010B ALLOM PRIVATE INTERRUPT 217 UO.CLK EQU 00000001B ALLOM PRIVATE INTERRUPT 219 XTEXT HÖSEQU								
212 * THESE BITS ARE SET IN CELL .MFLAG. 213 * THESE BITS ARE SET IN CELL .MFLAG. 214 UG.HLT EQU 10000000 DISABLE HALT PROCESSING 215 UG.ODV EQU 00000010B DISABLE DISPLAY UPDATE 217 UG.CLK EQU 00000010B ALLON PRIVATE INTERRUPT 219 XTEXT HÜSEQU		210	*	USER OF	TION BITS.			
212 * THESE BITS ARE SET IN CELL .WFLAG. 213		112	*					
214 UO.HLT EQU 1000000B DISABLE HALT PROCESSING 215 UO.NFR EQU CB.CLI NO REFRESH OF FRONT PANE 216 UO.ODU EQU 0000001B DISABLE DISPLAY UPDATE 217 UO.CLK EQU 00000001B ALLON PRIVATE INTERRUPT 219 XTEXT HÖSEQU		212	*		ARE SET	CELL	LAG.	
215 UO.NFR EQU CB.CLI NO REFRESH OF FRONT PANE 216 UO.000 EQU 00000010B DISABLE DISPLAY UPDATE 217 UO.CLK EQU 00000010B ALLON PRIVATE INTERRUPT 219 XTEXT HÖSEQU	200	416	TIME	600	10000000	OISARI	TIVE	
216 UG&DOV EQU 000000108 DISABLE DISPLAY UPDATE 217 UG&CLK EQU 000000018 ALLON PRIVATE INTERRUPT 219 XTEXT HÖSEQU	000,100	215	UD.NFR		CB.CLI	NO REF	RESH OF FRONT PANEL	
217 UG.CLK EQU 00000001B ALLOM PRIVATE INTERRUPT 219 XTEXT HÖSEQU	000.002	216	000,000	:	000000108	DISABL	E DISPLAY UPDATE	
21.9 XT EXT	000.001	217	UO.CLK	:	00000018	ALLOW	INTERRUPT	
219 XTEXT								
	000.000	219		XTEXT	HOSEOU	:		

	221X	* *	HDOS SY	STEM EQUIVALENC	ES.	
	XE22	,				
024.000	224X 225X	S. S.	E 0.0	24000A 25000A	SYSTEM AREA FOR GRTO SYSTEM AREA FOR GRTI	
05.6.000	226X	S. GR 12	EQU	26000A	AREA FOR	
030.000	228X 229X	ROMBOOT	EQU	30000A	ROM BOOT ENTRY	
040-100	230X 231X		ORG	40100A	FREE SPACE FROM PAN-8	
040.100	X32X	300	So	8	JUMP TO SYSTEM EXIT	
040.130	234X	SYDO	EQU	4 *	SYSTEM DISK ENTRY POINT	
040.130	235X	D.VE	98	24#3	ROM ENTRY V	
5.240	236X	D.RAM	SO	31	SYSTEM ROM WORK AREA	
040-343	738X	STAL	20	115	STATEM INTERNAL MORE AREAS	
1.126	239X		os os	16		
041.146	240X	S. SO VR	SO	2	STACK OVERFLOW WARNING	
041-150	241X		0.5	42200A-#	-	
001.032	242X	STACKL	600	*-5.50VR	STACK SIZE	
	X447	STACK	600	*	- MA+1 CYCTEM CTACK	
042.200	245X	USER	E 00	. *	Y Y	
	246		XTEXT	EDRAM		
	248X	*	EDRAM	- DISK RAM HORKAREA	AREA DEFINITION.	
	249X	* *		TOO A MOOIL		
	X162		0 10 10 10 10 10 10 10 10 10 10 10 10 10	5	•	
	252X	*	HOSEOU	MUST BE CHANGED	D WHEN THIS DECK IS CHANGED.	
	254X 254X					
040.240	255X		ORG	D.RAM		
0.46 0.40	256X	11			TABLE TO A CHORDEN TO DEFENT ON THE CO.	
040.241	258X		so	<b>-</b> 14	TARGET SECTOR (CURRENT OPERATION)	C.Z.
040.242	259X 260X	<b>-</b>	os	-		
•	261X	6			THIS AND SELECTION	
	263x	0	0.5	-	HEAD SETTLE DELAY COUNTER	
	Z64X	;				
040.245	265X	•	50	<b>~</b>	ADDRESS IN D.DRVIB FOR TRACK NUMBER	TRBER
	267X		3	7	10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	, was a
040.251	268X	D.DRVTB	0.5	5*4	TRACK NUMBER AND VOLUME NUMBER FOR 4 DRIVES	FOR 4 DRIVES
	X692				•	
040.261	X072	•	0.0	<b>-</b>	HAKD ERROR COUNT	
040.264	272X	0.0		7 1		
٠		:	i			

	•	*	GLOBAL	DISK ERROR COU	COUNTERS		
	2.						
040-265	9.7	D.E.E.E.E.E.E.E.E.E.E.E.E.E.E.E.E.E.E.E		0	BEGIN	ANING OF ERROR BLOCK	
40.26	- 8	D.E.HSY		٠.	SSIN	ING TEADER NAME	
40.26	2	D.E.CHK		1	DATA	CHECKSUM	
40.27	8:	D.E.HCK		1	HEADE	ER CHECKSUM	
40• 27 72 04	3	0.E. VOL			NON C	G VOLUME NUMBER	
040.273	o. OO	D.ERRL	os Os	0		LIMIT OF ERROR COUNTERS	
	8		-			:	
	82	*	I/0 0P	OPERATION COUNTS			
040-273	10. d	900		•			
Ģ	8	0.00 M M M	os o	2 7			
+	8						
000-037	290X	D.RAML	EQU	#-D.RAM			
	• .						
	XE62	:	JMP VECTORS	CTORS FOR ROM CODE			
	294X	*			: : : : :		
	2 95 X	*	SEE DISK	ROM FOR	ADDRESSES		
	296X		101000	100	1		
	298X	•		יייייייייייייייייייייייייייייייייייייי	E .	INTO LABLE TO ALLEKEU.	
040.130	X662		ORG	D.VEC			
	300%	3	2		1		
040-130	X105			7.7	5	A STUD (MUST BE FIRST)	
	303X	D.XOX		, "	A E 7	X • XOX • X	
•	304X	D.ABORT		m	d H C	R.ABORT	
•	305x	D.XIT		٣	JAP	ReXII	
040.147	30ex	D.READ		m (	d i	R.READ	
040,155	30.6X	2 - KINAU - C		۳.۳	È 9	T T T T T T T T T T T T T T T T T T T	
040-160	3008	0.CDE		n m	Ē	* CD = T	
040.163	310x	0.015		3	A P	R.OTS	
040.166	3118	0.501		<b>m</b> :	¥.	R SOT	
040-174	312X	0440	S	<b>.</b>	E E		
040.177	314x	D.LPS		3	Ę	R. LPS	
040.202	315x	D.R.08			AH.	R.KOB	
040.205	316	0.SDP		m (	A .	R.SOP	
040-013	X) 15	2.0		<b>7</b> (	È 1	X • VI V	
040-216	XOIE	7 10 1 V		<b>.</b>	E 1	K.512 P.101 V	
	320x	D. WSC			- E	P. USC	
040.224	321X	D.MSP		<b>m</b>	d H	R. KSP	
	322X	D. HNB			AH7	Z. Z.B.	
•	323		S	m:	E :	R. ERRT	
•	324		YTEXT	3 2056	Ē	K • UL T	
•			<.				

1000-117   1000-110	1978   1978			*	H17 CON	CONTROL INFORMATION	•NO
313X DE-HD 60U 000000108 HOLE DETECT 334X DE-HD 60U 000000108 WAITE OFFECT 334X DE-HD 60U 000001008 WAITE GATE ENGLE 335X DE-HD 60U 000010008 WAITE GATE ENGLE 335X DE-DS 60U 000010008 DAIVE SELECT 0 335X DE-DS 60U 000010008 DAIVE SELECT 0 345X DE-DS 60U 000010008 DAIVE SELECT 0 345X DE-DS 60U 000010008 DAIVE SELECT 0 345X DE-ST 60U 00010008 STEP CONNAND (CTIVE OFFE CONNAND CALVER OF SELECT 0 345X DE-ST 60U 000H STEP CONNAND CALVE AND SELECT OFFE CONNAND CALVE OF SELECT 0 355X UP-SC 60U 000H STEP CONNAND CALVE OF SELECT 0 355X UP-SC 60U 000H STEP CONNAND CALVE OF SELECT 0 355X UP-SC 60U 000H STEP CALVE OF SELECT OFFE CONNAND CALVE OF SELECT 0 355X UP-SC 60U 000H STEP CONNAND CALVE OF SELECT 0 355X UP-SC 60U 000H STEP CALVE OF SELECT 0 355X UP-SC 60U 000H STEP CALVE OF SELECT OFFE CONNAND CALVE OF SELECT 0 355X UP-SC 60U 000H STEP CALVE OF SELECT OFFE CONNAND CALVE OF SELECT OF S	33.X DF.ND EQU 00000018 HQLE DETECT 33.X DF.ND EQU 00000018 TALKO 0 DETECT 33.X DF.ND EQU 000001008 NATTE GATE ENABLE 33.X DF.ND EQU 000001008 NATTE GATE ENABLE 33.X DF.ND EQU 000000100 NATE SELECT 0.000000100 NATE SELECT 0.00000000 NATE SELECT 0.000000000 NATE NATITER DUFFER 0.000000000 NATE NATITER DUFFER 0.000000000 NATE NATE NATITER DUFFER 0.000000000 NATE NATE NATITER DUFFER 0.000000000 NATE NATE NATE NATE NATE NATE NATE NATE	000.177	329X	•:	EOU	7.	ISK CONTROL P
332 0F. FF E E 00 00001008 HRIE GATE ENABLE 333 0F. FF E E 00 00001008 HRIE GATE ENABLE 335 0F. FF E E 0 00001008 HRIE GATE ENABLE 335 0F. FF E E 0 00001008 NIVE SELECT 0 00001008 NIVE E E COMMAND (GLOUP) 000010008 NIVE E E E E COMMAND (GLOUP) 00001000 NIVE E E E E COMMAND (GLOUP) 00001000 NIVE E E E E COMMAND (GLOUP) 00001000 NIVE E E E E E E E E E E E E E E E E E E	133	1000	330X	DF. HD	Fou	00000018	HOLE DETECT
333 DE-NP EQU 00001000B NYLIC ENBLECT 335 DE-NP EQU 00001000B NYLIC ENBLECT 337 DE-DS EQU 0000010B DRIVE SELECT 2 338 DE-DS EQU 0000000B DRIVE SELECT 2 338 DE-DS EQU 0000000B DRIVE SELECT 2 348 DE-NR EQU 10000000B DRIVE SELECT 2 348 DE-NR EQU 10000000B DRIVE SELECT 2 348 DE-NR EQU 00000000B DRIVE EAGS 348 DE-NR EQU 00000000B DRIVE EAGS 348 DE-NR EQU 070H FILL CHARACTER (OUTPU) 358 UF-RE EQU 070H STATUS RANGER (OUTPU) 358 UF-RE EQU 070H STATUS RANGER (OUTPU) 358 UF-RE EQU 070H STATUS RANGER (OUTPU) 358 UF-RE EQU 0000000B RECEIVE DATA AVAILABI 358 UF-RE EQU 0000000B RECEIVE DATA AVAILABI 358 UF-RE EQU 10000000B TAANSHITTER BUFFER EQU 358 UF-RE EQU 358 UF-RE EQU 10000000B TAANSHITTER BUFFER EQU 358 UF-RE EQ	333 DE-NF EQU 00000100B NATIC ENABLE 335 DE-NF EQU 00000100B NATIC GATE ENABLE 335 DE-NF EQU 0000010B NATIC GATE ENABLE 336 DE-DS EQU 0000010B NATIC GATE ENABLE 336 DE-DS EQU 0000010B NATIC SELECT ON	000.002	332X	DF.10	EOU	00000000	TRACK 0 DETECT
334X DF-30 END 000000018 WRITE CARE EMBLE 334X DF-30 END 000000108 DRIVE SELECT 1 334X DF-30 END 000001008 DRIVE SELECT 2 334X DF-30 END 000010008 DRIVE SELECT 2 334X DF-30 END 000100008 DRIVE SELECT 2 344X DF-30 END 001000008 DRIVE SELECT 2 345X DF-30 END 001000008 DRIVE SELECT 2 345X DF-30 END 001000008 DRIVE SELECT 2 345X DF-30 END 001000008 DRIVE SELECT 3 345X DF-30 END 001000008 DRIVE SELECT 3 345X DF-30 END 001000008 DRIVE SELECT 3 345X DF-30 END 0010000008 STEP COMMAND (COUT OF DRIVE SELECT 3 345X UP-30 END 0010000008 STEP COMMAND (COUT OF DRIVE SELECT 3 355X UP-30 END 0010000008 STEP COMMAND (COUT OF DRIVE SELECT 3 355X UP-30 END 0010000008 STEP COMMAND (COUT OF DRIVE SELECT 3 355X UP-30 END 0010000008 STEP COMMAND (COUT OF DRIVE SELECT 3 355X UP-30 END 00100000008 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 00100000008 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 00100000008 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 00100000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 00100000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 00100000000 STEP COUT OF DRIVE SELECT 3 355X UP-30 END 0010000000000000000000000000000000000	33.4 DF-30 END 000000018 NRITE CRIE ENGLE  33.4 DF-32 EQU 00000018 NRITE CRIE ENGLE  33.4 DF-32 EQU 000000108 NRITE CRIE ENGLE  34.5 DF-32 EQU 000000018 NRITE CRIE ENGLE  34.5 DF-32 EQU 00000008 NRITE ENGLE  35.5 UP-32 EQU 00000008 NRITE ENGRE  35.5 UP-32 EQU 000000008 NRITE ENGRE  35.5 UP-32 EQU 00000008 NRITE ENGRE  35.5 UP-32 EQU 000000008 NRITE ENGR  35.5 UP-32 EQU 000000008 NRITE ENGR  35.5 UP-32 EQU 000000008	\$000°000		OF . N.	EQU.	000001008	WRITE PROTECT
1935   194	1975   1975	000.010		04.30	0	900010000	אות חבובנו
1972   3347 PF-D02   EQU   00000100B   DRIFE SELECT   0.0000100B   DRIFE SELECT   0.000100B   DRIFE SELECT   0.0001000B   DRIFE SELECT   DRIFE SELECT   0.0001000B   DRIFE SELECT   DRIFE SE	1972   337. PF-D5.0 EQU   00000109   00.0016	000.001	336X	DF.WG	EQU	000000018	WRITE GATE ENABLE
1992   1993   1994   1995	1904   334 OF-DIX EQU	000-000	337X	0F . D SO	EOU	000000108	DRIVE SELECT 0
100 340X OF-807 EQU 000100008 NUTUR ON (BOTH DRIVES OF SACK OF-807 EQU 000100008 NUTUR CONMAND (GPOUT) ON (BOTH DRIVES OF SACK OF-807 EQU 010000008 NUTUR ENABLE RAM (GPOUT) ON	100 3-40X OF-80 000100008	\$00°000		0F.0S1	200	000001008	ORIVE SELECT 1
100   341X DF.01   EQU   001000008   STEP COMMAND (ACTIVE   200   343X DF.87   EQU   100000008   STEP COMMAND (ACTIVE   343X DF.87   EQU   100000008   STEP COMMAND (ACTIVE   343X   3	100   341X DF_01   200   001000008   018ECTION (G-0UT)     100   343X DF_8T   200   010000008   018ECTION (G-0UT)     100   343X DF_8T   200   010000008   018ECTION (G-0UT)     110   340X   140	070		05.40	2.0	800001000	MOTOR ON (BOTH DRIVES)
100 342X DF-ST EQU 10000008 STEF COMMAND (ACTIVE 200 344X	100   342X 0F-87   60U   100000008   HRITE EMARLE RAM     344X   345X     345X   345X     355X   345X     365X     365X     365X     365X     365X     365X     365X     365	040-000	341X	ä	200	00100000	DIRECTION (0=0UT)
200 343X OF.WR EQU 10000000B WRITE EMBLE RAN 345X 345X 345X 345X 345X 345X 345X 345X	200 343X DF.NR EQU 10000000B NRITE ENABLE RAN 144X 375 4	000-100	342X	i d	600	010000008	IVE
34+X 34+X 34+X 34+X 34-X 34-X 34-X 34-X 34-X 34-X 34-X 34-	34+X 34+X 34-X 34-X 34-X 34-X 34-X 34-X 34-X 34-	000-200	343X	9	EQU	100000008	:
346X ** DISK UART PORTS AND CONTROL FLAGS.  347X ** DISK UART PORTS AND CONTROL FLAGS.  347X UP-FC EQU 07CH FILL CHARACTER  356X UP-FC EQU 07CH SYN CHARACTER (UUT)  354X UP-SC EQU 07CH SYN CHARACTER (UUT)  354X UP-SC EQU 07CH SYN CHARACTER (UUT)  354X UP-SC EQU 07CH SYN CHARACTER (UNT)  354X UP-SC EQU 07CH SYN CHARACTER ELECTOR (UUT)  354X UP-FC EQU 07CH SYN CHARACTER DEFINITIONS.  354X UP-FC EQU 010000008 RECEIVER PARITY ER BUFFER  354X UP-FC EQU 010000008 FILL CHAR TRANSHIT  354X UP-FC EQU 010000008 FILL CHAR TRANSHIT  354X UP-FC EQU 05CDH FIRL SYNC CHARACTER DEFINITIONS.  365X UP-FC EQU 05CDH PREFIX SYNC CHARACTER DEFINITIONS.	346X ** DISK UART PORTS AND CONTROL FLAGS. 348X 348X 348X 348X 348X 348X 348X 348X		344X				
347X ** DISK UART PORTS AND CONIRGL FLAGS.  340X UP-DF EQU 07CH DATA PORT 350X UP-FC EQU 07DH FILL CHARACTER 100T 351X UP-SC EQU 07DH SYNC RESET (INPUT) 353X UP-RC EQU 07EH SYNC RESET (INPUT) 354X UP-RC EQU 07CH SYNC RESET (INPUT) 354X UP-RC EQU 07CH SYNC RESET (INPUT) 354X UP-RC EQU 010000000 RECEIVE PORTRY EN 354X UP-RC EQU 0100000000 RECEIVE PORTRY EN 354X UP-RC EQU 010000000 RECEIVE PORTRY EN 354X UP-RC EQU 0100000000 RECEIVE PORTRY EN 354X UP-RC EQU 010000000 RECEIVE PORTRY EN 354X UP-RC EQU 0100000000 RECEIVE PORTRY EN 354X UP-RC EQU 010000000 RECEIVE PORTRY EN 354X	347X ** DISK UART PORTS AND CONINGL FLAGS.  348X UP-DF EQU 07CH DATA PORT 1349X UP-DF EQU 07DH FILL CHARGER 350X UP-SC EQU 07CH SYN CHARACTER (DUT 353X UP-SC EQU 07CH SYN CHARACTER (DUT 354X UP-SR EQU 07CH SYN CHARACTER (DUT 354X UP-SR EQU 07CH SYN CHARACTER (DUT 354X UP-SR EQU 07CH SYN CHARACTER UNITIONS.  356X UF-RP EQU 07CH SYN CHARACTER DEFINITIONS.  356X UF-RP EQU 07CH SYN CHARACTER DEFINITIONS.  356X UF-RP EQU 07CH SYN CHARACTER DEFINITIONS.  356X UF-RP EQU 07CH BANSHITTER BUFFER 356X UF-RP EQU 07CHARACTER DEFINITIONS.  366X ** HATDEF — HATDEF — HAT CONSTANT DEFINITIONS 365X ** HATDEF — HATDEF		346X				
348X WE EQU 07CH DATA PORT 51LL CHARACTER 350X UP.FC EQU 07DH STATUS GARGETER 351X UP.ST EQU 07DH STATUS GARGETER (OUT 351X UP.ST EQU 07EH SYN CHARACTER (OUT 351X UP.ST EQU 07EH SYN CHARACTER (OUT 351X UP.ST EQU 07EH SYN CHARACTER (OUT 351X UP.ST EQU 000000108 RECEIVE DATA AVAIL 351X UP.ST EQU 000000108 RECEIVER DATA AVAIL 351X UP.ST EQU 000000108 RECEIVER PARITY ER 351X UP.ST EQU 000000108 RECEIVER PARITY ER 351X UP.ST EQU 00000008 FILL CHAR PARITY ER 351X UP.ST EQU 00000008 TRANSMITTER BUFFER 351X ST CHARACTER DEFINITIONS.  364X CHARACTER DEFINITIONS.  365X UP.ST EQU 00000008 TRANSMITTER BUFFER 354X TRANSMITTER BUFFER 355X UP.ST HA7DEF — H47DEF	348X UP.DP EQU 07CH DATA PORT 350X UP.FC EQU 07DH STALL CHARACTER 351X UP.ST EQU 07DH STALL CHARACTER (QUIT 353X UP.ST EQU 07EH SYN CHARACTER (QUIT 353X UP.SR EQU 000000018 RECEIVE DATA AVAIL 355X UF.RD EQU 000000008 RECEIVE DATA AVAIL 355X UF.RD EQU 000000008 RECEIVE DATA AVAIL 355X UF.RD EQU 000000008 RECEIVE DATA AVAIL 355X UF.RD EQU 1000000008 FILL CHAR TRANSHIT ER BUFFER 360X UF.TBN EQU 1000000008 TRANSHITTER BUFFER 360X ATEXT H47DEF CONSTANT DOTINITIONS. 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER 147DEF CONSTANT TO SECOND TO		347X	*		PORTS	FLAGS
350 UP-FC EQU 070H FILL CHARACTER 351X UP-ST EQU 070H STATUS FLAGS 351X UP-ST EQU 070H STATUS FLAGS 352X UP-ST EQU 077EH STATUS FLAGS 352X UP-ST EQU 077EH SYNC RESET 10UT 354X UP-ST EQU 077EH SYNC RESET 10UT 354X UP-ST EQU 000000018 RECEIVE DATA AVAIL 355X UF-RPE EQU 000000018 RECEIVE DATA AVAIL 355X UF-RPE EQU 000000018 RECEIVE DATA AVAIL 355X UF-RPE EQU 000000018 RECEIVE PRETITE RUFFER 355X UF-RPE EQU 000000018 FLL. CHAR TRANSHIT ER 501X 360X UP-FC EQU 0100000018 TRANSHITTER BUFFER 364X 364X 400X EQU 075DH PREFIX SYNC CHARACTER DEFINITIONS.  3368X 44 HA7DEF — HA7DEF — HA7DEF SAGSTANT DOFINITIONS 366X 400X UP-REFIX SYNC CHARACTER BUFFER 365X 400X UP-REFIX SYNC CHARACTER DEFINITIONS.	350 UP-FC EQU 070H FILL CHARACTER 351X UP-ST EQU 070H STATUS FLAGS 351X UP-ST EQU 070H STATUS FLAGS 352X UP-SC EQU 076H SYN CHARACTER (0UT) 354X UP-ST EQU 000000018 RECEIVE DATA AVAIL 355X UF-RPE EQU 000001008 RECEIVE PARITY ER 355X UF-RPE EQU 000001008 RECEIVE PARITY ER 355X UF-RPE EQU 100000008 TRANSHITTER BUFFER 360X MANITTER BUFFER 365X WW MATTER HATTONS		3 4 8 X	:			
351X UP.ST EQU 078H 352X UP.ST EQU 078H 355X UP.SC EQU 08000018 355X UP.SPE EQU 0800018 355X UP.SPE EQU 08000018 355X UP.SPE EQU 080000018 355X UP.SPE EQU 08000018 355X UP.SPE EQU 080000018 355X UP.SPE EQU 080000018 355X UP.SPE EQU 0800000018 355X UP.SPE EQU	352X UP.ST EQU 07EH STATUS FLASS 352X UP.SC EQU 07EH SYN CHARACTER (BUT) 353X UP.SC EQU 07EH SYN CHARACTER (BUT) 355X UF.RDA EQU 000000108 RECEIVE DATA AVAIL 355X UF.RD EQU 000000108 RECEIVE OVERRUN 355X UF.RD EQU 000000108 RECEIVER OVERRUN 355X UF.RD EQU 00000008 FILL CHAR TRANSHITER BUFFER 360X 361X CHARACTER DEFINITIONS. 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DEFINITIONS. 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DEFINITIONS. 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DEFINITIONS.	000-174	X646	ີ :		0 7 CH	CALA PUKT
352X UP.SC EQU 07EH SYNC RESET (IMPUT) 353X UP.SR EQU 07EH SYNC RESET (IMPUT) 355X UF.RDA EQU 00000018 RECEIVER DATA AVAIL 355X UF.RDA EQU 000000108 RECEIVER PARITY ER 358X UF.RDA EQU 00000008 RECEIVER PARITY ER 359X UF.RDA EQU 100000008 FALL CHAR TRANSMIT SYNC CHARACTER DEFINITIONS. 363X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DATA AVAIL 364X SAX EQU 0FDH PREFIX SYNC CHARACTER DATA AVAIL 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DATA AVAIL 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DATA AVAIL 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DATA AVAIL 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DATA AVAIL 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DATA AVAIL 365X C.DSYN EQU 0FDH PREFIX SYNC CHARACTER DATA AVAIL 365X C.DSYN EQU DATA AVAIL 365X C.DS	352X UP.SC EQU 07EH SYNC RESET (INPUT) 353X UP.SR EQU 00000018 RECEIVE DATA AVAIL 355X UF.RDA EQU 000000108 RECEIVER DATA AVAIL 355X UF.RD EQU 00000108 RECEIVER OVERRUN 357X UF.RD EQU 010000008 FILL CHAR TANISHTY ER 358X UF.TEM EQU 10000008 FILL CHAR TANISHTY ER 358X UF.TEM EQU 10000008 FILL CHAR TANISHTY ER 358X UF.TEM EQU 10000008 FILL CHAR TANISHTY ER 358X UF.TEM EQU 010000008 FILL CHAR TANISHTY ER 364X 365X C.DSYN EQU 0FDH PREFIX SYNC CHARAC 368X ** HA7DEF — H47DEF — H47D CONSTANT DEFINITIONS 368X ** H47DEF — H47D CONSTANT DEFINITIONS	671.000	3204	5 3		10.00	CTATIS ELAGS
353X UP.SR EQU 076H SYNC RESET (IMPUT) 354X 355X UF.RDA EQU 000001010 RECEIVER DATA AVAIL 355X UF.RPE EQU 000001010 RECEIVER PARITY ER 35X UF.FCT EQU 00000100 RECEIVER PARITY ER 35X UF.FCT EQU 0000	353X UP-SR EQU 07EH SYNC RESET (INPUT) 354X 355X UF-RDA EQU 00000018 RECEIVE DATA AVAIL 355X UF-RP EQU 00000109 RECEIVER PARITY ER 358X UF-RP EQU 00000109 RECEIVER PARITY ER 358X UF-RP EQU 100000008 FILL CHAR TRANSMIT 359X 361X 360X 361X 365X CHARACTER DEFINITIONS. 364X 365X COSYN EQU 0FDH PREFIX SYNC CHARACTER DEFINITIONS. 365X COSYN EQU 0FDH BATDEF 365X COSYN EQU 0FDH PREFIX SYNC CHARACTER DEFINITIONS. 365X COSYN EQU 0FDH PREFIX SYNC CHARACTER DEFINITIONS.	77.	X 2 2 2	5:3		1100	CEARACTER (OUTPUT)
355x UF RDA EQU 00000018 355x UF RDA EQU 00000018 355x UF RDE EQU 00000108 358x UF RDE EQU 010000008 358x UF TEQU 010000008 359x UF TEN EQU 100000008 365x SA	355X UF RDA EQU 00000018 RECEIVE DATA AVAIL 355X UF RPE EQU 00000108 RECEIVER PARITY ER 355X UF RPE EQU 000001008 RECEIVER PARITY ER 359X UF FRE EQU 010000008 FILL CHAR TRANSMIT 359X UF FRE EQU 010000008 FILL CHAR TRANSMIT 365X UF FRE EQU 010000000 FILL CHAR TRANSMIT 365X UF FRE EQU 0100000000 FILL CHAR TRANSMIT 365X UF FRE EQU 0100000000000000000000000000000000000	8/1.000	355 X	5 3		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	CAN'S DENTI ( INDIA)
355X UF.RDA EQU 00000018 RECEIVE DATA AVAIL 356X UF.RDE EQU 00000108 RECEIVER DVERRUN 357X UF.RPE EQU 010000108 FILL CHAR TRAINT ER 359X UF.TEM EQU 100000008 FILL CHAR TRAINT ER 360X 360X 360X 360X 364X 365X G.DSYN EQU OFDH PREFIX SYNC CHARAC 366X 365X G.DSYN EQU OFDH PREFIX SYNC CHARAC 366X 365X G.DSYN EQU OFDH PREFIX SYNC CHARAC 366X 366X ** H470EF — H47 Constant Definitions 369X **	355X UF.RDA EQU 00000018 RECEIVE DATA AVAIL 356X UF.RPE EQU 00000108 RECEIVER OVERRUN 358X UF.RPE EQU 00000108 RECEIVER OVERRUN 358X UF.RPE EQU 010000008 FILL CHAR TRANSHIRE 358X UF.TBM EQU 100000008 TRANSHITER BUFFER 360X 360X 361X 365X C.DSYN EQU OFDH 365X C.DSYN EQU OFDH 365X ** H47DEF - H47 Constant Definitions 369X **	<u>-</u> :	X 40 E	5:	2	5	
356X UF.RDR EQU 00000100B RECEIVER DYERUN 357X UF.RPE EQU 010000100B FILL CHAR TANSHIT ER 359X UF.TBM EQU 10000000B TRANSHITTER BUFFER 361X 361X 361X 362X 363X ** CHARACTER DEFINITIONS. 363X ** CHARACTER DEFINITIONS. 365X C.DSYN EQU OFDH PREFIX SYNC CHARACTER BUFFER 364X 365X C.DSYN EQU OFDH PREFIX SYNC CHARACTER BUFFER 365X 365X ** H47DEF - H47DEF	356x UF.ROR EQU 00000010B RECEIVER DYERRUN 357X UF.RP EQU 01000000B FILL CHARTHANSHITTER BUFFER 360X UF.TBM EQU 10000000B TRANSHITTER BUFFER 364X 365X C.DSYN EQU 0FDH PREFIX SYNC CHARAC 366X 44 H47DEF - H47 Constant Definitions 369X 44 H47DEF - H47 Constant Definitions	100000	355X	5	600	000000018	RECEIVE DATA AVAILABLE
357x UF.RPE EQU 00000100B RECEIVER PARITY ER 358x UF.TBM EQU 10000000B TRANSMITTER BUFFER 360x 361x 361x 362x 365x 365x 365x 365x 365x 365x 365x 365	357X UF.RPE EQU 00000100B RECEIVER PARITY ER 358X UF.TBM EQU 10000000B FILL CHAR TRANSMIT 360X 361X 361X 364X 365X C.DSYN EQU OF DH 365X C.DSYN EQU OF DH 365X C.DSYN EQU OF DH 365X X X X X X X X X X X X X X X X X X X	000.002	356X	5	E 90	0000000	RECEIVER OVERRUN
359X UF.FGT EQU 010000008 FILL CHAR TRANSMITTER BUFFER 360X 361X 362X 363X ** CHARACTER DEFINITIONS. 364X 365X C.DSYN EQU OFDH PREFIX SYNC CHARACTER 368X ** H47DEF - H47 Constant Definitions 369X **	359X UF.FGT EQU 01000000B FILL CHAR TRANSMITTER 369X 4* CHARACTER DEFINITIONS. 363X 4* CHARACTER DEFINITIONS. 365X C.DSYN EQU OFDH PREFIX SYNC CHARACTER 368X 4* H47DEF 368	+00 -000	357X	3	E00	000001008	RECEIVER PARITY ERROR
353X UT-18N EUU LUUUUUUB 1863X 362X 364X 365X C-DSYN EQU OFDH 365X C-DSYN EQU OFDH 366X 44 H47DEF - H47 Constant Definitions 369X 4	363X UT-18N EUU LUUUUUUB IKANSALIIEK BUTTER 360X 361X 362X 363X 44 CHARACTER DEFINITIONS. 363X 44 STEXT H47DEF PREFIX SYNC CHARACTER 368X 44 H47DEF — H47 Constant Definitions 369X 4	000-100	358X	5	000	01000000	FILL CHAR TRANSMITTED
361X 362X 364X 364X 0.240 0.240 366X C.DSYN EQU OFDH 368X ** H47DEF PREFIX 368X ** H47DEF - H47 Constant Definit	362X 362X 364X 0.275 365X C.DSYN EQU OFDH 365X XTEXT H47DEF 368X ** H47DEF - H47 Constant Definit	000 • 000	3404E	5:	2	900000	
362X ** CHARACTER DEFINITIONS. 363X ** CHARACTER DEFINITIONS. 364X 365X C.DSYN EQU OFDH 366X ** H47DEF 369X ** H47DEF - H47 Constant Definit	362X ** CHARACTER DEFINITIONS. 364X 0.275 365X C.DSYN EQU OFDH 365 368X ** HA7DEF - H47 Constant Definit		361X				
363X ** CHARACTER DEFINITIONS. 364X 365X C.DSYN EQU OFDH 368X ** H47DEF - H47 Constant Definit 369X *	363X ** CHARACTER DEFINITIONS. 364X 365X C.DSYN EQU OFDH 368X ** H47DEF - H47 Constant Definit		362X	•		: ;	
365X C.DSYN EQU OFDH 0.240 368X ## H47DEF - H47 Constant Definit 369X #	3-5-5 3-5-5 3-5-5 X C.DSYN EQU OFDH 0-240 3-6-6 XTEXT H47DEF 3-6-8 4 H47DEF - H47 Constant Definit		36.3X		CHARAC	<b>*</b> :	•
368X ** H47DEF - H47 Constant Definition	0.240 366 XTEXT H47DEF 369X ** H47DEF - H47 Constant Definit	,	3648	Ċ	110	H010	
368X ** H470EF - H47 Constant Defin	368X ++ H47DEF - H47 Constant Defini	0.24	366	): }	XTEXT	H470EF	
## H470EF H47 Constant Defini	## H470EF H47 Constant Defini						
## HA7DEF - H47 Constant Defini	++ H470EF - H47 Constant Defini						
*	*		368×	*	H470EF	1	Defini
			X69E	*			

242.355   377.X   10.010   10.00   1		371X +	_	1-80 II	NSTRUCTIONS	
376.X ** 015X INTERFACE CONSTANTS  377.X 9-17.X 9-1		372X 373X 1 374X 1	•. •	E 0.0	101000108*256*111011018 101000118*256*111011018	
377X ** DISK INTERFACE CONSTANTS 377X ** THE TOTAL CONSTAN						
377X *   3		376X	*		CONSTANT	
317 x		377X 378X	_			
381X 5.5RR EQU 00000018 ERROR 5.5PR EQU 01000008 DATA TO 58 5.5PR EQU 000001008 DATA TO 58 5.5PR EQU 000001008 DATA TO 58 5.5PR EQU 000010008 DATA TO 58 5.5PR EQU 000000108 RESET 39 5X	000.000		•	E0.0	0	STATUS PORT
10.0000018	100 • 000		•	2	0.31A1V4	
394 X 5.00N EQU 00100000 INTERRED 385 X 5.0TR EQU 10000000 B DATA TO 385 X 5.0TR EQU 10000000 B DATA TO 385 X 5.8T EQU 000001000 B DATA TO 385 X 5.8T EQU 00001000 B DATA TO 58 395 X 5.8T EQU 00001000 B DATA TO 58 395 X 8.8T EQU 001000000 B DATA TO 58 395 X 8.8T EQU 001000000 B DATA TO 58 395 X 8.8T EQU 001000000 B DATA TO 58 395 X 8.8T EQU 001000000 B DATA TO 58 395 X 8.8T EQU 001000000 B DATA TO 58 395 X 8.8T EQU 001000000 B DATA TO 58 395 X 8.8T EQU 001000000 B DATA TO 58 395 X 8.8T EQU 001000000 B DATA TO 58 395 X 8.8T EQU 00000100 B DATA TO 58 395 X 8.8T EQU 00000100 B DATA TO 58 395 X 8.8T EQU 00000100 B DATA TO 58 395 X 8.8T EQU 00000100 B DATA TO 58 395 X 8.8T EQU 00000100 B DATA TO 58 395 X 8.8T EQU 00000100 B DATA TO 58 395 X 8.8T EQU 00000100 B DATA TO 58 395 X 8.8T EQU 00000000 B DATA TO 58 395 X 8.8T E	000.001	382X	S.ERR	EOU	000000018	~:
365 X 5.0TR EQU 100000008 DATA TE SEX 385 X 5.0TR EQU 00001008 DIP SW 384 X 5.8M EQU 000010008 DIP SW 384 X 5.8M EQU 000010008 DIP SW 394 X 5.8M EQU 000100008 DIP SW 395 X 4.8RE EQU 000100008 DIP SW 395 X 4.8RE EQU 000100008 DIP SW 395 X 4.8RE EQU 010000008 DIP SW 395 X 4.8RE EQU 010000008 DIP SW 395 X 4.8RE EQU 010000008 DIP SW 395 X 58.NLD EQU 010000008 DIP SW 395 X 58.NLD EQU 000100008 DIP SW 404 X 58.8TD EQU 000000008 DIP SW 405 X 58.8TD EQU 000000008 DIP SW 411X A.S.SL EQU 0001000008 DIP SW 510E TRACK A.D. A.S.SL EQU 0001000008 DIP SW 510E DIP SW	000.000	383X	N00.8	EOU		
386.X 5.540 EQU 000000108 DIP SW 389.X 5.541 EQU 000010008 DIP SW 399.X 5.543 EQU 000110008 DIP SW 391.X 4.RES EQU 000110008 DIP SW 391.X 4.RES EQU 000000108 RESET 395.X 4.RES EQU 000000108 WRITE 596.X 58.MED EQU 001000008 DELETE 400.X 58.MED EQU 000001008 DELETE 400.X 58.MED EQU 00000100 BYTE FLAGS 407.X 4.LEGA 410.X 58.8TD EQU 00000000 BYTE FLAGS 407.X 4.LEGA 410.X 5.51.D EQU 00000000 BYTE FLAGS 410.X 5.51.D EQU 000000011 B SECTOR	000-100	384X	SOIEN	300		ENABLE NAI
387X S.SWO EQU 000000108 DIP SW 388X S.SWI EQU 00001008 DIP SW 390X S.SWI EQU 000010008 DIP SW 390X S.SWI EQU 000100008 DIP SW 391X W.RES EQU 000100008 DIP SW 391X W.RES EQU 000100008 WRITE SPEX SWINE EQU 001000008 DIP SW 391X SWINE EQU 001000008 DIP SW 391X SWINE EQU 001000008 DIP SW 591X SWINE EQU 001000008 DIP SW 600X SWINE EQU 001000008 DIP SW 600X SWINE EQU 000010008 DIP SW 600X SWINE EQU 000010008 DIP SW 600X SWINE EQU 000001008 DIP SW 600X SWINE EQU 000001008 DIP SW 600X SWINE EQU 000000108 DIP SW 600X SWINE EQU 000000108 DIP SW 600X SWINE EQU 000000108 DIP SW 600X SWINE EQU 000000018 DIP SW 600X SWINE EQU 000000018 DIP SW 600X SWINE EQU 0000000018 DIP SW 600X SWINE EQU 000000000 DIP SW 600X SWINE EQU 0000000000 DIP SW 600X SWINE EQU 00000000000 DIP SW 600X SWINE EQU 0000000000 DIP SW 600X SWINE EQU 000000000 DIP SW 600X S	000.000	385X	× 0 ×			
388X S.SMI EQU 00000000B DIP SM 399X S.SM2 EQU 00001000B DIP SM 391X SECTOR EQU 00000010B RESET STATUS BYTE FLAGS STATUS BYTE EQU 01000000B WRITE STATUS BY SB.NP EQU 01000000B CELETE STATUS BY SB.NP EQU 00010000B CELETE STATUS BY SB.NP EQU 00010000B CELETE STATUS BY SB.NP EQU 000000100B CELETE STATUS BY SB.NP EQU 000000100B CELETE STATUS BY SB.NP EQU 00000010B CELETE STATUS BY SB.NP EQU 00000010B CELETE STATUS BY SB.NP EQU 00000000B CELETE STATUS BY SB.NP EQU 00000000B CELETE STATUS BY SB.NP EQU 001000000B CELETE STATUS BY STATUS BY STATUS BY SECTOR STATUS BY STATUS BY STATUS BY SECTOR STATUS BY SECTOR STATUS BY STATUS BY STATUS BY SECTOR STATUS BY STATUS BY STATUS BY SECTOR STATUS BY S	000.002	387X	S.SHO	EOU	000000108	SWITCHE
394X ** EQU 0001000B DIP SW 390X S.SW3 EQU 0001000B DIP SW 391X W.RES EQU 0001000B RESET 395X ** STATUS BYTE FLAGS 395X ** SALUM EQU 0100000B WRITE 395X ** SALUM EQU 0100000B WRITE 395X SALUM EQU 00100000B WRITE 395X SALUM EQU 00100000B WRITE 395X SALUM EQU 00100000B WRITE 595X SALUM EQU 00100000B WRITE 595X SALUM EQU 00010000B WRITE 595X SALUM EQU 00000100B WRITE 595X SALUM EQU 000000100B WRITE 595X SALUM EQU 00000010B WRITE 595X SALUM EQU 00000010B WRITE 595X SALUM EQU 00000001B WRITE 55X SALUM EQU 00000001B SYTE 55X SALUM EQU 0000001B STRACK 410X ASLUM EQU 0000001B STRACK 411X ASLUM EQU 0000001B SECTOR	000.000	388X	S.SW1	EQU	00000000	SWITCH:
394X ** STATUS BYTE FLACS 394X ** STATUS BYTE FLACS 395X ** SB_WP EQU 10000000B WINIT M 397X SB_WP EQU 10000000B WINIT M 397X SB_WP EQU 00100000B WINIT M 397X SB_WP EQU 00100000B WINIT M 405X SB_WP EQU 00010000B CRC ER 401X SB_RC EQU 00000100B LATE D 405X SB_RC EQU 00000100B LATE D 405X SB_RC EQU 00000010B BAD TR 406X ** AUXILLARY STATUS BYTE FLACS 406X ** AUXILLARY STATUS BYTE FLACS 406X ** AUXILLARY STATUS BYTE FLACS 406X ** EQU 00100000B TRACK 410X AS_SIA EQU 00100000B TRACK 410X AS_SIA EQU 00100000B SIDE 1 411X AS_SIA EQU 00000011B SECTOR	000.010	389X	S.SW2	200	00001000B	SWITCHS
394X ** STATUS BYTE FLAGS 394X ** STATUS BYTE FLAGS 395X ** STATUS BYTE FLAGS 395X ** STATUS BYTE FLAGS 395X ** SE-UNR EQU 10000000B UNIT M 395X ** SE-UNR EQU 00100000B UNIT M 405X \$8.0LTD EQU 00100000B CRC ER 402X \$8.0LTD EQU 00000100B CRC ER 403X \$8.0LTD EQU 00000100B CRC ER 404X \$8.0LTD EQU 00000010B CRC ER 405X \$8.0LTD EQU 00000010B CRC ER 405X \$8.0LTD EQU 00000010B CRC ER 405X \$8.0LTD EQU 000000010B CRC ER 405X \$8.0LTD EQU 000000010B CRC ER 405X \$8.0LTD EQU 000000010B CRC ER 407X ** AUXILLARY STATUS BYTE FLAGS 407X ** AUXILLARY STATUS BYTE FLAGS 407X ** EQU 001000000B TRACK 411X AS.SIA EQU 00000011B SIDE IRACK 411X AS.SIA EQU 000000011B SIDE IRACK 411X AS.SIA EQU 000000011B SECTOR	000.000	390X	2.543		90001000	20110
394X ** STATUS BYTE FLAGS 395X * 396X * 396X SB.WPD EQU 10000000B UNIT N 396X SB.WPD EQU 01000000B WRITE 399X SB.DLD EQU 00100000B WRITE 400X SB.RF EQU 0000100B CAFE CALX 401X SB.BTD EQU 00000100B CAFE DATE 404X SB.BTD EQU 00000001B LAFE DATE 406X ** AUXILLARY STATUS BYTE FLAGS 406	000.002	391X 392X	Æ	EQU	00000000	ESET
396X 396X 397X SB.WPR EQU 10000000B WILT N 397X SB.WPD EQU 00100000B WILETE 399X SB.WPR EQU 00100000B CRC ER 400X SB.WRF EQU 00001000B CRC ER 401X SB.CRC EQU 0000100B CRC ER 403X SB.LC EQU 0000010B ILLEGA 403X SB.LC EQU 00000010B BAD TR 404X SB.BTD EQU 00000001B BAD TR 406X 4** AUXILLARY STATUS BYTE FLAGS 407X ** AUXILLARY STATUS BYTE FLAGS 407X ** EQU 00100000B TRACK 408X AS.ODD EQU 00100000B TRACK 410X AS.STA EQU 000010000B SIDE ILLEGA 411X AS.SLA EQU 000010000B SIDE SECTOR		394X 395X	* *	STATUS	BYTE FLAGS	
394X 58-NPK E-00 010000008 WRITE 399X 58-NP E-00 001000008 DELETE 400X 58-NRF E-00 00100008 DELETE 401X 58-RF E-00 00000108 CRC ER 403X 58-RF E-00 000000108 LATE D 403X 58-RF E-00 000000108 LATE D 404X 58-RF E-00 000000108 BAD TR 404X 58-RF E-00 000000018 BAD TR 406X 4+ AUXILLARY STATUS BYFE FLAGS 407X 4+ AUXILLARY STATUS BYFE FLAGS 5070		396K		:		VORDA TON TIME
399X 58.0LD EQU 00100008 DELETE 400X 58.NRF EQU 00010008 CRC ER 401X 58.CRC EQU 00000108 LATE D 403X 58.LTD EQU 000000108 ILLEGA 404X 58.8TD EQU 000000108 BAD TR 404X 58.8TD EQU 000000018 BYFE FLAGS 406X 4** AUXILLARY STATUS BYFE FLAGS 407X 4 407X 4 408X 409X AS.ODD EQU 0010000008 TRACK 410X AS.STA EQU 0000100008 STDE 1 411X AS.STA EQU 000000118 SECTOR	000 - 200	39/X	SB.URK	:		WRITE PROTECTED DRIVE
400X 58.NRF EQU 00010008 CRC ER 601 000010008 CRC ER 602 00001008 CRC ER 603 S8.LTD EQU 000000108 LATE D 603 S8.LTD EQU 000000108 BAD TREF D 604 S8.BTD EQU 00000018 BYTE FLAGS 407X 407X 407X 4007X 4	000.000	399X	S8.DLD			DELETED DATA
### ##################################	000.000	X004	SB.NRF	:		NO RECORD FOUND
#05X SB-LID EQU 00000010B LAIE DO TO	000.010	¥104	SB.CRC	- :	000010008	CRC ERROR
404X 58.8TO EQU 0000001B BAD TR 406X ** AUXILLARY STATUS BYTE FLAGS 407X ** AUXILLARY STATUS BYTE FLAGS 408X ** AUXILLARY STATUS BYTE FLAGS 411X ** ASS.STATUS BYTE BYTE FLAGS 411X ** ASS.STATUS BYTE BYTE FLAGS 411X ** ASS.STATUS BYTE FLAGS	000°004	402X	58. LID		000000108	ILLEGAL COMMAND
407X * 407X * 407XLLARY STATUS BYTE FLAGS 407X * 408X 408X 408X 408X 410X AS.0DD EQU 01000000B 71XACK 411X AS.51A EQU 00010000B 510E 1 412X AS.5LM EQU 00000011B	000.001	X +0 +		EOU	000000018	FRACK
#00X ** AUXILLAKY STATUS BTTE FLAUS #08X #5.0DD EQU 01000000B TRACK #10X AS.1DD EQU 00100000B TRACK #11X AS.51A EQU 00010000B SIDE 1 #12X AS.5LM EQU 00000011B					i	
407X * 408X 409X AS.0D0 EQU 01000000B TRACK 410X AS.SIA EQU 00010000B SIDE 1 412X AS.SLM EQU 00000011B SECTOR		×90 <b>♦</b>	*	AUXIL	BYIE FLAG	
409% AS.0DD EQU 01000000B TRACK 410% AS.1DD EQU 00100000B TRACK 411% AS.51A EQU 00010000B SIDE 1 412% AS.5LM EQU 00000011B SECTOR		¥04 ×804	*			
411X AS.SLM EQU 000100008 SECTOR	000-100	X604	AS.000	:	010000008	TRACK O DOUBLE DENSITY
412X AS.SLM EQU 000000118 SECTOR	000 000	X014	A3.100	:	90000100	CTOC 1 AVATA AND E
	000.000	411X 412X	AS.SLA		000000118	SECTOR LENGTH MASK

000*000	YCTL	•			
00	416X				
000	417X	TOGALOO	ORG	0	ACOR
000.000	419X	DD.RST	So	1	READ STATUS
00.000	450X	DO.RAS	OS	1	READ AUX. STATUS
000.003	421X	00.150	SO		LUAD SECTUR COUNT PEAD ADDRESS OF LAST SECTOR ACCESSED
00-005	423X	00.REA	SO	4	
900-00	424X	DO.WRI	DS	Ī	WRITE SECTORS
00.007	425X	00.REAB	SO		READ SECTORS BUFFERED WRITE SECTIONS BIREEDED
000-011	427X	DO. WRD	0.5	4	DD.WRI + DELETED
100.012	428X	DO. WRBD	DS	1	DD.WRIB + DELETED
00.013	429X	00.CPY	DS		COPY
000.014	430X	DD. FRMO	0.5	<b>-</b>	CATAL LAT VO
000.015	431X	DD.FRM1	20	-	1
000*000	432X	DO FEET	200	4-	
000-000	434X	DD.RRDY		ī	eady
			- 6	4 6 6 6 7 6 7 6 7 6 7 6 7 6 7 6 7 6 7 6	
	436X	: .	Special	ne-end	
	438X				
000-000	¥36X	10	ORG	но 10	
000.020	X044	00.SPF0	So		SPECIAL FUNCTION O
000-021	X 2 7 7	00.585	200	4	FUNCTION
000.023	443X	00.SPF3	SO		FUNCTION
000.024		DD. SPF4	0.5		SPECIAL FUNCTION 4
000.069	YC		ŝ	4	
	X	* *	Special	Heath Function	
000	X644		500	1000	
000.200	451X	.0	DS		SET DRIVE CHARACTERISTICS
000.201	452X	Ο.	DS	-4	SEEK TO TRACK
000.202	453X	00.00	SO	<b>-</b>	DISK STATUS
000.204	455X	·. 🔾	DS		HRITE LOGICAL
000 - 205	456X	0;0	0.5		LUGICAL
000.202	458X	3 =	S	-	DATA LOG
000.210	X654		0.5		WRITE BUFFERED DELETED DATA LOGICAL

000.000 000.040 000.100 000.140	461X	*	Useful	Flags	
0000,000 0000,100 0000,140 0000,140	462X 463X	*			
000,100	464X	UNT.O	E0U	00000008	Unit: 0
000*140	X994	3	0.00	01000008	
000*000	X 8 9 4	5 1			
000-000	470X	5	2		
	472X	: 7	11011	8000000	
000.200	X424	\$10.1	000	10000008	Side: 1
000.200	X 2 2 4	N.OI.S	EOU	SID.0!SID.1	Side Mask
	X224 X924				
000.037	X084	SEC. M	EOU	000111118	Frack Nask
	482X				
000*+00	\$ 00 0	SSIZ.M	EQU	1024	Maximum Sector Size
	486X	:			
	X 8 4	ů.	EQU	128	
	4 8 8 8 8 8 8	*C.256 *C.26	) 0 0 0		
000.211	064	:	XTEXT	U8251 0EF	DEFINE 8251 USART BITS

100,000   100,	495X ** FORT ADDRESSES  496X ** FORT ADDRESSES  496X UNIT. B	100   100	100   100	1,000,000   1,00	995 995 995 995 995 995 995 995 995 995	10000000   10000000   10000000   10000000   100000000	10000000   10000000   10000000   10000000   100000000	
## PORT ADDRESSES  ## ## ## PORT ADDRESSES  ## ## ## ## ## ## ## ## ## ## ## ## ##	496X *** PORT ADDRESSES  497X VOR 600 0 0 10410 REGISTER IS EVEN  498X UOR 600 1 1 STATUS REGISTER IS NEXT  500X SC. MART 600 1 10000000 1 1 STOP BIT  500X WILLIAB 600 1 10000000 1 1 STOP BITS  504X WILLIAB 600 1 10000000 1 1 STOP BITS  504X WILLIAB 600 1 10000000 1 1 STOP BITS  504X WILLIAB 600 1 10000000	### ### ##############################	### POUT ADDRESSES  ### ### POUT ADDRESSES  ### ### POUT SCAURT EQU	### POUT ADDRESSES  ### ### POUT ADDRESSES  ### ### ### POUT ADDRESSES  ### ### ### ### ### ### ### ### #### ####	### ### ### ### ######################	### ### #### #########################	### ### ##############################	
499X USR EQU 0 DATA REGISTER IS WENT 499X USR EQU 1 DONOGOLE USART ADDRESS (IFF 9 50.2X 50.1X SC.UART EQU 17.20 CONSOLE USART ADDRESS (IFF 9 50.2X 50.3X 4.4 MODE INSTRUCTION CONTROL BITS. 50.3X 50.3X 6.4 MODE INSTRUCTION CONTROL BITS. 50.3X 6.4 MILLIB EQU 10.0000000 1 1.72 STOP BITS 50.3X MILLIB EQU 10.0000000 1 1.72 STOP BITS 50.3X MILLIB EQU 00.01000000 5 BIT CHARACTERS 50.3X MILLIP EQU 00.01000000 6 BIT CHARACTERS 51.3X MILLIS EQU 00.00010000 6 BIT CHARACTERS 51.3X MILLIS EQU 00.00010000 6 BIT CHARACTERS 51.3X MILLIS EQU 00.0001000 6 BIT CHARACTERS 51.3X MILLIS EQU 00.0000100 6 BIT CHARACTERS 51.3X MILLIS EQU 00.0000100 6 BIT CHARACTERS 51.3X MILLIS EQU 00.0000010 6 BIT CHARACTERS 61.3X MILLIS EQU 00.0000100 6 BIT CHARACTERS 61.3X MILLIS EQU 00.0000100 6 BIT CHARACTERS 61.3X MILLIS EQU 00.0000100 6 BITANISMITTER ERDY 61.3X MILLIS EQU 00.0000100 6 BITANISMITTER ERDY 61.3X MILLIS EQU 00.00000100 6 BITANISMITTER	499 x UOR	499X UOR EQU 1 STATUS REGISTER IS EVEN 499X USR EQU 1 STATUS REGISTER IS NEXT 500X USE EQU 1 STATUS REGISTER IS NEXT 503X 503X 503X 503X 503X 503X 503X 503X	4998 UDR EQU 1 STATUS REGISTER IS EVEN 4998 UDR EQU 1 3720 CONSOLE USART ADDRESS (IFF 9 501X SC. MART EQU 3720 CONSOLE USART ADDRESS (IFF 9 503X SC. WALL IS EQU 100000008 I STOP BIT 504X WILL IS EQU 100000008 I STOP BIT 505X WILL IS EQU 000000008 STIT CARACTERS 510X WILL IS EQU 000000008 STIT CARACTERS 511X WILL IS EQU 000000008 STIT CARACTERS 512X WILL IS EQU 000000008 STIT CARACTERS 513X WILL IS EQU 000000008 STAT CARACTERS 513X WILL IS EQU 000000008 STAT CARACTERS 513X WILL IS EQU 000000008 STAT CARACTERS 523X WILL IS EQU 000000008 STAT CARACTERS 523X WILL IS EQU 000000008 STAT WANNIT ERROR 523X WILL IS EQU 000000000 STAT WANNIT ERROR 523X WILL IS EQU 0000000000 STAT WANNIT ERROR 523X WILL IS EQU 00000000000000000000000000000000000	499   499	499x UOR   600	499x UOR   600   0	499x UOR EQU 0 DATA REGISTER IS WENT 50.0x	
901X SC.UART EQU 372Q CONSOLE USART ADDRESS (IFF 8 502X 504X 504X 504X 604 Mi B COU 010000000 1 57DP BIT 504X UNI B COU 100000000 1 1.2 STOP BIT 504X UNI B COU 100000000 1 1.2 STOP BIT 504X UNI B COU 000000000	500X   50X   5	500X   50X   5	500X 500X 500X 500X 500X 500X 500X 500X	500X SC.UART EQU 372Q CONSOLE USART ADDRESS (IFF 8 50X UNILLE EQU 010000008 1 STOP BIT 50XX UNILLE EQU 010000008 1 LAZ STOP BIT 50XX UNILLE EQU 010000008 1 LAZ STOP BIT 50XX UNILLE EQU 000000008 1 LARACTERS 51XX UNILLE EQU 00000000	90.X SC.UART EQU 372Q CONSOLE USART ADDRESS LIFF B 60.X WILLIS EQU 010000008 1 STOP BITS 50.X WILLIS EQU 010000008 1 STOP BITS 50.X WILLIS EQU 010000008 2 STOP BITS 50.X WILLIS EQU 0000100008 2 STOP BITS 50.X WILLIS EQU 0000100008 2 STOP BITS 50.X WILLIS EQU 0000100008 2 STOP BITS 50.X WILLIS EQU 000010000 6 SIT CHARCTERS 50.X WILLIS EQU 000010000 6 SIT CHARCTERS 50.X WILLIS EQU 000010000 6 SIT CHARCTERS 50.X WILLIS EQU 00001000 6 SIT CHARCTERS 50.X WILLIS EQU 00001000 B REDITE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE ENABLE FINE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE ENABLE FINE ENABLE FINE EDU 5.X WILLIS EQU 00001000 B REDITE ENABLE FINE ENABLE FINE EDU 5.X WILLIE EQU 00001000 B REDITE ENABLE FINE ENABLE FINE ENABLE FINE EDU 5.X WILLIE EQU 00001000 B REDITE ENABLE FINE ENABLE FINE EDU 5.X WILLIE EQU 00001000 B REDITE ENABLE FINE ENABLE FINE EQU 00001000 B REDITE ENABLE FINE ENABL	501X	900X 900X 900X 900X 900X 900X 900X 900X	
592X         592X           593X         400E INSTRUCTION CONTROL BITS.           504X         44         MODE INSTRUCTION CONTROL BITS.           505X         UNI.1B         EQU         10000000B         1 JZ STOP BITS           505X         UNI.2B         EQU         10000000B         2 STOP BITS           505X         UNI.2B         EQU         00010000B         EVEN PARITY           20         505X         UNI.2B         EQU         00010000B         SBIT         CHARACTERS           20         512X         UNI.2B         EQU         00000100B         SBIT         CHARACTERS           20         512X         UNI.2B         EQU         00000100B         SBIT         CHARACTERS           31         UNI.2B         EQU         00000100B         SBIT         CHARACTERS           40         512X         UNI.1B         EQU         00000100B         CLOCK         X 64           513         UNI.1B         EQU         00000100B         CLOCK         X 64           514         UNI.1B         EQU         00000100B         CLOCK         X 64           515         UNI.1B         EQU         000000100B         READER LOR	502X   502X   502X   502X   502X   502X   503X   504X   600X   60X   6	503X	503X   503X   503X   504X	903X 903X 903X 903X 903X 903X 903X 903X	504X	902X 903X 903X 903X 905X 905X 911	902X 903X 903X 904X 905X 905X 905X 905X 905X 905X 905X 905	: <b>•</b>
505X         UNIL-1B         GUD 000000B         1 STOP BITS           50         VMI-1B         EQU         10000000B         1 1/2 STOP BITS           50         VMI-1B         EQU         10000000B         2 STOP BITS           20         505X UMI-PE         EQU         00100000B         EVEN PARITY           20         510X UMI-PE         EQU         00100000B         6 BIT CHARACTERS           20         511X UMI-PE         EQU         0000100B         6 BIT CHARACTERS           20         512X UMI-PE         EQU         0000100B         6 BIT CHARACTERS           20         513X UMI-PE         EQU         00001100B         6 BIT CHARACTERS           20         513X UMI-PE         EQU         00001100B         6 BIT CHARACTERS           21         MIL-LA         EQU         00001100B         6 BIT CHARACTERS           21         MIL-LA         EQU         00001100B         6 BIT CHARACTERS           20         MIL-LA         EQU         00001100B         6 BIT CHARACTERS           21         MIL-LA         EQU         00001000B         CLOCK         X           21         MIL-LA         EQU         00000010B         CLOCK         X	50.5%         UNIT.1B         EQU         01000000B         1 STOP BITS           50.0         50.6X         UMI.2B         EQU         10000000B         1 1/2 STOP BITS           50.0         50.6X         UMI.2B         EQU         00100000B         EX STOP BITS           20         51.X         UMI.2B         EQU         00100000B         SIT <characters< td="">           10         51.X         UMI.1C         EQU         00000100B         SIT         CLOCK         X           11         51.X         UMI.1C         EQU         00000100B         CLOCK         X         A           20         51.X         UMI.1C         EQU         00000100B         CLOCK         X         A           51.X         UMI.1C         EQU         00000010B         READIR ESET         A         A         S         A         A         S         A         A         A         A</characters<>	50.X         UNITIB         GOUDODOODB         1 STOP BITS           50.0         50.X         UNIT.B         EQU         10000000B         1 1/2 STOP BITS           50.0         50.X         UNIT.P         EQU         10000000B         EVEN PARITY           20         50.X         UNIT.P         EQU         00010000B         EVEN PARITY           20         50.X         UNIT.P         EQU         0001000B         5 BIT         CHARACTERS           20         51.X         UNIT.P         EQU         0000100B         5 BIT         CHARACTERS           30         51.X         UNIT.C         EQU         00001100B         5 BIT         CHARACTERS           10         51.X         UNIT.C         EQU         0000110B         6 BIT         CHARACTERS           11         51.X         UNIT.C         EQU         0000110B         6 BIT         CHARACTERS           10         51.X         UNIT.C         EQU         0000110B         CLOCK         X         4           51.X         UNIT.C         EQU         0000100B         BIT         CHARACTERS         1           51.X         UNIT.C         EQU         0000100B         BIT         CHOC	100   500   1   100   1   100   1   100   1   1	1700   1700	10	100   5005X   100   100   100   10   10   10   10	17.0   17.0	
17.2   17.3   17.3	11/2   11/2	17.5 STOP BITS   17.6	100   507X UMI.*N   EQU	1	100   100	100   100	100   100	
100   500	100   500	100   200	100   100	100   100	1000   500X UNIT.E	100   508X UNI.2E EQU   00100008   EVEN PARITY     20	100   500	
0.20         510X UMIPA         EQU         00010000B         USE PARITY           0.00         511X UMILS         EQU         00000100B         5 BIT CHARACTERS           0.00         513X UMILS         EQU         0000100B         7 BIT CHARACTERS           0.10         513X UMILS         EQU         0000100B         8 BIT CHARACTERS           0.10         515X UMILS         EQU         0000010B         CLOCK         X 16           0.00         515X UMILS         EQU         00000010B         CLOCK         X 16           0.00         516X         CCCCK         X 16         CLOCK         X 16           0.00         518X         CCCCK         X 16         CLOCK         X 16           0.00         518X         CCCCK         X 16         CLOCK         X 16           0.00         521X         UCIRE         EQU         00100000B         READER-ON CONTROL           0.00         522X         UCIRE         EQU         000000100B         RARDR RESET           0.00         523X         UCIRE         EQU         000000100B         FRANING ERROR           0.00         525X         UCIRE         EQU         000000010B         FRANING ER	100   510X UNI.PA EQU   000100008   0.5 PARITY   0.000   0.000   0.00	DOCUMENT	100   100	100   510X UMI.PA EQU   00010000B   USE PARITY   CAGAGCERS   5 8.17 CHARACTERS   00000100B   5 8.17 CHARACTERS   010   00000100B   5 8.17 CHARACTERS   010   514X UMI.LT   EQU   00000100B   7 8.17 CHARACTERS   010   514X UMI.LT   EQU   00000100B   7 8.17 CHARACTERS   010   514X UMI.LT   EQU   00000100B   8 8.17 CHARACTERS   010   514X UMI.LT   EQU   00000010B   CLOCK X 16   CLOC	100   510X UHI.PA EQU   00010006   5 BT   CHARACTERS   5 BT   CH	100   510X UHI.PA   EQU	100   100   101	
December 2000   December 3	DOCUMENT   DOCUMENT	DOC	DOC   DOC	DOCO	DOCTOR   D	100   5.11 UII.15   EQU	100   511X UMI.LS   EQU	
004 512X UNI-LS EQU 00001000B 5 BIT CHARACTERS 014 513X UMI-L7 EQU 00001000B 8 BIT CHARACTERS 015 514X UMI-L8 EQU 0000100B 8 BIT CHARACTERS 001 515X UMI-LX EQU 0000001B CLOCK X 16 002 515X UMI-LAX EQU 0000001B CLOCK X 64 003 515X UMI-64X EQU 0000001B CLOCK X 64 004 523X UCI-R EQU 00100000B READER—UNCONTROL 020 522X UCI-R EQU 00100000B READER—UNCONTROL 030 523X UCI-R EQU 00000010B RECEIVE ENABLE 040 524X UCI-R EQU 00000010B RECEIVE ENABLE 052 525X UCI-R EQU 00000010B RECEIVE ENABLE 053 053 USR-F EQU 00000000B PARITY ERROR 054 533X USR-F EQU 00000010B PARITY ERROR 055 535X USR-F EQU 00000010B RECEIVER READP 056 535X USR-F EQU 00000010B RECEIVER READP 057 535X USR-F EQU 00000010B RECEIVER READP 058 535X USR-F EQU 00000010B RECEIVER READP 059 535X USR-F EQU 00000010B RECEIVER READP	100   513X UMI.LE 600 0000100B 0 BIT CHARACTERS 014   513X UMI.LE 600 0000100B 0 BIT CHARACTERS 014   513X UMI.LE 600 0000100B 0 BIT CHARACTERS 015   515X UMI.LE 600 0000010B   CLOCK X 16   CLOCK X	004 513X UMI-LS EQU 0000100B 5 BIT CHARACTERS 014 513X UMI-LS EQU 00001100B 6 BIT CHARACTERS 015 CLOCK X 15 CLOCK X 16 CL	100   513X UNITED   510   511 CHARACTERS   514 UNITED   510   513 UNITED   510   513 UNITED   510   513 UNITED   513 UNI	100   512X UNITED   510   511 CHARACTERS   514 UNITED   510   512 UNITED   51	014 513X UNIT-LE EQU 0000100B 7 BIT CHARACTERS 014 513X UNIT-LE EQU 00001100B 1 BIT CHARACTERS 014 515X UNIT-LE EQU 00001100B 1 BIT CHARACTERS 015 515X UNIT-LE EQU 0000010B 1 CLOCK X 14 515X UNIT-LE EQU 0000010B 1 CLOCK X 15 515X UNIT-LE EQU 00000010B 1 CLOCK X 16 515X UNIT-LE EQU 00000010B 1 CLOCK X 16 515X UNIT-LE EQU 00010000B READER-ON CONTROL 020 020 020 020 020 020 020 020 020 02	004 5313 UNIT-LB EUU 0000100B 7 BIT CHARACTERS 004 5133 UNIT-LB EUU 0000100B 7 BIT CHARACTERS 005 5143 UNIT-LB EUU 0000010B 6 BIT CHARACTERS 005 5143 UNIT-LB EUU 0000010B CLOCK X 14 5143 UNIT-LB EUU 0000010B CLOCK X 14 5143 UNIT-LB EUU 0000001B CLOCK X 14 5143 UNIT-LB EUU 0000001B CLOCK X 14 5143 UNIT-LB EUU 0000001B CLOCK X 14 5143 UNIT-LB EUU 00100000B READER-ON CONTROL 520 ULI-LB EUU 00100000B READER-ON CONTROL 524 ULI-LB EUU 00100000B RANITER-BROR 520 ULI-LB EUU 00100000B FAANING ERROR 530 USR-FE EUU 00100000B FAANING ERROR 530 USR-FE EUU 00100000B FAANING ERROR 530 USR-FE EUU 00010000B FAANING ERROR 530 USR-FE EUU 000100000B FAANING ERROR 530 USR-FE EUU 00010000B FAANING EUU 00010000B FAANING ERROR 530 USR-FE EUU 00010000B FAANING ERROR 530 USR-FE EUU 0001000	004 513X UNILLS ENU 00001100B 0 BIT CHARACTERS 014 513X UMILLS ENU 00001100B 0 BIT CHARACTERS 015 514X UMILLS ENU 00001100B 0 BIT CHARACTERS 015 514X UMILLS ENU 00000010B CLOCK X 14 515X UMILLS ENU 00000010B CLOCK X 64 515X UMILLS ENU 00100000B READER-ON CONTROL 052X UCI.R ENU 00100000B RECENT RESET 000 000 000 000 000 000 000 000 000 0	
014 00000100B 8 BIT CHARACTERS 001 515X UMI.1X EQU 00000010B CLOCK X 1 002 516X UMI.164 EQU 00000010B CLOCK X 16 003 516X UMI.164 EQU 00000010B CLOCK X 64 003 518X ++ COMMAND INSTRUCTION BITS. 040 522X UCI.1R EQU 00100000B ERROR RESET 040 522X UCI.1R EQU 00000010B RECEIVE ENABLE 004 524X UCI.1R EQU 00000010B RECEIVE ENABLE 005 525X UCI.1E EQU 00000010B RECEIVE ENABLE 006 526X UCI.1E EQU 00000010B RECEIVE ENABLE 007 526X UCI.1E EQU 00000010B RECEIVE ENABLE 007 526X UCI.1E EQU 00000010B RECEIVE ENABLE 008 526X UCI.1E EQU 00000010B RECEIVE ENABLE 009 526X UCI.1E EQU 000000010B TRANSHIT ENABLE 000 531X USR.1E EQU 00000100B PARITY ERROR 000 532X USR.1E EQU 00000100B RECEIVER READY 000 532X USR.1E EQU 00000010B RECEIVER READY 001 532X USR.1E EQU 00000010B RECEIVER READY 001 533X USR.1E EQU 00000010B RECEIVER READY 001 535X USR.1E EQU 00000010B RECEIVER READY	014 00000100B 8 BIT CHARACTERS 001 515X UMI.1X EQU 00000010B CLOCK X 1 002 516X UMI.164X EQU 00000010B CLOCK X 16 003 516X UMI.64X EQU 00000011B CLOCK X 64 004 520X UCI.R EQU 00100000B READER—UND CONTROL 004 522X UCI.R EQU 00100000B READER—ENERRUPTS 004 522X UCI.R EQU 00000010B RECEIVE ENABLE 005 525X UCI.R EQU 00000100B RECEIVE ENABLE 006 525X UCI.R EQU 00000010B RECEIVE ENABLE 007 526X UCI.R EQU 00000010B RECEIVE ENABLE 008 526X UCI.R EQU 00000010B RECEIVE ENABLE 009 526X UCI.R EQU 00000010B RECEIVE ENABLE 000 526X UCI.R EQU 00000010B RECEIVE ENABLE 000 526X UCI.R EQU 00000010B RECEIVE READR 000 531X USR.P E EQU 00000100B PARITY ERROR 000 532X USR.R EQU 00000100B RECEIVER READP 001 532X USR.R EQU 00000010B RECEIVER READP 002 535X USR.R EQU 00000010B RECEIVER READP 003 535X USR.R EQU 00000010B RECEIVER READP 004 535X USR.R EQU 00000010B RECEIVER READP	14	014 514X UMI.LB 60U 000001100B 8 BIT CHARACTERS 001 515X UMI.LK 60U 00000010B CLOCK X 16 002 517X UMI.64X 60U 00000010B CLOCK X 16 003 518X COMMAND INSTRUCTION BITS. 519X ** COMMAND INSTRUCTION BITS. 519X ** COMMAND INSTRUCTION BITS. 522X UCI.RB 60U 00100000B READER—ON CONTROL 522X UCI.RB 60U 00100000B READER—ON CONTROL 522X UCI.RB 60U 0001000B READER—ON CONTROL 523X UCI.RB 60U 000000100B READER INTERRUPTS 000 522X UCI.RB 60U 00010000B READER INTERRUPTS 001 525X UCI.RB 60U 000000100B READER INTERRUPTS 002 525X UCI.RB 60U 000000100B READER INTERRUPTS 003 525X UCI.RB 60U 000000100B READER INTERRUPTS 004 525X UCI.RB 60U 000000100B READER INTERRUPTS 005 526X WEITE 60U 00100000B FRANING ERROR 529X WINSTER EQU 00000100B RECEIVER READY 0010 533X USR.RK 60U 000000100B RECEIVER READY 0010 533X USR.RK 60U 000000010B RECEIVER READY 0010 533X USR.RK 60U 000000010B RECEIVER READY 0010 533X USR.RK 60U 0000000100B RECEIVER READY 0010 0010 0010 0010 0010 0010 0010 00	014 514X UNI.LB EQU 00000108 GLOCK X 1 515X UNI.1X EQU 000000108 CLOCK X 1 515X UNI.64X EQU 000000108 CLOCK X 64 515X UNI.64X EQU 000000108 CLOCK X 64 519X ** COMMAND INSTRUCTION BITS. 519X ** COMMAND INSTRUCTION BITS. 52X UCI.RE EQU 00100008 RECEIVE ENABLE OCCUPROL 524X UCI.RE EQU 000001098 RECEIVE ENABLE OCCUPROL 525X UCI.RE EQU 000000108 RECEIVE ENABLE OCCUPROL 525X UCI.RE EQU 000000108 FRANING ERROR 527X UCI.RE EQU 0000000108 FRANING ERROR 527X USR.FE EQU 000100008 PARITY ERROR 000000008 PARITY ERROR 00000000 BARITY ERROR 00000000 BARITY ERROR 000000000 BARITY ERROR 0000000000 BARITY ERROR 0000000000 BARITY ERROR 000000000000000000000000000000000	014 514X UMI8 EQU 00001100B 8 BIT CHARACTERS 001 515X UMI.1X EQU 00000010B CLOCK X 1 002 516X UMI.16 EQU 0000010B CLOCK X 1 003 516X EQU 0000010B CLOCK X 04 010 520X UCIR EQU 01000000B RECEIVE ENABLE 020 523X UCIR EQU 00100000B RECEIVE ENABLE 020 523X UCIR EQU 00100000B RECEIVE ENABLE 020 523X UCIR EQU 00100000B RECEIVE ENABLE 020 524X UCIR EQU 00000100B RECEIVE ENABLE 020 525X UCIR EQU 00000010B TRANSMIT ENABLE 031X USRPE EQU 00010000B PARITY ERROR 032 534X USRTXR EQU 00010000B TRANSMITER READY 040 534X USRTXR EQU 000000010B TRANSMITER READY 041 535X USRTXR EQU 000000010B TRANSMITTER READY 042 534X USRTXR EQU 000000010B TRANSMITTER READY	014 514X UHI.18 EQU 000001008 6 BIT CHARACTERS 001 515X UHI.1X EQU 000000108 CLOCK X 1 515X UHI.1X EQU 000000108 CLOCK X 1 515X UHI.1X EQU 000000108 CLOCK X 04 518X 519X 4** COMMAND INSTRUCTION BITS. 520X UCI.RE EQU 001000008 READER-ON CONTROL 0040 522X UCI.RE EQU 001000008 RECEIVE ENABLE EQU 000001008 RECEIVE ENABLE EQU 000001008 RECEIVE ENABLE EQU 000001008 RECEIVE ENABLE EQU 0000001008 RECEIVE ENABLE EQU 000001008 READITY ERROR 0010 522X US.R.FE EQU 0000010008 PARITY ERROR 0010 532X US.R.FE EQU 000010008 PARITY ERROR 0010 532X US.R.FE EQU 000010008 RECEIVER READITY 0010 533X US.R.TXE EQU 000001008 RECEIVER READITY 0010 533X US.R.TXE EQU 000001008 TRANSMITTER READITY 0010 0010000018 TRANSMITTER READITY 0010 00100000018 TRANSMITTER READITY 0010000000000000000000000000000000000	014 514X UMI.LB EQU 00001100B 8 BIT CHARACTERS 001 515X UMI.LX EQU 00000010B CLOCK X 1 002 516X UMI.LAK EQU 00000010B CLOCK X 64 518X EQU 00000010B CLOCK X 64 519X *** COMMAND INSTRUCTION BITS. 510X UCI.RE EQU 0010000B REDEE INTERRUPTS 0040 522X UCI.RE EQU 0010000B REDEE INTERRUPTS 0040 522X UCI.RE EQU 0000010B REDEE INTERRUPTS 0040 523X UCI.RE EQU 0000010B RECEIVE EMBLE 524X UCI.RE EQU 0000010B RECEIVE EMBLE 525X UCI.RE EQU 0000010B RECEIVE EMBLE 527X UCI.RE EQU 00000010B RECEIVE EMBLE 527X UCI.RE EQU 00000010B RECEIVE EMBLE 527X UCI.RE EQU 00000010B RECEIVE EMBLE 527X UCI.RE EQU 000000010B RECEIVE REROR 527X UCI.RE EQU 000000010B TRANSHITER EMPTY 527X USR.RE EQU 000000000B RARING ERROR 527X USR.RE EQU 000000000B RECEIVER READY 0040 533X USR.RE EQU 000000000B RECEIVER READY 0070 533X USR.RE EQU 0000000010B RECEIVER READY 0071 533X USR.RE EQU 0000000010B RECEIVER READY 0071 533X USR.RE EQU 0000000010B RECEIVER READY	
100   515X UHI-IX EQU   00000018   CLOCK X I L	100   515X UHI-IX EQU   00000018   CLOCK X I L	100   515X UHI-IX EQU   00000018   CLOCK X IS	1002   515X UMI.16X EQU   00000010B   CLOCK X 16	1002   515X UMI.1X EQU   00000010B   CLOCK X 16	515X UMI.15X EQU	0.01   0.00	515X UMI.16X EQU   000000018   CLOCK X 16	
.002 516X UMI.16X EQU 00000010B CLOCK X 16 .003 517X UMI.64X EQU 00000011B CLOCK X 64 .100 520X COMMAND INSTRUCTION BITS040 522X UCI.RE EQU 00010000B READER—OF ERROR RESET 0004 524X UCI.RE EQU 000001000B RECEIVE ENABLE END 525X UCI.RE EQU 000001000B RECEIVE ENABLE EQU 000001000B RECEIVE ENABLE EQU 000000100B RECEIVE ENABLE EQU 000000100B RECEIVE ENABLE EQU 000000100B RECEIVE ENABLE SZ6X UCI.RE EQU 000000100B RECEIVE ENABLE EQU 00000010B RECEIVE ENABLE EQU 00000010B RECEIVE ENABLE SZ6X UCI.RE EQU 00000010B FRANSHIT ENABLE SZ6X UCI.RE EQU 00000001B FRANSHIT ERROR 0000 0000 000 FRANSHIT ERROR 00000000 B FRANSHIT ERROR 000000010B RECEIVER READY 000000010B RECEIVER READY 0001 00000010B RECEIVER READY 0001 00000010B RECEIVER READY 0001 00000010B RECEIVER READY 0001 00000001B RECEIVER READY 0001 000000001B RECEIVER READY 0001 0000000001B RECEIVER READY 0001 0000000001B RECEIVER READY 0001 00000000001B RECEIVER READY 0001 00000000000000000000000000000000	.002 516X UMI.16X EQU 00000010B CLOCK X 16 .003 517X UMI.64X EQU 00000011B CLOCK X 64 .100 520X COMMAND INSTRUCTION BITS040 522X UCI.RE EQU 00000000B READER—OF ERON ESST COOL 523X UCI.RE EQU 000001000B RECEIVE ENABLE EQU 000001000B RECEIVE ENABLE EQU 000001000B RECEIVE ENABLE EQU 000001000B RECEIVE ENABLE EQU 000000100B RECEIVE ENABLE EQU 000000100B RECEIVE ENABLE EQU 000000100B RECEIVE ENABLE EQU 000000100B RECEIVE ENABLE EQU 00000010B RECEIVE ENABLE SZ5X UCI.TE EQU 00000010B TRANSHIT ENABLE SZ5X UCI.TE EQU 00000001B TRANSHIT ENABLE SZ2X USR.FE EQU 00010000B FRAMING ERROR 0000 00000100B FRAMING ERROR 000000010B RECEIVER READY 000000010B RECEIVER READY 000000010B RECEIVER READY 000000010B RECEIVER READY 000000001B TRANSHITTER READY 000100001B TRANSHITTER READY 0001000001B TRANSHITTER READY 000100001B TRANSHITTER READY 0001000001B TRANSHITTER READY 00010000001B TRANSHITTER READY 00010000001B TRANSHITTER READY 000100000001B TRANSHITTER READY 000100000001B TRANSHITTER READY 0001000000000000000000000000000000000	.002 516X UMI.16X EQU 00000010B CLOCK X 16 .003 517X UMI.64X EQU 00000011B CLOCK X 64 .100 520X COMMAND INSTRUCTION BITS040 522X UCI.RE EQU 00000000B READER—OF CONTROL COOLOGOOOB READER—OF SZAX UCI.RE EQU 000001000B RECEIVE ENABLE COOLOGOOOB READER—INTERRUPTS COOLOGOOOD READER—INTERRUPTS COOLOGOOOD READER—INTERRUPTS COOLOGOOOD READER—INTERRUPTS COOLOGOOOD READER—INTERRUPTS COOLOGOOOD READER—INTERRUPTY COOLOGOOOD RECEIVER READY COOLOGOOD RECEIVER	.002 516X UMI.16X EQU 00000010B CLOCK X 16 .003 517X UMI.64X EQU 00000011B CLOCK X 64 .100 520X COMMAND INSTRUCTION BITS040 521X UCI.RE EQU 00100000B READER—OF CONTROL004 522X UCI.RE EQU 000001000B RECEIVE ENABLE COMMAND BITS005 525X UCI.RE EQU 00000100B RECEIVE ENABLE COMMAND BITS006 525X UCI.RE EQU 000000100B RECEIVE ENABLE COMMAND BITS007 525X UCI.RE EQU 00000010B RECEIVE ENABLE COMMAND BITS008 525X UCI.RE EQU 00000001B TRANSHIT ENABLE SIZY USR.FE EQU 00010000B FRAMING ERROR OOVER COMMAND BITS009 531X USR.FE EQU 0001000B FRAMING ERROR OOVER COMMAND BITS000 532X USR.FE EQU 000000100B FRAMING ERROR OOVER COMMAND BITS000 532X USR.FE EQU 000000100B FRAMING ERROR OOVER COMMAND BITS000 533X USR.FX EQU 00000010B FRAMINTER READY OOVER COMMAND BITS000 535X USR.FX EQU 00000010B FRAMINTER READY	.002 516X UMI.16X EQU 00000010B CLOCK X 16 .003 517X UMI.64X EQU 00000011B CLOCK X 64 .004 520X WCI.R EQU 0100000B READER RESET .005 521X UCI.R EQU 00010000B READER RESET .006 522X UCI.R EQU 0001000B RECEIVE ENABLE .007 525X UCI.R EQU 0000000B RECEIVE ENABLE .008 524X UCI.R EQU 0000000B RECEIVE ENABLE .009 525X UCI.R EQU 0000000B RECEIVE ENABLE .009 525X UCI.R EQU 0000000B FRANING EROR .000 530X USR.FE EQU 00010000B FRANING EROR .000 531X USR.PE EQU 00010000B FRANING EROR .000 532X USR.R EQU 00001000B RECEIVER REDY .000 5334X USR.R EQU 000000100B RECEIVER REDY .000 535X USR.R EQU 0000000100B RECEIVER REDY	100   100	002 516X UNI.164 EQU 00000018 CLOCK X 16  518X UNI.64X EQU 00000018 CLOCK X 64  519X ** COMMAND INSTRUCTION BITS.  520X UCI.RE EQU 001000008 READER-ON CONTROL  521X UCI.RE EQU 00100008 READER-ON CONTROL  522X UCI.RE EQU 000000108 READER ENTERRUPTS  524X UCI.RE EQU 000000108 READER ENTERRUPTS  524X UCI.RE EQU 000000108 READER ENTERRUPTS  525X UCI.RE EQU 000000108 READER ENTERRUPTS  526X UCI.RE EQU 000000018 TRANSMIT ENABLE  526X UCI.RE EQU 000000018 TRANSMIT ENABLE  520X 520X 68.FE EQU 000000008 FRAMING ERROR  520X 531X USR.FE EQU 000010008 PARITY ERROR  532X USR.FE EQU 000001001 PARITY ERROR  533X USR.FE EQU 000001001 PARITY ERROR  534X USR.FE EQU 000001001 PARITY ERROR  535X USR.FE EQU 000000101 PARITY ERROR  535X USR.FE EQU 00000001 PARITY ERROR  535X USR.FE EQU 00000001 PARITY ERROR  535X USR	100   100	
100	517X UMI.64X EQU	100	5.17	517X UMI.64X EQU	1003   517X UMI.64X EQU   000000118   CLOCK X 64	517X UNI.64X EQU   000000118   CLOCK X 64	517X UMI.64X EQU	
518X 519X	518X 519X	518X 519X	518X 519X	518X 519X	518X   518X   520X   520X   520X   521X   CI.R E EQU	518X   518X   520X   520X   521X   CLI.R   600	518X ** COMMAND INSTRUCTION BITS. 520X UCIR EQU 0100000B READER—ON CONTROL 522X UCIR EQU 00100000B REGETY RESET 524X UCIR EQU 00000100B REGETY E ENABLE 524X UCIR EQU 00000100B REGETY E ENABLE 525X UCIT E EQU 00000010B TRANSMIT ENABLE 525X UCIT E EQU 00000010B TRANSMIT ENABLE 525X USRF EQU 00010000B PARITY ERROR 531X USRP EQU 00001000B PARITY ERROR 533X USRT E EQU 00000100B TRANSMITTER READY 535X USRT E EQU 00000100B TRANSMITTER READY 535X USRT E EQU 00000100B TRANSMITTER READY 535X USRT E EQU 00000100B TRANSMITTER READY	
520X 521X UCI.*IR EQU 001000008 READER—UCONTROL 522X UCI.*RE EQU 000100008 RECEIVE ENABLE 524X UCI.*RE EQU 000001008 RECEIVE ENABLE 525X UCI.*IE EQU 000000108 RECEIVE ENABLE 525X UCI.*IE EQU 000000108 TRANSHIT ENABLE 526X UCI.*IE EQU 000000108 TRANSHIT ENABLE 527X 528X ** STATUS READ COMMAND BITS. 528X ** STATUS READ COMMAND BITS. 528X ** STATUS READ COMMAND BITS. 538X USR.*FE EQU 000100008 FRAMING ERROR 531X USR.*FE EQU 000010008 TRANSHITY ERROR 534X USR.*TX EQU 000000108 RECEIVER READY 535X USR.*TX EQU 000000108 RECEIVER READY	520X 521X UCI.*IR EQU 01000008 INTERNAL RESET 522X UCI.*R EQU 000100008 RECEIVE ENABLE 524X UCI.*R EQU 000100008 RECEIVE ENABLE 524X UCI.*IE EQU 000000108 RECEIVE ENABLE 525X UCI.*IE EQU 000000108 RECEIVE ENABLE 526X UCI.*IE EQU 000000108 TRANSHIT ENABLE 527X 528X ** STATUS READ COMMAND BITS. 528X ** STATUS READ COMMAND BITS. 528X ** STATUS READ COMMAND BITS. 530X USR.*FE EQU 000100008 PARITY ERROR 531X USR.*TE EQU 0000100B TRANSHITER READY 535X USR.*TE EQU 00000010B TRANSHITER READY 535X USR.*TE EQU 00000010B RECEIVER READY	521X UCI.IR EQU 01000008 INTERNAL RESET 521X UCI.RD EQU 00100008 READER—UC CONTROL 523X UCI.RE EQU 000001008 RECEIVE ENABLE 524X UCI.RE EQU 000001008 RECEIVE ENABLE 525X UCI.IE EQU 000000108 RECEIVE ENABLE 525X UCI.IE EQU 000000108 TRANSHIT ENABLE 525X UCI.IE EQU 000000108 TRANSHIT ENABLE 527X ** STATUS READ COMMAND BITS. 528X ** STATUS READ COMMAND BITS. 530X USR.FE EQU 001000008 FAMING ERROR 531X USR.FE EQU 000010008 FAMITY ERROR 531X USR.FE EQU 000010008 TRANSHITER READY 535X USR.FX EQU 0000001008 TRANSHITER READY 535X USR.FX EQU 000000108 RECEIVER READY 535X USR.FX EQU 000000108 RECEIVER READY	521X UCI.IR EQU 01000008 INTERNAL RESET 521X UCI.RD EQU 001000008 READER—UCONTROL 523X UCI.RE EQU 0000010008 RECEIVE ENABLE 524X UCI.RE EQU 000001008 RECEIVE ENABLE 525X UCI.IE EQU 000000108 RECEIVE ENABLE 525X UCI.IE EQU 000000108 TRANSHIT ENABLE 525X UCI.IE EQU 000000108 TRANSHIT ENABLE 525X ** STATUS READ COMMAND BITS. 520X ** STATUS READ COMMAND BITS. 530X USR.FE EQU 00100000 OVERRUN ERROR 530X USR.FE EQU 000010008 FAMITY ERROR 530X USR.FE EQU 000010008 TRANSHITER READY 535X USR.FX EQU 0000001008 TRANSHITER READY 535X USR.FX EQU 000000108 RECEIVER READY 535X USR.FX EQU 000000108 TRANSHITER READY	521X UCI.IR EQU 01000008 INTERNAL RESET 521X UCI.RD EQU 001000008 READER—UCONTROL 523X UCI.RE EQU 000100008 RECEIVE ENABLE 524X UCI.RE EQU 000001008 RECEIVE ENABLE 525X UCI.TE EQU 000000108 TRANSHIT ENABLE 525X UCI.TE EQU 000000108 TRANSHIT ENABLE 525X UCI.TE EQU 000000108 TRANSHIT ENABLE 525X UCI.TE EQU 000000008 TRANSHIT ENABLE 525X UCI.TE EQU 000100008 FARING ERROR 530X USR.FE EQU 000010008 PARITY ERROR 530X USR.TX EQU 000010008 TRANSHITER READY 533X USR.TX EQU 000001008 TRANSHITER READY 535X USR.TX EQU 000000108 TRANSHITER READY 535X USR.TX EQU 000000108 TRANSHITER READY 535X USR.TX EQU 000000108 TRANSHITER READY	521X UCI. IR EQU 0100000B INTERNAL RESET 522X UCI. RE QU 0010000B ERROR RESET 524X UCI. RE QU 00010000B RECEIVE ENABLE 524X UCI. RE QU 00001000B RECEIVE ENABLE 524X UCI. IE EQU 00000100B RECEIVE ENABLE 525X UCI. IE EQU 00000010B TRANSNIT ENABLE 527X USR. FE EQU 000010000B TRANSNIT ENABLE 529X USR. PE EQU 00010000B PARITY ERROR 531X USR. PE EQU 000010000B TRANSNITER EMPTY 532X USR. TXR EQU 00001000B TRANSNITER READY 535X USR. TXR EQU 00000100B TRANSNITER READY 535X USR. TXR EQU 00000010B TRANSNITER READY 535X USR. TXR EQU 00000010B TRANSNITER READY 535X USR. TXR EQU 00000010B TRANSNITER READY	521x UC1.IR EQU 0100000B INTERNAL RESET 521x UC1.ER EQU 00100000B READER-ON CONTROL 522x UC1.ER EQU 00010000B RECEIVE ENABLE 524x UC1.RE EQU 00000100B RECEIVE ENABLE 525x UC1.IE EQU 00000010B RECEIVE ENABLE 527x UC1.TE EQU 00000010B RECEIVE ENABLE 527x UC1.TE EQU 00000001B TRANSHIT ENABLE 527x S27x S77x S77x S77x S77x S77x S77x S	521X UCIIR EQU 01000008 INTERNAL RESET 522X UCIRE EQU 001000008 ERADER—ON CONTROL 523X UCIRE EQU 000100008 RECEIVE ENABLE 524X UCIRE EQU 000001008 RECEIVE ENABLE 525X UCIIE EQU 000001008 RECEIVE ENABLE 525X UCIIE EQU 000000108 TRANSHIT ENABLE 528X ++ STATUS READ COMMAND BITS. 528X ++ STATUS READ COMMAND BITS. 529X USRFE EQU 00100008 PARITY EROR 531X USRFE EQU 000100008 PARITY EROR 531X USRFE EQU 000010008 TRANSHITTER EMPTY 534X USRTXR EQU 000010008 TRANSHITTER EMPTY 535X USRTXR EQU 000001008 TRANSHITTER READY 535X USRTXR EQU 000000108 TRANSHITTER READY 535X USRTXR EQU 0000000108 TRANSHITTER READY	
523X UCI.ER EQU 000100008 READER—ON CONTROL 523X UCI.ER EQU 000100008 ERROR RESET 524X UCI.RE EQU 00000100B ERROR RESET 525X UCI.TE EQU 00000010B TRANSMIT ENABLE 527X UCI.TE EQU 00000010B TRANSMIT ENABLE 527X S28X ** STATUS READ COMMAND BITS. 529X S30X USR.FE EQU 00010000B FRAMING ERROR 531X USR.PE EQU 00010000B PARITY ERROR 533X USR.PE EQU 000010000B TRANSMITTER EMPTY 534X USR.RE EQU 00001000B TRANSMITTER EMPTY 535X USR.RX EQU 00000010B TRANSMITTER READY	523X UCI.ER EQU 000100008 READER—ON CONTROL 523X UCI.ER EQU 000100008 ERROR RESET 524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.TE EQU 00000010B TRANSMIT ENABLE 526X WE STATUS READ COMMAND BITS. 528X ** STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00010000B FRAMING ERROR 531X USR.PE EQU 00010000B PARITY ERROR 533X USR.TXE EQU 00001000B TRANSMITTER EMPTY 534X USR.RX EQU 00001000B TRANSMITTER READY 535X USR.RX EQU 00000010B TRANSMITTER READY	523X UCI. RG EQU 000100008 READER—ON CONTROL 523X UCI. ER EQU 000100008 ERROR RESET 524X UCI. RE EQU 00001008 ERROR RESET 524X UCI. RE EQU 00000108 ENABLE 1NTERRUPTS 526X UCI. TE EQU 00000018 TRANSMIT ENABLE 527X USR. FE EQU 000000018 FRAMING ERROR 530X USR. FE EQU 000100008 FRAMING ERROR 531X USR. PE EQU 000010008 PARITY ERROR 534X USR. PE EQU 000010008 FRAMING ERROR 534X USR. RE EQU 000010008 FRAMITY ERROR 534X USR. TXE EQU 000010008 TRANSMITTER EMPTY 534X USR. TXE EQU 00000108 TRANSMITTER READY 535X USR. TXR EQU 000000108 TRANSMITTER READY 535X USR. TXR EQU 000000108 TRANSMITTER READY	523X UCI. RG EQU 00100008 READER—ON CONTROL 523X UCI. ER EQU 000100008 ERROR RESET 524X UCI. RE EQU 00001008 ERROR RESET 524X UCI. RE EQU 00000108 ENABLE 1NTERRUPTS 526X UCI. TE EQU 00000018 TAANSHIT ENABLE 528X ** STATUS READ COMMAND BITS. 529X S30X USR. FE EQU 000100008 FRAMING ERROR 531X USR. PE EQU 000100008 PARITY ERROR 532X USR. TXE EQU 000010008 TRANSMITTER EMPTY 534X USR. TXE EQU 00001008 TRANSMITTER EMPTY 534X USR. TXE EQU 000001008 TRANSMITTER READY 535X USR. TXR EQU 000000108 TRANSMITTER READY 535X USR. TXR EQU 000000108 TRANSMITTER READY	522X UCI. RO 600 000000 READER—ON CONTROL 523X UCI. ER 690 00010000 BEROR RESET 524X UCI. RE 690 00010000 BEROR RESET 524X UCI. RE 690 00000100 BEROR RESET 526X UCI. TE 690 00000010 TRANSHIT ENABLE 520X UCI. TE 690 00000010 TRANSHIT ENABLE 520X USR. FE 690 000100000 GOVERNOR 531X USR. PE 690 000100000 GOVERNOR 532X USR. PE 690 000100000 FARITY ERROR 534X USR. TXE 690 000010000 TRANSHITTER EMPTY 534X USR. TXR 690 000001000 TRANSHITTER READY 535X USR. TXR 690 000001000 TRANSHITTER READY 535X USR. TXR 690 000000100 TRANSHITTER READY	522X UCI. RO EQU 000100008 READER—ON CONTROL 523X UCI. ER EQU 000100008 ERROR RESET 524X UCI. RE EQU 000001008 ERROR RESET 524X UCI. RE EQU 000001008 ENABLE INTERRUPTS 526X UCI. TE EQU 000000108 TRANSHIT ENABLE 528X UCI. TE EQU 000000108 TRANSHIT ENABLE 528X UCI. TE EQU 000100008 FRAHING ERROR 531X USR. PE EQU 000100008 PARITY ERROR 533X USR. PE EQU 000010008 TRANSHITTER EMPTY 534X USR. TXR EQU 000010008 TRANSHITTER READY 535X USR. TXR EQU 0000000108 TRANSHITTER READY 535X USR. TXR EQU 0000000108 TRANSHITTER READY	522 X UCI. RG EQU 00100008 READER—ON CONTROL 523 X UCI. RE EQU 00010008 RECEIVE RESET 000001008 RECEIVE RABLE S25 X UCI. RE EQU 000000108 RECEIVE RABLE INTERRUPTS 526 X UCI. TE EQU 000000108 TAANSHIT ENABLE 528 X X STATUS READ COMMAND BITS. S30 X USR. FE EQU 000100008 FRANING ERROR 531 X USR. PE EQU 000100008 FRANING ERROR 532 X USR. RE EQU 000010008 FRANING ERROR 534 X USR. RE EQU 000010008 RECEIVER READ F S35 X USR. RT E EQU 000001008 RECEIVER READ F S35 X USR. RT E EQU 000001008 RECEIVER READ F S35 X USR. RT E EQU 000001008 RECEIVER READ F S35 X USR. RT E EQU 000000108 RECEIVER READ F S35 X USR. RT E EQU 000000108 RECEIVER READ F S35 X USR. RT E EQU 000000108 RECEIVER READ F S35 X USR. RT E EQU 000000108 RECEIVER READ F S35 X USR. RT E EQU 000000108 RECEIVER READ F S35 X USR. RT E EQU 000000018 RESENTING READ F S35 X USR. RT E EQU 000000018 RESENTING READ F S35 X USR. RT E EQU 0000000108 RECEIVER READ F S35 X USR. RT E EQU 0000000108 RESENTING READ F S35 X USR. RT E EQU 0000000108 RESENTING READ F S35 X USR. RT E EQU 0000000108 RESENTING READ F S35 X USR. RT E EQU 00000000108 RESENTING READ F S35 X USR. RT E EQU 00000000108 RESENTING READ F S35 X USR. RT E EQU 00000000108 RESENTING READ F S35 X USR. RT E EQU 00000000108 RESENTING READ F S35 X USR. RT E EQU 00000000108 RESENTING READ F S35 X USR. RT E EQU 000000000108 RESENTING READ F S35 X USR. RT E EQU 00000000000 RESENTING READ F S35 X USR. RT E EQU 0000000000 RESENTING READ F S35 X USR. RT E EQU 00000000000 RESENTING READ F S35 X USR. RT E EQU 000000000000 RESENTING READ F S35 X USR. RT E EQU 00000000000 RESENTING READ F S35 X USR. RT E EQU 000000000000000000000000000000000	523X UCI.ER EQU 000100008 READER—ON CONTROL 523X UCI.ER EQU 00001008 ERROR RESET 524X UCI.RE EQU 000001008 ENABLE 525X UCI.TE EQU 000000108 TRANSMIT ENABLE 527X UCI.TE EQU 000000108 TRANSMIT ENABLE 527X S28X ** STATUS READ COMMAND BITS. 529X S30X USR.FE EQU 000100008 FARITY ERROR 530X USR.PE EQU 000010008 FARITY ERROR 533X USR.TXE EQU 0000001008 TRANSMITTER READY 533X USR.TXE EQU 000000108 TRANSMITTER READY 535X USR.TXR EQU 000000108 TRANSMITTER READY 535X USR.TXR EQU 000000108 TRANSMITTER READY	
5.23 V.U.T.E.R. EQU. 00001008 ERROR RESET 5.24 V.U.T.R.E. EQU. 000001008 RECEIVE ENABLE 5.24 V.U.T.R.E. EQU. 000000108 ENABLE INTERRUPTS 5.24 V.U.T.T.E. EQU. 000000108 TRANSMIT ENABLE 5.27 V.T.R.E. EQU. 0000100008 FRAMING ERROR 5.30 V.U.S.R.P.E. EQU. 0000100008 PARITY ERROR 5.32 V.S.R.T.K.E. EQU. 0000100008 TRANSMITTER EMPTY 5.34 V.S.R.T.K.E. EQU. 0000100008 TRANSMITTER EMPTY 5.34 V.S.R.T.K.E. EQU. 000000108 TRANSMITTER READY 5.35 V.S.R.T.K.E. EQU. 000000108 TRANSMITTER READY	5.23 X UCI.ER EQU 00001008 ERROR RESET 5.24 X UCI.RE EQU 000001008 RECEIVE ENABLE 5.24 X UCI.RE EQU 000001008 ENABLE INTERRUPTS 5.26 X UCI.TE EQU 000000108 TRANSMIT ENABLE 5.27 X STATUS READ COMMAND BITS. 5.29 X STATUS READ COMMAND BITS. 5.30 X USR.FE EQU 000100008 FRAMING ERROR 5.31 X USR.FE EQU 000100008 PARITY ERROR 5.32 X USR.TE EQU 0000100008 TRANSMITTER EMPTY 5.34 X USR.RE EQU 000000108 TRANSMITTER EMPTY 5.35 X USR.RE EQU 000000108 TRANSMITTER READY 5.35 X USR.TX EQU 000000108 TRANSMITTER READY 5.35 X USR.TX EQU 000000108 TRANSMITTER READY	523X UCI.ER EQU 0000100B ERROR RESET 524X UCI.RE EQU 00000100B RECEIVE ENABLE 524X UCI.RE EQU 00000100B RECEIVE ENABLE 526X UCI.TE EQU 00000010B TRANSMIT ENABLE 527X UCI.RE EQU 00000001B TRANSMIT ENABLE 527X USR.FE EQU 00010000B FRAMING ERROR 530X USR.PE EQU 00010000B PARITY ERROR 530X USR.PE EQU 000010000B PARITY ERROR 530X USR.TXE EQU 00001000B TRANSMITTER EMPTY 534X USR.TXR EQU 00000010B RECEIVER READY 535X USR.TXR EQU 00000010B RECEIVER READY 535X USR.TXR EQU 00000010B TRANSMITTER READY	523X UCI.ER EQU 0000100B ERROR RESET 524X UCI.RE EQU 00000100B RECEIVE ENABLE 524X UCI.RE EQU 00000010B RECEIVE ENABLE 526X UCI.TE EQU 00000010B TRANSMIT ENABLE 527X UCI.RE EQU 00000001B TRANSMIT ENABLE 527X USR.FE EQU 00010000B FRAMING ERROR 530X USR.FE EQU 00010000B PARITY ERROR 534X USR.TXE EQU 00001000B TRANSMITTER EMPTY 534X USR.RX EQU 00001000B TRANSMITTER EMPTY 534X USR.RX EQU 00000010B RECEIVER READY 535X USR.TXR EQU 00000010B RECEIVER READY 535X USR.TXR EQU 00000010B TRANSMITTER READY	523X UCI.ER EQU 0000100B ERROR RESET 524X UCI.RE EQU 00000100B RECEIVE ENABLE 524X UCI.RE EQU 00000100B ENABLE INTERRUPTS 526X UCI.TE EQU 00000010B TRANSMIT ENABLE 527X USR.FE EQU 00010000B FRAMING ERROR 530X USR.FE EQU 00010000B PARITY ERROR 532X USR.TXE EQU 00001000B TRANSMITTER EMPTY 534X USR.TXE EQU 00001000B TRANSMITTER EMPTY 534X USR.TXE EQU 00000100B TRANSMITTER EMPTY 534X USR.TXR EQU 00000100B TRANSMITTER READY 535X USR.TXR EQU 00000100B TRANSMITTER READY	523X UCI.ER EQU 0000100B ERROR RESET 524X UCI.RE EQU 00000100B RECEIVE ENABLE 524X UCI.RE EQU 00000100B ENABLE INTERRUPTS 526X UCI.TE EQU 00000001B TRANSMIT ENABLE 527X UCI.RE EQU 00000000B TRANSMIT ENABLE 527X USR.PE EQU 00010000B DAVERUN ERROR 531X USR.PE EQU 00010000B PARITY ERROR 534X USR.PE EQU 00001000B TRANSMITTER EMPTY 534X USR.TXR EQU 00001000B TRANSMITTER READY 535X USR.TXR EQU 00000100B TRANSMITTER READY 535X USR.TXR EQU 00000001B TRANSMITTER READY	523X UCI.ER EQU 0000100B ERROR RESET 524X UCI.RE EQU 00000100B RECEIVE ENABLE 524X UCI.RE EQU 00000010B RECEIVE ENABLE 524X UCI.TE EQU 00000010B TRANSMIT ENABLE 527X UCI.TE EQU 00000001B TRANSMIT ENABLE 527X USR.PE EQU 00010000B GVERUN ERROR 531X USR.PE EQU 00010000B PARITY ERROR 532X USR.PE EQU 00001000B TRANSMITTER EMPTY 534X USR.RX EQU 00001000B TRANSMITTER READY 535X USR.TX EQU 00000100B TRANSMITTER READY 535X USR.TX EQU 00000010B TRANSMITTER READY 535X USR.TX EQU 00000010B TRANSMITTER READY	523X UCI.ER EQU 0000100B ERROR RESET 524X UCI.RE EQU 00000100B RECEIVE ENABLE 524X UCI.RE EQU 00000010B RECEIVE ENABLE 525X UCI.TE EQU 00000010B TRANSHIT ENABLE 527X 528X ** STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 0001000B FARITY ERROR 530X USR.PE EQU 00001000B FARITY ERROR 530X USR.RX EQU 00000100B TRANSHITER READY 534X USR.RX EQU 00000010B TRANSHITER READY 535X USR.TX EQU 00000010B TRANSHITER READY 535X USR.TX EQU 00000010B TRANSHITER READY	
524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 00000010B TRANSHIT ENABLE 527X US. 528X ** STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00100000B FAMING ERROR 531X USR.PE EQU 00010000B PARITY ERROR 532X USR.PX EQU 000010000B TRANSHITTER EMPTY 534X USR.RX EQU 00001000B TRANSHITTER READY 535X USR.RX EQU 00000010B RECEIVER READY 535X USR.TX EQU 00000010B RECEIVER READY	524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 00000010B TRANSHIT ENABLE 527X US.FE EQU 00000000B FRANING ERROR 530X USR.FE EQU 000010000B PARITY ERROR 531X USR.PE EQU 000010000B PARITY ERROR 532X USR.PE EQU 000010000B TRANSHITTER EMPTY 534X USR.RX EQU 00001000B TRANSHITTER EMPTY 534X USR.RX EQU 00000010B RECEIVER READY 535X USR.TX EQU 00000010B RECEIVER READY	524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 524X 527X 528X ++ STATUS READ COMMAND BITS. 529X 529X 530X USR.FE EQU 00010000B FRAMING ERROR 530X USR.PE EQU 00010000B PARITY ERROR 532X USR.PE EQU 000010000B FRAMING ERROR 532X USR.PE EQU 000010000B FRAMING ERROR 534X USR.TXE EQU 00001000B FRAMITY ERROR 534X USR.RX EQU 00000100B FREEIVER READY 535X USR.TX EQU 00000010B FREEIVER READY 535X USR.TX EQU 00000010B FREEIVER READY	524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 0000001B TRANSHIT ENABLE 527X USR.FE EQU 00100000B FRANING ERROR 530X USR.FE EQU 00010000B PARITY ERROR 531X USR.PE EQU 00001000B PARITY ERROR 534X USR.TXE EQU 00001000B TRANSHITTER EMPTY 534X USR.RX EQU 00000100B TRANSHITTER EMPTY 535X USR.RX EQU 0000010B RECEIVER READY 535X USR.TX EQU 00000010B RECEIVER READY	524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 0000001B TRANSHIT ENABLE 527X 528X ** STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 0010000B FRANING ERROR 531X USR.PE EQU 00011000B FARITY ERROR 533X USR.TXE EQU 0000100B TRANSHITTER EMPTY 534X USR.TXE EQU 00000100B TRANSHITTER EMPTY 535X USR.TXR EQU 0000010B RECEIVER READY 535X USR.TXR EQU 00000010B TRANSHITTER READY	524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 00000010B TRANSHIT ENABLE 527X UCI.TE EQU 00000001B TRANSHIT ENABLE 529X 530X USR.FE EQU 00010000B TRANING ERROR 531X USR.PE EQU 00010000B TRANSHITTER EMPTY 534X USR.TXE EQU 00001000B TRANSHITTER EMPTY 534X USR.TXR EQU 00000100B TRANSHITTER READY 535X USR.TXR EQU 00000010B TRANSHITTER READY 535X USR.TXR EQU 00000001B TRANSHITTER READY	524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.IE EQU 00000010B ENABLE INTERUPTS 526X UCI.TE EQU 0000001B TRANSHIT ENABLE 527X S28X ** STATUS READ COMMAND BITS. 529X 529X 0010000B FRANING ERROR 530X USR.PE EQU 00001000B PARITY ERROR 531X USR.PE EQU 00001000B TRANSHITTER EMPTY 534X USR.TXE EQU 00000100B TRANSHITTER READY 534X USR.TXR EQU 00000010B TRANSHITTER READY 535X USR.TXR EQU 00000001B TRANSHITTER READY	524X UCI.RE EQU 00000100B RECEIVE ENABLE 525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 00000010B TRANSHIT ENABLE 527X UCI.TE EQU 000000010B TRANSHIT ENABLE 529X 530X USR.FE EQU 00010000B TRANING ERROR 531X USR.PE EQU 00001000B PARITY ERROR 531X USR.TX EQU 00000100B TRANSHITTER READY 535X USR.TXR EQU 00000010B TRANSHITTER READY 535X USR.TXR EQU 00000010B TRANSHITTER READY 535X USR.TXR EQU 000000010B TRANSHITTER READY	
525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 0000001B TRANSHIT ENABLE 527X ** STATUS READ COMMAND BITS. 529X SAOX USR.FE EQU 00010000B FRAMING ERROR 530X USR.PE EQU 00001000B PARITY ERROR 531X USR.PIXE EQU 00001000B TRANSMITTER EMPTY 534X USR.RXR EQU 00000010B TRANSMITTER READY 535X USR.TXR EQU 00000010B TRANSMITTER READY	525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 0000001B TRANSHIT ENABLE 527X ** STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00010000B FRANING ERROR 531X USR.DE EQU 00001000B PARITY ERROR 532X USR.TXE EQU 00001000B TRANSHITER EMPTY 534X USR.TXE EQU 00000010B TRANSHITER READY 535X USR.TXR EQU 00000010B TRANSHITER READY	525X UCI.IE EQU 00000010B ENABLE INTERRUPTS 526X UCI.TE EQU 0000001B TRANSHIT ENABLE 527X ** STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00010000B DVERRUP ERROR 531X USR.PE EQU 00001000B PARITY ERROR 532X USR.TXE EQU 00001000B TRANSMITTER EMPTY 534X USR.TXR EQU 00000010B TRANSMITTER READY 535X USR.TXR EQU 00000010B TRANSMITTER READY	525X UCI.IE EQU 00000010B FRANSHIT ENABLE 526X UCI.TE EQU 00000001B TRANSHIT ENABLE 527X 528X ++ STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00010000B FRANING ERROR 531X USR.PE EQU 00010000B PARITY ERROR 532X USR.TXE EQU 00001000B TRANSHITTER EMPTY 534X USR.RXR EQU 00000010B TRANSHITTER EADY 535X USR.TXR EQU 00000010B TRANSHITTER READY	525X UCI.IE EQU 00000010B FRANSHIT ENABLE 526X UCI.TE EQU 00000001B TRANSHIT ENABLE 527X 528X ++ STATUS READ COMMAND BITS. 530X USR.FE EQU 00010000B GVERRUR 531X USR.PE EQU 00010000B PARITY ERROR 532X USR.TXE EQU 00001000B TRANSHITTER EMPTY 534X USR.TXR EQU 00000010B TRANSHITTER EMPTY 535X USR.TXR EQU 00000010B TRANSHITTER READY	525X UCI.IE EQU 00000010B TRANSHIT ENABLE 526X 4+ STATUS READ COMMAND BITS. 528X 4+ STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00010000B FRANING ERROR 531X USR.PE EQU 00001000B PARITY ERROR 532X USR.TXE EQU 00001000B TRANSHITTER EMPTY 534X USR.TXR EQU 00000100B TRANSHITTER READY 535X USR.TXR EQU 00000010B TRANSHITTER READY	525X UCI.IE EQU 00000010B TRANSHIT ENABLE 526X 4+ STATUS READ COMMAND BITS. 528X 4+ STATUS READ COMMAND BITS. 528X 530X USR.FE EQU 00010000B FRAMING ERROR 531X USR.PE EQU 00001000B PARITY ERROR 532X USR.TXE EQU 00000100B TRANSMITTER EMPTY 534X USR.TXR EQU 00000010B TRANSMITTER READY 535X USR.TXR EQU 00000001B TRANSMITTER READY	525X UCI.IE EQU 00000010B TRANSHIT ENABLE 526X 4* STATUS READ COMMAND BITS. 528X 4* STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 0010000B FRAMING ERROR 531X USR.PE EQU 00001000B PARITY ERROR 531X USR.RXR EQU 00000100B TRANSMITTER READY 535X USR.TXR EQU 00000010B TRANSMITTER READY 535X USR.TXR EQU 00000001B TRANSMITTER READY	
526X UCI.TE EQU 0000001B TKANSMIT EN 527X ** STATUS READ COMMAND BITS. 529X USR.FE EQU 00100000B FRAMING ERR 530X USR.FE EQU 00001000B PARITY ERRO 532X USR.TXE EQU 00000100B PARITY ERRO 534X USR.TXE EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B TRANSMITTER	526X UCI.TE EQU 0000001B TKANSMIT EN 527X	526X UCI.TE EQU 0000001B TRANSMIT EN 52X ** STATUS READ COMMAND BITS. 529X ** STATUS READ COMMAND BITS. 529X ** STATUS READ COMMAND BITS. 520X USR.FE EQU 0010000B FAMING ERR 531X USR.FE EQU 00001000B PARITY ERR 534X USR.TXE EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B TRANSMITTER	526X UCI.TE EQU 0000001B TRANSMIT EN 52X	526X UCI.TE EQU 0000001B TRANSMITEN 527X ** STATUS READ COMMAND BITS. 529X ** STATUS READ COMMAND BITS. 530X USR.FE EQU 00010000B FRANING ERR 531X USR.PE EQU 00001000B PARITY ERR 534X USR.RXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B TRANSMITTER	526X UCI.TE EQU 0000001B TRANSMIT EN 528X ** STATUS READ COMMAND BITS. 529X S29X S29X S29X S29X S29X S29X S29X S	526X UCI.TE EQU 0000001B TRANSMIT EN 52X ** STATUS READ COMMAND BITS. 529X ** STATUS READ COMMAND BITS. 530X USR.FE EQU 0001000B PARITY ERR 532X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000010B TRANSMITTER 535X USR.TXR EQU 00000010B TRANSMITTER 535X USR.TXR EQU 00000001B TRANSMITTER	526X UCI.TE EQU 0000001B TRANSMIT EN 527X	
529X 530X USR.FE EQU 001000008 FRAMING ERR 531X USR.DE EQU 000100008 OVERRUN ERR 531X USR.TXE EQU 000010008 PARITY ERRO 534X USR.TXE EQU 0000001008 TRANSMITTER 535X USR.TXR EQU 000000108 RECEIVER RE	528X ++ STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00010000B	528X ++ STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00100000B FRAMING ERR 531X USR.0E EQU 00010000B PARITY ERR 532X USR.TXE EQU 00001000B TRANSMITTER 534X USR.RXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B TRANSMITTER	529X 529X 530X USR-FE EQU 00100000B FRAMING ERR 531X USR-0E EQU 00010000B OVERRU ERR 531X USR-FE EQU 00001000B PARITY ERRO 533X USR-TXE EQU 00000010B TRANSMITTER 534X USR-TXR EQU 00000010B TRANSMITTER	529X 529X 530X USR-FE EQU 00100008	529X 529X 530X USR-FE EQU 00100008 OVERRUN ERR 531X USR-PE EQU 000100008 PARITY ERR 532X USR-TXE EQU 000010008 TRANSMITTER 534X USR-TXR EQU 000000108 TRANSMITTER 535X USR-TXR EQU 000000108 TRANSMITTER	528X ** STATUS READ COMMAND BITS. 529X 530X USR.FE EQU 00010000B OVERRUN ERR 531X USR.PE EQU 0001000B PARITY ERRO 532X USR.TXE EQU 0000010B RECEIVER RE 534X USR.TXR EQU 0000001B TRANSMITTER 535X USR.TXR EQU 0000001B TRANSMITTER	529X 529X 530X USR-FE EQU 00010000B OVERNING ERR 531X USR-DE EQU 00001000B OVERRUN ERR 533X USR-TE EQU 00000100B PARITY ERRO 534X USR-RXR EQU 00000010B RECEIVER RE 535X USR-TXR EQU 00000010B TRANSMITTER	
529X 530X USR.FE EQU 000100008 FRAMING ERR 531X USR.QE EQU 000010008 PARITY ERRO 532X USR.TXE EQU 000001008 TRANSMITTER 534X USR.TXR EQU 000000108 RECEIVER RE 535X USR.TXR EQU 000000108 TRANSMITTER	529X 530X USR.FE EQU 000100008	529X 530X USR.FE EQU 000100008 FRAMING ERR 531X USR.DE EQU 000010008 PARITY ERR 532X USR.TXE EQU 000010008 TRANSMITTER 534X USR.RXR EQU 000000108 RECEIVER RE 535X USR.TXR EQU 000000108 TRANSMITTER	529X 530X 530X 530X 531X 531X 531X 531X 532X 532X 532X 532X 533X 533X 533X 533	529X 530X USR.FE EQU 000100008	529X 530X 530X 530X 531X 531X 531X 531X 531X 531X 532X 533X 533X 533X 533X 533X 533X 533	529X 530X 530X 530X 530X 531X 531X 531X 531X 531X 531X 531X 531	530X USR.FE EQU 00100006 FRAMING ERR 531X USR.DE EQU 00010006 PARITY ERRO 532X USR.TXE EQU 000001008 PARITY ERRO 534X USR.RXE EQU 000000108 RECEIVER RE 534X USR.TXR EQU 000000108 RECEIVER RE 535X USR.TXR EQU 000000118 TRANSMITTER	
530X USR.FE EQU 00100000B FRAMING ERR 531X USR.DE EQU 00010000B 0VERRUN ERRO 533X USR.PE EQU 00001000B PARTITY ERRO 534X USR.TXE EQU 000000100B RECEIVER RE 535X USR.TXR EQU 00000010B RECEIVER RE	530X USR.FE EQU 00100000B FRAMING ERR 531X USR.DE EQU 00010000B OVERRUN ERRO 533X USR.PE EQU 00001000B PARTITY ERRO 534X USR.TXE EQU 000000100B RECEIVER RE 534X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B TRANSMITTER	530X USR.FE EQU 00100006 FRANING ERR 531X USR.DE EQU 00010008 PARITY ERRO 533X USR.PE EQU 000010008 PARITY ERRO 533X USR.TXE EQU 000001008 RECEIVER RE 534X USR.TXR EQU 000000108 RECEIVER RE 535X USR.TXR EQU 000000108 TRANSMITTER	530X USR.FE EQU 00100006 FRAMING ERR 531X USR.DE EQU 00010008 PARTITY ERRO 533X USR.TXE EQU 000001008 PARTITY ERRO 534X USR.TXE EQU 000000108 RECEIVER RE 535X USR.TXR EQU 000000108 RECEIVER RE	530X USR.FE EQU 0010000B FRAMING ERR 531X USR.DE EQU 0001000B PARITY ERRO 533X USR.TXE EQU 0000100B PARITY ERRO 534X USR.TXE EQU 00000100B RECEIVER RE 534X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 0000001B TRANSMITTER	530X USR.FE EQU 0010000B FRAMING ERR 531X USR.DE EQU 0001000B DVERRUN ERR 533X USR.TXE EQU 0000100B TANSMITTER 535X USR.TXR EQU 00000010B RECEIVER RE. 535X USR.TXR EQU 00000010B TRANSMITTER	530X USR.FE EQU 0010000B FRAMING ERR 531X USR.DE EQU 00001000B PARITY ERG 533X USR.TXE EQU 00000100B PARITY ERG 534X USR.TXE EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000011B TRANSMITTER	530X USR.FE EQU 0010000B FRAMING ERR 531X USR.DE EQU 0001000B PARITY ERRO 533X USR.TXE EQU 00000100B FRANSMITTER 534X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000010B TRANSMITTER	
531X USR.OE EQU 00010000B OVERUN ERRI 532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000001B TRANSMITTER	531X USR.OE EQU 00010000B OVERUN ERR 532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.RXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000001B TRANSMITTER	531X USR.OE EQU 00010000B OVERUN ERR 532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TERNSMITTER 534X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000001B TRANSMITTER	531X USR.OE EQU 00010000B OVERUN ERRI 532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TERNSMITTER 534X USR.TXR EQU 00000011B TRANSMITTER	531X USR.OE EQU 0001000B OVERUN ERR 532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000010B TRANSMITTER 535X USR.TXR EQU 00000001B TRANSMITTER	531X USR. DE EQU 0001000B DVERUN ERR 532X USR. TXE EQU 00000100B TEANSMITTY ERRO 534X USR. TXR EQU 00000010B RECEIVER RE 535X USR. TXR EQU 0000001B TRANSMITTER	531X USR. DE EQU 0001000B OVERUN ERR 532X USR. TXE EQU 00000100B TEANSMITTY ERRO 534X USR. RXR EQU 00000010B RECEIVER RE 535X USR. TXR EQU 00000001B TRANSMITTER	531X USR.OE EQU 0001000B OVERUN ERR 532X USR.TX EQU 00000100B TAAISTY ERRO 534X USR.TXR EQU 00000010B RECEIVER TE 535X USR.TXR EQU 00000010B TRANSMITTER	
532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000010B RECEIVER RE. 535X USR.TXR EQU 00000001B TRANSMITTER	532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.RXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000001B TRANSMITTER	532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000010B RECEIVER RE 535X USR.TXR EQU 00000001B TRANSMITTER	532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000011B RECEIVER RE 535X USR.TXR EQU 00000001B TRANSMITTER	532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 0000001B RECEIVER RE 535X USR.TXR EQU 00000001B TRANSMITTER	532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000010B TRANSMITTER 535X USR.TXR EQU 00000001B TRANSMITTER	532X USR.PE EQU 00001000B PARITY ERRO 533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000010B TRANSMITTER 535X USR.TXR EQU 000000010B TRANSMITTER	532X USR.PE EQU 0000100B PARITY ERRO 533X USR.TXE EQU 0000010B TRANSMITTER 534X USR.TXR EQU 0000001B TRANSMITTER 535X USR.TXR EQU 00000001B TRANSMITTER	
533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 0000001B RECEIVER RE. 535X USR.TXR EQU 0000001B TRANSMITTER	533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 0000001B TRANSMITTER	533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 0000001B TRANSMITTER	533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000001B TRANSMITTER	533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 0000001B TRANSMITTER	533X USR.TXE EQU 0000010B TRANSMITTER 534X USR.TXR EQU 00000011B TRANSMITTER TRANSMITTER	533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 00000011B TRANSMITTER	533X USR.TXE EQU 00000100B TRANSMITTER 534X USR.TXR EQU 0000001B TRANSMITTER	
534X USR.RXR EQU 00000010B RECEIVER RE. 535X USR.TXR EQU 0000001B TRANSMITTER	534X USR.RXR EQU 00000010B RECEIVER RE. 535X USR.TXR EQU 0000001B TRANSMITTER	534X USR.TXR EQU 00000010B RECEIVER RE.	534X USR.TXR EQU 0000001B RECEIVER RE.	534x USR.TXR EQU 0000001B RECEIVER RE.	534x USR.TXR EQU 0000001B RECEIVER RE.	534x USR.TXR EQU 0000001B RECEIVER RE.	534X USR.TXR EQU 00000018 RECEIVER RE.	
535X USR.TXR EQU 0000001B TRANSMITTER	535X USR.TXR EQU 0000001B TRANSMITTER	535X USR.TXR EQU 0000001B TRANSMITTER	535X USR.TXR EQU 0000001B TRANSMITTER	535X USR.TXR EQU Q000001B TRANSMITTER	535X USR.TXR EQU 0000001B TRANSMITTER	535X USR.TXR EQU 0000001B TRANSMITTER	535X USR.TXR EQU 0000001B TRANSMITTER	

* LEVEL O - RESET THIS "INTERRUPT" MAY NOT BE PRO	
542 ** LEVEL 0 - RESET 543 * THIS "INTERRUPT" MAY NOT BE PRO 545 0RG 00A 547 548 RAMBGO EQU *	
544	
545 546 0RG 00A 548 RAMBGO EQU *	ESSED BY A USER PROGRAM.
547 548 RAMBGO EQU * 549	
249	Address /Ram8Go 2/
550 <b>*</b>	
021 000 000 551 INITO LXI DARANGO 303 016 004 552 JAP XINIT	OF RAM (ROW) (ROW) (RAMES O 2)
553 JMP 554 ERRPL	10A MUST BE 0
556 ** LEVEL 1 - CLOCK	
INTI EQU 100 INTERRUPT	PT ENTRY POINT
559 560 FRRN #-110 INTO TAKES	
315 132 000 561 CALL SAVALL SAVE USER	ER REGISTERS
563 HVI D.O 563 JMP CLOCK PROCES 564 ERRPL CLOCK-1000A EXTRA	S CLOCK INTERRUPT BYTE MUST BE 0
566 ** LEVEL 2 - SINGLE STEP	
568 + IF THIS INTERRUPT IS RECEIVED WHEN NOT IN MC	HEN NOT IN MONITOR MODE,
* 025	NG). IN SUCH CASE, THE
000.020 573 INT2 EQU 20A LEVEL 2 ENTRY	ENTRY
575 ERRNZ #-21A INTI	KES EXTRA BYTE
CALL SAVALL SAVE	REGISTERS
032 577 LDAX U (A)	CILFLGJ
303 244 001 579	TURN

583			581	* *	NI 0/1	INTERRUPT VECTORS.	
1			583	• •	INTERR	з тнкоисн 7	AVAILABLE FOR GENERAL
586			585		THESE	INTERRUPTS ARE NO	SUPPORTED BY PAM/8, AND
303 045 040 590 1NT3 JHP UIVEC+6 JUMP TO USER ROUTINE  064 064 064 592 1NT3 JHP UIVEC+6 JUMP TO USER ROUTINE  100 112 107 594 1NT4 JHP UIVEC+9 JUMP TO USER ROUTINE  100 112 107 595 1NT4 JHP UIVEC+12 JUMP TO USER ROUTINE  100 112 107 595 1NT4 JHP UIVEC+12 JUMP TO USER ROUTINE  100 112 107 597 DB 1009,1129,1079,1149,1009 Support Code  100 112 107 597 DB 1009,1129,1129,1149,1099 Support Code  100 112 107 597 DB 1009,1129,1129,1149,1009 Support Code  100 112 107 597 DB 1009,1129,1129,1149,1109 Support Code  100 112 107 597 DB 1009,1129,1129,1149,1109 Support Code  100 112 107 597 DB 1009,1129,1149,1149,1199 Support Code  100 112 107 597 DB 1009,1129,1149,1149,1199 Support Code  100 112 107 597 DB 1099,1129,1149,1149,1199 Support Code  100 112 107 597 DB 1099,1129,1149,1149,1149,1149,1149,1149,11			586 587	* *	NEVER (THROU	OCCUR UNLESS THE GH UIVEC)	SER HAS SUPPLIED HANDLER
303 045 040 590 1N13 JHP UIVEC+6 JUNP TO USER ROUTINE  064 064 064 592 086 404  303 050 040 594 1M14 JNP UIVEC+9 JUNP TO USER ROUTINE  100 112 107 597 DB 1009,1129,1070,1140,1000 Support Code  100 112 107 597 DB 1009,1129,1070,1140,1000 Support Code  100 112 107 597 DB 1009,1120,1070,1140,1000 Support Code  100 112 107 597 DB 1009,1120,1120,1070,1140,1000 Support Code  100 112 107 597 DB 1009,1120,1120,1070,1140,1000 Support Code  100 112 107 597 DB 1009,1120,1120,1070,1140,1000 Support Code  100 112 107 597 DB 1009,1120,1120,1140,1100 DB 1009,1120,1140,1140,1140,1140,1140,1140,1140	000-030		588		ORG	30≜	
064 064 064 592 08 '44470' Heath Part Number  303 050 040 595 INT4 JRP UTVEC+9 JUMP TO USER ROUTINE  100 112 107 597 086 50A  303 055 040 600 INT5 JMP UTVEC+12 JUMP TO USER ROUTINE  603 ++	:	303 045 040		INT3	JAP	UIVEC+6	JUMP TO USER ROUTINE
303 050 040 595 INT4 JAP UIVEC+9 JUMP TO USER ROUTINE 100 112 107 596 100 112 107 597 08 1009+1129,1079,1149,1009 Support Code (Ram8Go 2 601 601 602 4 EXIT NONE 603 4 EXIT NONE 603 4 EXIT NONE 603 4 EXIT NONE 604 605 1 ENTRY (A) = MILLISECONO DELAY COUNT/2 605 4 EXIT NONE 606 5 ENTRY (A) = MILLISECONO DELAY COUNT/2 607 4 USES AFR 608 609 0LY PUSH FSW SAVE COUNT/2 303 143 002 611 JMP HAND PROCESS AS HORN 615 610 AND ACCESS AS HORN 615 611 0MP ACCESS AS HORN 616 615 0MP ACCESS AS HORN 617 618 MACCESS AS HORN 618 FROGERAM	000.033	<b>+90 +90 +90</b>			0.8	.02444.	Part Number
100 112 107 597 DB 1009,1120,1070,1140,1000 Support Code /Ram86o 2 303 053 040 600 INT5 JNP UIVEC+12 JUNP TO USER ROUTINE 601 + DLY - DELAY TINE INTERVAL. 603 + ENTRY (A) = MILLISECOND DELAY COUNT/2 605 + ENTRY (A) = MILLISECOND DELAY COUNT/2 606 + ENTRY (A) = MILLISECOND DELAY COUNT/2 607 + USES A NO		0.050	:	1NI	ORG	40A UI VEC+9	TO USER
303 053 040 600 INT5 JMP UIVEC+12 JUMP TO 601 602 603 ** DLY - DELAY TIME INTERVAL. 604 * ENTRY (A) = MILLISECOND DELAY 606 * EXIT NONE 606 * EXIT NONE 607 * EXIT NONE 608 * EXIT NONE 609 0LY PUSH PSH 257 610 XRA A DONT SO 303 143 002 611 JMP HRNO PRUCES 303 056 040 614 INT6 JMP HRNO 615 616 616 60A 303 235 001 618 JMP SSTILL RETURN 303 235 001 618 0RG 70A		100 112 107			90	1000,1120,1070	Support Code /Ram8Go 2
603 ** DLY - DELAY TIME INTERVAL. 604 * ENTRY (A) = MILLISECOND DELAY 606 * EXIT NONE 607 * USES A*F 608 DLY PUSH PSW CO 257 610 XRA A DONT SO 257 611 JMP RNO PRUCESS 303 143 002 611 JMP HRNO PRUCESS 303 056 040 614 INT6 JMP UIVEC+15 JUMP TO 615 616 615 616 JMP SSTI-CB-CLI+CB-SPK 303 235 001 618 JMP SSTI	000.000	303 053 040		:	ORG	50A UIVEC+12	JUMP TO USER ROUTINE
604			602	:	: :	DELAY TIME INTER	VAL.
365 600 0LY PUSH PSW SAVE CO 257 610 XRA A DONT SO 303 143 002 611 JMP HRNO PRUCESS 303 056 040 614 INT6 JMP UIVEC+15 JUMP TO 615 616 617 60• MVI A,CB.SSI+CB.CLI+Cb.SPK 303 235 001 618 JMP SSTI RETURN			606 606 606 607	* * * *	ENTRY EXIT USES	•	OND DELAY COUNT/2
257 303 143 002 611 JMP HRN0 PRUCESS 303 163 002 611 INT6 JMP UIVEC+15 JUMP TU 076 320 617 GO. MVI A.CB.SSI+CB.CL.I+Cb.SPK 303 235 001 618 JMP SST1 KETURN	:	365	809 909		PUSH	PSH	SAVE COUNT
303 056 040 614 INT6 JAP UIVEC+15 JUMP TU 615 60A 614 INT6 JAP UIVEC+15 JUMP TU 615 616 617 60. MVI A,CB.SSI+CB.CLI+CU.SPK 303 235 001 618 JMP SSTI RETURN 620 0RG 70A	:	143		:	XRA JAP	A HRNO	DOMI SOUND HORN PROCESS AS HORN
303 056 040 614 INT6 JMP UIVEC+15 JUMP TU 615 615 616 616 616 617 60. MVI A,CB.SSI+CB.CLI+CB.SPK 303 235 001 618 JMP SSTI KETURN 620 0RG 70A	: :				ORG	<b>60A</b>	
616 076 320 617 60. AVI A.CB.SSI+CB.CLI+CB.SPK 303 235 001 618 JMP SSTI RETURN 620 DRG 70A	:	950	:	IX.	dkr	UIVEC+15	JUMP TÜ ÜSER ROUTINE
620 ORG 70A	000.063	320		- : :	IVE	A+CB+SSI+CB+CL SST1	
303 061 040 621 INT/ JAP UIVEC+18	000.070	303 061 040	:	INT	ORG	70A UIVEC+18	JUMP TO USER ROUTINE

October   Octo	RAM.  * SETUP STACKPOINTER  * PRSROM* INTO RAM  BYTE EMENT DESTINATION EMENT SOURCE  DT DONE CH INCREMENT  = SEARCH INCREMENT  = FIRST RAM	CELLS IN RAM JRY EXISTS, S JOP. CLEAR AAIN LOUP HOVE BYT DECREMEN INCREMEN	CALLED WHENE AM/8 CONTROL HÖW MUCH WENG HE MONITOR LG FROM MASTER INTO PAM/8 r A000A	SETUP P DECODE ENTER T ENTRY EXIT TOX DOX DOX DOX DOX DOX DOX DOX DOX DOX D				0 673	
6.27	* SETUP STACKPOINTER, AND * PRSROM* INTO RAM BYTE HENT DESTINATION TOONE CH INCREMENT  SEARCH INCREMENT FIRST RAM	CELLS IN RAM JRY EXISTS, S JRY EXISTS, S CLEAR ANDY #PR MOVE BY DECRETER INCRETER INCRETER	ANY 8 CONTROL HEN WICH NEW HEN WICH NEW FROM MASTER INTO PANY 8 THE HEN HAA HAA HAA HAA HAA HAAA HAAAAAAAAAA	SETUP DÉCODÉ ENTER T ENTRY EXIT LDAX DOV DOX DOX LNZ LNZ LNZ				0 670	
629 * DECODE HOW HUCH MEMORY 630 + ENTER THE MONITOR LODG 631 * ENTER THE MONITOR LODG 632 * ENTER THE MONITOR LODG 633 * ENTER THE MONITOR LODG 634 634 ENTER THE MONITOR LODG 634 635 ENTER THE MONITOR LODG 634 635 ENTER THE MONITOR LODG 634 635 ENTER ED 645 1010 105 101 645 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 105 101 650 1010 101	** SETUP STACKPOINTER, AND ** PESRON* INTO RAM BYTE HENT DESTINATION ENENT SOURCE OF DONE CH INCREMENT  ** SEARCH INCREMENT  ** FIRST RAM	CLEAR AAIN LOUP COPY *PR NOVE BYI DECKEMEN INCREMEN IF NOT G	HOW HUCH HENCH HENCH HE MONITOR LO FROM MASTER INTO PAN/8 r INTO PAN/8 r H + A + A + A + A + A + A + A + A + A +	DECODE ENTER T EXIT HDAX HDV DCX INZ LNZ				0 520	
032 633	*PRSROM* INTO RAM BYTE AMENT DESTINATION EMENT SOURCE JT DONE CH INCREMENT  SEARCH INCREMENT  FIRST RAM	NE N						0 2 2 0	
032	*PRSROM* INTO RAM BYTE BYTE EMENT DESTINATION EMENT SOURCE JT DONE CH INCREMENT  SEARCH INCREMENT FIRST RAM	ALIN LOUP COPY + PR NOVE BY DECREMEN INCREMEN IF NOT C	FROM MASTER INTO PAN/8 r	EXIT EXIT EXIT EXIT EXIT EXIT EXIT EXIT				0 23 0	
032	# PRSROM# INTO RAM BYTE EMENT DESTINATION EMENT SOURCE DI DONE CH INCREMENT = SEARCH INCREMENT = FIRST RAM	COPY + PR COPY + PR HOVE BYT INCREMENT IF NOT 0	INTO PAM/8 7	LDAX DCX DCX INR INR				0 2 2 0	
032 635 INIT LDAX D 167 637 MOV H,A 037 639 DCX H 037 DCX H 302 073 000 640 JNZ INIT 041 000 040 644 MVI D,SINCR/256 041 000 040 645 LXI H,START 041 000 040 649 INITI JHP XINITI 110 105 101 651 ERNZ +-000117A 053 656 INIT2 DCX H 041 364 007 658 ERNZ +-000117A 053 656 INIT2 DCX H 041 364 007 659 PUSH H 051 CALL PATCHI 053 656 SOLL POSH 054 PATCHI	M* INTO RAM DURCE EMENT CH INCREMENT	D COPY D	E E E I NIT 40000	L L L L L L L L L L L L L L L L L L L				0 2 3 0	
032	ESTINATION OURCE EMENT CH INCREMENT	HOVE BYT HOVE BYT DECKETEN INCREMEN IF NOT	D H H H H H H H H H H H H H H H H H H H	LDAX PDCX FDCX FDC				0 223 0	
167 637 MOV H,A 054 638 DCX H 034 639 INR E 302 073 000 640 JNZ INIT 0642 SINCR EQU 4000A 0642 SINCR EQU 4000A 0640 644 WVI D,SINCR/256 061 000 060 645 INIT JHP XINIT 110 105 101 650 INIT JHP XINIT 110 105 101 651 ERRY +-000117A 053 656 INIT DX H 371 657 ERRY +-000117A 063 656 INIT DX H 315 254 007 659 PUSH H 315 254 007 660 GALL PATCH1 346 651 GALL PATCH1	OURCE COURCE CEMENT CH INCREMENT IT RAM	NOVE BYT	#*A H I I I MI MO000A	DOCX INN E DO				073 0	
034 302 073 000 640 302 073 000 640 302 073 000 640 642 SINCR EQU 4000A 663 664 664 664 WI D,SINCR/256 664 664 DETERMINE MENORY LIMIT 10 105 101 651 BS 1 BS 1 HEATH 110 105 101 652 BRR 2 4-000117A 655 INIT2 DCX H 655 656 INIT2 DCX H 657 857 LXI 657 BUSH 115 254 007 659 PUSH 115 BATCH1	GURCE EMENT CH INCREMENT T RAM	N N N N N N N N N N N N N N N N N N N	1 INIT +0000A	INR INZ EOU				073 0	
302 073 000 640 JNZ INIT 642 SINCR EQU 4000A 026 004 644 NVI D,SINCR/256 041 000 040 645 LXI H,START 648 648 DETERMINE MEMORY LIMIT 303 000 004 649 INITI JMP XINITI 110 105 101 650 DB "HEATH" 650 DB "HEATH" 651 ERNZ +-000117A 653 656 INITZ DCX H 653 656 INITZ DCX H 654 656 INITZ DCX H 655 656 INITZ DCX H 657 659 PUSH H 657 659 PUSH H	EMENT CH INCREMENT T RAM	SEARCH	1N11 40004	JNZ			-	073 0	
026 004 642 SINCR EQU 4000A 026 004 644 NVI D,SINCR/256 041 000 040 645 LXI H,START 648 647 DETERMINE MEMORY LIMIT 648 1011 JHP XINITI 110 105 101 651 DB "HEATH" 652 ERNZ +-000117A 653 655 ERNZ +-000117A 653 655 LMITZ DCX H 654 F RETURN TO INLINE CODE 655 F RETURN TO INLINE CODE	EMENT CH INCREMENT IT RAM	SEARCH	4000 <b>4</b>	EQU			000		
026 004 644 WVI D,SINCR/256 041 000 040 645 LXI H,START 646 647 # DETERMINE MEMORY LIMIT 648 303 000 004 649 INITI JMP XINITI 110 105 101 651 ERRAZ #-000117A 653 ERRAZ #-000117A 653 655 INIT2 DCX H 655 1NIT2 DCX H 657 87HL 1XI H,DEFPC 345 650 650 650 650 650 650 650 650 650 65	SEARCH INCREMENT FIRST RAM	5	5	2					:
026 004 644 WVI D,SINCR/256 041 000 040 645 LXI H,START 646 647	SEARCH INCREMENT FIRST RAM					i.m			
041 000 040 645 LXI H,START 646 647 * DETERNINE MEMORY LIMI1 303 000 004 649 INITI JMP XINITI 110 105 101 651 DB "HEATH" 653 ERRNZ *-000117A 653 654 * RETURN TO INLINE CODE 654 * RETURN TO INLINE CODE 655 INITZ DCX H 657 SPHL 657 LXI 657 LXI 345 254 007 659 PUSH H 1345 254 007 650 CALL PATCH1 346 PATCH1	FIRST RAM	•	D, SINCR/256	IAN		644		004	26 00
303 000 004 649 INITI JNP XINITI 110 105 101 650 10111 JNP XINITI 110 105 101 651 008 "HEATH" 653 ERRNZ *-000117A 653 655 RETURN TO INLINE CODE 053 655 INITZ DCX H 371 657 SPHL 041 364 007 658 LXI 345 254 007 658 LXI 345 254 007 650 CALL PATCH1 346 CALL PATCH1		•	H, START	ראו		645	040	000	41 00
303 000 004 649 INITI JMP XINITI 110 105 101 651		411.		DETERMI	*	647			:
303 000 004 649 INITI JHP XINITI 110 105 101 650 108 "HEATH" 651 108 653 ERRNZ #-000117A 653 654 RETURN TO INLINE CODE 053 655 INITZ DCX H 371 657 SPHL 041 364 007 658 LXI H, DEFPC 345 254 007 650 PUSH H 345 254 007 660 CALL PATCH1		,				648			
110 105 101 651	TO FREE SPACE /RAMBGO JUNBO/	JUMP TO	XINITI	d W 7	INITI	649	<b>\$00</b>	0.000	03 00
652 ERRNZ *-000117A 653	/RAMBGO		• HEATH	08		651	101	1.92.1	10 10
653	/RAM8GO		*-000117A	ERRNZ		652			
053 655 INITZ DCX H 371 657 SPHL 341 659 LXI H,DEFPC 345 650 CALL PATCH1 345 54 007 660 CALL PATCH1	SET TO FIRST NON-EXISTANT LOCATION	DE NITH HL SE	O INLINE	Z	•	653			
371 657 SPHL H, DEFPC SET STACKPOINTER = MEMORY 041 364 007 658 LXI H, DEFPC 561 FC value on stack 345 660 CALL PATCH1 Tape UART/Auto-Boot 345			I		INITZ	655			53
041 364 007 658 LXI H,DEFPC Set *PC* value on stack 345 660 CALL PATCH1 Tape UART/Auto-Boot 345 661 CALL PATCH1	STACKPOINTER - MEMORY LIMIT -1	SET STAC		SPHL		657			71
345 660 CALL PATCHI THE UART/Auto-Boo 315 254 007 660 CALL PATCHI TABE UART/Auto-Boo			H, DEFPC	LXI		658	007		
315 254 007 660 CALL PATCH1	tack	Set *PC	I	PUSH		629	,	i	,
	UART/Auto-Boot /Ram8Go	Tape UA	PATCH1	CALL		999	200	<b>*</b> :	٧.
257 AA2 AA2 AA2 AA2 AA2 AA2 AA2 AA2 AA2 AA		Set Keti	E <	107		100			
AKA AKA A A CONTRACT TO SAME	•	•	<	<b>*</b>		700			<b>6</b> 4

000-150-0 13 775 EERR GTCCC-NFCA-T (1 - MCLAS 000-150-0 000-150-0 01 775 EERR GTCCC-NFCA-T (1 - MCLAS 000-150-0 01 775 EERR GTCCC-NFCA-T (1 - MCLAS 000-150-0 01 775 EERR GTCC-NFCA-T (1 - MCLAS 000-150-	000.266					
012 777 ERRNY CHIEFGNFLAG-1  27 ERRNY COLHY-2000 ASSUME HIGH-DROER FOR HALT  782 W NOT TH MONITOR MODE: CHECK FOR HALT  783 W NOT TH MONITOR MODE: CHECK FOR HALT  784 M AT 10 (LA 10 CHE RESISTER MODESSS  785 M AT 1 ATTO (LA 10 CHE RESISTER MODESSS  785 M AT 1 ATTO (LA 10 CHE RESISTER MODESSS  785 M AT 1 ATTO (LA 10 CHE RESISTER MODESSS  785 M AT 1 ATTO (LA 10 CHE RESISTER MODESSS  785 M AT 1 ATTO (LA 10 CHE RESISTER MODESSS  785 M AT 1 M MITTOR MODE  785 M M M M M M M M M M M M M M M M M M M	***		377	DCX	æ	
927 779 EAKHZ UU-HT-2000 ASSUNE HICH-DRUBER 932 313 000 779 EAKHZ UU-HT-2000 ASSUNE HICH-DRUBER 932 313 000 779 EAKHZ UU-HT-2000 ASSUNE HICH-DRUBER 932 315 002 003 783	000.000	:	776 777	ERRN2 LDAX	CTLFLGMFLAG- B	
312 313 000 786   UC CLK4   SKIP IT	000.000	:	778	ERRNZ RAI	UD. HL T-2009	
776 012 775 775 401 14 FONITOR FRODE, CHECK FOR HALT 7 100 175 012 775	000.271	332	:	၁၅	CLK4	SKIP IT
906 012 784 NVI 1,10 (A) - IMOEX OF 497 REC 136 012 785 CALL LA. LUCATE RELISTER ADDRESS 136 786 CALL LA. LUCATE RELISTER ADDRESS 136 786 CALC LA. LUCATE RELISTER ADDRESS 136 789 CALC LOX D CALC D. LOX D CALC CALC LA. LUCATE RELISTER ADDRESS 137 322 300 792 CALC CONTENT CALC CALC CALC CALC CALC CALC CALC CAL			:		MONITOR MODE.	CHECK FOR HALT
135 782 604 644 644 644 664 604 644 604 664 604 645 644 604 645 644 644 644 644 644 644 644 644 64	000.274	076 012		IANI	A,10	(A) = INDEX OF *P* REG
126 788 607 604 607 604 607 607 607 607 607 607 607 607 607 607	000 301	136		NO.	E, A	LUCALE REGISTER AUDRESS
053 789 0CX 0 312 322 000 793 CHECK FOR 'KETOUN TO MONITOR' KEY EMTRY. 794 CPL 18, ALC ON 19 79 CPL 560 18, ALC ON 19 18 ACCESSING 0F 333 360 797 IN 18, ALC ON 19 18 ACCESSING 0F 302 165 000 799 JANE CULL IF NOT ALLON USER PROCESSING 0F	000 302	126	787	××××	I	(D.F) = PC CONTENTS
376 166 791 CDA NITH CHECK FOR HALT 312 322 000 793 1	000.304		789	X O		
312 322 000 792 JE ERROR IF FALL, BE IN HOUTOR HODE 794 CHECK FOR "RETORN TO HONITOR" KEY ENTRY. 794 CHECK FOR "RETORN TO HONITOR" KEY ENTRY. 795 CLK4 E0U * 1P.PAD 376 079 JM LD 1P.PAD 376 079 JM CULL IF PAD 376 056 799 CP CULL IF NOT, ALLOR USER PROCESSING OF CULL STAND OF THE CULL IF NOT, ALLOR USER PROCESSING OF	000 30	032	790	YVO7	0 1	Tive aga yourg
793 + CHECK FOR *RETURN TO HONITOR* KEY ENTRY. 795 CLK4 EGU + 797 CLK4 EGU + 338 360 799 CLK EGU + 376 056 799 CPI 560 CPI 560 SEE F '0' AND 'FF 376 056 799 CPI 560 CUIL TE NOT. ALLON USER PROCESSING OF	000.310	312 322		7 - F	ERROR	IF HALT, BE IN MONITOR MODE
333 360 799 FOUR PAD SEE IF "O" AND "F" BOCESSING OF CULL SGOOT 799 JOHN COULL SGOOT 799 JOHN COULT SGOOT 799 JOHN			:		FOR "RETURN TO	MONITOR" KEY ENTRY.
333 360 799	000		96	4	*	
376 056 799 CPI 560 SEE IF '0' AND 'FF PROCESSING OF LIF NOT, ALLOW USER PROCESSING OF	000.31	333	97	•	IP.PAD	
	000.31	376 056	_ '	CPI	560	10 07113313000

322 322 322 332 332 331 00 331 331 000 331	ODE, AND RESTORES THE STACKPOINTER,  MFLAG  AFLAG  AFLAG  AFLAG  AFLAG  AFLAG  AFLAG  AFLAG  AFLAG
807 * IT RESET 808 * ENTRY 808 * ENTRY 809 * EXIT 811 * EXIT 812 * USES 812 * USES 814 * USES 814 * USES 815 ERRUR EQU 815 ERRUR EQU 815 ERRUR 818 HUV 812 816 ERRUR 818 HUV 813 813 043 821 ERRUZ 813 043 823 ERRUZ 812 ERRUZ 812 ERRUZ 812 ERRUZ	RESTORES
809	MFLAG RE-EMABL
810 * EXIT 811 * USES 812 * USES 814 * USES 814 * USES 815 * USES 816 EROR EQU 825 176 818 HUV 830 167 820 NOV 831 043 820 NOV 821 1000 822 NOV 823 ERROZ	MFLAG RE-ENABL
812 # USES 814 # USES 814 815 822 041 010 040 816 ERRUR EQU 825 176 818 HUV 825 176 818 HUV 831 157 819 ANI 831 043 821 INX 832 066 360 822 ERRNZ	MFLAG RE-ENABL
814 815 815 816 816 816 817 818 818 810 810 820 820 821 821 820 821 821 821 821 821 821 821 821	AFLAG RE-ENABL
322 041 010 040 816 ERRÜR EQU 325 176 818 HUV 326 346 275 819 ANI 330 167 820 HOV 331 043 821 INX 332 066 360 822 RVI 300 373 823 ERRNZ	MFLAG RE-ENABL
225 176 818 HUV 326 346 275 819 ANI 330 167 820 HOV 331 043 821 INX 332 066 360 822 HVI 300 323 ERRNZ	MFLAG RE-ENABL
330 167 620 MOV 331 043 820 MOV 332 066 360 822 MVI 300 323 ERRNZ	KE-ENABL
331 043 821 INX 332 066 360 822 MVI 300 823 ERRNZ 324 373 824 ET	CALCOK
332 066 360 822 MVI 306 823 EKRNZ 34 373 824 ET	
324 272 824 ET	
225 052 025 040 825	
371 337 826 SPHL	STORE STACK POINTER TO EMPTY STATE
.341 315 136 002 827 CALL ALARM	
632 633 344 MTR	
344 373 835 EI	
345 041 345 000 837 HTRL LXI HONTRL	0 LT 0 1 0 A C
LXI B.DSPMOD	(BC) = #DSPMOD
554 046 001 841 ANÍ I (A)	) = 1 IF ALTER
357 057 842 CMA	6 C C C C C C C C C C C C C C C C C C C
844 READ KEY	- TLAC DI- 17 ALIEK
948	
363 315 260 003 847 CALL RCK 366 052 024 040 848 LHLD ABUSS	READ CONSOLE KEYSET
371 376 012 849 CPI 10	
376 137 001 851 HOV E9A	IN THINKS WALID' GROUP SAVE VALUE
007 852 . SET DSPMOD	
853 LDAX 8 854 RRC	(A) = 0SPM0D
001 332 051 001 855 JC MTR5	IF IN ALTER MODE

858	(A)
326 004 860 MTR4 SUI 332 160 004 861 JC 332 160 004 861 JC 345 863 PUSH 031 864 LXI 026 000 865 DAD 867 MVI 031 869 XTHL 021 005 040 870 LXTH 021 005 040 870 LXTH 022 871 ANI 346 002 872 LDAX 346 002 877 LDAX 347 LDAX 348	F TABLE ENTRY F PROCESSOR J = (ABUSS) F REG INDEX
332 160 004 861 JC 345 006 862 N0V 345 000 864 LXI 026 000 865 NVI 031 866 DAD 343 869 XTHL 021 005 040 870 LXI 012 872 LDAX 346 002 873 ANI 012 874 LDAX 311 875 874 LDAX 311 875 874 LDAX 311 875 874 LDAX 311 880 D8 141 880 D8 165 883 D8	F TABLE ENTRY F PROCESSOR F REG INDEX
147 662 HUV 345 001 664 LXI 026 000 665 HVI 031 866 DAD 313 869 CXTH 021 005 040 870 LXI 012 872 LDAX 346 002 873 ANI 012 874 LDAX 311 875 RET 876 879 LDAX 311 876 879 LDAX 311 876 879 LDAX 311 877 LDAX 311 878 RET 141 880 D8 142 881 D8 165 883 D8	F TABLE CABLE REGIN
041 035 001 864 LXI 026 000 865 MYI 031 866 DAD 136 867 MOV 031 869 XTHL 021 005 040 870 LXI 012 872 LDAX 343 002 873 LAN 311 875 RET 165 879 DB 141 880 DB 143 881 DB 165 883 DB	- ADDRESS OF TABLE - ADDRESS OF TABLE - ADDRESS OF REG IN - BSPHOD 2* IT MEMORY DSPHOD TO PROCESSOR TO PROCESSOR TABLE ONPUT NINGLE STEP ASSETTE DUMP EXT ASSETTE ASSE
026 000 865 MVI 031 866 DAD 033 869 XTHL 021 005 040 870 LXI 012 872 LDAX 346 002 873 LDAX 346 002 874 LDAX 311 875 RET 878 MTRA EQU 165 880 DB 141 880 DB 145 881 DB 165 883 DB	- ADDRESS OF TABLE - ADDRESS OF PROCES - ADDRESS OF REG IN DSPHOD DSPHOD TO PROCESSOR TO PROCESSOR TABLE ONPUT NITUT NIT
031 866 DAD 136 867 MOV 031 869 XTHL 021 005 040 870 LXI 022 002 872 LDAX 346 002 874 LDAX 310 874 LDAX 311 875 RET 165 880 DB 141 880 DB 145 881 DB 165 883 DB	- ADDRESS OF TABLE - ADDRESS OF PROCES - ADDRESS OF REG IN - ADDRE
130 867 MOV 343 869 XTHL 321 005 040 870 LXI 022 005 040 870 LXI 340 002 877 LDAX 340 002 877 LDAX 311 875 LDAX 312 874 LDAX 311 875 RET 876 RET 877 LDAX 311 875 RET 878 MTRA EQU 141 880 D8 141 880 D8 143 881 D8 143 881 D8	- ADDRESS OF PROCES - ADDRESS OF REG IN DSPHOD TO PROCESSOR TABLE ONPUT OYPUT
343 343 343 021 002 346 002 346 002 347 311 312 374 311 312 314 315 316 316 317 317 318 318 319 319 319 319 319 319 319 319	DORESSON  DORESSON  DOSPHOD  TO PROCESSOR  TABLE  ON PUT  INGLE STEP  A SSETTE DUMP  EXT  SAST  BORY  ANST  BORY  BORY  ANST  BORY  BORY  ANST  BORY
021 005 040 870 LXI 012 872 L0AX 314 002 873 ANI 012 874 LDAX 311 875 RET 876 RET 165 879 DB 141 880 DB 141 881 DB 145 882 DB 332 884 DB	DSPMOD  Z' IF HEMORY DSPMOD  TO PROCESSOR  TABLE OUTPUT OUTPUT OUTPUT ASSETTE DUMP EXT
012 871 . SET 012 872 LDAX 314 002 874 LDAX 311 875 RET 876 RET 876 RET 876 RET 877 878 RET 879 08 141 880 08 141 881 08 143 881 08 220 884 08	DSPMOD  SPMOD  OSPMOD  TO PROCESSOR  TABLE  OTHUT  UTPUT  UNFUT  NASETTE DUMP  A SSETTE DUMP  EXT  B OR  SOR  SOR  SOR  SOR  SOR  SOR  SOR
346 002 873 LDAX 346 002 873 ANI 012 874 LDAX 311 876 RET 877 RET 165 877 MTRA EQU 141 880 D8 141 881 D8 220 883 D8	P TABLE GO THE NEW TO UN PUT OUT PUT PUT OUT PUT OUT PUT OUT PUT OUT PUT OUT P
346 002 873 ANI 012 874 LDAX 311 875 RET 877 877 165 879 141 880 08 141 881 08 145 883 08 332 884 08	P TABLE GOOTPUT OUTPUT OUTPUT SINGLE ST CASSETTE NEXT LAST ABORTAN
311 874 LDAX 311 875 RET 876 876 877 877 878 978 08 141 880 08 143 881 08 220 883 08	TO PROCE TO PROCE TO PROCE SOUTHUT SOUTHUT CASSETTE CASSETTE NEXT LAST LAST ABORT
311 875 RET 876 876 877 878 EQU 165 879 D8 141 880 D8 143 881 D8 220 883 D8 332 884 D8	TO PROCE TABLE GOUTPUT SINGLE ST CASSETTE CASSETTE NEXT LAST LAST
35 165 878 MTRA EQU 35 165 879 MTRA EQU 36 141 880 D8 37 143 881 D8 41 220 883 D8 42 332 884 D8	UMP TABLE  - GO - INPUT - OUTPUT - OUTPUT - CASSETTE - CASSETTE - CASSETTE - LAST - LAST - OISPLAY/A
35 165 878 MTRA EQU 35 165 879 D8 36 141 880 D8 37 143 881 D8 40 165 982 D8 42 332 884 D8	- GO - GO - INPUT - OUTPUT - OUTPUT - CASSETTE - CASSETTE - CASSETTE - CASSETTE - CASSETTE - CASSETTE - CASSETTE - OISPLAY/A
35 165 879 08 36 141 880 08 37 143 881 08 40 165 882 08 41 220 883 08	- 60 - 60 - 100 TPUT - 51 NGLE ST - CASSETTE - CASSETTE - LAST - LAST - ABORT
36 141 860 08 37 143 861 08 40 165 882 08 41 220 883 08	- 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
37 143 661 06 40 165 682 08 41 220 683 06 42 332 884 06	- 00 TPUT - 00 TPUT - 01 TPUT - 02 SETTE - 02 SETTE - 02 SETTE - 03 SETTE - 01 SPLAYA
40 165 682 08 41 220 883 08 42 332 884 08	- SINGLE ST CASSETTE - CASSETTE - NEXT - LAST - ABORT
41 220 683 06 42 332 884 D8	- 24 NGCE 2 - CASSETTE - CASSETTE - NEXT - NEXT - AB ORT - 015PLAY/A
42 332 884 08	- CASSETTE - NEXT - LAST - ABORT - 01SPLAY/A
	- NEXT - LAST - ABORT - DISPLAY
•043 067 885 D8	LAST ABORT DISPLAY
.044 104 886 08	ABORT
045 102 887 08	DISPLAY
•046 060 888 <b>0</b> 8	
047 116 889 08 N	MORY H
r.	• - REGISTER MODE
892 ** PROCESS MEMORY/REGISTER	GISTER ALTERATIONS.
•	
894 * THIS CODE IS ENTERED IF	RED IF
	AND
	<b>)</b> :
017 899 MTR5	
173 900	(A) = VALUE
ပ္	
067 902 STC	INDICATE 1ST DIGIT IS IN (A)
315 066 003	BYTE
043 904 INX	

		927	* *	REGM -	ENTER REGISTER D	DISPLAY MODE.
		929	*	ENTRY	(A) - DSPMOU	
076	002	931	REGM	IVI	2.4 2.4	SET DISPLAY REGISTER MODE
~		932 933		SET STAX	05PM00 8	SET
: '		934		ERRNZ	DSPMOD-DSPROT-1	TO 0000 - 1017
001-100 257		0369		XXX		
005		937		STAX	. 40	SET ALL PERIODS ON
315	260 003	938		CALL	× ×	READ KEY ENTRY
116 376	90	940		CPI	9	
322	322 000	941		) N	ERROR	NOT 1-6
123 007		945		RLC	Q	SET NEW REG INO
040.005		944	•	SET	REGI	
		047	*	1 2 2	TOGGLE DISPLAY/ALTER MODE	TEX MODE.
		• •	•	1		
		4.1	*	ENTRY	(A) = DSPHO0	
		ດ	•		• AUUKESS	UF USFRUU
	. (	952	• 6	SET	DSPMOD	
920	100	<b>v</b> .v	¥	STAX	<b>4</b> 6	
001-131 311		· 10		RET		
		957	*	NEXT	INCREMENT DISPLAY	AY ELEMENT.
		958 959	* *	ENTRY	- (ABUSS)	
		096	*		(DE) = ADDRESS	OF REGIND
-		962	NEXT	×××	I	
001-133 312	063 001	963	:	75	SAE	IF MEMORY, STORE ABUSS AND EXIT
		964	*	IS REG	IS REGISTER MODE.	
400		996		100	1000	
001-136 032		968	•	X VOT	100	(A) * REGI
	002	696		104	2	~ oz ⋅
001-141 022	-	970		STAX	0	ARAP TO #SP#
:	• 10	972		בים בי		IF NOT TOO LARGE, EXIT
		973		XX	4	OVERFLOM
		426	:	STAX	0	
:		975	ABORT	RET		

977 ** LAST - DECREMENT DISPLAY ELEMENT.	979 * ENTRY (HL) = (ABUSS) 980 * (DF) = ADDRFS OF REGIND	LAST 0CX H		+ 15 REGIS	SET SET	SUI 2 STAX D	ZZ		998 ** MEMM - ENTER DISPLAY MEMORY MODE.	* ENTRY (BC) * ADDRESS O	:	XATS	006 B CX B COO	1008 LXI 1009 JAP	1011 ** IN - INPUT DATA BYTE.	1013 1013 1014 ## OUT - DITPUT DATA SYTE	1016 * ENTRY (HL) = (ABUSS) 1017	1016 IN MVI 6, MI. IN	OUT MVI BAMI.OUT	1021 MOV A,H (A) = VALUE	- (1) - (1) - (1)	1024 SHLD IDWRK	1025 CALL
			.151		032	001.155 326 002 001.157 022		001.163 022 001.164 311			001.165 257 040.007	001.166 002	1.167 013	001.171 041 025 040 001.174 303 062 003				001-177 006 333	<b>-</b>	174	001.206 150	.207	200 616 217

Advance - 196 - 5001 - 196 - 1	102 - 103 - 104 - 104 - 105 -			
SOL. 217 303 063 001 1028 Ame SAE STORE ABUSS AND EXIT	9201. 102 00 00 3 00 1. 102 00 1. 10		1.02.00.	Unix H8ASM VI.4.1 5-Jui-80 Page 16:53:03 11-SEP-80
		001.217 303 063 001 1028	Q.E.	STORE ABUSS AND EXIT
		:		

303 063 000 363 011 040 372 011 040 373 360 062 011 040 341 341 303 344 000 302 344 000 303 344 000 042 040 042 031 040		:			1033	:	GO - RE	- RETURN TO USER MODE	MODE
225 303 063 000 1037 GD JHP GD.  1039 ** SSTEP - SINGLE STEP INST 1040 * ENTRY NOME 1041 ** ENTRY NOME 1042 1043 SSTEP EDU ** 225 0.02 0.11 040 1045 241 363 172 000 1056 244 366 020 1056 245 323 360 1055 246 323 360 1055 247 366 020 1056 248 020 1069 SSTI SSTEP EDU 249 020 1069 SSTI SSTEP EDU 240 341 10 040 1069 241 303 172 000 1056 244 366 020 1056 245 046 020 1056 251 346 020 1056 252 302 044 000 1059 253 302 045 040 1061 255 303 042 040 1061 256 042 031 040 1063 256 042 031 040 1066 257 045 046 060 1066 258 047 047 047 060 1066 258 048 048 060 1066 259 048 048 060 1066 250 048 049 060 1066 250 048 049 060 1066 251 049 060 1066 252 048 060 1066 253 048 060 1066 254 045 046 060 1066 255 046 047 048 060 1066 256 047 048 060 1066 257 048 060 1066 258 048 060 1066 258 048 060 1066 259 048 060 1066 259 048 060 1066 250 048 060 1066 250 048 060 1066 250 048 060 1066 250 048 060 1066 250 048 060 1066 260 048 060 106					1035	• •	ENTRY	NONE	
1039 ** SSTEP - SINGLE STEP INST  1041 * ENTRY NONE  1042 SSTEP - SINGLE STEP INST  1043 SSTEP - SINGLE STEP INST  363 020 1040 1045 SSTEP  364 020 1046 SSTEP - SINGLE STEP RET  1052 ** STEPTN - SINGLE STEP RET  1053 ** STEPTN - SINGLE STEP RET  1054 020 1056 01  305 020 1056 01  305 020 1056 01  306 020 1056 01  307 022 1058 01  308 042 040 1067 040  309 042 040 1067 040  309 042 040 1067 040  309 042 040 1067 040  309 042 040 1067 040  309 042 040 1067 040  309 042 040 1067 040  309 042 040 1067 040  309 042 040 1067 040  309 042 040 1067 040  309 042 040 040 060  309 042 040 040 060  309 042 040 040 060  309 042 040 040 060  309 042 040 040 060  309 042 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040  309 040 040 040 040 040  309 040 040 040 040 040  309 040 040 040 040 040 040 040 040 040 0	01.222		m.	8	1036		d#C	<b>.</b> 00	IS IN HASTE
1040 * ENTRY NONE  1041 * ENTRY NONE  1042 1044					1039	:		STEP	
36.3 11 040 1042 1044 1014 1044 1014 1044 1014 1044 1014 1044 1014 1044 1014					1040	* *			•
363 011 040 1045 LOA CTLFLG 356 020 1046 SST1 STA CTLFLG 362 011 040 1049 SST1 STA CTLFLG 362 011 040 1050 44 SST1 STA CTLFLG 366 020 1055 44 STPRTH - SINGLE STEP RET 1053 STPRTH = 00	001.22	:			1042		EQU	*	SINGLE STEP
356 020 1046 XRI C8.551 323 360 1047 001 00, CTL 062 011 040 1059 017 00, CTL 1049 971 1049 303 172 000 1050 1050 1055 001 00, CTL 1053 1054 1055 001 00, C6.351 323 360 1056 001 00, C6.351 324 000 1059 010 01 00, CE.ML 302 344 000 1060 1060 1060 1060 1060 1060 106	001.22			: '	1044		10	0.11.01.0	DISABLE INTERRUPTS UNTIL THE RIGHT TIME
323 360 1047 591 574 571 601 662 011 040 1048 5871 574 CTLELG 303 172 000 1050 JMP HITXIT 1052 44 51PRTN EQU 4 1055 1055 1055 001 0P.CTL 022 1055 574 00 1050 JMZ TRR 3 302 344 000 1050 JMZ TRR 3 303 344 000 1059 574X D 3042 041 244 002 1066 KMEN LXI H,TPABT 042 031 040 1067 541 244 002 1066 KMEN LXI H,TPABT 042 031 040 1067 541 244 002 1067 541 241 241 241 241 241 241 241 241 241 2	001.23	•		> :	1046		XRI	C8.55I	CLEAR SINGLE STEP INHIBIT
303 172 000 1049 5571 577 CTLFLG 303 172 000 1050 JMP HIXIT 1053 ++ STPRTN = 0U + 1054 57PRTN = 0U + 1055 1055 0UT CB.5SI 323 360 1055 0UT CB.7TLFLG 022 1055 577 SET CTLFLG 022 1055 77 SET CTLFLG 022 1055 77 SET CTLFLG 302 344 000 1060 JNP UNEC+3 303 042 040 1060 34* RMEH - LOAD HEMORY FROM 1064 4 RMEH - LOAD HEMORY FROM 1065 1065 RMEH LXI H,TPABT 042 031 040 1067 SHLD TPERRX 1068 4 JMP LOAD	001.23		360		1047		100	OP.CTL	PRIME SINGLE STEP INTERRUPT
303 172 000 1050 JMP INTXIT  1052 ** STPRTN - SINGLE STEP RET  1054 STPRTN - SINGLE STEP RET  1055 STPRTN GOU  323 360 1055 OUT  022 1055 STAX D  1057 STAX D  302 344 000 1050 JNZ  303 042 040 1050 JNZ  1064 ** RMEN - LOAD MEMORY FROM  1064 ** RMEN - LOAD MEMORY FROM  1065 KMEN  041 244 000 1066  1065 KMEN  1065 JNP  1065 KMEN  1065 KMEN  1066 KMEN  1068 ** JMP  1060	001.23		9	040	1048	SST1	STA	CTLFLG	SET NEW FLAG VALUES
1052 ** STPRIN - SINGLE STEP RET 1054 STPRIN - SINGLE STEP RET 1054 STPRIN EQU	001.24				1050		<b>E 5</b>	INTXIT	RETURN TO USER ROUTINE FOR STEP
1053 STPRTN EQU					1052	:	STPRIN	SINGLE	. A
366 920 1055 0RI CB.SSI 323 360 1056 0UT OP.CTL 022 1057 STAX 0 346 040 1059 ANI CB.NTL 302 344 000 1060 JNZ ANI CB.NTL 303 344 000 1060 JNZ ANI CB.NTL 303 344 000 1060 JNZ ANI CB.NTL 1063 ** RMEM - LOAD MEMORY FROM 1065 WE'N LXI H,TPABT 042 031 040 1067 SHLD TPERX 1068 * JMP LOAD	001.24			:	1053	STPRIN	EQU	:	:
323 360 1056 0UT OP-CTL 022 1057 5 STAX 0 346 040 1059 ANI CB.NTL 302 344 000 1060 JNZ MTR 303 042 040 1061 JNP UIVEC+3 1064 # RMEM - LOAD MEMORY FROM 1065 RMEM LXI H,TPABT 042 031 040 1067 SHLD TPERRX 1068 # JMP LOAD	001.24	:	366 020	_	1055		URI	C8.551	DISABLE SINGLE STEP INTERRUPTION
022 1058 STAX D 346 040 1059 ANI CB.MTL 302 344 000 1060 JNZ MTR 78 303 042 040 1061 JMP UIVEC+3 1064 ** RMEM - LOAD MEMORY FROM 1064 ** RMEM - LOAD MEMORY FROM 1065 RMEM LXI H,TPABT 042 031 040 1067 SHLD TPERRX JMP LOAD	001.24			_	1056		001	OP.CTL	TURN OFF SINGLE SIEP ENABLE
346 040 1059 ANI CB-NTL 302 344 000 1060 JN2 MTR 303 042 040 1061 TO	001.25				1058		STAX	ם ברבי	
303 042 040 1061 JAP UIVEC+3 1063 ++ RMEH - LOAD MEMORY FROM 1064 * 1065 WEN LXI 042 031 040 1067 SHLD TPERRX 1068 * JAP LOAD	001.25	:	•		1059		INY	CB.MTL	SEE IF IN MUNITUR ADDE
1063 ** RMEM - LOAD MEMORY FROM TAPE. 1064 * 1065 042 031 040 1066 RMEM LXI H;TPXBT SETUP ERROR EXIT 5MLD TPERRX SETUP ERROR EXIT JAPP LOAD	001.25			040	1901		JAP	UIVEC+3	TRANSFER TO USER®S ROUTINE
1064 * 1065 RMEM LXI H;TPXBT SETUP ERROR EXIT 042 031 040 1067 SHLD TPERRX SETUP ERROR EXIT 1068 * JMP LOWD		Ė			1063	:	MEN	DAD MENORY	TAP
041 244 002 1066 RMEM LXI H4TPABT 042 031 040 1067 SHLD TPERRX 1068 * JAP LONO		:			:				
	001.26	: :		4 002 1 040	: :	RHEH	LXI SHLD JAP	H, TP XBT TPERRX LOXO	ERROR EXIT

001-137 - 353 232 001 - 1129						
	001.366	272	1126 1127	.T 3A0	ALL DONE - TURN OFF TAPE REAJ ANOTHER RECORD	
	:					
		:				

		11330 11331 11332 11333 11	1131   1132   1134	1134 + 1135   1135   1135   1135   1135   1135   1145   1145   1145   1145   1145   1145   1145   1155   11
--	--	--	---	--

		٥.		20	7.4	
.~	63	1187		ORA	F	
02.117	102 104 002	1188		245	NME2	IF MORE TO GO
		1190	*	HRITE C	CHECKSUM	
:		1611				
002-122 0	052 027 040	1192		LHLD	CRCSUM	
	710	1194		באר נאר	. Q	ELICH CHECKSIN
:		1195	•	JAP	TFT	
		1197	* .	TFT - T	TFT - TURN OFF TAPE.	
		1199		STOP TH	STOP THE TAPE TRANSPORT	- 1
		1200				
		1201				
002-133 2	25 / 323 371	1202	1-1	XRA OUT	A 0P.TPC	TURN DEF TAPE
		1205	*	HORN -	HORN - MAKE NOTSE.	
		1206	•			
		1207	* 1	ENTRY		(MILLISECOND COUNT)/2
		1209		IISES	AURC	
		1210	,			
	076 144	1211	ALARM	IAN	A-20072	OUD MY AFFD
002.140		1213	HORN	PUSH	PSH	
:	076 200	1214		NV I	A,CB.SPK	TURN ON SPEAKER
	343	1215	HRNO	XTHL		SAVE (HL). (H) = COUNT
	325	1217		PUSH	0	SAVE (DE)
		1218		ХСНС	7	(D) = LOOP COUNT
002-146	041 011 040 256	1219		LXI XBA	H, CTLFLG	
	136	1221		NO.	E,A	(E) = OLD CTLFLG VALUE
		1222		MOV	H, A	TURN ON HORN
:	056 033	1223		HVI	L, #TICCNT	
:	172	1225	:	AU.	0.4	14.1 = CVC E COUNT
	902	1226		ADD		
:		1221	HRN2	CMP		WAIT REQUIRED TICCOUNTS
:	302 160 002	1228		LNE	HRN2	
	011	1229		IAM	L,#CTLFLG	
- :	163	1230		A04	<b>7</b> 9€	TURN HORN OFF
	341	1232		904		
002.171	311	1233		RET	:	

1242   1242	1240	
1241 * EXIT TO TATERER IF BK  1243 * USES TO TATERER IF BK  1244 * USES TAFER* IF BK  1244	1241	
1243 * USES A,F,H,L  1244  11244  11244  11245  11246  11246  11249  11249  11249  11249  11249  11249  11250  11250  11251  11252  11254  11252  11254  11255  11255  11255  11256  11256  11256  11257  11256  11257  11258  11258  11258  11259  11259  11259  11259  11259  11259  11259  11259  11259  11259  11259  11250  11259  112	1243 * USES A*F*H*L  1244  315 325 002 1246 CTC CALL RNP 265 027 040 1247  210 1249 CRCSUM 265 1249 DRA 267 1250 RZ  310 1251 A*H TPERR 268 1250 RZ  310 1251 * TPERR PROCESS TAPE ERR 1255 * USPLAY ERR NUMBER IN LO 1255 * TPERR PROCESS TAPE ERR 1255 * USPLAY ERR NUMBER IN LO 1257 * TPERR PROCESS TAPE ERR 1255 * USPLAY ERR NUMBER IN LO 1257 * TPERR PROCESS TAPE ERR 1256 * DISPLAY ERR NUMBER IN LO 1257 * TPERR PROCESS TAPE ERR 1256 * DISPLAY ERR NUMBER IN LO 1257 * TPERR PROCESS TAPE ERR 1256 * TPERR PROCESS TAPE ERR 1257 * TPERR PROCESS TAPE ERR 1258 * TPERR PROCESS TAPE ERR 1259 * TPERR PROCESS TAPE ERR 1250 * TPERR PROCESS TAPE ERR 1257 * TPERR PROCESS TAPE ERR 1258 * TPERR PROCESS TAPE ERR 1257 * TPERR PROCESS TAPE ERR 1258 * TPERR PROCESS TAPE ERR 1259 * TPERR PROCESS TAPE ERR 1250 * TPERR PROCESS TAPE PROCESS TAPE ERR 1259 * TPERR PROCESS TAPE PROCESS TAP	
1244  315 325 002 1246  052 027 040  1249  076 001 1250  1250  1250  1254  076 001 1251  1255  1255  1255  1256  1256  1257  1257  1258  1258  1259  1259  1259  1260  1270  1	1244  315 325 002 1246  052 027 040 1247  174 1248  175 1250 1247  175 1250 1250  177 1250 1250  178 1250 1250  179 1250 1250  170 1251 140 140  170 1255 140  170 1255 150  170 1255 150  170 1265 150  170 1265 160  170 1265 160  170 1265 160  170 1265 160  170 1265 160  170 1265 160  170 1265 160  170 1265 160  170 1265 160  170 1265 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 1270 160  170 170 160  170 170 160  170 170 160  170 170 160  170 170 160  170 170 160  170 170 160  170 170 160  170 170 160  170 170 160  170 170 160  170 170 170  170 170  170 170 170  17	
315 325 002 1246 CTC CALL RNP 052 027 040 1247 CHLD CRCSUM 265 1249 0RA L 310 1250 RZ L 076 001 1251 + MVI A,1 1252 + MVI A,1 1255 + MVI A,1 1256 + DISPLAY ERR NUMBER IN LO 1257 + FERROR NUMBER EVEN, DOI 1258 + FERROR NUMBER EVEN, DOI 1259 + FERROR NUMBER EVEN, DOI 1250 + FERROR NUMBER EVEN, DOI 1251 + FERROR NUMBER EVEN, DOI 1251 + FERROR NUMBER EVEN, DOI 1252 + FERROR NUMBER EVEN, DOI 1254 + FERROR NUMBER EVEN, DOI 1257 + FERROR NUMBER EVEN, DOI 1258 + FERROR NUMBER EVEN, DOI 1257 + FERROR NUMBER EVEN, DOI 1258 + FERROR NUMBER EVEN, DOI 1259 + FERROR NUMBER EVEN, DOI 1250 + FERROR NUMBER EVEN, DOI 1250 + FERROR NUMBER EVEN, DOI 1257 + FERROR NUMBER EVEN, DOI 1258 + FERROR NUMBER EVEN 1258 + FERROR NUMBER EVEN 1259 + FERROR NUMBER EVEN 1250 + FERROR NUMBER EVEN 125	315 325 002 1246 CTC CALL RNP 1052 027 040 1247 HOW 265 1249 DRA L 266 1249 DRA L 310 1251 RN L 310 1251 RN L 315 1251 RN L 315 133 002 1256 PRER RROR NUMBER EVEN, DO 316 1251 RROR NUMBER EVEN, DO 316 1250 PRER RROR NUMBER EVEN, DO 317 1250 PRER RROR NUMBER EVEN, DO 318 133 002 1261 PRER STA RETURN (IF PARITY 315 133 002 1265 PRER STA RETURN (IF PARITY 315 135 002 1270 RRC 316 1271 PRER RRC 317 RRC 318 126 RRC 318 127 RRC 319 127 RRC 310	
052 027 040 1247  1248  1249  1250  076 001 1251  1255  1256  1257  1256  1257  1257  1257  1259  1257  1259  1259  1259  1250  1259  1260  1261  1262  1263  1263  1264  1264  1265  1264  1267  1267  1268  1269  1269  1269  1269  1269  1269  1270	174 27 040 1247	
265 1249 0RA L 1250 1249 0RA L 1250 1250 RZ APP	265 1249 0RA L 1250 1250 RZ ORA L 1250 RZ ORA L 1255 + JMP TPERR L 1255 + JMP TPERR L 1255 + JMP TPERR L 1255 + TPERR - PROCESS TAPE ERR 1255 - TPERR - TPERR - PROCESS TAPE ERR 1255 - TPERR - TPERR - PROCESS TAPE ERR 1255 - TPERR - TPERR - PROCESS TAPE ERR 1255 - TPERR	
310 1250 RZ	310 076 001 1250 471 076 001 1251 471 1255 4	
076 001 1251 + HVI A,1  1252 + JMP TPERR  1254 + TPERR - PROCESS TAPE ERR 1255 + DISPLAY ERR NUMBER IN LO 1256 + DISPLAY ERR NUMBER IN LO 1256 + TPERR - PROCESS TAPE ERR 1255 + TFEROR NUMBER EVEN, DO 1256 + TFEROR NUMBER EVEN, DO 1256 + TFEROR NUMBER EVEN, DO 1260 + TFEROR NUMBER EVEN, DO 1271 + TFEROR NUMBER EVEN NU	076 001 1251 #WI A,1  1252 * JMP TPERR  1254 ** TPERR - PROCESS TAPE ERRI 1255 * DISPLAY ERR NUMBER IN LO 1257 * TPERR - PROCESS TAPE ERRI 1250 * TERROR NUMBER ODO, ALL 1260 * TERROR NUM TERROR NUM 1270 * TERROR NUM 1271 * TERROR NUM 1272 * REC 1274 * REC 1275 * REC 1276 * GEL  TPRIT 1277 * REC 1	
1254 ** TPERR - PROCESS TAPE ERR 1255 ** DISPLAY ERR NUMBER IN LO 1257 ** DISPLAY ERR NUMBER IN LO 1257 ** IF ERROR NUMBER EVEN, DO 1259 ** IF ERROR NUMBER EDD9, ALL 1260 ** IF ERROR NUMBER EDD9, ALL 1261 ** ENTRY (B) - PATTERN 1262 ** IF ERROR NUMBER EDD9, ALL 1263 ** IF ERROR NUMBER EDD9, ALL 1264 ** IF ERROR NUMBER EDD9, ALL 1265 ** IF ERROR NUMBER EDD9, ALL 1266 ** IF ERROR NUMBER EDD9, ALL 1267 ** IS ** RETURN (IF PARITY 1268 ** IS ** RETURN (IF PARITY 1269 ** IS ** RETURN (IF PARITY 1269 ** IS ** RETURN (IF PARITY 1260 ** IZ ** BEEP AND FLASH ERROR NUM 1274 ** BEEP AND FLASH ERROR NUM 1275 ** BEEP AND FLASH ERROR NUM 1276 ** BEEP AND FLASH ERROR NUM 1276 ** BEEP AND FLASH ERROR 1277 ** CALL FPXIT 1278 ** BEEP AND FLASH ERROR 1279 ** BEEP AND FLASH ERROR 1270 ** DIE 1271 ** BEEP AND FLASH ERROR 1271 ** DIE 1272 ** BEEP AND FLASH ERROR 1273 ** DIE 1274 ** BEEP AND FLASH ERROR 1275 ** BEEP AND FLASH ERROR 1276 ** DIE 1277 ** BEEP AND FLASH ERROR 1277 ** BEEP AND FLASH ERROR 1278 ** BEEP AND FLASH ERROR 1279 ** BEEP AND FLASH ERROR 1270 ** DIE 1271 ** BEEP AND FLASH ERROR 1271 ** BEEP AND FLASH ERROR 1271 ** BEEP AND FLASH ERROR 1272 ** BEEP AND FLASH ERROR 1274 ** BEEP AND FLASH ERROR 1275 ** BEEP AND FLASH ERROR 1276 ** BEEP AND FLASH ERROR 1277 ** BEEP AND FLASH ERRO	1254 ** TPERR - PROCESS TAPE ERR 1255 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1256 * 1261 * 1261 * 1261 * 1261 * 1261 * 1261 * 1261 * 1261 * 1262 * 1262 * 1263 * 1263 * 1263 * 1263 * 1264 * 1264 * 1264 * 1264 * 1266 * 1266 * 1266 * 1266 * 1266 * 1266 * 1270 * 1270 * 1269 * 1270	
1254 ** TPERR - PROCESS TAPE ERR 1255 * 1255 * 1256 * 1256 * 1256 * 1256 * 1257 * 1257 * 1257 * 1258 * 15 ERROR NUMBER DDD, ALL 1259 * 15 ERROR NUMBER DDD, ALL 1260 * 1270 * 1260 * 127	1254 ** TPERR - PROCESS TAPE ERR 1255 ** 1255 ** 1256 ** 1256 ** 1257 ** 1257 ** 1257 ** 1257 ** 1257 ** 1257 ** 1259 ** 15 ERROR NUMBER ODD, ALL 1259 ** 15 ERROR NUMBER ODD, ALL 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1260 ** 1270 ** 1260 ** 1270 ** 1260 ** 127	
1256 * DISPLAY ERR NUMBER IN LD 1257 * 1256 * 1 F ERROR NUMBER EVEN, DO 1259 * IF ERROR NUMBER EVEN, DO 1250 * IF ERROR NUMBER EVEN, DO 1260 * ENTRY (B) = PATTERN 1260 * ENTRY (B) = PATTERN 1261 * ENTRY (B) = PATTERN 1262 1264	1256 * DISPLAY ERR NUMBER IN LD 1257 * IF ERROR NUMBER EVEN, DO 1259 * IF ERROR NUMBER EVEN, DO 1250 * IF ERROR NUMBER EVEN, DO 1260 * ENTRY (B) = PATTERN 1263 * ENTRY (B) = PATTERN 1264 * ENTRY (B) = PATTERN 1265 * ENTRY (B) = PATTERN 1265 * ENTRY (B) = PATTERN 1265 * ENTRY (B) = PATTERN 1266 * ENTRY (B) = PATTERN 1265 * ENTRY (B) = PATTERN 1266 * ENTRY (B) = PATTERN 1267 * ENTRY (B) = PATTERN 1268 * ENTRY (B) = PATTERN 1269 * ENTRY (B) = PATTERN 1269 * ENTRY (B) = PATTERN 1269 * ENTRY (B) = PATTERN 1260 * I260 * ENTRY (B) = PATTERN 1260 * ENTRY (ENTRY	
1257 * 1568 * 1F ERROR NUMBER EVEN, DOI 1259 * 1F ERROR NUMBER EVEN, DOI 1259 * 1F ERROR NUMBER EDD9, ALL 1250 * 1560 * ENTRY (B) = PATTERN 1262 1263 1264 1765 874 89.4 89.4 1265 1265 CALL 1767 1268 * 1270 008 MI.ANI 1270 1271 1269 008 MI.ANI 1270 1271 1272 RRC 1274 RRC 1275 86EEP AND FLASH ERROR NUM 1276 1276 86EEP AND FLASH ERROR NUM 1276 1277 CALL 1771 1277 1277 1277 1277 1277 1277 12	1257 * 156 * 15 ERROR NUMBER EVEN, DOI 1259 * 15 ERROR NUMBER EDD9, ALL 1260 * 1270 * 1260 * 1270 *	
1258 * IF ERROR NUMBER EVEN, DOD 1259 * IF ERROR NUMBER GDD, ALL 1260 * ENTRY (B) = PATTERN 1263 1263 1263 1264 1264 1265 1266 1266 1266 1266 1266 1266 1266	1258 * IF ERROR NUMBER EVEN, DOD 1259 * IF ERROR NUMBER GDD, ALL 1260 * ENTRY (B) = PATTERN 1262 1263 1264 1265 1265 1265 1266 315 133 002 1266 CALL FFT 1268 * IS #, RETURN (IF PARITY 1269 008 MI.ANI 170 1271 TER3 MOV A,8 1272 RRC 336 1276 & BEEP AND FLASH ERROR NUM 1275 834 136 002 1276 CALL TPXIT 334 136 002 1277 CC ALARH 1277 1278 1279 CALL TPXIT 335 25 002 1279 TERI CC ALARH 1277 1278 1279 TERI CC ALARH 1277 1279 1270 1270 1270 1277 1277 1277 1277 1277 1277 1277 1277	
1259 * IF ERROR NUMBER DDD, ALL. 1266 * ENTRY (B) = PATTERN 1262 1263 1263 1264 1265 1265 1265 1266 1266 1266 1266 1266	1259 * IF ERROR NUMBER DDD, ALL 1266 * ENTRY (B) = PATTERN 1262	
1260	1262	
1262 1263 1263 1264 1265 1265 1265 1266 1266 1266 1266 1267 1268 1270 1270 1271 1271 1272 1273 1274 1275 1275 1276 1277 1277 1277 1277 1277 1277 1277	1262 1263 1263 1264 1265 1265 1266 1267 1268 1270 1271 1271 1272 1273 1274 1274 1275 1274 1275 1276 1276 1277 1277 1278 1277 1278 1277 1278 1277 1278 1277 1278 1277 1278 1279 1277 1277 1278 1277 1278 1279 1270	
062 024 040 1264 TPERR STA ABUSS 107 11265 HOV B.A 315 133 002 1266 CALL FFT 1268 * 15 % RETURN (IF PARITY 1269 08 MI.ANI 1270 1271 TER3 MOV A.8 1272 RRC 330 1274 RC 334 136 002 1275 * 8EEP AND FLASH ERROR NUM 1275 RC 334 136 002 1276 * 8EEP AND FLASH ERROR NUM 1277 CALARM 333 360 1270 CALARM 333 360 1271 CFI 001011118 312 215 002 1280 JE 312 215 002 1284 CALARM 313 220 002 1285 JE 303 220 002 1285 JHP TER1	062 024 040 1264 TPERR STA ABUSS 107 11265 HOV B.A 315 133 002 1266 CALL FFT 1268 * IS #, RETURN (IF PARITY 1269 DB MI.ANI 170 1271 TER3 MOV A.B 1272 RRC 330 1274 RC 334 136 002 1276 * BEEF AND FLASH ERROR NUM 1275 BEEF AND FLASH ERROR NUM 1277 CALARH 315 252 002 1278 TER1 CC ALARH 315 252 002 1278 TER1 CC ALARH 315 250 002 1278 TER1 CC ALARH 316 002 1278 TER1 CC ALARH 317 252 002 1280 CAL TPAIT 318 252 002 1280 CIN 1118 316 052 1282 US 051 1281	
002 024 040 1264 17ekk 51A A6035 315 133 002 1266 CALL FFT 1268 * 15 % RETURN (1F PARITY 1269 08 MI.ANI 1270 08 MI.ANI 1271 08 MI.ANI 1272 RRC 330 1274 RC 334 136 002 1275 * 8EEP AND FLASH ERROR NUM 1277 CALL TPXIT 335 250 002 1278 CALL TPXIT 335 360 1280 CALL TPXIT 336 057 1281 CPI 00101118 312 215 002 1282 JE 10A TICCNT+1 037 057 1284 RAR	002 024 040 1264 17ekk 51A A6055 315 133 002 1266 CALL FFT 1268	
315 133 002 1266 CALL FFT 1268 * 15 *, RETURN (IF PARITY 1269 08 MI.ANI 170 1271 TER3 MOV A,8 330 1273 RRC 334 136 002 1275 * BEEP AND FLASH ERROR NUM 1275 8EP AND FLASH ERROR NUM 1277 CALARM 333 360 1279 CAL TPXIT 313 360 1280 CPI 001011118 312 215 002 1281 CPI 001011118 312 215 002 1281 CPI 001011118 312 215 002 1281 CPI 001011118 312 215 002 1283 LDA TICCNT+1 037 220 002 1285 JMP TER1	315 133 002 1266 CALL FFT 1267 1267 15 #, RETURN (IF PARITY 1269 08 MI.ANI 170 1271 TER3 MOV A.8 1272 RRC 330 1274 RC 334 136 002 1275 # BEEF AND FLASH ERROR NUM 1277 25 002 1279 CALL TPXIT 334 136 002 1279 CALL TPXIT 335 350 1280 IN IP.PAD 315 215 002 1282 UE CAL 315 215 002 1282 UE TER3 316 050 1282 UE TER3 317 218 1281 CAL 318 1281 001011118	
1266 * 15 #, RETURN (IF PARITY 1268 * 15 #, RETURN (IF PARITY 1269 08 MI.ANI 1270 08 MI.ANI 1271 TER3 MOV A:8 RC 1272 RC 1274 RC 1275 * 8EEP AND FLASH ERROR NUM 1275 * 8EEP AND FLASH ERROR NUM 1277 CAL TPXIT 315 252 002 1279 CAL TPXIT 315 252 002 1279 CAL TPXIT 315 252 002 1279 CAL TPXIT 315 252 002 1280 CPI 00101118 0010111118 001011118 001011118 001011118 001011118 001011118 001011118 001011118 001011118 001011	1266 * 15 #, RETURN (IF PARITY 1269 08 MI.ANI 170 1271 TER3 MOV A.8 1270 017 1271 TER3 MOV A.8 1277 RC	
1268 * 15 #, RETURN (IF PARITY 1269	1268 * 15 #, RETURN (IF PARITY 1269	
346 1270 0B MI.ANI 170 1271 TER3 MOV A,8 017 1272 RRC 330 1274 RC 1275 RC 334 136 002 1275 BEEP AND FLASH ERROR NUM 1277 CC ALARM 334 136 002 1279 CALL TPXIT 333 360 1280 CPI 001011118 312 215 002 1281 CPI 001011118 312 215 002 1281 JE TER3 072 034 040 1283 LDA TICCNT+1 037 1281 RAR	346 1270 08 MI.ANI 170 1271 TER3 MOV A,8 017 1272 RRC 330 1274 RC 1275 # BEEP AND FLASH ERROR NUM 1275 # BEEP AND FLASH ERROR NUM 1277 CC ALARM 315 252 002 1279 TER1 CC ALARM 315 252 002 1279 CALL TPXIT 334 136 057 1281 CPI TEXIT 312 215 002 1280 DF TEXIT	
170 1271 TER3 MOV A.8  1272 RRC 330 1274 RC 1275 RC 334 136 002 1276 BEEP AND FLASH ERROR NUH 1277 CC ALL TPXIT 333 360 1280 IN IP-PAD 376 057 1281 CPI 001011118 312 215 002 1282 JE 376 057 1281 CPI 001011118 037 057 1281 JE TER3 072 034 040 1283 LDA TICCNT+1 037 1285 JMP TER1	170 1271 TER3 MOV A,8  1272 RRC 330 1274 RC 334 136 002 1275 BEEP AND FLASH ERROR NUM 1277 CALL TPXIT 315 252 002 1279 TER1 CC ALARM 315 252 002 1279 CALL TPXIT 315 250 1280 IN IP,PAD 316 057 1281 JE TER3 072 034 040 1283 LDA TICCNT+1	r CLEAR
017 1272 RRC 330 1274 RC 1276 RC 1275 BEEP AND FLASH ERROR NUH 1277 CAL TARM 315 252 002 1279 CAL TARM 315 252 002 1280 IN IP-PAD 316 057 1281 CPI 001011118 312 215 002 1282 JE TER3 072 034 040 1283 LDA TICCNT+1 037 1284 RAR	017 1273 RRC 330 1274 RC 1275 RC 1275 BEEP AND FLASH ERROR NUM 1277 CALL TPAIT 313 360 1279 CALL TPAIT 313 360 1280 IN IP-PAD 312 215 002 1282 JE TER3 312 215 002 1282 JE TER3 072 034 040 1283 LDA TICCNT+1	
330 1274 RC 1275 # BEEP AND FLASH ERROR NUM 1276 # BEEP AND FLASH ERROR NUM 1277 CC ALARM 315 252 002 1279 CALL TPXIT 313 360 1280 IN IP-PAD 376 057 1281 CPI 001011118 312 215 002 1282 JE TEK3 072 034 040 1283 LDA TICCNT+1 037 1284 RAR	330 1274 RC 1275 BEEP AND FLASH ERROR NUM 1277 CC ALARM 315 252 002 1279 CALL TPXIT 313 360 1280 IN IP-PAD 316 057 1281 CP 10011118 312 215 002 1283 LDA TICCNT+1	
1275	1275	
1276 * BEEP AND FLASH ERROR NUM 1277 1277 CC ALARM 315 252 002 1278 TERI CC ALARM 313 360 1280 IN IP.PAD 376 057 1281 CPI 001011118 312 215 002 1282 JE TER3 072 034 040 1283 LDA TICCNT+1 037 1285 JMP TER1	1276 * BEEP AND FLASH ERROR NUM 1277 334 136 002 1278 TERI CC ALARH 315 252 002 1279 CALL TPXIT 333 360 1280 IN IP-PAD 376 057 1281 O01011118 072 034 040 1283 LDA TICCNT+1	
334 136 002 1278 TERI CC ALARM 315 252 002 1279 CALL TPXIT 333 360 1280 IN IP.PAD 376 057 1281 CPI 001011118 312 215 002 1282 JE TER3 072 034 040 1283 LDA TICCNT+1 037 1285 JMP TERI	334 136 002 1276 TER1 CC ALARM 315 252 002 1279 CALL TPXIT 333 360 1260 IN IP-PAD 376 057 1281 CPI 001011118 312 215 002 1282 JE TER3 072 034 040 1283 LDA TICCNT+1	
315 252 002 1279 CALL TPXIT 333 360 1280 IN IP-PAD 312 255 002 1282 JF TER3 072 034 040 1283 LDA TICCNT+1 037 1285 JMP TER1	315 252 002 1279 CALL TPXIT 333 360 1280 IN IP-PAD 316 057 1281 CPI 001011118 316 215 002 1282 JE TEX3 317 215 040 1283 LDA TICCNT+1	
333 360 1280 IN IP.PAD 376 057 1281 CPI 001011118 312 215 002 1282 JE TEK3 072 034 040 1283 LDA TICCNT+1 037 1284 RAR	333 360 1280 IN IP.PAD 376 057 1281 CPI 001011118 312 215 002 1282 JE TEK3 072 034 040 1283 LDA TICCNT+1	
376 057 1281 CPI 001011118 312 215 002 1282 JE TEK3 037 034 040 1283 LDA TICCNT+1 037 1284 RAR 303 220 002 1285 JMP TER1	376 057 1281 CPI 001011118 312 215 002 1282 JE TER3 072 034 040 1283 LDA TICCNT+1	
312 213 002 1282 JF 1EK3 072 034 040 1283 LDA TICCNT+1 037 1284 RAR 303 220 002 1285 JMP TER1	312 213 002 1282 35 15K3 072 034 040 1283 LDA TICCNT+1	
037 1284 RAR 303 220 002 1285 JMP TER1		
303 220 002 1285 JMP TER1	037 1284 RAR	
	303 220 002 1285 JMP TER1	

1330 # USES A FED F H - 1
---------------------------

25 25 25 25 25 25 25 25 25 25 25 25 25 2	277 315 331 002 1337 SRS2 CALL F   275 024	HAVE SYN  HAVE SYN  NOT SIX - START DVER  SEE IF ENDUCH SYN CHARACTERS  NOT ENDUCH  CLEAR CRC-16  READ COUNT  READ COUNT  BYTES FROM THE INPUT DEVICE.
275 376 026 1339 277 312 271 002 1340 302 376 002 1344 304 302 265 002 1344 311 272 050 002 1346 320 315 325 002 1348 321 327 040 1347 320 315 325 002 1348 321 124 137 1359 1356 4 1356 4 1356 4 1356 4 1357 4 1358 4 1358 4 1359 4 1359 4 1360 137 1360 137 1360 1360 1360 1360 1360 1360 1360 1360 1360 1360 1360 1360	275 376 026 1339 CPI 277 312 271 002 1340 302 376 002 1341 304 302 265 002 1343 307 076 012 1344 MVI 311 272 1349 MVI 323 124 137 1359 CALL 324 137 1359 CALL 325 315 325 002 1348 324 137 1355 CALL 335	HAVE SYN  NOT STX - START OVER  SEE IF ENOUGH SYN CHARACTERS  NOT ENOUGH  CLEAR CRC-16  READ LEADER  READ COUNT  BYTES FROM THE INPUT DEVICE.
277 312 271 002 1340 302 376 002 1341 304 302 265 002 1345 311 272 1345 312 322 265 002 1346 313 32 265 002 1346 314 37 134 32 124 137 1355 4 1354 4 32 137 1355 4 1355 4 1356 4 1357 4 1358 4 1359 4 1360 137 1365 1360 1364 4 1364 4 1366 4	277 312 271 002 1340 302 376 002 1343 304 302 265 002 1343 311 272 1345 312 272 265 002 1346 313 222 265 002 1346 313 222 265 002 1349 320 315 325 002 1349 320 315 325 002 1349 321 137 1353 ** RNP READ 1354 ** RNP READ 1355 ** RNP READ 1355 ** RNP READ 1356 ** RNP READ 1367 ** RNB - RE 1368 ** RNB READ 1369 ** LNP 1371 ** EXIT 1372 ** EXIT 1372 ** EXIT 1372 ** EXIT 1373 ** USES	HAVE SYN  NOT SIX - START OVER  SEE IF ENOUGH SYN CHARACTERS  NOT ENOUGH  CLEAR CRC-16  READ COUNT  READ COUNT  BYTES FROM THE INPUT DEVICE.
304 302 265 002 1342  310 776 012 1343  311 272 1346  312 322 265 002 1346  315 042 027 040 1347  320 124 325 002 1348  324 137 1355 4  1354 4  1355 4  1356 4  1356 4  1356 4  1357 4  1358 4  1368 4  1369 4  1369 4	304 302 265 002 1342 JNE  307 076 012 1346 312 322 265 002 1346 315 042 027 040 1347 320 315 325 002 1348 321 137 1350 4 HOV 324 137 1351 4 JMP  1355 4 RNP READ 1355 4 RNP READ 1356 4 EXIT 1356 4 EXIT 1351 4 JMP  1362 315 331 002 1362 1363 4 RNB READ 1363 147 1363 4 HOV 1364 4 RNB - RE 1365 4 EXIT 1366 4 EXIT 1366 4 RNB - RE 1366 4 RNB READ 1369 4 HOV 1371 4 EXIT 1371 4 EXIT 1372 4 EXIT 1372 4 EXIT	NOT STX - START OVER  SEE IF ENOUGH SYN CHARACTERS  NOT ENOUGH CLEAR CRC-16  READ COUNT  READ COUNT  BYTES FROM THE INPUT DEVICE.
307 076 012 1344 311 272 1345 312 322 265 002 1346 320 315 325 002 1348 321 124 137 1359 1354 137 1359 1355 1 1357 1 1357 1 1357 1 1357 1 1358 1 1358 1 1359 1 1350 1 1364 1 1364 1 1364 1 1366 1	307 076 012 1344 HVI 311 272 1345 312 272 060 1346 312 022 265 002 1346 320 315 325 002 1349 320 315 325 002 1349 324 137 1350 HOV 1351 ** RNP READ 1355 ** RNP READ 1355 ** RNP READ 1356 ** RNP READ 1357 ** EXIT 1358 ** RNP - RE 1356 ** RNP - RE 1356 ** RNP - RE 1357 ** EXIT 1358 ** RNP - RE 1359 ** USES 1360 ** RNB - RE 1366 ** RNB READ 1369 ** THE CHEC 1369 ** THE CHEC 1369 ** THE CHEC 1370 ** EXIT 1371 ** EXIT 1372 ** EXIT 1373 ** USES	SEE IF ENDUGH SYN CHARACTERS NOT ENDUGH CLEAR CRC-16 READ LEADER READ COUNT BYTES FROM THE INPUT DEVICE.
311	311 272 1345 CMP 312 322 265 002 1346 320 315 325 002 1346 323 124 137 1350 MOV 324 137 1351 ** MOV 325 137 1351 ** MOV 326 137 ** MOV 327 040 1347 MOV 328 137 ** RNP - RE 1354 ** RNP READ 1355 ** RNP READ 1355 ** RNP READ 1356 ** RNP READ 1356 ** RNP READ 1357 ** EXIT 1364 ** RNB READ 1364 ** RNB READ 1365 ** RNB READ 1366 ** RNB READ 1369 ** THE CHEC 1370 ** EXIT 1370 ** EXIT 1370 ** EXIT 1371 ** EXIT 1372 ** EXIT 1372 ** EXIT 1373 ** USES	SEE IF ENOUGH SYN CHARACTERS NOT ENOUGH CLEAR CRC-16 READ LEADER READ COUNT READ COUNT BYTES FROM THE INPUT DEVICE.
325 315 331 002 1369 324 137 135 042 027 040 1347 324 137 135 02 1348 324 137 135 135 135 135 135 135 135 135 135 135	325 315 320 002 1347 SHLO 320 315 325 002 1348 CALL 321 124 137 1350 #00V 1351 ** RNP - RE 1355 ** RNP READ 1355 ** RNP READ 1356 ** RNP READ 1357 ** EXIT 1358 ** RNP READ 1364 ** RNB READ 1364 ** RNB READ 1369 ** THE CHEC 1370 ** EXIT 1370 ** EXIT 1370 ** EXIT 1371 ** EXIT	READ COUNT  READ COUNT  BYTES FROM THE INPUT DEVICE.
320 315 325 002 1348 323 124 137 1350 1351 ** 1353 *** 1354 ** 1355 ** 1355 ** 1356 ** 1358 ** 1358 ** 1358 ** 1358 ** 1358 ** 1358 ** 1358 ** 1358 ** 1358 ** 1358 ** 1364 ** 1366 ** 1366 ** 1366 ** 1366 ** 1366 ** 1366 **	320 315 325 002 1348 CALL 323 124 137 1359 ** RNP - RE 1355 ** RNP - RE 1355 ** RNP - RE 1355 ** RNP READ 1356 ** EXIT 1358 ** RNP READ 1359 ** USES 1361 1361 1364 ** RNB - RE 1364 ** RNB - RE 1364 ** RNB READ 1369 ** THE CHEC 1369 ** THE CHEC 1370 ** EXIT 1370 ** EXIT 1370 ** EXIT 1370 ** EXIT 1371 ** EXIT 1372 ** EXIT	READ COUNT  BYTES FROM THE INPUT DEVICE.
1354 137 1350 • 1350 • 1350 • 1351 • • 1350 • 1355 • 1356 • 1356 • 1356 • 1356 • 1356 • 1356 • 1356 • 1366 • 1366 • 1366 • 1366 • 1366 • 1366 • 1366 • 1366 • 1366 • 1366 • 13	1354 127 1350 HOV 1354 HOV 1351	READ COUNT  BYTES FROM THE INPUT DEVICE.
1353 ** 1354 ** 1355 * 1355 * 1355 * 1356 * 1357 * 1358 * 1358 * 1358 * 1360 ** 1361 ** 1362 ** 1364 ** 1366 ** 1366 ** 1366 ** 1366 ** 1366 ** 1366 ** 1366 **	L 1353 ** RNP - RE L 1354 * RNP - RE L 1355 * RNP READ L 1356 * ENTRY L 1358 * ENTRY L 1358 * EXIT L 1359 * USES L 1361 CALL L 2.330 L 147 L 1361 L 1366 ** RNB - RE L 1367 * THE CHEC L 1368 * THE CHEC L 1369 * THE CHEC L 1372 * EXIT L 1372 * EXIT	BYTES FROM THE INPUT DEVICE.
1353 ** 1354 * 1355 * 1356 * 1356 * 1357 * 1358 * 1358 * 1359 * 1359 * 1359 * 1359 * 1350 147 1363 * 1364 * 1364 * 1366 ** 1368 * 1369 *	1353 ** RNP - RE 1354 * RNP READ 1355 * RNP READ 1356 * ENITY 1358 * ENITY 1358 * ENITY 1362 1362 RNP CALL 2,330 147 1364 * RNB - RE 1366 ** RNB - RE 1366 ** RNB READ 1369 * THE CHEC 1369 * THE CHEC 1370 * ENITRY 1371 * ENITRY 1372 * ENITRY 1372 * ENITRY 1372 * ENITRY	BYTES FROM THE INPUT DEVICE.
1355 + 1356 + 1356 + 1356 + 1357 + 1357 + 1358 + 1359 + 1360 + 1361 + 1361 + 1361 + 1364 + 1366 + 13	1355	BYTES FROM THE INPUT DEVICE.
1356 * 1356 * 1356 * 1359 * 1359 * 1359 * 1350	1356	
1358 * 1359 * 1359 * 1350 1360 1361 2,325 315 331 002 1362 RNP 1364 * 1364 * 1364 * 1364 * 1366 ** 1368 *	1359	
1359 * 1359 * 1360 1360 1361 2,325 315 331 002 1362 RNP 1364 * 1364 * 1366 ** 1366 ** 1366 **	1359 * USES 1360 1361 1361 2.325 315 331 002 1362 1363 * HOV 1364 * JMP 1366 * RNB - RE 1366 * RNB READ 1369 * THE CHEC 1370 * ENTRY 1371 * ENTRY 1372 * EXIT	¥I'
2,325 315 331 002 1362 RNP 2,330 147 1363 1364 * 1366 ** 1366 ** 1366 ** 1366 ** 1366 **	2.325 315 331 002 1362 RNP CALL 2.330 147 1364 \$ MP 1364 \$ MP 1366 \$ RNB - RE 1367 \$ RNB READ 1369 \$ THE CHEC 1370 \$ ENTRY 1372 \$ ENTRY 1372 \$ EXTR	
2,330 147 1364 • 1364 • 1366 •	2.330 147 11363 HOV 1364 + JNP 1364 + JNP 1365 + RNB - RE 1367 + RNB READ 1368 + THE CHEC 1370 + ENTRY 1372 + ENTRY 1372 + EXIT	
1364 * 1366 * 1366 * 1366 * 1366 * 1368 * 1369 * 13	1364 + JNP 1366 ++ RNB - RE 1367 + RNB READ 1369 + THE CHEC 1370 + ENTRY 1372 + EXIT 1372 + EXIT 1373 + USES	NEX
366 * * * * * * * * * * * * * * * * * *	366 ++ RNB - RE 368 + THE CHEC 369 + THE CHEC 370 + ENTRY 371 + ENTRY 372 + EXIT 373 + USES	READ NEXT BYTE
366 ** 367 * 368 *	366 ++ RNB - RE 367 + RNB READ 369 + THE CHEC 370 + ENTRY 371 + EXT 373 + USES	
368 <b>*</b> 369 <b>*</b>	368 + RNB READ 369 + THE CHEC 370 + ENTRY 371 + EXIT 373 + USES	
	370 * ENTRY 371 * EXIT 372 * EXIT 373 * USES	SLE BYTE FROM THE INPUT DEVICE.
370 *	371 * ENTRY 372 * EXIT 373 * USES	
371 * ENTRY	373 * USES	
373 * USES		
	374	
31 076 064 1376 RNB MVI	331 076 064 1376 RNB MVI	(+UCI.RE TURN ON READER FOR NEXT BYTE
335 315 252 002 1378 RNB1 CALL	335 315 252 002 1378 RNB1 CALL	CHECK FOR * READ STATUS
340 346 002 1379 ANI	340 346 002 1379 ANI	
342 312 335 002 1380 JZ RNB1 345 333 370 1381 IN IP_TPD	342 312 335 002 1380 JZ 345 333 370 1381 IN	IF NOT READY
1382 *	1382 + JAP	CHECKSUM

1386 * 1387 * 1388 * 1389 * 1480 * 14		1000	•	ر د د	COMPUTE CRC-16	9
1389   1389   1389   1389   1389   1389   1389   1389   1389   1389   1389   1389   1389   1389   1389   1391   1391   1391   1391   1391   1391   1391   1391   1392   1393   1393   1394   1393   1394   1395   1394   1395   1394   1395   1394   1395   1395   1396		1386	* *		UTES A CRC	6 CHECKSUN FROM THE
1390   1390		1368	* *	(x + 1	* (XA15 +	•
1391 * 1 CHECKSOMED DIA SEUDNEC CREATER DE STORMED 1392 * 1393 * 1394 * 1395 * 1395 * 1395 * 1395 * 1395 * 1395 * 1395 * 1395 * 1395 * 1395 * 1395 * 1395 * 1395 * 1396 *		1390		SINCE	THE CHECKSUM	:
1393 + CHECKSUM THROUGH CRC. THE RESULTANT CHECKSUM SHOULD BE 1395 + ENTRY (CRCSUM) = CUARENT CHECKSUM SHOULD BE 1395 + EXIT (CRCSUM) UPDATED 1399 + EXIT (CRCSUM) UPDATED 1399 + EXIT (CRCSUM) UPCAMGED.  1399 + EXIT (CRCSUM) UPCATED 1399 + EXIT (CRCSUM) UPCAMGED.  1399 + EXIT (CRCSUM) UPCATED 1400		1391 1392	* *	A CHE(	TESTATED DATA STATES	ō
1395 # ENTRY (CRCSUH) = CURRENT CHECKSU 1396 # EXIT (CRCSUH) UPDATED 1398 # EXIT (CRCSUH) UPDATED 1398 # EXIT (CRCSUH) UPDATED 1399 # USES F 1400		1393	* *	CHECK	UM THROUGH CR	ш.
1396 * (A) = BYTE  1397 * EXIT (CRCSUM) UNCHANGED.  1399 * USES F  1400  1400  1400  1401  1401  1402  1402  1402  1402  1402  1403  1404  1404  1406  1407  1406  1407  1407  1407  1408  1408  1408  1408  1409  1409  1409  1409  1411  1411  1411  1412  1414  1414  1414  1414  1415  1416  1419		1395		ENTRY	(CRCSUM) =	-
1399		1396	*		(A) = 8YTE	
1399		139	• •	EXII	CRCSON) UP	.0A1ED
47 305 1402 CRC PUSH 8 SAVE (8C) 50 006 010 1403 MV1 8.8 (81 = 817 53 25 027 040 1403 MV1 8.8 (81 = 817 55 007 1405 CRC1 RLC CRCSUH CRC			*	USES	u.	
47 305 1402 CRC PUSH 8 SAVE (8C) 50 006 010 1403 HVI 8.8 (8) - 61T 52 345 1404 1405 HLC CRCSUM 53 052 027 040 1405 HLC CRCSUM 54 027 1406 CRCI RLC 55 17 1409 H0V C,A 61 207 1410 H0V A,H 64 027 1412 RAL 65 147 1413 H0V H,A 66 027 1415 RRC 67 027 1416 RRC 67 027 1416 RRC 67 027 1416 RRC 67 027 1418 H0V A,H 68 027 1418 H0V A,H 69 027 1418 H0V A,H 69 027 1418 H0V A,H 60 027 1419 RRI 60 027 1419 RRI 60 027 1420 H0V H,A 60 027 1421 RRI 60 027 RRI 60 0		140				
50 006 010 1403	347 305	140	- :	PUSH		AVE (BC)
353 052 027 040 1405 CRC1 RLC CRC5UH 356 007 117 1406 CRC1 RLC C.A 361 117 1408 MDV A.L 362 1207 1409 MDV L.A 362 157 1410 MOV A.H 364 027 1411 RAL 364 027 1414 RAL 365 027 1414 RAL 367 251 1415 RAL 367 251 1416 RAL 377 356 200 1419 RRC 378 356 200 1419 RRC 377 177 1420 MDV A.H 378 356 005 1421 MDV A.H 377 356 005 1421 SOU	350 006	140		IAN	•	8) - BIT
56 007 1406 CRC1 RLC 57 117 1407 MOV C,A 60 175 1408 MOV A,L 61 207 1409 ADD A 62 157 1410 MOV L,A 64 027 1412 RAL 65 147 1413 RAL 66 027 1415 RAL 67 027 1415 RAL 67 027 1416 RAL 68 005 1419 ARI 75 356 005 1419 ARI 76 356 005 1419 ARI 77 147 148 MOV A,H 78 005 1420 MOV A,H 79 01 1421 ARI 70 155 1422 ARI 70 155 1421 ARI 71 1421 ARI 72 157 1421 ARI 73 157 1421 ARI 74 174 ARI 75 156 005 1419 ARI 75 156 005 1420 ARI 75 157 1421 ARI 76 175 1421 ARI 77 1421 ARI 78 157 1421 ARI 78	352 343	:			TIN COL	
57 117 1407 MUV C,A (C) = 60 175 1408 MUV A,L 61 207 1409 MDD A 62 1207 1409 MDV L,A 63 174 1410 MUV A,H 64 027 1412 RAL 65 147 1413 RAL 66 027 1415 RAL 67 251 1416 RRC 68 1419 RRC 69 1419 RRI 69 175 1420 RRU 60 1420 R	56 007					
6.0 175 1408 MOV A,L 6.1 207 1409 ADD A 6.2 157 1410 MOV L,A 6.4 027 1412 MOV H,A 6.4 027 1412 RAL 6.5 147 1413 RAL 6.5 027 1415 RAL 6.7 251 1415 RRC 6.7 251 1415 RRC 6.7 174 1416 RRC 6.7 174 1418 RRC 6.7 175 147 1418 RRC 6.7 175 147 1418 RRC 6.7	. 2	140			C. A	- 0
61 207 1409 ADD A 62 157 1410 MOV L,A 64 027 1411 MOV A,H 65 027 1412 RAL 65 027 1414 RAL 66 027 1415 RAL 67 251 1415 RAC 67 125 004 003 1417 JNC CRC2 71 322 004 003 1417 JNC CRC2 71 325 004 1419 KRI 2009 72 175 356 005 1419 KRI 2009 73 157 142 MOV H,A 76 005 142 KRI 2009 74 174 1418 KRI 2009 75 142 KRI 2009 76 1419 KRI 2009 77 142 KRI 2009	9	140	œ	AOM.	A . L	
6.3 157 1410 MOV L.A 6.3 174 1411 MOV A.H 6.4 0.27 1412 RAL 6.5 147 1414 RAL 6.7 251 1415 RRC 7.1 322 004 003 1415 RRC 7.1 322 004 003 1417 JNC CRC2 7.1 322 004 1419 KRI 2009 7.2 175 1419 KRI 2009 7.3 150 005 1419 KRI 2009 7.4 174 1419 KRI 2009 7.5 142 KRI 2009 7.6 1420 KRI 2009 7.7 147 1420 KRI 2009 7.8 1420 KRI 2009 7.9 1420 KRI 2009 7.9 1420 KRI 2009	19	140	6	00V	⋖	
65 177 1411 NUV AND	7	141	•	AOK.	٨, ١	
65 147 1413 HDV H,A 66 027 1414 RAL 67 251 1415 RRC 70 017 1416 RRC 71 322 004 003 1417 JNC CRC2 74 174 174 1418 HDV A,H 75 356 200 1419 KRI 2009 77 147 1419 KRI 2009 77 147 1421 KRI 2009 78 156 005 1421 KRI 50	n 4		7 2	A 8	E. 4	
66 027 1414 RAL C   67 251 1415	: 59	141		VON	H,A	
67 251 1415 KRA C 70 017 1416 RRC CRC2 IF NOT 71 322 004 003 1417 JNC CRC2 IF NOT 74 174 174 1418 RDV A;H 75 356 200 1419 KRI 2009 77 147 1420 RDV H;A 190 175 1421 RUV A;L 01 356 005 1421 KRI 59	99	141		RAL		
71 322 004 003 1417 JNC CRC2 IF NOT 74 174 174 MOV A,H 75 356 200 1419 XRI 2000 175 1420 MOV A,L 00 175 1421 MUV A,L 01 356 005 1421 XRI 59 005 1423 MOV L,A	29	141	<b>د</b> د	X X A	ပ	
74 174 1418 MDV 75 356 200 1419 XRI 77 147 1420 HDV 100 175 1421 HDV 101 356 005 1422 XRI 1423 HOV	71 322 004 0	33	7	227	CRC2	NOT
75 356 200 1419 XRI 17 147 1420 HOV 100 175 1421 HUV 101 356 005 1422 XRI 103 157 1423 HOV	74 174	:	80	YON	A,H	
100 175 1421 MUV 100 175 1421 MUV 1422 XRI 1923 157 1423 MOV	75 356	141	<b>o</b> c	XRI	2000	
01 356 005 1422 XRI 03 157 1423 HOV	:0	142		AOR	A.L	
103 157 1423 MOV	01 356 00	142	7	XRI	50	
	03 157	145			L, A	
104 171 1424 CKC2 MUV	•	142	:	:	٠ • ۲	
105 005 1425 007 1NZ	105 005		ي ر	727	CRCI	AORE TO
111 042 027 040 1427 SHLD CRCSUM	111 042 02	:	7	SHLD	CRCSUM	:
114 341 1428 POP H	114 341		80	POP	I	RESTORE (HL)
115 301 1429 POP B RESTORE	115	145	6	90 <b>9</b>	80	RESTORE (BC)
116 311 1430 KEI	9.	14	2	A		EAII

			1432	:	I den	MNP - WRITE NEXT PAIR.	
			1433				
			1434	* *	- X	MAL MALLEY THE NEXT INC	BTIES TO THE CASSELLE DKIVE.
			1436	*	ENTRY	(H,L) = BYTES	
			1437	* •	EXIT	WRITTEN.	
			1430	•	USES	A)F	
			1440				
003.017			1441	dNA	¥0×	М, М	
003-020	315 02	003	1442		CALL	9Z	
003.023	5		144	•	A # .	7.1 1.7.8	ARITE NEXT BYTE
			1447	: .	0	1 Y Y I C	
			1448	* *	WNB WRITES	THE NEXT	BYTE TO THE CASSETTE TAPE.
	:		1450	• •	FNTRY	(A) = RYTE	
			1451		EXIT	NONE.	
			1452	*	USES	u.	
			1454				
003.024	365		1455	ENB	PUSH	PSM	
003.025	315	252 002	1456	MNB1	CALL	TPXIT	CHECK FOR *, READ STATUS
003-032	312	003	1458		777	ENB1	IF MORE TO GO
003.035	076		1459		IVI	A, UCI, ER+UCI,TE	ENABLE TRANSMITTER
003.037	323		1460		00 P	OP.TPC PSH	TURN ON TAPE
003.042	323	370 347	1462		00.1 P P	OP.TPD CRC	OUTPUT DATA COMPUTE CRC
	:						
:							
	:						

1,000   1,00			1467	* *	LRA -	LOCATE REGISTER AL	ADDRESS.
1470 • EXIT (A.) - STORGE ADDRESS  1471 • 1472 • 1056  1472 • 1056  1473 • 1056  1474 • 1056  1475 • 1057  1475 • 1056  1476 • 1057  1475 • 1056  1476 • 1076  1476 • 1076  1476 • 1076  1476 • 1076  1477 • 1076  1476 • 1076  1476 • 1076  1476 • 1076  1477 • 1076  1477 • 1076  1478 • 1076  1478 • 1076  1478 • 1076  1478 • 1076  1478 • 1076  1478 • 1076  1478 • 1077  1478 • 1078  14			1469	• •	ENTRY		
1472 • USES (0FE) = (0ATE) = (0ATE) = (1474)			1470	* •	EXIT	٠.	INDEX
1473 * USES A.O.E.H.L.F.  1474				• •			AUUKESS
1477 1477 1477 1477 1477 1477 1477 1477			1473	*	USES		
1.052 137 005 040 1477 LAA LAA REGI 1.052 137 005 040 1477 LAA LAA REGI 1.053 025 030 040 1477 LAA HOV E,A 1.055 025 035 040 1480 LHV0 LHV0 E,A 1.055 025 035 040 1480 LHV0 LHV0 E,A 1.056 031 1481 LAB RET 1.056 031 1482 ENTRY (H,L) - ADDRESS. 1.056 031 1492 LAA LAA LAA LAA LAA LAA LAA LAA LAA LA							
1.052 1.05 0.05 0.00 1.179 KA HOV E.A. 1.052 1.052 0.05 0.05 0.05 0.05 0.05 0.05 0.05			-	:			
1,052 026 000 1479 1479 1470 1470 1470 1470 1470 1470 1470 1470	3.04	72 005 04	- ·	LR.	LDA	REGI	
1.055 032 040 1460	3.05	26 0		7	N N	0.0	
3.062 315 066 003 1495 ** 104 - INPUT OCTAL ADDRESS.  1.894 ** 104 - INPUT OCTAL ADDRESS.  1.895 ** EXIT TO *REF*! IF ERROR.  1.895 ** EXIT TO *REF*! IF ERROR.  1.895 ** USES A.D.E.H.L.F.  3.062 315 066 003 1491  3.065 053 1493 ** 108 - INPUT OCTAL BYTE.  1.495 ** RAD ONE OCTAL BYTE FROM THE KEYSET.  1.495 ** EXIT TO *READ THE KEYSET.  1.495 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.496 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.497 ** EXIT TO *READ THE KEYSET.  1.498 ** EXIT TO *READ THE KEYSET.  1.499 ** EXIT TO *READ THE KEYSET.  1.490 ** EXIT TO *READ T	3.05	52 035 04	-		LHLD	REGPTR	
1466 + 10A - IMPUT OCTAL ADDRESS.  1466 + ENTRY (H-L) - ADDRESS OF RECEPTION DOUBLE BYTE  1467 - EXIT TO *RET*-1 IF OK, VALUE IN NEHORY.  1488 + USES A.D.*E*+1.FF  1489 + USES A.D.*E*+1.FF  1490 - LAPE	3.06	311	1482		RET	2	יייי - ייייי - ייייי - ייייי - ייייי - ייייי - יייייי
1486				*		OCTAL	525•
1488				* *	YOLNU	SSECTA - CLAM	
1488 * 10 *RET**1 IF OK, VALUE II 1489 * USES A,D,E,H,L,F 1489 * USES A,D,E,H,L,F 1490  065 053 1492 10A CALL 10B INPUT BYTE  1495 * READ ONE OCTAL BYTE.  1496 * READ ONE OCTAL BYTE.  1496 * READ ONE OCTAL BYTE.  1499 * READ ONE OCTAL BYTE.  1500 * NOT SET IF FIRST DIGIT IN  1501 * NOT SET IF FIRST DIGIT IN  1502 * NOT SET IF FIRST DIGIT IN  1504 * READ ONE OCTAL BYTE.  1504 * READ ONE OCTAL BYTE.  1505 * NOT SET IF ONE OCTAL BYTE.  1506 * NOT SET IF ONE OCTAL BYTE.  1507 * NOT SET IF ONE OCTAL BYTE.  1508 * NOT SET IF ONE OCTAL BYTE.  1509 * NOT SET IF ONE OCTAL BYTE.  1509 * NOT SET IF ONE OCTAL BYTE.  1500 * NOT SET IF ONE OCTAL BYTE.  1600 * NOT SET IF ONE OC			. •	•	EXIT	TO *RET* IF ERR	2
1489 * USES A,D,E,H,L,F  1490 1491 1490 1491 1490 1491 1490 1492 10A 065 053 1492 10A 065 053 1492 1493 1496 1496 1497 1499 1499 1499 1499 1499 1499 1500 1500 1500 1500 1500 1500 1500 15			•	*		TO *RET*+1 IF O	K, VALUE IN MEMORY.
065 315 066 003 1492 10A CALL 10B INPUT BYTE  065 053 1493 0CX H  1499			•	*	USES	A,D,E,H,L,F	
0.65 0.53 0.03 1492 10A CALL 10B 0.65 0.53 1492 10A CALL 10B 0.65 0.53 1492 4 10B - INPUT OCTAL BYTE. 1496 4 1497 4 10B - INPUT OCTAL BYTE FROM THE KEYS! 1499 4 ENTRY (H,L) - ADDRESS OF BYTE T 1500 4 1500 4 1500 1500 1500 1500 1500 1			1	: '			
1495 ** 108 - INPUT OCTAL BYTE.  1496 * READ ONE OCTAL BYTE FROM THE KEYSI  1498 * ENTRY (H,L) - ADDRESS OF BYTE TI  1500 * EXIT TO *RET* IF FIRST DIGIT IN  1500 * EXIT TO *RET* IF FIRST DIGIT IN  1501 * EXIT TO *RET* IF FIRST DIGIT IN  1502 * A.D.E.H.L.F  1503 * USES A.D.E.H.L.F  1504	003.065	053		-	DCX	e I	
1496 # READ ONE OCTAL BYTE FROM THE KEYSI 1499 # ENTRY (H,L) = ADDRESS OF BYTE TI 1500 # EXIT TO #RET# IF ALL UK 1501 # EXIT TO #RET# IF ALL UK 1502 # OSES A,D,E,H,L,F 1503 # USES A,D,E,H,L,F 1504 1505 1506 1506 1509 1500 1500 1500 1500 1500 1500 1500			1495	*	90	OCTAL	
1499 # ENTRY (H,L) = ADDRESS OF BYTE TO 1499 # ENTRY (H,L) = ADDRESS OF BYTE TO 1500 # C C SET IF FIRST DIGIT IN 1501 # EXIT TO *ERROR* IF ERROR 1503 # USES A,D,E,H,L,F ERROR 1504 # USES A,D,E,H,L,F ERROR 1504 1504   1504   1505   1504   1506   1506   1506   1506   1507   108   HVI D,3			1496	* •			
1500 * CV SET IF FIRST DIGIT IN 1501 * CV SET IF FIRST DIGIT IN 1501 * CV SET IF FIRST DIGIT IN 1502 * CV SET IF FIRST DIGIT IN 1503 * USES A.D.E.H.L.F 1504 1505 1504 108 HVI D.3 (D) = DIG 0073 374 010 1510 003 1508 1081 CWC RCK READ CONS 1509 CPI 8 100 137 1510 USE RROR IF ILLEGA 1501 176 1513 HOV E.A (E) = VAL 1101 176 1514 HOV E.A (E) = VAL 1102 007 1516 1516 RCC RCC READ CONS 1517 1513 HOV E.A (E) = VAL 1102 007 1516 RCC RCC READ CONS 1516 RCC RCC READ CONS 1517 1518 HOV E.A (E) = VAL 1101 176 1516 RCC RCC READ READ CONS 1516 RCC RCC READ READ CONS 1517 1518 HOV E.A (E) = VAL 1103 007 1516 RCC RCC READ READ CONS 1516 RCC RCC READ READ CONS 1517 1518 HOV E.A (E) = VAL 1103 007 1516 RCC RCC READ READ READ READ READ READ READ READ			1498	*	KCAU	7	THE KETS
1502 # EXIT TO #RET# IF FIX51 DIGIT IN 1502 # 1502 # 1502 # 1502 # 1503 # 1503 # 1504   1504   1504   1504   1504   1505   1504   1505   1506   1506   1506   1507   108   1041   10,3   1507   108   1041   10,3   1509   1061   1061   1062   1509   1061   1062   1063			1499	*	ENTRY	(H,L) - ADDRESS	E 1
1502 * 10 * 10 * ERROR* IF ER 1503 * 1504 * 1505 * 1504 * 1505 *			1501	• •	EXIT	TO *RET* IF ALL	NI 119
1504 1505 1505 1506 004 0056 0056 0070 324 260 0073 1508 1508 1509 1071 1509 1071 1509 1072 1073 1075 1075 1075 1075 1075 1075 1075 1075			1502	* *	USES	TO *ERROR* IF E	KROR
066 026 003 1506 HVI D,3 070 324 260 003 1509 1081 CNC RCK 073 376 010 1510 CPI 8 075 322 322 000 1511 JNC ERROR 100 137 1513 MOV E,A 101 176 1514 MOV A,H 103 007 1515 RLC			1504				
0066 026 003 1507 108 MVI D <sub>2</sub> 3 070 324 260 003 1509 1081 CNC RCK 073 376 010 1510 CPI 8 075 322 322 000 1511 JNC ERROR 100 137 1513 MUV A,M 101 176 1515 RLC 103 007 1516 RLC		:	1506	:			
1509 1509 1509 1509 1509 1509 1509 1509		026 003	7.	:	IAU	0,3	(D) = DIGIT COUNT
373 376 010 1510 CPI 8 375 322 322 000 1511 JNC ERROR IF ILLEGAL 100 137 1513 NOV E,A 101 176 1514 NOV A,H 102 007 1516 RLC 103 007 1516		75.	, –		) E	4	KEAU CUNSULE RETSEI
10 137 1513 MOV E,A  101 176 1513 MOV E,A  101 176 1514 MOV A,M  102 007 1516 RLC  103 007 1516	0.0	376 010	-		CP I	<b>a</b>	
100 137 1513 MOV E,A (E) = 101 176 1514 MOV A,M SHET 102 007 1515 RLC SHIFT 103 007 1516	3:	322 322	`:		285	EKKOK	ILLEGAL
101 176 1514 MOV A,M 102 007 1515 RLC SHIFT 103 007 1516 RLC	003.100	:	1513		MOV	E,A	• :
103 007 1516 RLC	003-101		1514		Y04	A, A	C F0100
	003.103	:	1516		ארכי		

003-112 325 1538 ANI 3709 003-110 167 1520 MOV N.A 003-111 025 1521 DCR D 003-112 025 070 003 1522 JAP
167 1520 HOV N.A 025 1521 DCR D 302 070 003 1522 JNZ TOB1 303 140 002 1524 HVI A,30/2 BEEP FOR 30 MS 303 140 002 1524 HVI A,30/2 BEEP FOR 30 MS 303 140 002 1524 HVI HORN 1526 HVI DOD - DECODE FOR OCTAL DISPLAY. 1527 HVI DOD - DECODE FOR OCTAL DISPLAY. 1528 HVI (H,L) - ADDRESS OF LED REFRESH 1530 HVI (H,L) - MEX DIGIT ADDRESS 1532 HVI (H,L) - MEX DIGIT ADDRESS 1533 HVI D,000A/256
025 1521 DCR D
302 070 003 1522 JNZ 1081  076 017  303 140 002 1524 JMP HORN  1526 ** DOD - DECODE FOR OCTAL DISPLAY.  1527 * ENTRY (H,L) = ADDRESS OF LED REFRESH  1528 * ENTRY (H,L) = ADDRESS OF LED REFRESH  1530 * EXIT (H,L) = NEX DIGIT ADDRESS  1531 * EXIT (H,L) = NEX DIGIT ADDRESS  1532 * USES A,8,C,D,H,L  1533 ** USES A,8,C,D,H,L  1534 ** USES A,8,C,D,H,L  1535 ** USES A,8,C,D,H,L  1535 ** USES A,8,C,D,H,L  1535 ** USES A,8,C,D,H,L  1536 ** USES A,8,C,D,H,L  1537 ** USES A,8,C,D,H,L  1538 ** USES A,8,C,D,H,L  1539 ** USES A,8,C,D,H,L  1530 **
303 140 002 1524 JMP HORN  1526 ** DOD - DECODE FOR OCTAL DISPLAY.  1527 * ENTRY (H,L) = ADDRESS OF LED REFRESH 1529 * (A) = 6074 PATTERN TO FURCE ON 1530 * EXIT (H,L) = NEX DIGIT ADDRESS 1531 * USES A,8,C,D,H,L 1533 1535 DOD PUSH D 026 003 1536 MVI D,DODDA/256
1526 ** DOD - DECODE FOR OCTAL DISPLAY. 1527 * ENTRY (H,L) - ADDRESS OF LED REFRESH 1529 * (A) - OCTAL VALUE 1530 * (A) - OCTAL VALUE 1531 * EXIT (H,L) - NEX DIGIT ADDRESS 1531 * OSES A.8.C.D.H.L 1533 * OSES A.8.C.D.H.L 1534 * OSES A.8.C.D.H.L 1535 DOD PUSH D.
1526
1529 * (8) = *0R* PATTERN TO FURCE ON 1530 * (A) = 0CTAL VALUE 1531 * EXIT (H;L) = NEX DIGIT ADDRESS 1532 * USES A,8,C,D,H;L 1533 1534 1535 DOD PUSH D 0.0DDA/256
1531 # EXIT (H,L) 1532 # USES A,8,C, 1533 # 1535 22 325 1535 DOD PUSH D
1533 1534 1535 000 PUSH 0 23 026 003 1536 HVI 0
22 325 1535 000 PUSH 23 026 003 1536 HVI
123 026 003 1536 HVI
(2) 016 003 1537 MVI C,3
130 027 1539
131 027 1540 RAL
132 365 1541 133 346 007 1542
135 306 356 1543
137 137 1544 HOV
(4) 032 1545 LDAX D (A) = (A)
142 346 177 1547 ANI
1548 XRA B
146 043 1550 INX
147 170 1551 HOV
51 107 1553 MOV
152 361 1554
(53 015 1555 0CR C
57 321 1557
160 311 1558 RET

561 ## IJED - IJEDATE ERDNT PANEL DISPLAYS	1561 ** UFD - UPDATE FRONT PANEL DISPLAYS.	SVA 19310 LIBOAT CONTRACT CONTRACT OFFI	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	+ 700		4 ( ) 4 (							1000	570 # UCEC	570 <b>*</b> USES	570 * USES	570 * USES	3/0 • 0353		1572		
1562 • 1562 • 1660 IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IT IT IS 1564 • 1660 IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IT IT IS 65 • 1684 • 1684 IS INTERRUPTS, DIRPAR, ODNIENTS, CURRENTLY, HIS 15 1565 • 1684 IS INTERRUPTS, DIRPAR, ODNIENTS, CURRENTLY, HIS 15 1566 • 1684 IS INTERRUPTS, DIRPAR, ODNIENTS, CURRENTLY, HIS 15 1566 • 1684 IS INTERRUPTS, DIRPAR, ODNIENTS, CURRENTLY, HIS 15 15 15 15 15 15 15 15 15 15 15 15 15	1564 • UPO IS CALLEO BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1564 • UPO IS CALLEO BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1565 • EVERT 32 INTERRUPT'S, DRAGOT 32 TIMES A SECONO. 1566 • EVERT 32 INTERRUPT'S, DRAGOT 32 TIMES A SECONO. 1567 • EVIT NOME 1569 • EVIT NOME 1569 • EVIT NOME 1570 • USES ALL 1571 • USES ALL 1572 • EXIT NOME 1573 • EXIT NOME 1574 • WHI A, UG.DOU 1574 • WHI A, UG.DOU 1575 • EXIT NOME 1576 • EVIT NOME 1577 • WHI A, UG.DOU 1578 • EXIT NOME 1577 • WHI A, UG.DOU 1578 • EXIT NOME 1579 • EXIT NOME 1570 • EXIT NOME 1570 • EXIT NOME 1570 • EXIT NOME 1570 • EXIT NOME 1571 • WHI A, UG.DOU 1571 • WHI A, UG.DOU 1572 • EXIT NOME 1574 • EXIT NOME 1575 • EXIT NOME 1576 • EXIT NOME 1576 • EXIT NOME 1577 • WHI A, UG.DOU 1570 • WHI A, UG.DOU 15	1562   1562   1564   1564   1671	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECONO. 1568 + EWRY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECONO. 1568 + EWRY (i.i.) - ADDRESS OF REFORT 1570 - USES ALL 1571 - OSES ALL 1572 - ANA B ANA	1564   1	1564 • UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS DONE 1567 • FTER TO UPDATE THE DISPACE COLOR. THIS IS DONE 1567 • FTER TO UPDATE THE DISPACE COLOR. THIS IS DONE 1567 • FTER TO UPDATE COLOR THIS IS DONE 1567 • FTER TO UPDATE COLOR THIS IS DONE 1567 • FTER TO UPDATE COLOR THIS IS DONE 1567 • FTER TO USES ALL 1567 • FTER TO USES ALL 1571 • MAY 8, UG. DOU 1572 • MAY 8, UG. DOU 1573 • MAY 8, UG. DOU 1574 • MAY 1774 • MAY 1574 • MAY 1774 • MAY 177	1564 THE TO UPDATE THE UITSCRIPT PROCESSOR WHEN IT IS 1567 FEWER 28 THERROPTS, CORRENILY, THIS IS DONE 1569 FEWER 28 THERROPTS, OR ABOUT 32 TIMES A SECONO. 1569 FEMITY (H.L.) - ADDRESS OF REFORT 1569 FEMITY (H.L.) - ADDRESS OF REFORT 1570 USES ALL 1571 HAN HANDED THE UITSCRIPT OF THE UITSCRIPT	1570	1570	1570   USES ALL	1570   USES ALL     1571   WINDER   WINDER     1572   WINDER   WINDER     1573   WINDER   WINDER     1574   WINDER   WINDER     1575   WINDER   WINDER     1576   WINDER   WINDER     1577   WINDER   WINDER     1577   WINDER   WINDER     1580   WINDER   WINDER     1581   WINDER   WINDER     1581   WINDER   WINDER     1582   WINDER   WINDER     1583   WINDER   WINDER     1584   WINDER   WINDER     1585   WINDER   WINDER     1585   WINDER   WINDER     1586   WINDER   WINDER     1586   WINDER   WINDER     1587   WINDER   WINDER     1588   WINDER   WINDER     1589   WINDER   WINDER     1590   WINDER	1570   USES ALL     1571   WAY   A,UO.DDU     1572   WAY   A,UO.DDU     1573   WAY   A,UO.DDU     1574   WAY   A,UO.DDU     1575   WAY   A,UO.DDU     1576   WAY   A,UO.DDU     1577   WAY   A,UO.DDU     1580   WAY   A,UO.DDU     1581   WAY   A,UO.DDU     1582   WAY   A,UO.DDU     1583   WAY   A,UO.DDU     1584   WAY   A,UO.DDU     1585   WAY   A,UO.DDU     1585   WAY   WAY     1586   A,UO.DDU     1586   A,UO.DDU     1586   A,UO.DDU     1587   WAY   A,UO.DDU     1588   WAY   WAY     1589   WAY   WAY     1590   WAY   WAY     1500   WAY   WAY     1601   WAY   WAY     1602   WAY   WAY     1603   WAY   WAY     1604   WAY   WAY     1606   WAY   WAY     1607   SETUP DISPLAY     1608   WAY   WAY     1609   WED   PUSH   PSH     1600   WED   WAY     1600	1571   1572   1573   1574   1574   1575   1775	1571   0.55.5   0.5	1571   1572   1573   1574   1575   1775	1571   1572   1573   1574   1474   1474   1574   1575   1775	1571   1572   1573   1574   1574   1575	1571   1571   1571   1571   1571   1572   1572   1573   1574   1474   1474   1575   1575   1575   1474   1474   1575   1575   1474   1474   1575   1575   1474   1474   1775	1571   1572   1572   1573   1574   1574   1575	1573   UFO   GOU   +	1573   UFD   EQU   +	1573   UFD   EQU   +	1575
1562   1562   1662	1562 4 1562 4 1562 4 1562 4 1562 4 1563 4 1663 4 16	1562 - 1167 O SALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 15 156 - 1167 O SALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 15 156 - 1167 O SALLE SA	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECONO. 1568 + EWERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECONO. 1568 + EWIRY (H.L.) - ADDRESS OF REFORT 1570	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1565 • FUREY OUTDAIR THE DISPLAY CONTENTS. CURRENTLY, THIS IS DONE 1566 • EVERY 32 INTERRUPTS, OR RED. 34 THES A SECOND. 1567 • EVERY 32 INTERRUPTS, OR RED. 34 THES A SECOND. 1568 • EXIT NOWE 1569 • EXIT NOWE 1570 • USES ALL 1571 • NAT A.UO.DDU 1573 • WILL A.UO.DDU 1574 • WAY A.MA B. WAY B. WAY A.MA B. WAY B.MA B.MA B.WAY	1565 • TIME TO UPDATE THE OLDCK INTERRUPT PROCESSOR WHEN IT IS SOME 1565 • TIME TO UPDATE THE OLDCK INTERRUPT PROCESSOR WHEN IT IS SOME 1567 • TIME TO UPDATE THE OLDCK THES A SECOND. 1567 • ENTRY IT HONE 1569 • ENTRY IT HONE 1569 • ENTRY IT HONE 1571 • TO THE TO UPDATE THE OLDCK THE STANDARD AND THE STANDARD AN	1564	1570 • USES ALL  1571  1571  1572  1573  1574  1575  1574  1575  1574  1575  1577  157  1577  1577  1577  1577  1577  1577  1577  1577  1577  1577  15	1570 • USES ALL  1571 1571 1572 1573 1573 1574 1574 1575 1575 1574 1575 1575 1575	1570   USES ALL	1570	1570   USES ALL     1571   USES ALL     1571   USES ALL     1572   USES ALL     1573   UFO   60   6     1574   WAY   A,UO.DDU     1575   WAY   A,UO.DDU     1576   WAY   A,UO.DDU     1577   RAZ     1577   RAZ     1587   WAY   A,UO.DDU     1580   RUC   A,H     1581   RAZ   RAZ     1581   RAZ   RAZ     1582   RUC   A,H     1583   RUC   A,H     1583   A,H     1584   A,H     1584   A,H     1585   A,H     1585   A,H     1586   A,H     1586   A,H     1587   RUC   A,H     1589   A,H     1580   A,H     1580   A,H     1580   A,H     1590   A,H     1590   A,H     1590   RUC   RUC     1590   RUC   RUC   RUC     1590	1570   0.25.5 ALL   1571   0.25.5 ALL   1571   0.05.5 ALL   1572   0.05.5 ALL   1573   0.05.5 ALL   0.05.5	1571   1572   1573   1574   1574   1574   1575   1775   1575	1571   1572   1573   1574   1575   1775	1571   1572   1573   1574   1575   1775	1571   1571   1572   1573   1574   1574   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1775	1571   1571   1572   1573   1574   1574   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1775	1571 1572 1573 160 1574 1774 1774 1774 1774 1775 1777 1777	1573   150   150   1573   150   1574   170   1	1575   UFD   EQU	1573   UFD   EQU   4, UG.DDU     1574   ANA   8, UG.DDU     1575   ANA   8, UG.DDU     1576   ANA   8, UG.DDU     1577   ANA   A, UG.DDU     1578   ANA   A, M     1580   R.C.     1581   ANA   A, M     1582   AND   B, A     1583   AND   B, A     1584   AND   B, A     1585   AND   B, A     1586   AND   B, A     1586   AND   B, A     1587   AND     1588   AND   B, A     1589   AND   B, A     1589   AND   B, A     1589   AND   B, A     1589   AND   B, A     1590   AND   B, A     1590   AND   B, A     1590   AND   B, A     1590   AND   A, A     1500   AND   A, A     15	1574   WAY
1545   1545   1445	156.2 * 1. 156.2 * 1. 156.2 * 1. 156.2 * 1. 156.2 * 1. 156.3 * 1.	1562 - 116	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECONO. 1567 - ENTRY (H.L) - ADDRESS OF REFORT 1570 - ENTRY (H.L) - ADDRESS OF REFORT 1571 - OSES ALL 1573 UFD EQU	1564   1   15   16   10   10   10   10   10   10   10	1565 • TITRE TO UPDATE THE CLOCK INTERRUPT PROCESSOR WHEN IT IS SOME 1567 • TITRE TO UPDATE THE DISPAY CONTENTS. CURRENILY. THIS IS DONE 1567 • TITRE TO UPDATE THE DISPAY CONTENTS. CURRENILY. THIS IS DONE 1567 • TITRE TO UPDATE THE DISPAY CONTENTS. A SECOND. 1569 • EXIT NONE 1569 • EXIT NONE 1570 • USES 1571 • WAI A NOT CONTENT TO THANDLE UPDATE 1571 • WAI A NOT CONTENT TO THANDLE UPDATE 1572 • WAI A NOT CONTENT TO THANDLE UPDATE 1573 • WAI A NOT CONTENT TO THANDLE UPDATE 1574 • WAI A NOT CONTENT TO THANDLE UPDATE 1575 • WAI A NOT CONTENT TO THANDLE UPDATE 1576 • WAI A NOT CONTENT TO THANDLE UPDATE 1580 • WAI A NOT CONTENT TO THANDLE UPDATE 1580 • WAI A NOT CONTENT TO THANDLE UPDATE 1580 • WAI A NOT CONTENT TO THANDLE UPDATE 1580 • WAI DISPLAYING RECISTERS XOURESS OF REG NAME PATTERNS 1580 • WAI DISPLAYING RECISTERS OF REG NAME PATTERNS 1580 • WAI DISPLAYING RECISTERS OF REG NAME PATTERNS 1580 • WAI DISPLAYING RECISTER PAIR CONTENT 1580 • WAI DISPLAY WAI HANDLE UPDATESS OF REG NAME PATTERNS 1580 • WAI DISPLAYING WAI HANDLE UPDATESS OF REG NAME PATTERNS 1580 • WAI DISPLAYING WAI HANDLE UPDATESS OF REG NAME PATTERNS 1580 • WAI DISPLAYING WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI DISPLAYING WAI HANDLE UPDATESS OF REG NAME PATTERNS 1580 • WAI DISPLAYING WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI DISPLAYING WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATESS OF REGISTER PAIR CONTENT 1580 • WAI HANDLE UPDATES	1564 THE OF THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1567 FEWER 12 STATER DISPARENTS. CORREST A SECONO. 1567 FEWER 12 STATER MAY (11.1.) - ADDRESS OF REFCAT 1569 FEMILY (11.1.) - ADDRESS OF REFCAT 1571 CORP. 1571 CORP. 1572 CORP. 1573 GEO. 1574 MAY 8.1 MAY 1.1 M	1570	1570 • USES ALL   1571 • USES ALL   1570 • USES ALL   1571 • USES ALL   1572 • USES ALL   1572 • USES ALL   1573   UFO	1570   USES ALL	1570	1570   USES ALL     1571   WIT   A.W. DDU     1571   WIT   WIT   A.W. DDU     1572   WIT   WIT   A.W. DDU     1573   WIT   WIT   A.W. DDU     1574   WIT	1571   1571   1572   1573   160   6   1573   160   6   1573   160   6   1573   160   6   1573   160   6   1573   160   6   1573   160   6   1573   160   6   1573   160   6   1573   160   6   1573   160   6   1573   160   1573   160   160   1573   160   160   1573   160   160   1573   160   160   1573   160   160   1573   160   160   1573   160   160   1573   160   160   1573   160   160   1573   160   1	1571   1572   1573   1574   1574   1575   1775	1571   1572   1574   1575   1576   1577   1577   1577   1577   1577   1577   1577   1577   1577   1577   1577   1577   1577   1777	1571   1572   1573   1574   1575   1775	1571   1571   1572   1573   1574   1574   1575   1775	1571   1571   1572   1573   1574   1574   1575   1775	1571   1572   1572   1573   1574   1574   1575	1573   150	1573   UFD   EQU   +	1573   UFD   EQU   +	1575   ANA   B
1562   1562   1562   1662	156. 1 15	1562   1562   1563   1564   1565   15	1566 * EYERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECONO.  1568 * ENTRY (H,L) - ADDRESS OF REFORT  1570 * OSES ALL  1571 * OSES ALL  1572 * OSEO * OSES OF REFORT  1573 UFD 60U * OSES OF REFORT  1574 MY A, UO.DDU  1575 MY A, UO.DDU  1576 MY A, UO.DDU  1577 MY A, UO.DDU  1578 MY A, UO.DDU  1579 MY A, UO.DDU  1570 MY A, UO.DDU  1570 MY A, UO.DDU  1571 MY A, UO.DDU  1571 MY A, UO.DDU  1571 MY A, UO.DDU  1572 MY A, UO.DDU  1573 MY A, UO.DDU  1574 MY A, UO.DDU  1575 MY A, UO.DDU  1576 MY A, U. CALL  1585 MY A, U. CALL  1585 MY A, U. CALL  1586 MY A, U. CALL  1586 MY A, U. CALL  1587 MY A, U. CALL  1587 MY A, U. CALL  1588 MY A, U. CALL  1589 MY A, U. CALL  1580 MY A	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1565 • EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 • EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1569 • EXIT MONE 1570 • USES ALL 1571 1570 • USES ALL 1571 1573 • EXIT MONE 1574 1575 • MAI 1575 • MAI 1576 • USES ALL 1577 1577 1576 • USES ALL 1577 1577 1578 • USES ALL 1579 • USES ALL 1579 • USES ALL 1570 • USES ALL 1570 • USES ALL 1570 • USES ALL 1570 • USES ALL 1571 • USES ALL 1571 • USES ALL 1571 • USES ALL 1571 • USES ALL 1572 • USES ALL 1573 • USES ALL 1574 • USES ALL 1575 • USES ALL 1575 • USES ALL 1576 • USES ALL 1577 • USES ALL 1	1565 • THE TO UPDATE THE DISCRIPTIST CONTRESS OF WHEN IT ISS DONE 1567 • FIREY 12' UNITERATED CONTRESS. COND. 1567 • FIREY 12' UNITERATED CONTRESS. COND. 1567 • FIREY 12' UNITERATED CONTRESS. SECOND. 1569 • EXIT NONE 1570 • USES ALC. 1571 • WAY 8	1564 THE TO UPDE THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1567 FIRE TO UPDE THE DISPARCH THE STATE OF THE STA	1570 • USES ALL  1571  1572  1573  1574  1575  1574  1575  1574  1575  1576  1577  157  1577  1577  1577  1577  1577  1577  1577  1577  1577  1577  15	1570 • USES ALL  1571 1572 1573 1573 1573 1574 1575 1575 1575 1574 1575 1575 1575	1570	1570 • USES ALL  1571 • USES ALL  1571 • USES ALL  1572 • WI  1573 UFO • OSES ALL  1574 ANA 6 USES OF RECOMME DESTRUCT  1576 RAT 6  1576 RAT 6  1577 HVI C., *BOSPROT IF NOT TO HANDLE UPDATE  1577 ANA 1  1581 HOV A,M A,M A,M A,M A,M B,M A,M A,M A,M A,M A,M A,M A,M A,M A,M A	1570   USES ALL     1571   USES ALL     1571   USES ALL     1572   USES ALL     1573   USES ALL     1574   USES ALL     1575   USES ALL     1576   USES ALL     1577   USES ALL     1578   USES ALL     1581   USES ALL     1582   USES ALL     1583   USES ALL     1584   USES ALL     1585   USES ALL     1586   USES ALL     1587   USES ALL     1588   USES ALL     1589   USES ALL     1590   USES ALL     1590	1570   0.55.5 ALL   1571   0.55.5 ALL   1572   0.05.5 ALL   1573   0.05.5 ALL   1573   0.05.5 ALL   1574   0.05.5 ALL   1574   0.05.5 ALL   1575   0.05.5 ALL	1571   1572   1573   1574   1574   1575	1573 UFO	1571 1572 1674 1774 1774 1774 1775 1775 1775 1775 17	1571   1571   1572   1574   1774   1775	1571   1571   1572   1574   1774   1775	1571   1571   1571   1571   1571   1571   1572   1574   1774   1774   1775	1573	1575   UF D   EQU   +	1573 UFD   FOU	002 1575 ANA 8 100.00U 1576 ANA 8 17.00.00U 1577 ANA 8 17.00.00U 1578 ANA 1
1554   1564   15 CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1564   1564   1564   1564   1564   1564   1565   1565   1565   1565   16	1564 • 1562 • 1564 • 1564 • 1565 • 1565 • 1565 • 1566 • 1660 • 1566 • 1660 • 1566 • 1660 • 1566 • 1660 • 16	1562 - 1563 - 1564 - 1564 - 1565 - 15	1566 + EYERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECONO.  1567 + ENTRY (H,L) - ADDRESS OF REFCRI  1570 - USES ALL  1571 - USES ALL  1571 - WAL  1572 - WAL  1573 - WE  1574 - WAL  1575 - WAL  1576 - WAL  1577 - WAL  1576 - WAL  1577 - WAL  1577 - WAL  1578 - WAL  1578 - WAL  1578 - WAL  1579 - WAL  1570 - WAL  1571 - WAL  1571 - WAL  1571 - WAL  1572 - WAL  1573 - WAL  1574 - WAL  1575 - WAL  1575 - WAL  1576 - WAL  1576 - WAL  1577 - WAL  1570 - W	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1565 • EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 • EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 • EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1570 • EXIT NONE 1570 • USES ALL 1571 • WIT NONE 1571 • WIT NONE 1572 • WIT NONE 1573 • WIT NONE 1574 • WIT NONE 1575 • WIT NONE 1575 • WIT NONE 1576 • USES ALL 1577 • WIT NONE 1577 • WIT NONE 1578 • WIT NONE 1579 • WIT NONE 1570 • WIT NONE 1571 • WIT NONE 1571 • WIT NONE 1571 • WIT NONE 1572 • WIT NONE 1573 • WIT NONE 1574 • WIT NONE 1575 • WIT NONE 1576 • WIT NONE 1577	1565 • TIME TO UPDE THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1567 • FIRE TO UPDE THE DISPACE A SECOND. 1568 • FIRE TO UPDE THE DISPACE A SECOND. 1569 • ENTRY 12 'INTERRUPT'S, OR ABOUT 32 TIMES A SECOND. 1569 • ENTRY 12 'INTERRUPT'S, OR ABOUT 32 TIMES A SECOND. 1571 • COLD INTERRUPT SOLITION OF THE SECOND OF T	1564 TO THE TO UPDATE THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1567 TO UPDATE THE UISCALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS DONE 1568 TO UPDATE THE UISCALL THE STATEMENTS. CURRENTLY. THIS IS DONE 1569 TO USES ALL 1569 TO USES ALL 1571 TO USES ALL 1571 TO USES ALL 1572 TO USES ALL 1573 TO USES ALL 1574 TO USES ALL 1574 TO USES ALL 1574 TO USES ALL 1575 TO USES ALL 1575 TO USES ALL 1576 TO USES ALL 15	1570 • USES ALL  1571 • USES ALL  1571 • USES ALL  1572 • USES ALL  1573 UFD EQU • And And Burd  1575 And Burd  1575 And Burd  1576 And And And Burd  1576 And And And And Burd  1580 HC L+BSPROT  1581 HWY And	1570 • USES ALL  1571	1570 • USES ALL  1571	1570 • 0355 ALL  1571	1570 - 0355 ALL  1571 - 0755 ALL  1573 - 076	1571   1572   1573   1574   1775   1575   1575   1775	1572   1573   1574   1774   1775	1571   1572   1573   1574   1474   1474   1575   1775	1571   1572   1573   1574   1774	1571   1571   1572   1574   1774   1774   1775   1774   1775	1571   1571   1572   1574   1774   1774   1775   1774   1775	1571   1571   1571   1571   1571   1571   1572   1574   1774   1775   1774   1775	1273   UFD   EQU   H     1573   UFD   EQU   H     1574   ANI	1573   UFD   EQU   #   WI	1573   150	002 1575 JPD HVI A:00.00U 1575 ANA B.U0.00U 1576 ANA B.U0.00U 1577 ANI B.U1.00U 1577 ANI C.; BDSPROT 1580 ANI C.; BDSPROT 1581 AND A; A 1582 ANI DSPLAYING RECISTER XDDRESS 1584 ANI DISPLAYING RECISTER XDDRESS 1589 ANI DISPLAYING RECISTER PAIR CONTENT 1590 ANI AIR AIR CONTENT
1552 *	1562 * UFD 15 CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1T 15 1563 * THE TO UPDATE THE DISPLAY CONTENTS. CURRENLLY. THIS 15 DONE 1564 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 * EXIT NONE 1569 * EXIT NONE 1560 * EXIT NONE 1570 * USES ALL 1571 * NVI L.**DSPROT 1572 * NVI L.**DSPROT 1573 * NVI L.**DSPROT 1574 * NVI L.**DSPROT 1575 * NVI L.**DSPROT 1576 * NVI L.**DSPROT 1577 * NVI L.**DSPROT 1580 * NVI NA HANDLE UPDATE 1581 * NVI NA HANDLE UPDATE 1582 * NVI NA HANDLE UPDATE 1583 * NVI NA HANDLE UPDATE 1584 * NVI NA HANDLE UPDATE 1585 * NVI NA HANDLE UPDATE 1585 * NVI NA HANDLE UPDATE 1585 * NVI NA HANDLE UPDATE 1586 * NVI NA HANDLE UPDATE 1587 * NVI NA HANDLE UPDATE 1589 * NVI NA HANDLE UPDATE 1580 * NVI NA	1562 - 1562 - 1563 - 1564 - 1565 - 15	1566 + EYERY 32 INTERRUPTS, DR ABOUT 32 TIMES A SECONO. 1567 + ENTRY (H+L) - ADDRESS OF REFCRI 1570 - CALT NONE 1570 - CALT NONE 1571 - CALL L. ADDRESS OF REFCRI 1572 - CALL LRA 1586 - HOY H,A 1587 - HOY H,A 1586 - HOY H,A 1587 - HOY H,A 1586 - HOY H,A 1686 - H	1564 * UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1565 * FINE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, FILS IS DONE 1566 * ENTRY (H.L.) - ADDRESS OF REFORM 1568 * ENTRY (H.L.) - ADDRESS OF REFORM 1570 * USES ALL 1571 * USES ALL 1571 * USES ALL 1572 * USES ALL 1573 * USES ALL 1574 * WY A, UD, DDU 1575 * WY A, UD, DDU 1575 * WY A, UD, DDU 1576 * WY A, UD, DDU 1577 * WY A, UD, DDU 1577 * WY A, UD, DDU 1578 * WY A, UD, DDU 1579 * WY A, UD, DDU 1571 * WY A, UD, DDU 1571 * WY A, UD, DDU 1571 * WY A, UD, DDU 1572 * WY A, WY A, UD, DDU 1573 * WY A, WY A, UD, DDU 1574 * WY A, UD, DDU 1575 * WY A, WY A, UD, DDU 1575 * WY A, WY	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1565 * FURRY 32 INTERRUPTS, COMPRIST, FHIS IS DONE 1566 * FURRY 32 INTERRUPTS, COMPRIST, FHIS IS DONE 1567 * ENTRY 11.1 * ADDRESS OF REFORT 1569 * EXIT NOME 1570 * USES ALL 1571 * FOR SOLUTION OF SECOND 1577 * USES ALL 1577 * USES ALL 1578 * WILL 1578 * WILL 1578 * WILL 1579 * USES ALL 1571 * WILL 1570 * USES ALL 1571 * WILL 1571 * WILL 1571 * WILL 1572 * WILL 1573 * WILL 1574 * WILL 1574 * WILL 1575 * WILL 1575 * WILL 1576 * WILL 1577 * WIL	1569 * UFD 15 CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1T 15 1569 * EVERY 32 INTERRUPT'S, COMERTY. FINIS 15 DONE 1569 * EXIT NONE 1569 * EXIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * ANY 1	1570 • USES ALL  1571 • USES ALL  1571 • USES ALL  1572 • USES ALL  1573 UFD EQU • A., M.	1570 • USES ALL  1571 • USES ALL  1571 • USES ALL  1572 • USES ALL  1573 UFD 600 •  1575 ANA 8  1575 ANA 8  1575 ANA 8  1576 ANA 1  1580 RLC  1580 RLC  1580 RLC  1580 RLC  1581 HNY H, A HOLD-DSPROT-1  1581 HNY H, A HOLD-DSPROT-1  1582 NOV H, A HOLD-DSPROT-1  1583 HNY H, A HOLD-DSPROT-1  1584 NOV H, A HOLD-DSPROT-1  1585 NOV H, A HOLD-DSPROT-1  1585 NOV H, A HOLD-DSPROT-1  1585 NOV H, A HOLD-DSPROT-1  1580 RLC  15	1570 • USES ALL  1571	1570 • 0355 ALL  1571	1570	1570 • USES ALL  1571 • USES ALL  1572 • UFD • USES ALL  1573 • UFD • USES  1574 • WI A, UO. DOU  1575 • WA B  1576 • WA B  1576 • WA B  1580 • WA MISSPROTE  1580 • WA DISPLAYING RECISTERS  1580 • WA DISPLAYING RECISTER WODRESS  1580 • WA DISPLAYING RECISTER WODRESS  1580 • WA DISPLAYING RECISTER WODRESS  1580 • WA WA CLEAR 'L'  1580 • WA WA WA CLEAR 'L'  1580 • WA WA WA CLEAR 'L'  1580 • WA WA WA WA CLEAR 'L'  1580 • WA WA WA WA WA CLEAR 'L'  1580 • WA	1571	1571 1573 1573 1574 1575 1577 1577 1577 1577 1577 1577	1571 1572 1573 1574 1575 1574 1575 1575 1575 1575 1577 1577	1571 1571 1572 1573 1574 1574 1575 1575 1576 1577 1577 1577 1577 1577	1571 1571 1572 1573 1574 1574 1575 1575 1576 1577 1577 1577 1577 1577	1571   1571   1571   1572   1573   1574   1574   1575   1775	1277   UFD   EQU   +	1573   UFD   EQU   +	002 1573 UFD 600 4 1576 ANA 8 1576 ANA 8 1577 ANY L. **DSPROT 1577 ANY L. **DSPROT 1580 ACC ACC ACC ACC ACC ACC ACC ACC ACC AC	002 1574 ANA 8 4.00.00U  1575 ANA 8 4.00.00U  1576 ANA 8 4.00.00U  1577 ANA 8 4.00.00U  1580 RC A.M A.M ADTATE PATTERN  1581 NOV 8.A ADDRESS OF RECISTER XOURESS  1582 NOV 4.M ADDSPHOT—1  1584 RECISTER XOURESS  227 003 1588 JZ UPO1  1590 AN DISPLAYING RECISTERS.  1590 AN DISPLAYING RECISTER ADDRESS OF REC NAME PATTERNS  1590 AND A.M (H-LL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT  1600 AND H-M H-M (HL) - ADDRESS OF RECISTER PAIR CONTENT
1562 - 1564 - 1060 IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1564 - 1667 - 1667 IN THE JOSPAN CONTENTS. CORRECTION. 1565 - 16767 - 16767 IN THE JOSPAN CONTENTS. CORRECTION. 1566 - 1570 - 1556 - 1677 IN THE JOSPAN CONTENTS. CORRECTION. 1570 -	1562 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 * EVERY 32 INTERRUPTS, OR REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 * UFD IS ALL 1571 * UFD IS ALL 1572 * UFD IS ALL 1573 * UFD * USES ALL 1573 * UFD * USES ALL 1574 * UFD IS ALL 1575 * UFD IS ALL 1575 * UFD IS ALL 1576 * USES ALL 1577 * UFD IS ALL 1576 * USES ALL 1577 * UFD IS ALL 1578 * USES ALL 1579 * USES ALL 1570 * USES ALL 1571 * UFD IS ALL 1571 * UFD IS ALL 1571 * UFD IS ALL 1572 * UFD IS ALL 1573 * UFD IS ALL 1574 * USES ALL 1575 * USES ALL 1575 * USES ALL 1576 * USES ALL 1577 * USES ALL 1577 * USES ALL 1577 * USES ALL 1578 * USES ALL 1579 * USES ALL 1570 * USES ALL 1570 * USES ALL 1571 * USES ALL 1571 * USES ALL 1571 * USES ALL 1572 * USES ALL 1573 * USES ALL 1574 * USES ALL 1575 * USES ALL 1575 * USES ALL 1575 * USES ALL 1576 * USES ALL 1577 * USES ALL 1577 * USES ALL 1578 * USES ALL 1578 * USES ALL 1579 * USES ALL 1570 * USES ALL 1570 * USES ALL 1570 * USES ALL 1571 * USES ALL 1571 * USES ALL 1572 * USES ALL 1573 * USES ALL 1574 * USES ALL 1575 *	1562   1563   1564   1565   1665	1566	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1565 * TIME TO UPDATE THE DISPLAY CONTENTY, FILS IS DONE 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECONO. 1567 * ENTRY (H.L.) - ADDRESS OF REFCNI 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 USE 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1576 ANA B IF NOT TO HANDLE UPDATE 1577 ANA B IF NOT TO HANDLE UPDATE 1578 ANA B IF NOT TO HANDLE UPDATE 1579 AND ANA B IF NOT TO HANDLE UPDATE 1570 ANA B IF NOT ANA B IF NOT TO HANDLE UPDATE 1571 ANA B IF NOT ANA B IF NOT TO HANDLE UPDATE 1570 AND ANA B IF NOT ANA B IF NOT TO HANDLE UPDATE 1580 AND ANA B IF NOT ANA B IF NOT TO HANDLE UPDATE 1580 AND ANA B IF NOT ANA B IF NOT TO HAND B IF NOT ANA B IF NOT	1565 • UFD IS CALLED BY THE CLOCK INVERRUPT PROCESSOR WHEN IT IS 1567 • FIRE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS DONE 1566 • FIRE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS DONE 1567 • EXIT NONE 1569 • EXIT NONE 1569 • EXIT NONE 1570 • USES ALL 1570 • USES ALL 1570 • USES ALL 1570 • USES ALL 1571 • NA A A A A A A A A A A A A A A A A A	1569 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS 1569 * FIRE TO UPDATE TIE DISPLAY CONTENTS. CURRENTLY, THIS IS DONE 1560 * EVERY 32 LIVERRUPTS, CONTENTS. CURRENTLY, THIS IS DONE 1567 * EVERY 32 LIVERRUPTS, CONTENTS. A SECOND. 1569 * EXIT NOTE 1569 * EXIT NOTE 1569 * EXIT NOTE 1570 * USES ALL 1571 * NAT A *UD.*DDU 1570 * USES ALL 1571 * NAT A *UD.*DDU 1572 * NAT A *UD.*DDU 1573 * NAT A *UD.*DDU 1574 * NAT A *UD.*DDU 1575 * NAT A *UD.*DDU 1576 * NAT A *UD.*DDU 1577 * NAT A *UD.*DDU 1578 * NAT A *UD.*DDU 1579 * NAT A *UD.*DDU 1570 * UND A *UD 1571 * NAT A *UD.*DDU 1571 * NAT A *UD.*DDU 1572 * NAT A *UD.*DDU 1573 * NAT A *UD.*DDU 1574 * NAT A *UD.*DDU 1575 * NAT A *UD.*DDU 1575 * NAT A *UD.*DDU 1576 * NAT A *UD.*DDU 1577 * NAT A *UD.*DDU 1578 * NAT A *UD.*DDU 1579 * NAT A *UD.*DDU 1570 * NA	1570 + USES ALL  1571	1570 • USES ALL  1570 • USES ALL  1571 • USES ALL  1572 • USES ALL  1573 UFD EQU •  1574 ANA B WI A, UO. DDU  1575 ANA B WI I, E WOT TO HANDLE UPDATE  1576 ANA C A, M  1581 HOV A, M  1582 HOV B, A  1583 HOV A, M  1583 HOV A, M  1584 HOV A, M  1585 HOV A, M  1585 HOV A, M  1586 AN DISPLAYING RECISTERS.  1580 AN DISPLAYING RECISTERS.  1590 HOV A, M  1	1570 * USES ALL  1570 * USES ALL  1571	1570 * USES ALL  1570 * USES ALL  1571 UFD 60U * USES ALL  1573 UFD 60U * USES ALL  1575 ANA 8	1970 * USES ALL  1970 * USES ALL  1971	1770   1785   ALL     1771   1770   1785   ALL     1771   1770   1785   ALL     1771   1771   AND BOD     1772   AND BOD     1775   AND BOD     1775   AND BOD     1776   AND BOD     1776   AND BOD     1777   AND BOD	1570 - 0.550 ALL  1571	1571 1571 1571 1573 160 1574 177 177 177 177 177 177 177 177 177 1	1571 1573 160 1574 177 177 177 177 177 177 177 177 177 1	1571 1572 1573 1574 1575 1576 1577 1577 1577 1577 1577 1577	1571 1572 1573 1574 1575 1576 1577 1577 1577 1577 1577 1577	1571 1571 1573 1580 1574 1580 1575 1580 1580 1580 1580 1580 1580 1580 158	1577 UFD 600	1574   WFD   EQU   +	1573 UFD   640	002 1574 ANA 8.100.00U 1575 ANA 8.100.00U 1576 RNZ 1577 RNZ 1577 RNZ 1577 RNZ 1577 RNZ 1578 RNZ 1580 RCC 1580 R
1562   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1565   1565   1565   1566   1666   1566   1666   1566   1666   1566   1666	1562 *	1562 + 1562 + 1562 + 1562 + 1563 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1566 + 15	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 + ENTRY (H,L) - ADDRESS OF REFCAIT  1568 + EXIT NONE  1570 + USES ALL  1571 - USES ALL  1571 - ANA B. IF NOT TO HANOLE UPDATE  1572 - ANA B. IF NOT TO HANOLE UPDATE  1573 - ANA B. IF NOT TO HANOLE UPDATE  1574 - ANA B. IF NOT TO HANOLE UPDATE  1575 - ANA B. IF NOT TO HANOLE UPDATE  1576 - ANA B. IF NOT TO HANOLE UPDATE  1577 - ANA B. IF NOT TO HANOLE UPDATE  1578 - ANA B. IF NOT TO HANOLE UPDATE  1580 - AND A.A.  1581 - AND A.A.  1581 - AND A.A.  1584 - AND A.A.  1584 - AND A.A.  1584 - AND SPLATICRA  1584 - AND SPLATICRA  1586 - AND A.A.  1587 - AND A.A.  1589 - AND SPLATICRA  1590 - AND SP	1564 * UJED IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * ITHE TO UPDATE THE DISPLAY CONTENTS, CURENILY, THIS 1568 * ENTRY 12 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY 14.1) - ADDRESS OF REFCAT 1568 * EXIT NOWE	1564 * 1 UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1564 * 1 TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILLY, THIS 1567 * EVERY 32 INTERRUPTS, OR ABOUT 32 ITHES A SECOND. 1568 * ENTRY (H-L.) = ADDRESS OF REFORT 1569 * ENTRY (H-L.) = ADDRESS OF REFORT 1569 * ENTRY (H-L.) = ADDRESS OF REFORT 1573 UFD EQU * O.55 ALL 1573 UFD EQU * O.55 ALL 1574 ANA 8 NOT A ***  1570 * 1574 ANA 8 ***  1571 * ANA 8 ***  1572 * ANA 8 ***  1573 * ANA 8 ***  1574 * ANA 8 ***  1575 * ANA 8 ***  1576 * ANA 8 ***  1577 * ANA 8 ***  1578 * ANA 8 ***  1579 * ANA 8 ***  1570 * ANA 8 ***  1570 * ANA 8 ***  1570 * ANA 8 ***  1571 * ANA 8 ***  1571 * ANA 8 ***  1571 * ANA 8 ***  1572 * ANA 8 ***  1573 * ANA 8 ***  1574 * ANA 8 ***  1574 * ANA 8 ***  1575 * ANA 8 ***  1576 * ANA 1574 ANA 8 ***  1577 * ANA 8 ***  1578 * ANA 1574 ANA 8 ***  1570 * ANA 1574 ANA 8 ***  1780 * ANA 1774 ANA 8 ***  1780 *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 ITHES A SECOND. 1567 * EVERY 32 INTERRUPTS, OR ABOUT 32 ITHES A SECOND. 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 * USES ALL 1572 * USES ALL 1573 * USES ALL 1574 * USES ALL 1574 * USES ALL 1574 * USES ALL 1575 *	1570 * USES ALL  1571 1572 1573 1574 1574 1575 1577 1577 1577 1577 1577	1570 * USES ALL  1571	1570 * USES ALL  1571	1570 * USES ALL  1571	1570 • 0355 ALL  1570 • 0355 ALL  1571	1570 - 0355 ALC  1571	1571 1571 1572 1573 1574 1574 1575 1575 1575 1575 1575 1576 1576 1577 1577	1571 1572 1573 1574 1575 1575 1575 1575 1576 1576 1577 1577	1571	1571   1571   1572   1572   1572   1573   1573   1574   1574   1574   1575   1775	1571   1571   1572   1572   1572   1573   1573   1574   1574   1574   1575   1775	1571 1572 1573 1690 1574 1575 1575 1575 1576 1576 1576 1576 1576	1572   UFD   EQU   +	1573   UFO   EQU   +	1573 UFD   EQU   +	002 1574
1562 • 1562 • 1564 • 160 IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1564 • 1664 • 1714 IO UPDATE THE DISPLAY CONTENTS, CURRENTY, THIS 1566 • 1714 IO UPDATE THE DISPLAY CONTENTS, CURRENTY, THIS 1566 • 1714 IO UPDATE THE DISPLAY CONTENTS, CURRENTY, THIS 1566 • 1714 IO UPDATE THE DISPLAY CONTENTS OF REFORM 1550 • 1574 • 1055 •	1562 *	1562   1562   1563   1564   1565   1564   1565   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1564   1565	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + ENTRY (H,L) - ADDRESS OF REFCAT 1568 + EXIT NONE 1570 - USES ALL 1571 - USES ALL 1571 - WA A, UO.DOU 1572 - WA B 1573 - WPO EQU + 1574 - WAI A, UO.DOU 1575 - WA B 1576 - WA B 1577 - WA B 1578 - WA B 1579 - WAI A, UO.DOU 1570 - WAI A, UO.DOU 1570 - WAI A, WA B 1571 - WA B 1571 - WAI A, WA B 1580 - WAI B 1580 - WAI B 1580 - WAI B 1580 - WAI DISPLAYING RECISTERS 1590 - WAI WAI H, DSPA (H,L) - REG NAME PATTERN 1590 - WAI WAI H, DSPA (H,L) - REG NAME PATTERN 1590 - WAI WAI H, DSPA (H,L) - REG NAME PATTERN 1590 - WAI H, DSPA H, H, H, H, DSPA H, H, H, DSPA H, H, H, H, DSPA H, H, H, H, DSPA H,	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS 1566 * ENTRY 12 INTERRUPTS, OR ABOUT 32 ITHES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1569 * ENTRY (H.L.) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * ANA 1	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1564 • ITAE TO UPDATE THE DISPLAY CONTENTS. CURRENILLY, THIS 1566 • EVERY 32 INTERRUPTS, OR ABOUT 32 ITHES A SECONO. 1567 • EVERY 32 INTERRUPTS, OR ABOUT 32 ITHES A SECONO. 1568 • ENTRY (H.L.) - ADDRESS OF REFCAT 1570 • USES ALL 1571 • USES ALL 1571 • USES ALL 1572 • USES ALL 1573 UFD EQU • USES ALL 1574 • MAI AND BOOK 1574 • MAI MAI AND BOOK 1574 • MAI MAI MAI HOSPA (H-L.) - ADDRESS OF REG NAME 1574 • MOY HAN HAN HAND HAND HAND HAND HAND HAND H	1564 * 1 UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * 17.04	1570 * USES ALL   1570 * USES ALL   1570 * USES ALL   1572   USES ALL   1572   USES ALL   1572   USES USES USES USES USES USES USES US	1570 * USES ALL   1570 * USES ALL   1570 * USES ALL   1572   USES ALL   1572   USES ALL   1572   USES ALL   USES ALL   USES WAY	1570 • USES ALL  1570 • USES ALL  1571	1570 * USES ALL 1570 * USES ALL 1571 UFO EQU * USES ALL 1572 UFO EQU * USES ALL 1574 ANA B	1570 • USES ALL  1571	1570	1571 1572 1574 1574 1575 1575 1577 1577 1577 1577	1571 1572 1674 1674 1677 1677 1677 1677 1677 1677	1571   1571   1572   1572   1574   1574   1574   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1575   1775	1571   1572   1572   1573   1574   1575	1571   1572   1572   1573   1574   1575	1571 1573 1670 1574 1574 1575 1577 1577 1577 1577 1577	1573 JFD 600 **  1573 JFD 600 **  1574 HVI A, UO. DDU  1575 ANA B  1576 ANA B  1577 ANA B  1577 ANA B  1580 ANA B  1581 AND A, M  1581 AND A, M  1581 AND A, M  1582 AND A, M  1584 AND A, M  1585 AND A, M  1585 AND A, M  1586 AND A, M  1589 AN DISPLAYING RECISTERS ADDRESS OF REG NAME  1590 AND A, M  1590 AND A, M  1590 AND B, A  1590 A	1573   UFD   EQU   **   1574   MVI	1573 UFD   EQU   +	002 1574
1562 +   1562 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1567 +   1568 +   114E TO UPDATE THE DISPLAY CONTENTS, CURRENILY, THIS Liber	1562 *	1562 + 1562 + 1564 + 1664 + 16	1566   *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * EVERY 32 INTERRUPTS, CONTENTS, CURRENTLY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H+L) - ADDRESS OF REFCAT   1569 * ENTRY (H+L) - ADDRESS OF REFCAT   1570 * USES ALL   1571 * NONE	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1564 * ITAE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1567 * EVERY 32 INTERRUPTS, OR ABOUT 34 ITAES A SECONO. 1569 * ENTRY (HLI) = ADDRESS OF REFORT 1569 * ENTRY (HLI) = ADDRESS OF REFORT 1570 * USES ALL 1571 * USES ALL 1572 * USES ALL 1573 UFD EQU * USES ALL 1574 ANA B * USES ALL 1575 ANA B * USES ALL 1575 ANA B * USES ALL 1577 ANA B * USES ANA B * U	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILLY, THIS 1566 * ENTRY (H.L.) = ADDRESS OF REFORT 1569 * ENTRY (H.L.) = ADDRESS OF REFORT 1569 * ENTRY (H.L.) = ADDRESS OF REFORT 1573 UFD EQU * USES ALL 1573 UFD EQU * USES ALL 1574 ANA B WILL A, UG, DDU WALL 1574 ANA B WALL 1574 ANA B WALL 1575 ANA B WAL	1570 * USES ALL     1572 * USES ALL     1573 UFD EQU *	1570	1570 • USES ALL  1570 • USES ALL  1571 • USES ALL  1572 UFD EQU •  1573 UFD EQU •  1574 ANA B  1576 ANY C  1576 ANY C  1577 ANY C  1578 ANY C  1570 ANY C  1570 ANY C  1571 ANY C  1571 ANY C  1571 ANY C  1572 ANY C  1573 ANY C  1574 ANA B  1575 ANY C  1576 ANY C  1577 ANY C  1578 ANY C  1578 ANY C  1578 ANY C  1581 ANY C  1582 ANY C  1584 ANY C  1584 ANY C  1585 ANY C  1586 ANY D.SPHOD-DSPROT-I  1584 ANY C  1585 ANY C  1586 ANY D.SPHOD-DSPROT-I  1586 ANY DISPLAYING REGISTERS  1586 ANY D.SPHOD-DSPROT-I  1587 ANY D.SPHOD-DSPROT-I  1588 ANY D.SPHOD ANY  1589 ANY D.SPHOD ANY  1580 ANY D.SPHOD ANY ANY  1580 ANY	1570 * USES ALL  1570 * USES ALL  1571 UFD EQU * USES  1573 UFD EQU * USES  1574 ANA B UG. DDU  1576 ANY B IF NOT TO HANDLE UPDATE  1577 AVI L.* DSPROT  1581 ANY B IF NOT TO HANDLE UPDATE  1582 ANY B AND ESPROT-I  1583 HOV A.M AND	1570	1570	1571 1572 1574 1574 1575 1575 1577 1577 1577 1577	1571 1571 1572 1573 1600 1574 1575 1575 1575 1575 1576 1577 1576 1577 1577	1571 1572 1574 1575 ANA 8 1575 ANA 8 1576 ANA 8 1577 ANA 8 1577 ANA 8 1577 ANA 1587 ANA 1587 ANA 1587 ANA 1588 AND	1571 1572 1573 1600 1574 1575  ANA  B 1575  ANA  B 1576  ANA  B 1576  ANA  B 1577  ANA  B 1580  1577  ANA  B 1580  1581  ANY  ANY  ANA  1581  ANY  ANA  ANA  ANA  ANA  ANA  ANA  AN	1571 1572 1573 1600 1574 1575  ANA  B 1575  ANA  B 1576  ANA  B 1576  ANA  B 1577  ANA  B 1580  1577  ANA  B 1580  1581  ANY  ANY  ANA  1581  ANY  ANA  ANA  ANA  ANA  ANA  ANA  AN	1571 1572 1574 1575 1577 1577 1577 1575 1576 1577 1576 1577 1576 1577 1576 1577 1577	1573 JFD 60U **  1573 JFD 60U **  1574 HVI A,UO.DDU  1575 ANA B IF NOT TO HANOLE UPDATE  1576 ANI C,*DSPROT  1581 HOV A,H  1583 HOV A,H  1583 HOV A,H  1584 ANI Z  227 003 1588 JZ  227 003 1588 AN DISPLAYING RECISTERS  1589 AN DISPLAYING RECISTERS  1590 AN DISPLAYING RECISTERS  1500 AND DISPLAYI	1573   UFD   EQU   **   1574   MVI	1574   FOUNTY   FOU	1574   1574   1574   1574   1574   1575   1575   1575   1577
1562 +   1562 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1567 +   1570 +   1	1562 +  1564 +  1564 +  1564 +  1564 +  1564 +  1566 +  1716 TO UPDATE THE CLOCK INTERRUPT PROCESSOR WHEN 1566 +  1566 +  1566 +  1566 +  1716 TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS 1569 +  1560 +  1560 +  1560 +  1560 +  1560 +  1560 +  1560 +  1570 -  1571	1562 + 1562 + 1564 + 1564 + 1564 + 1565 + 15	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571 * NONE  1572 ANA B  1573 UFD EOU *  1574 ANA B  1575 ANA B  1575 ANA B  1576 ANA B  1577 ANA B  1578 ANA B  1579 ANA ANA B  1580 INV	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 • EVERY 32 INTERRUPTS, GARGATZ, CURRENTLY, THIS 1566 • EVERY 32 INTERRUPTS, GARGATZ, CURRENTLY, THIS 1567 • ENTRY (H+L) - ADDRESS OF REFCRIT 1569 • ENTRY (H+L) - ADDRESS OF REFCRIT 1570 • USES ALL 1571 • USES ALL 1571 • USES ALL 1572 • USES ALL 1573 • USES ALL 1574 • WILL A+UO.DDU 1574 • WILL A+UO.DDU 1574 • USES ALL 1574 • USES ALL 1575 • USES ALL 1576 • USES ALL 1576 • USES ALL 1576 • USES ALL 1577 • USES ALL	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 • ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS 1566 • ENTRY 12 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 • ENTRY 14.11) - ADDRESS OF REFCAT 1568 • EXIT NONE 1570 • USES ALL 1571 • USES ALL 1572 ANA B IFF NOT TO HANDLE UPDATE 1573 UFD EOU • ANA B IFF NOT TO HANDLE UPDATE 1574 ANA B IFF NOT TO HANDLE UPDATE 1575 ANA B IFF NOT TO HANDLE UPDATE 1576 ANA B IFF NOT TO HANDLE UPDATE 1577 ANA B IFF NOT TO HANDLE UPDATE 1578 ANA B IFF NOT ANA B IFF NOT TO HANDLE UPDATE 1579 AND ANA ANA B IFF HENDRY 1580 BISS OF BEAND OF THE HANDLE UPDATE 1581 AND ANA B IFF HENDRY 1582 AND DISPLAYING RECISTERS. 1584 AND DISPLAYING RECISTERS. 1585 AND DISPLAYING RECISTERS. 1589 AND DISPLAYING RECISTERS. 1589 AND AND BOOM AND THE HANDLE PATTERN 1589 AND THE HANDLE PATTERN 1589 AND THE HANDLE DADRESS OF REC NAME 1589 HOV AND THE HANDLE PATTERN 1589 HOV AND HOV AND THE HANDLE PATTERN 1589 HOV AND HOV AND HOV AND HANDLE PATTERN 1589 HOV AND HANDLE PATTERN 1589 HOV AND HANDLE PATTERN 1580 HOV AND HAND	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS 1560 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1560 * ENTRY (H.I.) * ADDRESS OF REFCAT 1560 * ENTRY (H.I.) * ADDRESS OF REFCAT 1570 * USES ALL	1570 * USES ALL  1572	1570 * USES ALL  1572 USES ALL  1573 UFD EQU *  002 1574 HVI A.UG.DDU  1575 AND B	1570 • USES ALL  1570 • USES ALL  1571 • USES ALL  1571 • USES ALL  1572	1570 • USES ALL  1570 • USES ALL  1571 • USES ALL  1571 • USES ALL  1572 • USES ALL  1573 UFD EOU • USES ALL  1574 AND B	1570 • 03E3 ALL  1571 1570 • 03E3 ALL  1571 1571	1572	1570	1571 1572 1573 1574 1575 1575 1575 1576 1577 1577 1577 1577	1571 1572 1573 1574 1575 1575 1575 1576 1576 1577 1577 1577	1571 1572 1573 1574 1575 1577 1577 1577 1577 1577 1577	1571 1572 1573 1574 1575 1577 1577 1577 1577 1577 1577	1571 1571 1573 160 1574 1775 1776 1777 1777 1777 1777 1777 1777	1574 UFD EQU + 1574 MAI A,UO.DOU  1575 ANA B 1575 ANA B 1576 ANA C, L. & DSPROT 1577 ANA B, TF NOT TO HANDLE UPDATE 1587 AND A,M 1581 AND A,M 1583 AND A,M 1584 ERRNZ DSPHOD-DSPROT-I 1585 AND A,M 1585 AND A,M 1585 AND A,M 1586 JZ UFD I EMENSY 1589 AM DISPLAYING REGISTERS. 1589 AM DISPLAYING REGISTERS. 1590 AM DISPLAYING REGISTERS.	1573 UFD	1574 HVI 4,00.00U  1575 ANA B  1575 ANA B  1576 ANA B  1577 ANA B  1577 ANI L.,05PROT  1580 HOV A,H  1581 HOV A,H  1581 HOV B,A  1584 ERRNZ DSPROT-1  1584 HOV A,H  1585 ANI Z  1584 ANI Z  1585 ANI Z  1586 ANI Z  1589 AN DISPLAYING RECISTERS  1590 AN DISPLAYING RECISTERS  1591 CALL LRA  1594 HOV A,H  1590 AN DISPLAYING RECISTERS  1591 CALL LRA  1591 HOSPA  1596 HOV H,H  1597 HOV H,H  1596 HOV H,H  1597 HOV	1574   1774   1774   1774   1774   1774   1774   1774   1774   1774   1774   1774   1774   1774   1774   1774   1774   1774   1775
1562 +   1562 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1566 +   1566 +   1566 +   1184   10   10   10   10   10   10   10   1	1562 + 1563 + 1564 + 1564 + 1564 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1566 + 15	1562 +   1562 +   1564	1566   *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1566 * EVERY 32 INTERRUPTS, CONTENTS, CURRENTLY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H+L) - ADDRESS OF REFCAT   1569 * EXTT NONE   1570 * USES ALL   1571 * NONE   1572 * ANA	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 • ENTRY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 • ENTRY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 • ENTRY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 • ENTRY NOWE 1590 • AND 1571 • USES ALL 1570 • USES ALL 1570 • USES ALL 1571 • USES ALL 1571 • USES ALL 1572 • USES ALL 1572 • USES ALL 1574 • USES ALL 1574 • USES ALL 1574 • USES ALL 1574 • USES ALL 1575 • USES ALL 1574 • USES ALL 1575 • USES ALL 1570 • USES AL	1564	1570 * USES ALL  1571 1572 1573 1574 1574 1575 1575 1575 1576 1576 1577 1577 1577	1570 * USES ALL  1571 1572 1573 1574 1577 1577 1577 1578 1579 1579 1570 1570 1570 1570 1570 1570 1570 1570	1570 • USES ALL  1571	1570 • USES ALL  1570 • USES ALL  1571 1572 • USES ALL  1573 UFD EQU • • • • • • • • • • • • • • • • • • •	1570	1570	1572	1571 1572 1573 002 1574 006 1575 ANA B ANA	1571 1571 1571 1572 002 1574 ANA 8	1571   1571   1571   1572   1573   UFD   600   4   1574   MVI   A,00.000   1574   MVI   A,00.000   1575   ANA   B   IF NOT TO HANGLE UPDATE   1575   ANY   L,055PROT   1577   ANY   L,055PROT   1580   ANY   L,055PROT   1581   ANY   L,057PROT   1581   ANY   A,00.000   1585   ANY   A,00.000   1585   ANY   A,00.000   1580   ANY   A,00.000   1580   ANY   A,00.000   1580   ANY   A,00.000   1580   ANY   A,00.000   A,00.0000   A,00.00000   A,00.0000   A,00.0000   A,00.0000   A,00.0000   A,00.00000   A,00.0	1571   1571   1571   1572   1573   UFD   600   4   1574   MVI   A,00.000   1574   MVI   A,00.000   1575   ANA   B   IF NOT TO HANGLE UPDATE   1575   ANY   L,055PROT   1577   ANY   L,055PROT   1580   ANY   L,055PROT   1581   ANY   L,057PROT   1581   ANY   A,00.000   1585   ANY   A,00.000   1585   ANY   A,00.000   1580   ANY   A,00.000   1580   ANY   A,00.000   1580   ANY   A,00.000   1580   ANY   A,00.000   A,00.0000   A,00.00000   A,00.0000   A,00.0000   A,00.0000   A,00.0000   A,00.00000   A,00.0	1571 1571 1573 1600 1574 1773 1774 1774 1774 1774 1774 1774 17	1574 UFD EQU	1573   UFO   EQU   +	1573 UFD	002 1575 UPD END TAYLOLODU ALUOLODU CO
1562 +   1562 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1564 +   1565 +   1565 +   1566 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1567 +   1570 +   1	1562 +  1563 +  1564 +  1564 +  1565 +  1710	1562	1566	1564 1 UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 1 EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 • EMTRY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 • EMTRY 111 - ADDRESS OF REFORT 1569 • EMTRY 1111 - ADDRESS OF REFORT 1569 • EMTRY 1111 - ADDRESS OF REFORT 1570 • USES ALL 1570 • USES ALL 1571 • WAY A, WAY B B DORD 1577 • WAY B B DORD 1570 • USES ALL 1570 • WAY A, WAY B B DORD 1570 • WAY A, WA	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS 1566 * THE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS 1566 * THE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS 1566 * THE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS IS69 * THIRD NONE SALL SALL SALL SALL SALL SALL SALL SAL	1570 * USES ALL  1571 1572 1573 1574 1576 1577 AND A, UG.DDU 1575 AND B 1576 AND A, M 1580 AND B, A 1584 AND B, A 1584 AND B 1584 AND A, M COOZ 1585 AND A, M COOZ 1585 AND A, M COOZ 1585 AND A, M COOZ 1586 AND A, M COCZ 1586 ANDRESS OF REG NAME 1596 AND A, M COCZ 1596 AND A, M COCZ 1596 ANDRESS OF REG NAME 1597 ANDRESS 1596 ANDRESS 15	1570 * USES ALL  1571 USES ALL  1572 USES ALL  1573 UF GOU	1570 • USES ALL  1570 • USES ALL  1571 • USES ALL  1571 • USES ALL  1572 • USES ALL  1573 UFD EOU • OOZ  1574 ANA B I IF NOT TO HANDLE UPDATE  1575 ANA B IF NOT TO HANDLE UPDATE  1576 ANY A.M ROTATE PATTERN  1580 RUC N.A ROTATE PATTERN  1581 HOV N.A ROTATE PATTERN  1582 HOV A.M Z  1584 AND Z  1585 AND Z  1585 AND Z  1586 AND Z  1580 AND ISPLAYING RECISTERS.  1580 AND ISPLAYING RECISTERS  1590 AND ISPLAYING RECISTERS  1591 HOV A.M  1595 HOV A.M  1596 HOV A.M  1599 HOV A.M  1599 HOV L.A RODRESS OF REC NAME PATTERN  1599 HOV L.A RODRESS OF REC NAME PATTERN	1570 • USES ALL  1570 • USES ALL  1571 1571 1572 1573 002 1575 ANA B 1575 ANA B 1576 ANZ ANA B 1577 ANZ B 1580 ANZ ANA B 1581 ANZ ANA B 1582 ANZ	1570	1571 1570	1570 - 0.555 ALC  1572	1571 1571 1572 1573 1574 ANA A, 100.000 1575 ANA B 1576 ANA B, 100.000 1576 ANA A, 100.000 1578 ANY C, 100.000 1581 ANY C, 100.000 1581 ANY C, 100.000 1584 ANY C, 100.000 A, 100.000	1571 1571 1571 1572 1573 UFO 60U *  002 1574 ANA B 4,00.000U 1576 ANA B 7,00.000U 1576 ANY C, **D\$PROT 1581 ANY C, **D\$PROT 1582 ANY C, **D\$PROT 1583 ANY C, **D\$PROT 1584 ANY C, **D\$PROT 1585 ANY C, **D\$PROT 1586 ANY A, **D\$PROT 1589 AN DISPLAYING RECISTERS. 1590 AN DISPLAYING RECISTERS. 1590 AN DISPLAYING RECISTERS. 1590 ANY A, **H 1590 ANY A, **H 1590 ANY A, **H 1590 ANY A, **H 1590 ANY H, **H 1590 ANY H, **H 1599 ANY H, *	1571   1571   1571   1572   1573   1573   1573   1574   1574   1574   1574   1574   1574   1574   1575   1574   1775	1571   1571   1571   1572   1573   1573   1573   1574   1574   1574   1574   1574   1574   1574   1575   1574   1775	1571 1572 1573 1670 1574 1574 1575 1574 1575 1575 1576 1576 1576 1577 1577 1577	1572 UFD EQU #  002 1574 MVI A,UQ.DDU  1575 ANA B  1575 ANA B  1576 RNZ  1577 AVI L,#DSPROT  1580 HOV A,H  1581 HOV B,A  1583 HOV B,A  1584 RRNZ OSPHOD-DSPROT-I  1584 RRNZ OVPHOD-DSPROT-I  1585 ANI Z  227 003 1586 ANI Z  227 003 1590 # AN DISPLAYING RECISTERS  047 003 1594 CAL LRA  1589 AN DISPLAYING RECISTERS  1590 AND A,H  1590 # AN DISPLAYING RECISTERS  1590 # AN DISPLAYING RECISTER ADDRESS  1590 # AN DISPLAYING RECISTERS  1590 # AN DISPLA	1574   WYI   A, UO. DOU     1574   WYI   A, UO. DOU     1575   RNZ   B   IF NOT TO HANDLE UPDATE     1576   RNZ   RNZ   IF NOT TO HANDLE UPDATE     1577   WYI   L, EDSPRÜT   IF NOT TO HANDLE UPDATE     1580   RLC   A, M	1573 UFD	002 1574 HW 41 4,00.00U  1575 ANA B 1576 ANA B 1577 HW 1 L,#DSPROT 1580 HOV A,N 1581 HOV A,N 1582 HOV B,A 1584 ERRNZ OSPHOD-DSPROT-1 1584 HOV A,N 1585 HOV A,N 1584 HOV A,N 1585 HOV A,N 1586 ANI Z 227 003 1588 JZ OFFO DE CALL LRA 1589 AN DISPLAYING RECISTERS. 1590 AN DISPLAYING RECISTERS OF REC NAME 1599 HOV A,N 1590 HOV A,N 1590 HOV A,N 1590 HOV H,N
1562 * 1562 * 1562 * 1563 * 1563 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1565 * 1565 * 1566 * 1666 * 1566 * 1666 * 1566 * 16	1562 +  1563 +  1564 -  1564 -  1564 -  1564 -  1566 -  1714 TO UPDATE THE UISPLAY CONTENTS, CURRENTLY, THIS Libbor to UPDATE the UPDATE to USES ALL LIBBOR THE UPDATE TO USE TO US	1562	1566	1564 1 UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 1 LINE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 1 EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568	1564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 # FINE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS 1566 # FURRY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 # ENITY (H.L.) = ADDRESS OF REFCRIT 1560 # EXIT NOWE USES ALL 1570 # USES ALL 1570 # USES ALL 1570 # USES ALL 1570 # USES ALL 1571 # USES ALL 1572 # USES ALL 1574 # WAN A HOND TO HANDLE UPDATE 1574 # WAN A HOND TO HANDLE UPDATE 1570 # USES ALL 1570	1564 * UFD IS CALLED BY THE CLUCK INTERRUPT PROCESSOR WHEN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * ENITY (H+L) = ADDRESS OF REFCRIT 1569 * EXIT MONE USES ALL 1570 * USES ALL 1570 * USES ALL 1571 * ANA 8	1570 * USES ALL  1571 1571 1572 1573 1674 1574 ANA B 1575 000 1576 ANA 1580 1580 1581 1581 1581 1581 1581 1581	1570 • USES ALL  1571 1572 1573 1574 002 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANI 1580 1577 HDV A,H 1581 HDV A,H 1582 1584 HDV A,H 1583 1NX HDV A,H 1584 ANI 1584 ANI 1584 ANI 1584 ANI 1584 ANI 1590 AN DISPLAYING RECISTERS 1594 AN DISPLAYING RECISTERS 1595 AN DISPLAYING RECISTERS 1594 AN DISPLAYING RECISTERS 1594 AN DISPLAYING RECISTERS 1595 AN DISPLAYING RECISTERS 1595 AN DISPLAYING RECISTERS 1596 AN DISPLAYING RECIST	1570 • USES ALL  1571 1572 1573 1697 1574 ANA B	1570 • USES ALL  1571	1570 • 0355 ALL  1571	1570 • 0355 ALL  1571	1570	1571 1571 1572 1573 000	1571 1571 1571 1571 1572 1573 1574 1575 1575 1575 1575 1576 1577 1577 1577	1571   1571   1571   1571   1571   1571   1571   1572   1573   1574   1771   1574   1771   1574   1771   1771   1772	1571   1571   1571   1571   1571   1571   1571   1572   1573   1574   1771   1574   1771   1574   1771   1771   1772	1571 1571 1572 1573 160 1574 1574 1575 ANA  8	1572 UFD EQU * 1573 UFD EQU * 1574 ANA B 160 EQU * 1575 ANA B 174 A.UO.DDU 1576 ANA B 175 IF NOT TO HANOLE UPDATE 1577 HVI L.#DSPROT 1580 HOV A.H 1581 HOV B.A. 1583 INX HOV B.A. 1584 AND A.H 1585 ANI Z 227 003 1586 ANI Z 227 003 1589 AN DISPLAYING REGISTERS. 1590 * AND DISPLAYING REGISTERS. 1591 AND DISPLAYING REGISTERS. 1592 AND DISPLAYING REGISTERS. 1593 HOV A.H 1594 AND DISPLAYING REGISTERS. 1596 AND DAD D CALL LRA 1596 HOV A.H 1596 HOV H.H 1597 HOV H.H 1596 HOV H.H	1573 UFD EQU **  002 1574 MVI A,UO.00U  1576 ANA B  1576 ANA B  1577 IF NOT TO HANDLE UPDATE  1587 MVI L, #DSPROT  1581 MOV A,N  1583 MOV B,A  1584 ERNX DSPHOD-DSPROT-1  1584 MOV A,N  002 1586 ANI  227 003 1586 ANI  227 003 1586 ANI  1590 * AN DISPLAYING RECISTERS  1590 AN DISPLAYING RECISTERS  1590 WOV A,N  1591 H,DSPA  142 003 1596 WOV A,N  1596 HOV A,N  1597 INX H,DSPA  1597 INX H,DSPA  1697 INX H,DSPA  1698 HOV A,N  1698 HOV A,N	1573 UFD   EQU   #   W   M   M   W   M   M   M   M   M   M	002 1574 ANA 8 IF NOT TO HANOLE UPDATE 1575 ANA 8 IF NOT TO HANOLE UPDATE 1576 ANA 8 IF NOT TO HANOLE UPDATE 1577 AVI L'. **DSPROT 1580 RLC A.** 1581 MOV A.** 1584 MOV A.** 1584 MOV A.** 1584 MOV A.** 1585 MOV A.** 1585 AND A.** 1585 AND A.** 1590 * AN DISPLAYING RECISTERS. 1591 CALL LRA LOCATE RECISTER ADDRESS OF REC NAME 1595 MOV A.** 1591 AND MOV A.** 1593 HOV A.** 1594 HOV A.** 1595 HOV A.** 1596 HOV A.** 1597 HOV H.** 1597 HOV H.** 1598 HOV H.** 1598 HOV H.** 1599 HOV H.** 1590 HOV H.**
1562 * 1562 * 1562 * 1562 * 1563 * 1563 * 1563 * 1563 * 1563 * 1564 * 1564 * 1564 * 1564 * 1565 * 1565 * 1565 * 1566 * 15	1562 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1563 ** ENTRY (H,L) ** ADDRESS OF REFCNT 1569 ** EXIT NONE 1569 ** EXIT NONE 1570 ** USES ALL 1571 ** USES ALL 1571 ** USES ALL 1572 ** USES ALL 1574 ** WYI A,UO.DOU ** IF NOT TO HANOLE UPDATE 1574 ** WYI A,UO.DOU ** IF NOT TO HANOLE UPDATE 1574 ** WYI A,UO.DOU ** AOTATE PATTERN 1570 ** WYI A,UO.DOU ** AOTATE PATTERN 1580 ** AND ISPLAYING RECISTERS.** IF HENDRESS OF REG NAME 1590 ** AND ISPLAYING RECISTERS.** IF HENDRESS OF REG NAME 1590 ** AND ISPLAYING RECISTERS.** IF HENDRESS OF REG NAME 1590 ** AND ISPLAYING RECISTERS.** IN HIDSPLAYING A,H INDSPACE IN HENDRESS OF REG NAME 1590 ** HOV A,H INDSPACE IN HENDRESS OF	1562 + 1562 + 1563 + 1564 + 1564 + 1565 + 1565 + 1565 + 1565 + 1670 IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN I 1565 + 170	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) = ADDRESS OF REFCRI  1568 * EXIT NONE  1571 USES ALL  1572 USES ALL  1573 UFO EQU * *  1574 MYI A,UQ.DDU  1574 ANA B  1575 ANA B  1576 ANA B  1577 ANA CANA  1580 ANA CANA  1580 ANA CANA  1581 MOV A,M  1584 AND CANA  1584 AND CANA  1586 AND CALL LRA  1589 AN DISPLAYING RECISTERS.  1590 AN DISPLAYING RECISTERS.  1591 CALL LRA  1592 CALL LRA  1594 AND SPACE ANDRESS OF REC NAME  1595 AND CANA  1596 AND CANA  1596 AND A,M  1597 AND CANA  1598 AND CANA  1599 AND CANA  1599 AND CANA  1590 ANDRESS OF REC NAME	1564	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN I 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN I 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * ENTRY (H-L) - ADDRESS OF REFCAT 1569 * ENTRY (H-L) - ADDRESS OF REFCAT 1569 * ENTRY (H-L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * AN A B B B B B B B B B B B B B B B B B	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN I 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I 1560 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1564 * ENITRY (H-L) - ADDRESS OF REFCRIT I 1560 * ENITRY (H-L) - ADDRESS OF REFCRIT I 1570 * USES ALL I 1571 A 100	1570 * USES ALL  1571 1571 1572 1573 002 1574 177 006 1578 007 1579 007 1584 1584 1585 107 107 1081 1587 1587 1081 1587 1081 1588 1081 1589 1091 1091 1091 1091 1091 1091 1091 10	1570 * USES ALL  1571 1571 1572 1573 1673 UFD EQU *  002 1574 NVI A,U0.00U 1575 ANA B 1 IF NOT TO HANDLE UPDATE 1575 ANA A,N 1580 RLC 1580 RLC 1581 HOV A,N 1581 HOV B,A 1581 HOV B,A 1584 ERRNZ DSPHOD-DSPROT-I 1584 ERRNZ DSPHOD-DSPROT-I 1585 AND A,N 1585 AND A,N 1586 JZ UHLD ABUSS 227 003 1588 JZ UFDI IF HEHJRY 1589 AN DISPLAYING RECISTERS. 1590 AN DISPLAYING RECISTERS. 1590 AN DISPLAYING RECISTERS. 1590 AND A,N 1591 AND A,N 1592 BOD D 1596 HOV A,N 1597 HADSPA	1570 • USES ALL  1571 1572 1573 1574 002 1574 006 1578 006 1578 007 1580 007 1581 007 1581 007 1582 007 1584 007 1585 007 1585 007 1585 007 1585 007 1586 007 1589 00	1570 • USES ALL  1571	1570 • USES ALL  1571	1570 • 0355 ALL  1571 • 0355 ALL  1572 • 0355 ALL  1573 UFD EQU • • 0002 1574 ANA B	1570	1571 1572 1573 160 1574 1575 1576 1577 1577 1577 1577 1577 1577	1571 1571 1572 1573 1570 1571 1573 1570 1574 1575 1576 1577 1577 1577 1577 1577 1577	1571   1571   1571   1572   1572   1573   UFD   EQU   #	1571   1571   1571   1572   1572   1573   UFD   EQU   #	1571 1572 1572 1574 400 4002 1574 401 400000 1575 401 1576 401 1580 400 400 400 400 400 400 400 400 400 4	1573 UFD EQU +  1574 MVI A,UG.DDU  1575 ANA B  1575 ANA B  1576 ANA B  1577 AVI L, #DSPRUT  1581 HOV A,M  1582 HOV B,A  1583 HOV A,M  1584 ERRN DSPHOD-DSPROT-1  1584 ERRN DSPHOD-DSPROT-1  1584 ANI  227 003 1586 JJZ UFD1  1589 AM DISPLAYING REGISTERS.  1590 AM DISPLAYING REGISTERS.  1591 CALL LRA  242 003 1592 DAD  1596 HOV A,M  342 003 1595 DAD  1596 HOV A,M  1597 HOSPA  1597 HOSPA  1597 HOSPA  1598 HOV A,M  1599 HOV A,M  1596 HOV A,M  1596 HOV A,M  1597 HOSPA  1597 HOSPA  1597 HOSPA  1597 HOSPA  1596 HOV A,M  1597 HOSPA  1597 HOSPA  1597 HOSPA  1597 HOSPA  1596 HOV A,M  1597 HOSPA  1598 HOSPA  1598 HOSPA  1598 HOSPA  1596 HOSPA  1597 HOSPA  1596 HOSPA  1597 HOSPA  1597 HOSPA  1597 HOSPA  1597 HOSPA  1597 HOSPA  1598 HOSPA  1588	002 1574 UFD EQU **  1575 ANA B IF NOT TO HANDLE UPDATE 1576 ANA B IF NOT TO HANDLE UPDATE 1577 AVI L.*BSPROT 1579 ANY A.** 1580 ANY A.** 1584 ANY A.** 1584 ANY A.** 1584 ANY A.** 1585 ANY A.** 1586 ANY A.** 1586 ANY A.** 1589 ANY DEPLAYING RECISTERS. 1590 ** 15	1573 UFD	12/3
1562 *  1564 *  1565 *  1564 *  1565 *  1566 *  1566 *  1566 *  1566 *  1566 *  1566 *  1566 *  1566 *  1566 *  1567 *  1568 *  1568 *  1568 *  1569 *  1569 *  1569 *  1570 *  1570 *  1570 *  1570 *  1570 *  1570 *  1570 *  1570 *  1570 *  1570 *  1570 *  1570 *  1571 *  1571 *  1572 *  1573 *  1574 *  1574 *  1574 *  1575 *  1576 *  1577 *  1578 *  1579 *  1570 *  1570 *  1570 *  1570 *  1570 *  1571 *  1571 *  1572 *  1573 *  1574 *  1574 *  1575 *  1576 *  1577 *  1577 *  1578 *  1570 *	1562 **  1563 **  1564 **  1565 **  1566 **  1566 **  1566 **  1566 **  1567 **  1568 **  1568 **  1568 **  1568 **  1569 **  1569 **  1570 **  1570 **  1570 **  1570 **  1570 **  1571 **  1571 **  1571 **  1572 **  1573 **  1574 **  1574 **  1574 **  1574 **  1575 **  1576 **  1577 **  1578 **  1579 **  1570 **  157	1562 + 1563 + 1564 + 1564 + 1564 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1565 + 1667 + 16	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H+L) - ADDRESS OF REFCAT  1570 * USES ALL  1571 * HONE  1572 * USES ALL  1573 * USES ALL  1574 * MAI  1574 * MAI  1574 * MAI  1575 * USES  1576 * EXIT  NOT A, UG. DDU  1576 * MAI  1577 * MAI  1577 * MAI  1578 * MAI  1579 * MAI  1570 * MAI  1580 * MAI  1581 * MAI  1581 * MAI  1581 * MAI  1581 * MAI  1582 * MAI  1584 * MAI  1584 * MAI  1584 * MAI  1584 * MAI  1589 * AM DISPLAYING RECISIERS.  1590 * AM DISPLAYING RECISIERS.  1591 * CALL  1591 * CALL  1591 * MAI  1594 * MAI  1595 * MAI  1596 * MAI  1597 * MAI  1596 * MAI  1597 * MAI  1597 * MAI  1598 * MAI  1599 * MAI  1590 * MAI	1564	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN I 1564 * EVERY 32 INTERRUPTS, CURRENILY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 INNES A SECOND.  1567 * ENTRY (H+L) - ADDRESS OF REFCNT I 1569 * ENTRY (H+L) - ADDRESS OF REFCNT I 1569 * ENTRY (H+L) - ADDRESS OF REFCNT I 1570 * USES ALL I 1571 * USES ALL I 1572 * USES ALL I 1573 * USES ALL I 1574 * USES ALL I 1575 * USES A	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * ENTRY (H+L) - ADDRESS OF REFCAT 1569 * ENTRY (H+L) - ADDRESS OF REFCAT 1569 * ENTRY (H+L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * NONE 1572 * NONE 1573 * NONE 1574 * NONE 1574 * NONE 1575 * NONE 1576 * NONE 1576 * NONE 1576 * NONE 1576 * NONE 1577 * NONE 1577 * NONE 1578 * NONE 1579 * NONE 1570 * NO	1570 • USES ALL  1571	1570 * USES ALL  1570 * USES ALL  1571	1570 • USES ALL  1572	1570 * USES ALL 1571 1572 1573 1574 1575 002 1574 1576 1576 1576 1576 1577 1577 1577 1577	1570 - 0355 ALL 1571	1570	1571 1572 1573 160 1574  ANA B 1576  ANA B 1576  ANA B 1577  ANA B 1580  ANA A.M  A.M  A.M  A.M  A.M  A.M  A.M  A.M	1571 1571 1572 1573 160 1574 1574 1575 1575 1575 1575 1575 1575	1571 1571 1572 1573 1574 1574 1575 1574 1575 1575 1576 1577 1571 1576 1576 1577 1576 1577 1576 1577 1576 1577 1577	1571   1571   1571   1572   1572   1573   UFD   EQU	1571   1571   1571   1572   1572   1573   UFD   EQU	1572 1573 1574 1575 1575 1577 1577 1576 1576 1576 1576	1573 UFD 60U + 1574 HVI A,U0.0DU  1575 ANA B IF NOT TO HANDLE UPDATE 1577 AVI C, 105PROT 1580 HC A,H 1581 HOV A,H 1584 HOV A,H 1585 HOV A,H 1586 ANI Z 227 003 1586 AN DISPLAYING RECISTERS. 1590 + AM DISPLAYING RECISTERS. 1590 + AM DISPLAYING RECISTERS. 1590 + AM DISPLAYING RECISTERS. 1591 HOSPH H 342 003 1594 AN DISPLAYING RECISTERS. 1592 HOSPH H 1593 HOSPH H 1594 H 1595 HOSPH H 1595 HOSPH H 1596 HOSPH H 1597 HANDSPA	1573   UFD   EQU   +	1573 UFD   69U   4   4   4   4   4   4   4   4   4	1574   WY
1562 + 1563 + 1563 + 1563 + 1564 + 1563 + 1564 + 1564 + 1564 + 1564 + 1566 + 1666 + 16	1562 **  1563 **  1564 **  1565 **  1566 **  1566 **  1566 **  1566 **  1567 **  1568 **  1568 **  1568 **  1568 **  1570 **  1570 **  1570 **  1570 **  1571 **  1571 **  1571 **  1571 **  1572 **  1573 UFD EQU **  1574 MAI A, UQ. DDU  1574 MAI B, A, UQ. DDU  1575 MAI B, A, UQ. DDU  1576 MAI C, EDSPROT  1577 MAI B, A, UQ. DDU  1578 MAI B, A, UQ. DDU  1578 MAI B, A, UQ. DDU  1579 MAI C, EDSPROT  1570 MAI C, EDSPROT  1570 MAI C, EDSPROT  1571 MAI C, EDSPROT  1571 MAI C, EDSPROT  1572 MAI C, EDSPROT  1573 MAI B, A, UQ. DDU  1574 MAI C, EDSPROT  1575 MAI C, EDSPROT  1576 MAI C, EDSPROT  1577 MAI C, EDSPROT  1578 MAI C, EDSPROT  1579 MAI C, EDSPROT  1580 MAI DISPLAYING REGISTERS.	1562 +  1563 +  1564 +  1564 +  1565 +  1664 +  1666 +  1566 +  1666 +  1566 +  1567 +  1668 +	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H-L) - ADDRESS OF REFCAT  1568 * EXIT NONE  1570 • USES ALL  1571 • USES ALL  1572 • USES ALL  1574 • WYI A,UG,DDU  1574 • WYI L, &DDRESS OF REFCAT  1575 • WYI L, &DDRESS OF REFCAT  1576 • WYI L, &DDRESS OF REFCAT  1577 • WYI L, &DDRESS OF REFCAT  1578 • WYI L, &DDRESS OF REFCAT  1579 • WYI L, &DDRESS OF REFCAT  1570 • WYI L, &DDRESS OF REFCAT  1570 • WYI L, &DDRESS OF REG NAME  1580 • AM DISPLAYING REGISTERS.  1580 • AM DISPLAYING REGISTERS.  1580 • AM DISPLAYING REGISTERS.  1590 • AM DISPLAYING REGISTERS.  1591 • AM DISPLAYING REGISTERS.  1592 • AM DISPLAYING REGISTERS.  1593 • AM DISPLAYING REGISTERS.  1594 • AM DISPLAYING REGISTERS.  1595 • AM DISPLAYING REGISTERS.  1596 • AM DISPLAYING REGISTERS.	1565 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IS65 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * ENTRY (H+L) = ADDRESS OF REFCNT  1569 * EXIT NONE  1570 * USES ALL  1571 * NOSES ALL  1572 * AN B  1573 AN B  1574 AN B  1575 AN B  1576 AN C  1580 AN C  1580 AN DISPLAYING RECISTERS.  1580 AN DISPLAYING RECISTERS.  1580 AN DISPLAYING RECISTERS.  1590 AN DISPLAYING RECISTERS.  1590 AN DISPLAYING RECISTERS.  1591 CALL  1592 AN DISPLAYING RECISTERS.  1593 AN DISPLAYING RECISTERS.  1594 AN DISPLAYING RECISTERS.  1595 AND DISPLAYING RECISTERS.  1596 AN DISPLAYING RECISTERS.  1597 AND DISPLAYING RECISTERS.  1598 AND DISPLAYING RECISTERS.  1599 AN DISPLAYING RECISTERS.  1599 AN DISPLAYING RECISTERS.  1599 AN DISPLAYING RECISTERS.  1599 AN DISPLAYING RECISTERS.	1564	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN 1565 * EVERY 32 INTERRUPTS, CONTENTS, CURRENTY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A SECOND.  1567 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A SECOND.  1568 * ENTRY (H,L) - ADDRESS OF REFORT 1569 * ENTRY (H,L) - ADDRESS OF REFORT 1569 * ENTRY (H,L) - ADDRESS OF REFORT 1570 * USES ALL 1570 *	1570 * USES ALL  1571	1570 * USES ALL  1572	1570 * USES ALL  1570 * USES ALL  1571	1570 * USES ALL 1571	1570 * USES ALL 1571 1572 1573 UFD 60U *  1573 UFD 60U *  1574 MVI A,U0.DDU  1575 MVI A,U0.DDU  1576 RNZ 1577 MVI L,#DSPRDT 1580 RLC 1581 MOV A,M 1582 MOV B,A 1583 MOV B,A 1584 MOV A,M 1584 MOV A,M 1584 MOV A,M 1585 MOV A,M 1586 ANI 1587 MOV B,A 1589 ANI 1589 ANI 1589 ANI 1589 ANI 1589 ANI 1589 ANI 1590 ANI	1570 * USES ALL  1571	1571 1572 1573 160 1573 160 1574 1574 1574 1574 1575 1575 1575 1576 1576 1577 177 177 177 177 177 177 177 177 17	1571 1572 1573 1574 1574 1574 1574 1574 1574 1574 1575 1576 1577 1577 1577 1577 1577 1577	1571 1572 002 1573 0F0 002 1574 ANA B 1576 ANA B 1576 ANA B 1577 ANA B 1577 ANA B 1578 ANA B 1580 BC 1580 BC 1580 BC 1580 BC 1581 BC 1581 BC 1582 BC 1584 BC 1	1571 1572 1573 1574 1574 1575 1575 1576 1576 1576 1576 1576 1577 1577	1571 1572 1573 1574 1574 1575 1575 1576 1576 1576 1576 1576 1577 1577	1571 1572 1573 1600 **  002 1574 4V1 A,U0.00U  006 1575 ANA B 1575 ANA B 1576 ANA B 1577 AVI L,*65PROT 1580 ANO A,N 1581 ANO A,N 1584 ANO A,N 1589 AN 1590 A,N 1590 AN 1594 AN 1595 BAD A 150 BAD	1576 1577 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1577 ANA B IF NOT TO HANDLE UPDATE 1577 AND ANA B IF NOT TO HANDLE UPDATE 1580 AND ANA B IF NOT TE PATTERN AND BNA AND AND AND AND AND AND AND AND AND A	1573   UFD   EQU   +	1573 UFD   EQU   +	002 1574 NV1 4,00.000 1576 NV1 1,05PROT 1577 ANA B IF NOT TO HANOLE UPDATE 1577 AND CALL LAND ROTALE PATTERN 1580 RLC ANA ROTATE PATTERN 1581 HOV A,M 1584 HOV A,M 1584 HOV A,M 1584 HOV A,M 1585 AND SPHOD-DSPROT-1 (A) = 0SPHOD 024 040 1586 JZ UFDI IF MENJRY 1589 AN DISPLAYING RECISTERS. 1590 AN DISPLAYING RECISTERS. 1590 AN DISPLAYING RECISTERS. 1590 AND DSPLAYING RECISTERS.
1562 + 1563 + 1564 + 1564 + 1564 + 1564 + 1566 + 1666 + 16	1562 * 1564 * 1564 * 1564 * 1565 * 1665 * 16	1562 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1664 * 1664 * 1664 * 1664 * 1664 * 1664 * 1664 * 1666 * 16	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + ENTRY (H.L) - ADDRESS OF REFCAT 1569 + EXIT NOME 1570 - USES ALL 1571 - USES ALL 1571 - USES ALL 1572 - USES ALL 1573 UFD EQU + 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1576 ANA B IF NOT TO HANDLE UPDATE 1577 ANI L. BOSPROT 1578 HOV A.M ROTATE PATTERN 1581 HOV B.A HOV A.M ROTATE PATTERN 1582 HOV B.A HOV A.M ROTATE PATTERN 1584 HOV A.M A.M ROTATE PATTERN 1585 HOV B.A M DISPLAYING REGISTERS. 1589 AM DISPLAYING REGISTERS. 1590 + AM DISPLAYING REGISTERS. 1591 CALL LRA LOCATE REGISTER ADDRESS DAD POR SEG NAME	1565	1564	1564	1570 * USES ALL  1571  1571  1571  1572  1574  ANA 8 IF NOT TO HANOLE UPDATE  1575  ANA 8 IF NOT TO HANOLE UPDATE  1576  ANA 8.4  1580  AND A.M  1581  AND A.M  1583  AND A.M  2024  0040  1589  AND A.M  2024  0040  1590  AND DEPLATION RECISTERS  1590  AND DEPLATION RECISTERS  1590  AND DEPLATION RECISTERS  1590  AND OF DEPLATION RECISTERS  1590  AND DEPLATION RECISTERS  1500  AND DEPLATION RECISTERS  1500  AND DEPLATION RECISTERS  1500  AND DEPLATION RECISTERS	1570 * USES ALL  1570 * USES ALL  1571	1570 * USES ALL  1571	1570 * USES ALL 1571	1570 * USES ALL 1571	1571	1571 1572 1573 160 1573 160 1574 1774 1775 1775 1775 1775 1776 1776 1777 1777	1572 1573 1600 1573 1600 1574 1600 1575 1600 1576 1600 1577 1600 1577 1600 1580 1600 1580 1600 1580 1600 1580 1600 1580 1600 1600 1600 1600 1600 1600 1600 16	1571 1572 1573 UFD 60U 1574 ANA B 1574 ANA B 1575 ANA B 1575 ANA B 1577 ANA B 1577 ANA B 1580 ANA B 1580 ANA B 1580 ANA B 1580 ANA ANA B 1580 ANA ANA B 1580 ANA ANA ANA B 1580 ANA ANA ANA ANA ANA ANA ANA ANA ANA AN	1571 1572 1573 1574 1574 1574 1574 1574 1575 1575 1576 1577 1577 1577 1577 1577	1571 1572 1573 1574 1574 1574 1574 1574 1575 1575 1576 1577 1577 1577 1577 1577	1571 1572 1573 UFD EQU + NVI A,UD.DDU  002 1574 NVI C, #DSPROT 1576 NVI L, #DSPROT 1580 NOV N,A NOV NOV N,A NOV N,A NOV N,A NOV NOV N,A NOV NOV NOV NOV NOV NOV NOV NO	1976 1977 002 1574 NVI A,UO,DDU 1575 ANA B 1575 ANA B 1577 ANI L,iDSPROT 1580 1581 NDV A,M 1583 1882 1583 1NDV A,M 1584 1585 ANI 2 002 1586 ANI 2 024 040 1587 ANI 2 024 040 1587 ANI 2 027 003 1590 AN DISPLAYING REGISTER'S OF REG NAME 1599 AN DISPLAYING REGISTER'S OF REG NAME 1594 AN DISPLAYING REGISTER'S OF REG NAME 1594 AN DISPLAYING REGISTER'S OF REG NAME 1595 AND DAD DAD DAD DAD DAD DAD DAD DAD DAD	1573 UFD	1573 UFD   EQU   +	1574   WY
1562 +  1563 +  1564 -  1564 -  1565 +  1705 -  1706 -  1706 -  1707 -  1708 -  1709 -	1562 * 1564 * 1564 * 1564 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1566 * 1566 * 1566 * 1566 * 1566 * 1567 * 1566 * 1567 * 1566 * 1567 * 1567 * 1567 * 1568 * 1567 * 1568 * 1577 * 1577 * 16	1562 * 1563 * 1564 * 1565 * 1564 * 1565 * 1565 * 1565 * 1565 * 1566 * 1576 * 1666 * 16	1566	1565 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1569 * ENITRY (H,L) - ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 * AND	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFCMT 1569 * EXIT NONE 1559 * EXIT NONE 1550 * USES ALL 1571 * OSES ALL 1572 * OSES ALL 1573 UFD EQU * OSES OF REFCMT 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANY CANA B IF NOT TO HANDLE UPDATE 1576 * OSES OF ANY CANA ANA ANA ANA ANA ANA ANA ANA ANA AN	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 * TIME TO UPDATE THE UISPLAY CONTENTS, CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * EXIT NONE 1569 * EXIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 * ONES ALL 1571 * ONES ALL 1572 * ONES ALL 1573 UFD 60U * ONES AND TO HANDLE UPDATE 1574 ANA B TO THE NOT TO HANDLE UPDATE 1575 AND	1570 * USES ALL  1571  1571  1571  1572  ANA B  1574  ANA B  1575  ANA B  1576  ANA B  1577  ANA B  1578  ANI C  1580  ANI C  1581  ANI C  1584  ANI C  1584  ANI C  1585  ANI C  1586  ANI C  1586  ANI C  1587  ANI C  1588  ANI C  1589  ANI DISPLAYING RECISTERS.	1570 * USES ALL  1571  1571  1572  1573  UFD  EQU	1570 • USES ALL  1571 • USES ALL  1571 • USES ALL  1572 • USES  1573 UFD EQU • •  1574 ANA B IF NOT TO HANDLE UPDATI  1575 ANA B IF NOT TO HANDLE UPDATI  1577 ANI L, #DSPROT  1581 HOV B, A  1582 INX H  1583 INX H  1584 ERRNZ A, H  1584 ANI Z  227 003 1586 ANI Z  227 003 1590 • AN DISPLAYING RECISTERS  1591 CALL LRA  1559 LXI H-105PA	1570 • USES ALL  1571	1570	1570	1572	1572 1573 002 1573 006 1574 007 1575 008 1576 008 1577 008 1577 008 1580 007 1581 008 1581 008 1581 008 1581 008 1581 008 1581 008 1581 008 1581 008 1581 008 1581 008 1581 008 1581 008 1581 1581	1571 1572 1573 UFD EQU * 1574 ANA B B COS 1575 ANA B B COS 1576 ANA B B COS 1577 ANI B COS 1578 ANI C C C C C C C C C C C C C C C C C C C	1571 1572 1573 1574 1574 1575 1574 1575 1575 1575 1576 1577 1577 1577 1577	1571 1572 1573 1574 1574 1575 1574 1575 1575 1575 1576 1577 1577 1577 1577	1571 1572 1573 0F0 602 1574 MVI A,U0.DDU  002 1575 ANA B 1576 ANA 1580 1577 HVI L,#DSPROT 1580 1581 HOV B,A 1582 1584 HOV B,A 1584 1585 ANI 2 024 047 003 1590 AN DISPLAYING RECISTERS AND 1590 AN DISPLAYING RECISTER ADDRES 1591 LXI H,055A	1576 1577 002 1574 ANA B 1575 ANA B 1576 ANA B 1577 006 1578 MOV A,H 1580 1581 MOV A,H 1584 1584 MOV A,H 1584 MOV A,H 1585 ANI B 227 003 1589 AH DISPLAYING RECISTERS 1589 AH DISPLAYING RECISTER ADDRES 1591 HOSPA	002 1573 UFD EQU + 100.000	002 1573 UFD EQU +  1574 MVI A,UQ.DDU  1575 ANA B B.D.DDU  1577 ANA B COND. 1577 ANA B COLORATE PATTERN  1581 HOV A,H  1582 HOV B,A  1583 INX H  1584 ERRNZ A,H  1585 ANI Z  227 003 1586 AN DISPLAYING RECISTERS  047 003 1590 + AN DISPLAYING RECISTERS  1589 AN DISPLAYING RECISTERS  1589 AN DISPLAYING RECISTERS  1591 CALL LRA  14.056A	002 1574
1562 + 1563 + 1564 + 1564 + 1564 + 1564 + 1566 + 16	1562 * 1564 * 1564 * 1564 * 1565 * 1565 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1567 * 1568 * 1568 * 1568 * 1569 * 1570 * 1570 * 1570 * 1571 * 1571 * 1571 * 1572 * 1573 * 1574 * 1674 * 1774 * 1775 * 1776 * 1777 * 17	1562 * 1563 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1566 * 1666 * 16	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + ENTRY (H+L) = ADDRESS OF REFCNT 1569 + ENITRY (H+L) = ADDRESS OF REFCNT 1570 + USES ALL 1571 - USES ALL 1572 - USES ALL 1573 UFD EQU + AND B	1565	1565	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 * TIME TO UPDATE THE UISPLAY CONTENTS, CURRENTLY, THIS ITS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFORT 1569 * EXIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 * NONE 1570 * WILL 1571 * NONE 1571 * NONE 1571 * NONE 1572 * NONE 1573 * UFD 1574 * NONE 1574 * NONE 1575 * NONE 1576 * NONE 1577 * NONE 1577 * NONE 1577 * NONE 1578 * NONE 1579 * NONE 1570 * NONE	1570 * USES ALL  1571  1571  1571  1572  ANA B  1574  ANI C. *** *** *** *** *** *** *** *** ***	1570 * USES ALL  1571  1571  1572  1573  1574  ANA  B  1575  ANA  1576  ANA  1577  ANI  1580  1581  ANY  1581  ANY  1581  ANY  1581  ANY  1581  ANY  1583  ANY  1584  ANY  1585  ANY  1589  ANY  ANY  ANY  ANY  ANY  ANY  ANY  AN	1570 * USES ALL  1571	1570 * USES ALL  1571	1570 • USES ALL  1571	1570	1572	1572 UFD EQU + 1573 UFD EQU + 1573 UFD EQU + 1573 UFD EQU + 1573 UFD EQU + 1574 ANA B CONTRIBER IS TO THE PRICE OF THE PRI	1571 1572 1573 0F0	1571 1572 1573 1692 1573 1690 1574 1574 1574 1575 1575 1575 1575 1577 1577	1571 1572 1573 1692 1573 1690 1574 1574 1574 1575 1575 1575 1575 1577 1577	1571 1572 1573 002 1574 ANA B 1575 ANA B 1576 ANA 1580 1579 MOV A,M C, 40 SPROT 1581 MOV A,M C, 40 SPROT 1582 MOV A,M CO2 1584 MOV A,M CO2 1584 MOV A,M CO2 1585 MOV A,M CO2 1586 MOV A,M CO2 1589 MOV A,M COCATE REGISTER NOURES	1576 1574 ANA B 1575 ANA B 1575 ANA B 1576 ANI L, BSPROT 1580 ANI L, BSPROT-1 1584 ANI Z 227 003 1586 ANI Z 227 003 1589 AN DISPLAYING REGISTER X NDRES 1590 AN DISPLAYING REGISTER X NDRES	002 1573 UFD EQU + 1573 MAI A,UO.DDU 1574 ANA B 1575 ANA B 1575 ANA B 1576 ANI L, #DSPROT 1577 HVI L, #DSPROT 1579 HOV H,A 1581 HOV H,A 1583 INX H 1584 ERRNZ OSPHOD-DSPROT-1 1585 ANI Z 1585 ANI Z 1586 ANI Z 1587 OO3 1589 AN DISPLAYING REGISTERS 1590 * AN DISPLAYING REGISTERS 1591 AN DISPLAYING REGISTERS 1593 PUSH H	002 1573 UFD EQU +  1574 NAVI A,UQ.DDU 1575 ANA B IF NOT TO HANDLE UPDATI 1577 NVI L,#DSPROT 1577 NVI L,#DSPROT 1580 RLC 1581 NOV N,A 1582 NOV B,A 1583 NOV N,A 1584 NOV N,A 1584 NOV N,A 1584 NOV N,A 1585 ANI 1585 ANI 1586 ANI 1586 ANI 1586 ANI 1586 ANI 1587 CALL LRA 1589 AN DISPLAYING REGISTERS. 1590 * AN DISPLAYING REGISTER ADDRES 1593 PUSH H	002 1574
1562 * 1562 * 1563 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1566 * 1666 * 16	1562 + 1563 + 1563 + 1563 + 1564 + 1564 + 1564 + 1564 + 1566 + 1566 + 1567 - 1567 - 1567 - 1569 - 1570 - 1570 - 1570 - 1570 - 1570 - 1571 - 1571 - 1572 - 1573 UFD EQU + 1574 ANA B IFF NOT TO HANDLE UPDATE 1574 ANA B IFF NOT TO HANDLE UPDATE 1575 ANA B IFF NOT TO HANDLE UPDATE 1576 AND A.M ROTAL 1577 AND B IFF NOT TO HANDLE UPDATE 1577 AND B IFF NOT TO HANDLE UPDATE 1578 AND A.M ROTALE PATTERN 1580 AND A.M ROTAL 1581 AND A.M ROTAL 1581 AND A.M ROTAL 1582 AND A.M ROTAL 1583 AND A.M ROTAL 1584 AND A.M ROTAL 1585 AND A.M ROTAL 1586 AND A.M A.M ROTAL 1586 AND A.M A.M ROTAL 1586 AND A.M A.M ROTAL 1589 AND A.M A.M ROTAL 1589 AND A.M A.M ROTAL 1589 AND A.M ROTAL	1562 + 1562 + 1563 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1564 + 1566 + 15	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 + ENTRY (H+L) = ADDRESS OF REFCMT 1569 + EXIT NONE 1570 + USES ALL 1571 - USES ALL 1572 - USES ALL 1573 UFD EQU + ANA B	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENITY (H,L) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1570 * USES ALL 1571 AND AND B IF NOT TO HANOLE UPDATE 1572 AND B IF NOT TO HANOLE UPDATE 1574 AND AND B IF NOT TO HANOLE UPDATE 1575 AND	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * FVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A SECOND. 1566 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1568 * ENIT NOME 1559 * EXIT NOME 1550 * USES ALL 1570 * USES ALL 1571 * HAI A,UG,DDU 1572 * ANA B 1573 UFD EQU * IF NOT TO HANOLE UPDATE 1574 * ANA B 1575 * ANA B 1576 * RNZ 1577 * ANA B 1578 * HVI L, *DSPROT 1577 * HVI L, *DSPROT 1578 * HVI L, *DSPROT 1579 * HOY A,N 1570 * HOY A,N 1580 * ANI Z 1580 * ANI DISPLAYING REGISTERS. 1590 * AN DISPLAYING REGISTERS.	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN III 1565 * TIME TO UPDATE THE UISPLAY CONTENTS. CURRENILY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1569 * EXIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 * NONE 1571 * NONE 1572 * NONE 1573 * UFD EQU * * * * * * * * * * * * * * * * * * *	1570 * USES ALL  1571  1571  1571  1572  ANA B  1575  ANA B  1576  ANA B  1576  ANA B  1577  AVI L.#DSPROT  1580  ANY L.#DSPROT  1581  MOV A.M  1582  1002  1584  ANO A.M  AND A.M  A.M  A.M  A.M  A.M  A.M  A.M  A.M	1570 * USES ALL  1571	1570 • USES ALL  1571 • USES ALL  1572 • USES ALL  1573 UFD EQU • • • • • • • • • • • • • • • • • • •	1570 * USES ALL  1571	1570 • USES ALL  1571	1570	1572	1572	1571 1572 002 1573 UFD EQU ** MVI A**UO.DDU B GOOD 1576 ANA B GOOD 1576 ANA B GOOD 1577 ANA B GOOD 1578 AND ANA 1581 AND	1571 1572 1573 060 1573 060	1571 1572 1573 060 1573 060	1571 1572 1573 1573 1574 1574 1574 1574 1575 1575 1575 1575	1576 JFD EQU + 1576 MVI A,UQ.DDU   1574 MVI A,UQ.DDU   1575 ANA B   1575 ANA B   1575 ANA B   1577 AVI L, #DSPROT   1577 AVI L, #DSPROT   1577 AVI L, #DSPROT   1580 AVI   1581 ADV A, M   1582 AVI   1583 INX H   1584 ADV A, M   1584 ADV A, M   1585 AVI	1573 UFD   EQU   +	002 1573 UFD EQU +  1574 MVI A,UQ.DDU 1575 ANA B 18 1576 RNZ 1577 AVI L, & DSPROT 1577 AVI L, & DSPROT 1580 RCV 1580 RCV 1581 HOV B, A 1582 HOV B, A 1583 INX H 1585 ANI 1586 ANI 1586 ANI 1586 ANI 1587 CHLD ABUSS 227 003 1588 JZ UFOI 1589 AN DISPLAYING RECISTERS. 1591 CALL LRA COCATE RECISTER ADDRES	002 1574
1562 * 1562 * 1563 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1566 * 1666 * 16	1562 *	1562 + 1563 + 1564 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + ENTRY (H,L) - ADDRESS OF REFCAT 1568 + ENTRY (H,L) - ADDRESS OF REFCAT 1570 + USES ALL 1571 + NONE 1571 + NONE 1572 AND 8 1574 AND 8 1575 AND 1576 AND 1576 AND 1576 AND 1577 1576 AND 8 1577 AND 17 L, 4059ROT 1578 AND 17 L, 4059ROT 1579 AND 17 L, 4059ROT 1570 AND 17 L, 4059ROT 1570 AND 17 L, 4059ROT 1571 AND 17 L, 4059ROT 1571 AND 17 L, 4059ROT 1572 AND 17 L, 4059ROT 1573 AND 17 L, 4059ROT 1574 AND 17 L, 4059ROT 1575 AND 17 L, 4059ROT 1576 AND 17 L, 4059ROT 1577 AND 17 L, 4059ROT 1577 AND 17 L, 4059ROT 1578 AND 17 L, 4050LAYING RECISTERS.	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 + ENTRY (H+L) = ADDRESS OF REFCMT 1569 + EXIT NONE 1570 + USES ALL 1571 - USES ALL 1572 - ANA B IF NOT TO HANDLE UPDATE 1574 - ANA B IF NOT TO HANDLE UPDATE 1575 - ANA B ISPHOD-DSPROT 1576 - HOY A.M ROTALE PATTERN 1580 - RRZ OSPHOD-DSPROT-1 1584 - ERRNZ OSPHOD-DSPROT-1 1584 - HOY B.A ROTATE PATTERN 1585 - ANI Z OSPHOD-DSPROT-1 1586 - ANI Z UFD1 IF NEMBERS 1590 + AN DISPLAYING REGISTERS.	1565	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * FERRY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCNT 1569 * EXIT NONE 1550 * USES ALL 1570 * USES ALL 1571 * NONE 1572 * USES ALL 1573 * UFD * USES ALL 1574 * NONE 1575 * NONE 1576 * NONE 1577 * NONE 1578 * NONE 1579 * NONE 1570 * USES ALL 1571 * NONE 1570 * USES ALL 1571 * NONE 1571 * NONE 1572 * NONE 1573 * NONE 1574 * NONE 1575 * NONE 1576 * USES 1576 * USES 1577 * NONE 1578 * NONE 1579 * NONE 1570 * USES 1570 * US	1565 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN III 1566 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A SECOND. 1566 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A SECOND. 1567 * ENTRY (H+L) - ADDRESS OF REFCNT 1569 * ENTRY (H+L) - ADDRESS OF REFCNT 1570 * USES ALL 1571 * NONE * ALL 1572 * ANA * B	1570 * USES ALL  1571  1571  1571  1572  ANA B  1575  ANA B  1576  ANA B  1577  ANA L. & DSPROT  1580  1580  1581  1581  ANY A.H  ROV A.H  1584  1584  1585  ANY A.H  COZ 1586  ANY A.H  A.H  A.D A.H  A.H  A.D A.H  A.H  A.H  A.H  A.H  A.H  A.H  A.H	1570 * USES ALL  1571	1570 * USES ALL  1571	1570 * USES ALL  1571	1570 • USES ALL  1571	1570 • USES ALL 1571	1572	1572	1571 1572 002 1573 UFD EQU ** ANA B B B B B B B B B B B B B B B B B B	1571 1572 1573 1574 1574 1574 1575 1575 1575 1575 1575	1571 1572 1573 1574 1574 1574 1575 1575 1575 1575 1575	1571 1572 1573 0F0 002 1573 0F0 1574 ANA B 1576 ANA B 1577 ANA B 1577 ANA B 1578 ANA B 1580 1580 1580 HOV H,A A,U0,DDU HANOLE UPDATI B 1580 HOV H,A A,M B 1580 HOV H,A A,M B 1581 HOV A,M B 1582 ANI C 1583 ANI C 1583 ANI C 1583 ANI C 1584 ANI C 1585 ANI C 1586 ANI C 1586 ANI C 1586 ANI C 1589 A	1576 1577 4NVI A,UO.DDU 1575 4NA B 1575 4NA B 1576 4NO. 1577 4NVI L,#DSPROT 1580 1580 1581 1582 1583 100 1584 1585 1585 4NI 2 1585 4NI 2 1585 4NI 2 1586 4NI 2 1586 4NI 2 1589	1573 UFD EQU	1573 UFD   EQU   +	002 1574 WY A,UO,DOU  002 1576 ANA B 1575 ANA B 1577 ANY L,#DSPROT 1577 MYI L,#DSPROT 1580 RLC A,M 1581 MOV A,M 1583 INX H 1583 INX H 1584 ERRNZ A,M 002 1585 ANI 2 227 003 1586 ANI 2 1589 ANI 2 1589 ANI 2 1589 ANI 1589 1590 * ANI DISPLAYING RECISTERS.
1562 * 1563 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1566 * 1566 * 1566 * 1676 * 16	1562 * 1563 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1676 * 1676 * 1566 * 1676 * 16	1562 * 1562 * 1564 * 1564 * 1565 * 1565 * 1566 * 15	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571 * USES ALL  1571 * ANA B IF NOT TO HANDLE UPDATE  1572 * ANA B IS  1574 * ANA B IS  1576 * ANA B IS  1576 * ANA B IS  1577 * ANA B IS  1578 * ANA B IS  1578 * ANA B IS  1580 * RLC AAM  1581 * MOV AAM  1581 * MOV BA  1582 * MOV BA  1583 * MOV BA  1584 * AND AAM  1584 * AND AAM  1585 * MOV BA  1586 * AND AAM  1588 * AND AAM	1565 + 10FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN III 1565 + 11ME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 + ENTRY (H+L) = ADDRESS OF REFCAT 1570 + USES ALL 1571 - USES ALL 1572 - ANA B TO TO HANDLE UPDATE 1574 - ANA B TO TO HANDLE UPDATE 1575 - ANA B TO TO HANDLE UPDATE 1576 - ANA B TO TO TO HANDLE UPDATE 1577 - ANA B TO TO TO HANDLE UPDATE 1578 - ANA B TO	1564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 # ENTRY (H+L) = ADDRESS OF REFCAT 1568 # ENTRY (H+L) = ADDRESS OF REFCAT 1570 # USES ALL 1571 # AND # B B B B B B B B B B B B B B B B B B	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H+L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 * ENTRY (H+L) - ADDRESS OF REFCNT 1573 UFD & ALL 1574 * MVI A+U0.DDU 1575 * MVI A+U0.DDU 1576 * RNZ 1577 * MVI A+U0.DDU 1578 * RNZ 1578 * MVI L,*DSPRÖT 1579 * MVI L,*DSPRÖT 1579 * MVI L,*DSPRÖT 1580 * RLC 1581 * MOY A+M B*,A 1581 * MOY A+M B*,A 1584 * ERRNZ OSPHOD-DSPRÖT-I 1584 * ERRNZ OSPHOD-DSPRÖT-I 1585 * MOY A+M B*,A 1586 * MOY A+M B*,A 1586 * LHD ABUSS 227 003 1588 * JZ UFOI IF MENDRY	1570 * USES ALL  1571 1573 1574 1574 1575 1574 1575 1576 1576 1576 1577 1577 1577 1577	1570 * USES ALL  1571 1573 1573 1573 1574 1574 1575 1574 1576 1576 1577 1577 1577 1577 1577 1577	1570 * USES ALL  1571	1570 * USES ALL 1571	1570	1570	1572	1572 UFD 60U 4 1573 UFD 60U 4 1574 NVI A, UO. DDU 1575 ANA B 80.0DU 1576 RNZ 1577 AVI L, EDSPROT 1580 RVI A, H 1580 RNZ 1581 ROV A, H 1582 ROV H, A 1582 ROV B, A 1584 RRNZ OSPHOD-DSPROT-1 1584 RRNZ OSPHOD-DSPROT-1 1585 ANI 227 003 1586 JZ UFO1 1589 JZ UFO1	1571 1572 1573 002 1573 006 1574 MVI A, UG. DDU 1575 ANA B B B B B B B B B B B B B B B B B B	1571 1572 1573 002 1574 002 1574 004 1576 006 1577 006 1579 007 1581 007 1582 1084 1584 1585 1084 1585 1084 1087 1087 1088 1088 1089 1089 1089 1089 1089 1089	1571 1572 1573 002 1574 002 1574 004 1576 006 1577 006 1579 007 1581 007 1582 1084 1584 1585 1084 1585 1084 1087 1087 1088 1088 1089 1089 1089 1089 1089 1089	1571 1572 1573 1574 1574 1574 1575 1576 1577 006 1577 006 1579 1580 1581 1581 1582 1583 100 1584 1584 1585 1585 1586 1589 1002 1589 1002 1589 1003 1004 1009 1009 1009 1009 1009 1009 1009	1572 1574 1575 ANA 8 1575 ANA 8 1576 ANA 8 1577 ANI 1577 ANI 1580 1581 ANY 1581 ANY 1582 ANY 1583 ANY 1583 ANY 1584 ANY 1586 ANY	002 1573 UFD EQU * 1573 UFD EQU * 1574 A*UG*DDU   1575 ANA B   1576 ANZ   1577 ANZ   1577 ANZ   1577 ANZ   1580 ANZ   1581 ANZ   1582 ANZ   1583 ANZ   1583 ANZ   1584 ANZ   1585 ANZ   1586 ANZ   1586 ANZ   1586 ANZ   1586 ANZ   1586 ANZ   1587 ANZ   158	002 1573 UFD EQU 4  002 1574 NVI A, UQ. DDU 1575 ANA B 1576 RNZ 1577 NVI L, EDSPROT 1579 NOV A, H 1580 RLC 1581 MOV B, A 1582 NOV B, A 1585 ANI 2 227 003 1586 JZ 1589 JZ 1589 LHLD ABUSS 227 003 1588 JZ	002 1574 MVI A,UG.DDU  1575 ANA B 1576 ANA B 1577 AVI L, EDSPROT 1579 MVI L, EDSPROT 1579 MVI A,M 1580 RLC 1581 MOV A,M 1582 INX H 1584 ERRNZ OSPHOD-DSPROT-1 1585 ANI 2 227 003 1588 JZ UFOI
1562 *  1563 *  UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 *  1566 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A SECOND.  1567 *  ENTRY (H,L) - ADDRESS OF REFCNT  1569 *  ENTRY (H,L) - ADDRESS OF REFCNT  1570 *  USES ALL  1571 *  1572 *  NVI A,UQ,DDU  1574 *  ANA B  1575 *  NVI L, #DSPROT  1576 *  NVI L, #DSPROT  1580 *  NOV A,M  1581 *  NOV A,M  1582 *  NOV A,M  1583 *  NOV A,M  1583 *  NOV A,M  1583 *  NOV A,M  1584 *  NOV A,M  1585 *  NOV A,M  1586 *  NOV A,M  1586 *  NOV A,M  1589 *  NOV A,M  1589 *  NOV A,M  1580 *  NOV A,M	1562 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCNT	1562 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * UFD IS CALLED BY THE USPLAY CONTENTS, CURRENLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * ENTRY (H,-L) = ADDRESS OF REFCAT 1568 * EXIT NONE 1570 * USES ALL 1571 * OSES ALL 1572 * OSES ALL 1573 * UFD * OSES 1574 * ANA B * USES 1574 * ANA B * USES 1575 * OSES 1576 * OSES 1577 * OSES 1578 * OSES 1578 * OSES 1579 * OSES 1579 * OSES 1570 * O	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567	1564	1564 • UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN III 1565 • EVERY 32 INTERRUPTS, OR ABOUT 34 THES A SECOND. 1566 • EVERY 32 INTERRUPTS, OR ABOUT 34 THES A SECOND. 1567 • ENTRY (H+L) - ADDRESS OF REFCNT 1569 • EXIT NONE 1570 • USES ALL 1571 • USES ALL 1572 • USES ALL 1573 UFD EQU • AND B IF NOT TO HANDLE UPDATE 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1576 ANA ANA ANA ANA ANA ANA ANA ANA ANA AN	1564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN III 1565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 # ENITY (H,L) = ADDRESS OF REFCAT 1568 # ENITY (H,L) = ADDRESS OF REFCAT 1570 # USES ALL 1571 #WI A,UO.DOU 1571 #WI A,UO.DOU 1572 ANA B 1574 ANA B 1575 ANA B 1576 ANA A,M 1581 HOV B,A 1581 HOV B,A 1581 HOV B,A 1581 HOV A,M	1570 * USES ALL  1571 1573 1574 1573 1574 1575 1574 1575 1576 1576 1576 1577 1577 1577 1580 1580 1581 1582 1000 1584 1583 1000 1584 1585 1000 1586 1586 1586 1586 1586 1587 1587 1588 1588 1588 1588 1588 1588	1570 * USES ALL  1571 1572 1573 UFD EQU *  1574 NVI A, UG, DDU 1575 ANA B 1576 ANA B 1576 ANA B 1577 AVI L, #DSPROT 1579 MDV A, M 1580 MDV H, A 1581 MDV H, A 1582 MDV H, A 1584 MDV H, A 1586 MDV H, A 1588 MDV H,	1570 * USES ALL 1571 1573 UFD EQU * 1573 UFD EQU * 1574 ANA B 1575 ANA B 1576 ANA B 1577 AVI L, *DSPROT 1579 HOV A, H 1580 HOV H, A 1581 HOV H, A 1581 HOV H, A 1584 HOV H, A 1584 ANI 2 227 003 1588 USS	1570 * USES ALL 1571 UFD EQU * 1573 UFD EQU * 1573 UFD EQU * 1574 ANA B 1575 ANA B 1576 ANA B 1577 AVI L, BDSPROT 1579 MOV A, M 1580 MOV B, A 1581 MOV B, A 1584 MOV B, A 1585 ANI 2 227 003 1588 UF OI	1570	1570	1572 UFD USES ALL 1573 UFD EQU 4 1574 MVI A, UO. DDU 1575 MVI A, UO. DDU 1576 MNZ 1577 MVI L, EDSPROT 1579 MOV A, M 1580 MOV M, A 1581 MOV M, A 1584 ERRNZ OSPHOD-DSPROT-1 1585 MOV A, M 0024 040 1587 MVI 2 227 003 1588 UFD UFD1	1572 UFD 60U 4 1573 UFD 60U 4 1574 MVI A, UO.DDU 1575 ANA B 8 1576 ANA B 1577 006 1578 AVI L, #DSPROT 1580 RLC A, M 1581 MOV M, A 1582 MOV B, A 1584 ERRNZ A, M 1584 ANI 2 227 003 1588 JZ UFOI	1571 1572 1573 0F0 002 1573 0F0 1574 NVI 1576 NVI 1576 NVI 1577 NVI 1580 NDV N,A 1581 NDV N,A 1584 1584 1584 1584 1584 1584 1584 1584	1571 1572 1573 060 1574 NVI NVI 1576 NVI 1576 NVI 1577 NVI 1580 RLC 1581 NOV 1581 NOV 1582 1584 1584 1584 1584 1584 1584 1584 1584	1571 1572 1573 060 1574 NVI NVI 1576 NVI 1576 NVI 1577 NVI 1580 RLC 1581 NOV 1581 NOV 1582 1584 1584 1584 1584 1584 1584 1584 1584	1571 1572 1573 1574 WVI WVI WVI MVI MA,UG.DDU 1575 MVI	1572 1574 1574 1575 1575 1575 1576 1576 1577 1577 1577	002 1573 UFD EQU * 1573 LFD EQU * 1574 A, UD. DDU   1575 ANA B   1576 ANZ   1577 AVI   1, ebsprot   1577 AVI   1, ebsprot   1579 AVI   1, ebsprot   1580 AVI   1582 AVI   1584 AVI   1584 AVI   1585 AVI   1585 AVI   1585 AVI   1585 AVI   1585 AVI   1585 AVI   1586 A	002 1573 UFD EQU	002 1574 MVI 64,00.000 1575 ANA 8 1576 ANZ 8 1577 AVI L, EDSPROT 1579 MVI L, EDSPROT 1580 RLC ANA 1581 1581 MOV A,M A,M A,M B 1582 ANI 2 ANI 3
1562 #  1563 #  1564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 #  1566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 # ENTRY (H+L) = ADDRESS OF REFCNT  1569 # EXIT NONE  1570 # USES ALL  1571 # NONE  1572 # NONE  1573 UFD EQU #  1574 ANA B B B B B B B B B B B B B B B B B B	1562 + 1563 + 1564 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + ENTRY (H,L) - ADDRESS OF REFCAT 1569 + EXIT NONE 1570 + USES ALL 1571 - USES ALL 1572 - USES ALL 1573 UFD EQU + 1574 ANA B RANA 1575 ANA B RANA 1576 ANA B RANA 1577 HVI L, #DSPROT 1578 HOV A,H 1580 HOV B,A 1581 HOV B,A 1581 HOV A,H 1584 HOV A,H 1586 HOV A,H 1586 ANI 2 1586 ANI 2 1586 THE DISPROT 1586 ANI 2 1586 THE DISPROT 1586 ANI 2	1562 *	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) = ADDRESS OF REFCNT  1569 * ENTRY (H,L) = ADDRESS OF REFCNT  1570 * USES ALL  1571 * USES ALL  1572 UFD EQU * AND B IF NOT TO HANOLE UPDATE  1574 ANA B IF NOT TO HANOLE UPDATE  1575 ANA B IF NOT TO HANOLE UPDATE  1576 RNZ B IF NOT TO HANOLE UPDATE  1577 ANI L, #DSPROT  1580 RLC AN AND AND AND AND AND AND AND AND AND	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1570 * USES ALL 1571 * HVI A,UG,DDU 1572 UFD EQU * IF NOT TO HANOLE UPDATE 1574 ANA B IF NOT TO HANOLE UPDATE 1575 ANA B IF NOT TO HANOLE UPDATE 1576 HVI L,#DSPROT 1577 HVI L,#DSPROT 1578 HOV A,M 1580 RLC 1578 HOV A,M 1580 RLC 1581 HOV A,M 1581 HOV A,M 1582 HOV B,A 1583 HOV A,M 1584 HOV A,M 1584 HOV A,M 1585 HOV A,M 1586 HOV A,M 1586 HOV A,M 1586 AMI 2	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IIMES A SECOND. 1567 * ENTRY (H+L) - ADDRESS OF REFCAT 1568 * ENTRY (H+L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * USES ALL 1572 * ANA B 1574 * ANA B 1575 * ANA B 1576 * ANA B 1577 * ANA B 1578 * ANA B 1580 * ANI CANA B 1581 * AND CANA B 1582 * ANA B 1584 * ANA B 1584 * ANI CANA B 1584 * ANI CANA B 1585 * ANI CANA B 1586 * ANI CANA B 1587 * ANI CANA B 1588 * ANI CAN	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN III 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H+L) - ADDRESS OF REFCAT 1569 * ENTRY (H+L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * USES ALL 1572 * ANA B 1574 * ANA B 1575 * ANA B 1576 * ANA B 1577 * AVI L.*DSPROT 1578 * AVI L.*DSPROT 1579 * AVI L.*DSPROT 1579 * AVI HANDE DISPROT-I 1581 * HOV H,A AVI (A) = OSPHOD 1584 * AVI C) ANA C) 1584 * AVI C) ANA C) 1585 * AVI C) ANA C) 1585 * AVI C) ANA C) 1586 * AVI C) ANA C) 1586 * AVI C) ANA C) 1586 * AVI C) ANA C) 1587 * AVI C) ANA C) 1588 * AVI C) ANA C) 1588 * AVI C) ANA C) 1588 * AVI C) ANA C) 1589 * AVI C) ANA C) 1589 * AVI C) ANA C) 1580 * AVI C) ANA C) 1580 * AVI C) ANA C) 1581 * ANA C) 1581 * ANA C) 1582 * ANI C) 1583 * ANI C) 1584 * ANI C) 1585 * ANI C) 1585 * ANI C) 1585 * ANI C) 1586 * ANI C) 1586 * ANI C) 1587 * ANI C) 1588 *	1570 • USES ALL  1571 1572 1573 002 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANI L, BSPROT  1580 RLC 1581 MOV A, M A 1582 MOV B, A 1583 INX H 1583 MOV A, M A 1584 ANI C ABUSS	1570 * USES ALL  1571	1570 * USES ALL 1571 1573 UFD EQU * 1573 UFD EQU * 1574 ANA B 1575 ANA B 1577 ANA B 1577 ANA B 1577 ANA B 1577 ANI L.*BDSPROT 1579 ANI L.*BDSPROT 1581 ANO A, A 1584 ANI 2 1585 ANI 2 1586 ANI 2 1586 ANI 2 1586 ANI 2 1587 ANI 2	1570 * USES ALL 1571 1572 1573 UFD EQU * 1573 UFD EQU * 1574 ANA B 1575 ANA B 1577 ANA B 1577 ANA B 1577 ANI L.*BDSPROT 1579 ANI A** 1581 ANO A** 1582 ANI 2 1584 ANI 2 1586 ANI 2 1586 ANI 2	1570	1570	1572 USES ALL 1573 UFD EQU * 1573 UFD EQU * 1574 MVI A, UQ. DDU 1575 ANA B 1576 ANA B 1577 AVI L, EDSPROT 1579 HOV A, M 1580 RLC A, M 1581 HOV B, A 1584 HOV B, A 1584 HOV A, M 1584 HOV A, M 1585 HOV A, M 1585 HOV A, M 1585 HOV A, M 1586 ANI 2 1586 ANI 2	1572 UFD EQU * 1573 UFD EQU * 1574 MVI A, UG. DDU 1575 ANA B NO. DDU 1576 ANA B NO. DDU 1576 ANA B NO. DDU 1577 MVI L, EDSPROT HOV A, M NO. DSPROT 1581 MOV A, M OV A,	1571 1572 1573 002 1573 006 1574 NVI 00.000 1575 006 1576 006 1577 007 1580 007 1581 007 1582 007 1583 007 1585 007 1585 007 008 1586 007 1586	1571 1572 1573 1574 1574 1574 1575 1575 1576 1577 006 1577 006 1579 1580 1581 1581 1582 1583 1007 1584 1585 1584 1584 1584 1588 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1008 1008 1008 1008 1008 1008 1008	1571 1572 1573 1574 1574 1574 1575 1575 1576 1577 006 1577 006 1579 1580 1581 1581 1582 1583 1007 1584 1585 1584 1584 1584 1588 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1585 1007 1008 1008 1008 1008 1008 1008 1008	1571 1572 1573 1574 002 1574 002 1575 006 1576 006 1578 006 1578 007 1580 007 1581 007 1581 007 1582 008 1583 007 1584 007 1584 007 1585 002 1585 007 1586 0	1576 60U + 1576 1574 WI A, UO. DDU   1575 ANA B   B   B   B   B   B   B   B   B   B	002 1573 UFD EQU * 1574 HVI A, UD. DDU 1575 ANA B 1575 ANA B 1577 ANZ 1577 HVI L, #DSPROT 1579 HOV A, H A 1582 HOV H, A 1582 HOV H, A 1582 HOV H, A 1582 HOV H, A 1582 HOV A, H CON 1585 ANI 2 0224 040 1587 LHLD ABUSS	002 1573 UFD EQU	002 1574 MVI 4,00.000 1575 ANA 8 1576 RNZ B 1576 RNZ B 1577 MVI L, BOSPROT 1579 RUC A,M 1581 ROV A,M 1581 ROV B,A 1583 INX H 1584 ERRNZ OSPHOD-DSPROT-1 1584 ANI 2 0024 040 1587 LHLD ABUSS
1562 #  1563 #  UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 #  1566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 # ENTRY (H+L) = ADDRESS OF REFCNT  1569 # ENTRY (H+L) = ADDRESS OF REFCNT  1570 # USES ALL  1571 # NONE  1572 # NONE  1573 UFD EQU #  1574 MAI A HUG-DDU  1575 ANA B IF NOT TO HANDLE UPDATE  1576 # NONE  1577 ANA B IF NOT TO HANDLE UPDATE  1578 HOV A H  1579 HOV A H  1580 RLC  1581 HOV A H  1584 ERRNZ DSPHOD-DSPROT-1  1584 HOV A H  1585 HOV A H  1586 ANI A H  1587 AND A H  1588	1562 + 1563 + 1564 + 1567 + 1564 + 1566 + 1566 + 1566 + 1566 + 1566 + 1567 + 1567 + 1568 + 1567 + 1567 + 1568 + 1570 + 1570 + 1570 + 1570 + 1571   1571   1572   1573   1574   1674   1676   1575   1576   1577   1578   1579   1570   1570   1570   1570   1570   1570   1571   1571   1572   1573   1574   1674   1676   1677   1677   1677   1677   1678   1670   16	1562	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) = ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571 * USES ALL  1572 * ANA B	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * ENTRY (H,L) - ADDRESS OF REFCAT 1570 * USES ALL 1570 * USES ALL 1571 * ANA B IF NOT TO HANDLE UPDATE 1572 ANA B IF NOT TO HANDLE UPDATE 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT A,M 1576 ANA A,M 1577 AVI L,#DSPROT 1578 ANA A,M 1580 RLC 1581 MOV A,M 1581 MOV A,M 1582 AND A,M 1584 ERRNZ OSPHOD-DSPROT-1 1584 ANI Z 1585 AND A,M 1586 AND A,M 1	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IIHES A SECOND. 1568 * ENIT (H.L.) = ADDRESS OF REFCRIT 1568 * ENIT NONE 1570 * USES ALL 1571 * NONE 1571 * ANA B IF NOT TO HANDLE UPDATE 1572 * ANA B IF NOT TO HANDLE UPDATE 1574 * ANA B IF NOT TO HANDLE UPDATE 1575 * ANA B IF NOT TO HANDLE UPDATE 1576 * ANA C	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN III 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IIMES A SECOND. 1567 * ENITY (H.L.) * ADDRESS OF REFCAT 1568 * ENITY (H.L.) * ADDRESS OF REFCAT 1570 * USES ALL 1571 * NONE 1572 * ANA B * IF NOT TO HANOLE UPDATE 1574 * ANA B * IF NOT TO HANOLE UPDATE 1575 * ANA A** A** A** A** A** A** A** A** A	1570 * USES ALL 1571 1571 1572 1573 002 1574 ANA 8 1575 ANA 8 1577 ANI 1580 RLC 1579 HDV A,M 1580 RLC 1581 HDV A,M H 1584 HDV A,M H H H 1584 HDV A,M H H H H H H H H H H H H H H H H H H H	1570 * USES ALL  1571 1573 1574 1574 1575 1574 1576 1576 1577 1577 1577 1577 1581 1581 1582 1583 1000 1584 1583 1000 1584 1586 1586 1586 1586 1586 1586 1586 1586	1570 * USES ALL  1571	1570 * USES ALL  1571	1570	1570	1572	1572 UFD EQU * 1573 UFD EQU * 1574 MVI A, UD. DDU   1575 MVI A, UD. DDU   1575 MVI A, UD. DDU   1575 MVI A, UD. DDU   1576 MVI C, #DSPROT   1579 MDV A, H C	1572 1573 1574 NVI A, UG. DDU 1575 NVI A, UG. DDU 1575 NVI A, UG. DDU 1577 NVI L, ØDSPROT 1579 NDV A, H 1582 NDV B, A 1583 INX H 1584 NDV B, A 1583 INX H 1584 NDV B, A 1585 NDV B, A 1586 NDV B, A 15	1571 1572 1573 1574 1574 1575 1576 1577 1576 1577 1577 1578 1579 1580 1581 1582 1583 1584 1583 1584 1584 1585 1586 1586 1586 1586 1586 1586 1586	1571 1572 1573 1574 1574 1575 1576 1577 1576 1577 1577 1578 1579 1580 1581 1582 1583 1584 1583 1584 1584 1585 1586 1586 1586 1586 1586 1586 1586	1571 1572 1573 002 1574 002 1575 006 1576 006 1577 006 1580 007 1581 007 1582 007 1584 007 1584 007 1584 007 1586 007 1586 007 1586 007 1586	1572 1574 NVI A, UG. DDU 1575 ANA B 1575 ANA B 1577 ANI A, UG. DDU 1577 ANI A, UG. DDU 1577 ANI A, H 1580 ANI A, H 1584 ANI A, H 1584 ANI A, H 1584 ANI A, H 1585 ANI A, H	002 1573 UFD EQU * 1574 MVI A, UO. DDU 1574 MVI A, UO. DDU 1575 ANA B 1576 ANZ B 1577 AVI L, EDSPROT 1579 MOV A, M A 1581 MOV B, A 1581 MOV B, A 1583 INX H 1583 INX H 1584 ERRNZ OSPHOD-DSPROT-1 1585 AND A, M A 1585 AND A, M A, M A 1585 AND A, M A,	002 1573 UFD EQU	002 1574 MVI A,U0.00U 002 1576 ANI B 1576 ANI B 1577 ANI C, DSPROT 1578 ANI C, DSPROT 1580 ANI H 1584 ERRNZ OSPHOD-DSPROT-1 1585 AND A,M
1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT ISOS **  1565 ** TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS ISOS **  1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 ** ENTRY (H,L) ** ADDRESS OF REFCAT  1569 ** EXIT NONE  1570 ** USES ALL  1571 **  1572 **  1574 **  AND A, UG. DDU  1574 **  AND A, UG. DDU  1575 **  AND A, UG. DDU  1576 **  AND A, H  1577 **  AND A, H  1578 **  AND A, H  1578 **  AND A, H  1581 **  AND A, H  1582 **  AND A, H  1583 **  AND A, H  1584 **  AND A, H  1584 **  AND A, H  1585 **  AND A, H  1586 **  AND A, H  1587 **  AND A, H  1588 **  AND A, H  1589 **  AND A, H  A, H	1562 #  1563 #  1564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 # ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1560 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1560 # EXIT NONE 1560 # EXIT NONE 1570 # USES ALL 1570 # USES ALL 1571 AND A, UD. DDU 1571 AND A, UD. DDU 1572 AND B IF NOT TO HANOLE UPDATE 1573 MYI L, #DSPROT 1574 MOY A, M 1581 MOY B, A 1582 MOY B, A 1583 MOY B, A 1584 MOY B, A 1584 MOY B, A 1585 MOY A, M 1586 MOY A, M 1587 MOY A, M 1587 MOY A, M 1588 MOY A, M 1588 MOY A, M 1588 MOY A, M 1588 MOY A, M 1589 MOY M 1589 MOY A, M 1589 MOY A, M 1589 MOY A, M 1589 MOY A, M 1589 MOY M 1580 MOY M 1	1562 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCNT  1568 * ENTRY (H,L) - ADDRESS OF REFCNT  1570 * USES ALL  1571 * NONE  1572 * ANA B  1574 * ANA B  1575 * ANA B  1576 * MOY A,H  1580 * MOY A,H  1580 * MOY A,H  1580 * MOY A,H  1581 * MOY A,H  1581 * MOY A,H  1583 * INX H  1584 * ERRNZ OSPHOD-DSPROT-I  1585 * MOY A,H  1586 * ERRNZ OSPHOD-DSPROT-I  1586 * MOY A,H  1587 * MOY A,H  1588 * MOY A,H  1588 * MOY A,H  1588 * MOY A,H  1589 * MOY A,H  1580 * MOY A,H  1680 * MOY A,H  1780 * MOY A,H  1780 * MOY A,H	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571  1572  1573  ANA B IF NOT TO HANDLE UPDATE  1575  ANA B IF NOT TO HANDLE UPDATE  1576  ANA B IF NOT TO HANDLE UPDATE  1577  AND A,M  1580  RLC  1580  RLC  1581  AND A,M  1581  AND A,M  1583  AND A,M  1583  AND A,M  1583  AND A,M  1584  AND A,M  1585  AND A,M  1585  AND A,M  1585  AND A,M  1585  AND A,M  1586	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1568 * ENTRY (H,L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * WYI A,UO.DDU 1572 * ANA B 1574 * ANA B 1575 * ANA B 1576 * ANA B 1576 * ANA B 1577 * ANA B 1578 * ANA B 1579 * ANA B 1570 * ANA	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IINES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCRI 1568 * ENTRY (H.L.) - ADDRESS OF REFCRI 1570 * USES ALL 1571 NONE 1572 ANA B IF NOT TO HANDLE UPDATE 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B RC 1576 ANY A.M 1580 ANY A.M	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENITRY (H,L) - ADDRESS OF REFCAT 1568 * ENIT NONE 1550 * USES ALL 1571 * USES ALL 1572 * ANA B IF NOT TO HANOLE UPDATE 1574 * ANA B IF NOT TO HANOLE UPDATE 1575 * ANA B RC 1576 * MOY A,M 1580 * RC 1577 * HOY A,M 1580 * HOY A,M 1580 * HOY A,M 1581 * HOY A,M 1581 * HOY A,M 1583 * INX H 1584 * ERRNZ OSPHOD-DSPROT-I 1584 * ERRNZ OSPHOD-DSPROT-I 1585 * HOY A,M 1585 * HOY A,M 1586 * ERRNZ OSPHOD-DSPROT-I 1586 * ERRNZ OSPHOD-DSPROT-I 1587 * HOY A,M 1588 * HOY A,M 1588 * HOY A,M 1588 * HOY A,M 1588 * HOY A,M 1589 * ERRNZ OSPHOD-DSPROT-I 1589 * HOY A,M 1580	1570 * USES ALL  1571 1573 1574 1573 002 1574 ANA 1575 ANA 1577 AVI A,UO,DDU 1576 ANA B 1577 AVI A,UO,DDU 1579 ANZ BAC 1579 ANY 1580 ANY A,M	1570 * USES ALL  1571 1573 1573 1574 002 1574 002 1575 006 1576 006 1577 006 1578 007 1580 007 1580 007 1581 007 1583 10X 1584 1585 007 1585	1570 * USES ALL  1571	1570 * USES ALL 1571 1572 1573 UFD EQU * 1573 UFD EQU * 1574 ANA B 1576 ANA B 1576 ANA B 1577 ANI CLEDSPROT 1579 ANY CLEDSPROT 1579 ANY CLEDSPROT 1581 ANY H 1582 ANY H 1583 INX H 1584 ERRNZ OSPHOD-DSPROT-1	1570	1570 USES ALL 1571 1572 UFD EQU * 1573 UFD EQU * 1573 UFD EQU * 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANI L, #DSPROT 1579 ANY A, H 1580 RLC RLC 1581 MOV B, A 1583 INX H 1583 INX H 1584 MOV B, A 1585 MOV A, H	1572	1572 UFD EQU * 1573 UFD EQU * 1574 MVI A, UO.DDU   1575 MVI A, UO.DDU   1575 MVI B B B B B B B B B B B B B B B B B B B	1571 1572 1573 1574 MVI A, UG. DDU 1575 MVI A, UG. DDU 1576 MVI L, EDSPROT 1579 MOV A, M 1582 MOV H, A 1582 MOV B, A 1583 MOV B, A 1583 MOV B, A 1583 MOV A, M 1583 MOV A, M 1583 MOV A, M 1583 MOV A, M 1585 MOV A, M	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MVI A, UG. DDU   1574 MVI B B UG. DDU   1576 MVI C, EDSPROT   1579 MOV A, M   1582 MOV B, A   1583 MOV B, A   1583 MOV B, A   1583 MOV A, M   1585 MO	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MVI A, UG. DDU   1574 MVI B B UG. DDU   1576 MVI C, EDSPROT   1579 MOV A, M   1582 MOV B, A   1583 MOV B, A   1583 MOV B, A   1583 MOV A, M   1585 MO	1571 1572 1573 1574 1574 1574 1575 1575 1576 1577 1577 1577 1579 1579 1580 1581 1582 1683 1683 1684 1585 1685 1685 1685 1685 1685 1685 1685	1572 1574 HVI 4,00.000 1575 ANA 8 1575 ANA 8 1577 ANZ 006 1578 HUV A,M 1580 RLC 1581 HOV A,M 1581 HOV A,M 1582 HOV A,M 1583 INX H 1583 INX H 1584 ERRNZ OSPHOD-DSPROT-1	002 1573 UFD EQU * 1574 HVI A,UG.DDU 1575 ANA B 1576 ANA B 1577 ANZ 006 1578 HVI L,#DSPROT 1580 RLC 1581 MOV H,A 1583 INX H 1583 INX H 1584 RRNZ OSPHOD-DSPROT-1 1585 HOV	002 1573 UFD EQU	002 1574 MVI 6400 1575 ANA 8 B.00.000 1576 RNZ 1577 AVI L.#DSPROT 1579 ADV A.M 1580 RLC 1581 MOV M.A 1582 MOV B.A 1583 INX H 1584 ERRNZ OSPHOD-DSPROT-1 1585 ADV
1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1560 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1560 ** EXIT NONE 1570 ** USES ALL 1570 ** USES ALL 1571 ** USES ALL 1571 ** ANA B 1572 ** ANA B 1574 ** ANA B 1576 ** RNZ 1576 ** RNZ 1577 ** ANA B 1576 ** RNZ 1577 ** ANA B 1577 ** ANA B 1578 ** ANA B 1570 ** ANA B 1571 ** ANA B 1571 ** ANA B 1571 ** ANA B 1572 ** ANA B 1573 ** ANA B 1574 ** ANA B 1574 ** ANA B 1575 ** ANA B 1576 ** ANA B 1577 ** ANA B 1578 ** ANA B 1580 ** ANA ANA ANA B 1580 ** ANA ANA ANA ANA ANA ANA ANA ANA ANA	1562 +  1563 +  1564 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 +  1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 + ENTRY (H,L) - ADDRESS OF REFCAT  1569 + EXIT NONE  1570 + USES ALL  1571 - USES ALL  1572 - USES ALL  1573 UFD EQU +  1574 ANA B IF NOT TO HANGLE UPDATE  1575 ANA B ROLL IF NOT TO HANGLE UPDATE  1576 HVI L, BDSPROT  1577 HVI L, BDSPROT  1578 HOV H,A ROTATE PATTERN  1580 HOV H,A ROTATE PATTERN  1580 HOV H,A ROTATE PATTERN  1581 HOV H,A ROTATE PATTERN  1582 HOV B,A HOV H,A ROTATE PATTERN  1583 HOV H,A ROTATE PATTERN  1584 HOV H,A ROTATE PATTERN  1585 HOV B,A HOV H,A ROTATE PATTERN  1585 HOV B,A HOV H,A ROTATE PATTERN  1586 HOV H,A ROTATE PATTERN  1586 HOV H,A ROTATE PATTERN  1586 HOV H,A ROTATE PATTERN	1562 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCRIT 1568 * EXIT NONE 1570 * USES ALL 1571	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) = ADDRESS OF REFCNT  1569 * EXIT NONE  1570 * USES ALL  1571 * USES ALL  1572 * USES ALL  1573 UFD EQU * * *******************************	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * EXIT NONE ENDRESS OF REFCAT 1570 * USES ALL 1571 * WYI A,UG.DDU 1572 UFD EQU * A,UG.DDU 1574 ANA B IF NOT TO HANOLE UPDATE 1575 ANA B ISTO HANOLE UPDATE 1576 NOT A,H	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENITY (H,L) - ADDRESS OF REFCAT 1569 * ENITY (H,L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 USES ALL 1572 UFD EQU * 1573 UFD EQU * 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1576 MOY A.M ROTATE PATTERN 1580 RLC 1577 MOY A.M ROTATE PATTERN 1581 MOY A.M ROTATE PATTERN 1582 MOY B.A.M ROTATE PATTERN 1584 MOY B.A.M ROTATE PATTERN 1585 MOY A.M ROTATE PATTERN 1586 MOY A.M ROTATE PATTERN 1587 MOY A.M ROTATE PATTERN 1588 MOY A.M ROTATE PATTERN 1588 MOY A.M ROTATE PATTERN 1588 MOY A.M ROTATE PATTERN 1589 MOY A.M ROTATE PATTERN 1588 MOY A.M ROTATE PATTERN	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IIHES A SECOND. 1567 * ENITRY (H,L) = ADDRESS OF REFCRI 1568 * ENITRY (H,L) = ADDRESS OF REFCRI 1570 * USES ALL 1571 * USES ALL 1572 * ANA B IF NOT TO HANDLE UPDATE 1574 * ANA B IF NOT TO HANDLE UPDATE 1575 * ANA B * ANA ROTATE PATTERN 1580 * RLC 1579 * MOY * A.* A 1581 * MOY * A.* A 1582 * MOY * A.* A 1582 * MOY * A.* A 1583 * MOY * A.* A 1584 * MOY * A.* A 1585 * MOY * A.* A 1586 * MOY * A.* A 1587 * MOY * A.* A 1588 * MOY * A.* A 1	1570 * USES ALL 1571 1572 1573 1574 ANA 8 1575 ANA 8 1575 ANA 1577 006 1578 ANY L, #DSPROT 1580 RLC 1581 MOY A, M 1582 MOY B, A 1583 1184 ERRNZ 006 05PHOD-DSPROT-1 1584	1570 * USES ALL  1571 1573 1574 1575 002 1574 ANA 8 1575 ANA 8 1577 ANI 1577 ANI 1580 1580 1581 MOY A*H 1583 1184 1584 1585 MOY A*H 1583	1570 * USES ALL 1571	1570 * USES ALL 1571 1572 UFD EQU * 1573 002 1574 MVI A, UD. DDU 1575 RNZ 1576 RNZ 1577 HVI L, #DSPROT 1579 RLC 1580 RLC 1581 ROV N,A 1582 RDC 1581 ROV N,A 1583 INX HSA 1584 ROV A,A	1570	1570	1572 UFD USES ALL 1573 UFD EQU * 1573 UFD EQU * 1574 MVI A, UQ. DDU 1575 ANA B B B B 1576 ANA B B B B 1577 ANVI L, #DSPROT 1579 HOV A, M 1581 HOV N, A 1581 HOV B, A 1584 ERRN OSPHOD-DSPROT-1 1584 ERRN A, M	1571 1572 1573 UFD EQU * 1574 NVI A*,UG.DDU 1575 ANA B B B B B B B B B B B B B B B B B B	1571 1572 1573 1600 1574 1600 1574 1600 1575 1576 1577 1577 1580 1580 1581 1581 1581 1583 1000 1584 1585 1584 1585 1584 1585 1585 1584 1585 1584 1585 1584 1585	1571 1572 1573 1574 1574 1575 1575 1576 1577 006 1577 006 1580 1580 1581 1581 1581 1583 1007 1583 1007 1583 1007 1584	1571 1572 1573 1574 1574 1575 1575 1576 1577 006 1577 006 1580 1580 1581 1581 1581 1583 1007 1583 1007 1583 1007 1584	1571 1572 1573 002 1574 MVI A,u00,000 1576 ANA B 1576 ANA B 1577 AVI C, BSPROT 1580 RCC 1581 MOV A, M A, M B B B B B B B B B B B B B B B B B B B	1576 1576 FOU * 1574 ANA B B B B B B B B B B B B B B B B B B	002 1573 UFD EQU + 1573 UFD EQU + 1574 A,UG.DDU   1575 ANA B   1577   1577 AVI   L.#DSPRÜT   1580   1581   1582   1583   1584   1583   1584   1585   1584   1585   1584   1585	002 1573 UFD EQU * 1574 MVI A, UQ. DDU 1575 ANA B B B B B B B B B B B B B B B B B B	002 1574 MVI 4,00.000 1575 ANA B 1576 RNZ 1577 ANI 1,805PROT 1579 MVI L,805PROT 1580 RLC M,A 1581 MOV M,A 1582 MOV B,A 1584 ERRNZ OSPHOD-DSPROT-1 1584 AND A,M
1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 **  1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) - ADDRESS OF REFCMT  1569 ** ENTRY (H,L) - ADDRESS OF REFCMT  1569 ** ENTRY (H,L) - ADDRESS OF REFCMT  1570 ** USES ALL  1571 **  1572 UFD EQU **  1574 ANA B  1575 ANI A,UG,DOU  1576 ANI B  1577 ANI L,#DSPROT  1578 ANI B  1578 ANI B  1579 ANI A,UG,DOU  1570 ANI B  1570 ANI C,#DSPROT  1570 ANI B  1571 ANI C,#DSPROT  1571 ANI C,#DSPROT  1572 ANI B  1573 ANI B  1574 ANI B  1574 ANI B  1575 ANI B  1576 ANI C,#DSPROT  1576 ANI B  1577 ANI C,#DSPROT  1578 ANI B  1578 ANI C,#DSPROT  1580 ANI A,M  1580 ANI A,	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 **  1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) = ADDRESS OF REFCMT  1569 ** ENTRY (H,L) = ADDRESS OF REFCMT  1570 ** USES ALL  1571 USES ALL  1572 UFD EQU **  1573 UFD EQU **  1574 ANA B  1575 AND A,M  1576 AND A,M  1577 HANOLE UPDATE  1577 HOT IN HANOLE UPDATE  1578 HOY A,M  1580 RLC  1580 HOY A,M  1580 HOY B,A  1581 HOY B,A  1584 ERRNZ OSPHOD-DSPROT-1	1562	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) = ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571 * USES ALL  1572 * ANA B  1574 * ANA B  1574 * ANA B  1575 * ANA B  1576 * ANA B  1577 * ANA B  1570 * OSPHOD-DSPROT-1  1580 * RLC  1	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) * ADDRESS OF REFCRI 1569 * EXIT NONE	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IIMES A SECOND. 1567 * ENITRY (H,L) = ADDRESS OF REFCRI 1568 * ENITRY (H,L) = ADDRESS OF REFCRI 1570 * USES ALL 1571 UFD EQU * ONDE 1572 UFD EQU * ONDE 1573 UFD EQU * ONDE 1574 MVI A,UO.DDU 1575 ANA B 1576 MOY A,H 1580 RLC 1578 MOY A,H 1580 RLC 1581 MOY A,H 1581 MOY A,H 1581 ERRNZ OSPHOD-DSPROT-I	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENITY (H,L) - ADDRESS OF REFCRI 1568 * ENITY (H,L) - ADDRESS OF REFCRI 1570 * USES ALL 1571 UFD EQU * 1572 UFD EQU * 1574 MVI A,UO.DDU 1575 ANA B 1576 RNZ 1577 AVI L,#DSPROT 1578 HOV A,H 1580 RLC 1580 RLC 1580 RLC 1580 RLC 1581 HOV A,H 1581 HOV A,H 1581 HOV B,A 1584 ERRNZ USPHOD-DSPROT-I	1570 * USES ALL 1571 1572 1573 UFD EQU * 1574 ANA B ANA B 1575 ANA B ANA B 1577 ANI A,UG.DDU 1577 ANI A,UG.DDU 1579 ANY A,H 1580 ANY A,H 1580 ANY A,H 1581 ANY A,H 1581 ANY A,H 1581 ANY A,H 1581 ANY A,H 1583 ANY A,H 1583 ANY A,H 1584	1570 * USES ALL  1571 1572 1573 1574 1574 1575 ANA B 1575 ANA B 1577 ANI A,UO.DDU 1577 ANI A,UO.DDU 1577 ANI A,UO.DDU 1579 ANI A,UO.DDU 1579 ANI A,H 1580 ANI A,H 1580 ANI A,H 1581 ANI A,H 1581 ANI A,H 1584	1570 * USES ALL 1571	1570 * USES ALL 1571	1570	1570	1572	1572	1571 1572 1573 1574 1574 1575 1576 1576 1577 1578 1578 1580 1580 1580 1581 1582 1581 1582 1583 1584	1571 1572 1573 1574 1574 1574 1574 1576 1576 1577 1577 1577 1580 1580 1581 1582 1583 1584 1584 1584	1571 1572 1573 1574 1574 1574 1574 1576 1576 1577 1577 1577 1580 1580 1581 1582 1583 1584 1584 1584	1571 1572 1573 002 1574 002 1575 ANA 8 1576 ANA 8 1577 ANI 1578 ANI 1579 ANI 1580 ANI 1580 ANI 1581	1572 1573 002 1574 MVI A,UQ.DDU 1575 ANA B 1575 ANI B 1577 ANI L,BDSPROT 1579 ANY A,H 1580 ACC 1581 ADV A,H 1581 ADV A,H 1581 ADV A,H 1581 ADV A,H 1583 ALC 1581 ADV A,H 1583 ALC 1581 ADV A,H 1583 ADV A,H 1583 ADV A,H	002 1573 UFD EQU * 1574 MVI A,UO.DDU 1575 ANA B 1576 ANZ B 1577 ANZ 1577 AVI L,#DSPRUT 1580 A,M 1580 A,M 1581 ANX H 1583 INX H 1584 EKRNZ OSPHOD-DSPRUT-1	1573 UFD EQU	002 1574 MVI 6400 1574 MVI 74,00.000 1575 ANA 8 1570 AVI L, 60SPROT 1579 MOY A, M 1580 RLC A, M 1581 MOY A, M 1581 MOY A, M 1581 MOY B, A 1583 INX H 1584 ERRNZ 0SPHOD-DSPROT-1
1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS S. TIME TO UPDIS CALLED BY THE CLOCK INTERRUPTY, THIS IS S. TIME TO UPDATE THE UISPLAY CONTENTS. CURRENTLY, THIS IS S. TIME A SOUT 32 TIMES A SECOND.  1560 ** ENTRY (H,L) - ADDRESS OF REFCAT  1560 ** EXIT NONE  1570 ** USES ALL  1571 **  1572 **  1574 **  1574 **  1574 **  1575 **  1576 **  1577 **  1576 **  1577 **  1578 **  1577 **  1578 **  1579 **  1570 **  1570 **  1571 **  1571 **  1571 **  1572 **  1573 **  1574 **  1575 **  1576 **  1577 **  1578 **  1578 **  1579 **  1570 **  1570 **  1571 **  1571 **  1572 **  1573 **  1574 **  1575 **  1576 **  1577 **  1578 **  1579 **  1570	1562 #  1563 #  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) = ADDRESS OF REFORT 1570 * USES ALL 1571 * NONE 1572 * ANA B * UFD TO HANOLE UPDATE 1574 * MYI	1562 * 1564 * UFD USALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * 1186 * 1564 * 1186 * 1564 * 1186 * 1566 * 1666	1566	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * FVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H+L) - ADDRESS OF REFCMT 1569 * ENTRY (H+L) - ADDRESS OF REFCMT 1570 * USES ALL 1571 * NONE * USES ALL 1572 * ANA B * USES OF REFCMT 1574 * NVI A, UO.DDU 1575 * ANA B * USES OF REFCMT 1576 * TO HANDLE UPDATE 1576 * NVI L, *DSPROT 1577 * NVI L, *DSPROT 1578 * NOV H, A ROTATERN 1580 * MOV H, A ROTATERN 1581 * MOV H, A ROTATERN 1582 * MOV H, A ROTATE PATTERN 1583 * INX H	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * ENTRY (H,L) - ADDRESS OF REFCNT 1570 * USES ALL 1571 * NONE 1572 * ANA B * USES 1573 * UFD EQU * 1574 * NVI A,UO.DDU 1575 * ANA B * USES 1576 * NVI L,*DSPROT 1577 * NVI L,*DSPROT 1580 * HOV A,H 1581 * HOV H,A 1582 * HOV B,A 1583 * INX H	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCAT 1568 * ENTRY (H,L) = ADDRESS OF REFCAT 1570 * USES ALL 1571 ** NONE 1572 ** ANA B 1574 ** ANA B 1575 ** ANA B 1576 ** ANA B 1577 ** ANA B 1577 ** ANA B 1578 ** ANA B 1579 ** ANA AAN 1580 ** ANA AAN 1580 ** ANA AAN 1581 ** ADDRESS OF REFCATERN 1582 ** ANA AANA AANA AANA AANA AANA AANA A	1570 * USES ALL 1571 1572 1573 1573 1574 002 1574 ANA 1575 ANA 1576 ANA 1576 ANA 1577 ANI 1580 ANI 1580 ANI 1581 ANI 1583	1570 * USES ALL 1571 1573 1573 1573 1574 002 1574 1575 ANA 8 1576 ANA 8 1576 ANA 1576 ANA 1576 ANA 1576 ANA 1577 AVI C.#DSPROT 1579 ANY A*A 1580 ANY A*A 1581 ANY A*A 1583 ANY A*A	1570 * USES ALL 1571	1570 * USES ALL 1571 1572 1573 002 1573 0FD EQU * 1573 ANA B 1576 ANA B 1576 ANA B 1577 ANI A, UG, DDU 1570 ANI A, UG, DDU 1570 ANI A, UG, DDU 1570 ANI L, #DSPROT 1580 ANI L, #DSPROT 1580 ANI L, #DSPROT 1580 ANI A, A	1570	1570	1570 0355 ALL 1571 1572 0565 ALL 1573 UFD EQU * 1573 UFD EQU * 1575 ANA B 1575 ANA B 1577 AVI L, BDSPROT 1579 HOV A, H 1580 RRC 1580 HOV H, A 1582 HOV H, A	1572 UFD COU # 1572 UFD EQU # 1573 UFD EQU # 1574 MVI A,UO.DDU   1575 ANA B B B. DDU   1577 AVI C,#DSPROT   1579 AVI C,#DSPROT   1580 AVI A,A   1582 AUX H,A   1583 INX H	1571 1572 UFD EQU * 1573 UFD EQU * 1574 NVI A, UG. DDU 1575 ANA B 1576 ANA B 1577 AVI L, EDSPROT 1579 AVI L, EDSPROT 1580 AC  A, A 1581 AUV B, A 1583 INX H	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MVI A, UG. DDU 1575 ANA B 1575 ANA B 1577 AVI L, EDSPROT 1579 AVI L, EDSPROT 1580 RC A, M 1581 AUV B, A 1583 INX H	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MVI A, UG. DDU 1575 ANA B 1575 ANA B 1577 AVI L, EDSPROT 1579 AVI L, EDSPROT 1580 RC A, M 1581 AUV B, A 1583 INX H	1571 1572 UFD EQU * 1573 UFD EQU * 1574 NVI A,UO.DDU 1574 NVI B B B B B B B B B B B B B B B B B B B	1576 1573 UFD EQU * 1574 HVI A,UQ.DDU 1575 ANA B 1576 RNZ 1577 HVI L,#DSPRDT 1579 HOV A,M 1580 RLC 1581 HOV H,A 1582 HOV B,A	002 1573 UFD EQU * 1574 UFD EQU * 1574	002 1573 UFD EQU * 1574 MVI A,UQ.DDU A)02 1575 ANA B B B B B B B B B B B B B B B B B B	002 1574 MVI 6,00.000 1575 ANA 8 1576 RNZ 1577 NVI L,#DSPROT 1579 NOV A,M 1580 RLC 1581 NOV H,A 1582 NOV B,A
1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1566 ** ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1560 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1560 ** ENTRY (H.L.) ** ADDRESS OF REFCAT  1570 ** USES ALL  1571 ** ANA B  1572 ** ANA B  1574 ** ANA B  1576 ** ANA B  1576 ** ANA B  1577 ** ANA B  1578 ** ANA B  1579 ** ANA B  1570 ** ANA B  1570 ** ANA B  1571 ** ANA B  1571 ** ANA B  1572 ** ANA B  1573 ** ANA B  1574 ** ANA B  1575 ** ANA B  1576 ** ANA B  1577 ** ANA B  1578 ** ANA B  1578 ** ANA B  1578 ** ANA B  1571 ** ANA B  1571 ** ANA B  1572 ** ANA B  1573 ** ANA B  1574 ** ANA B  1575 ** ANA B  1576 ** ANA B  1577 ** ANA B  1578 ** ANA B  1788 ** ANA B  1789 ** ANA B  18	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 **  1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 ** ENTRY (H,L) - ADDRESS OF REFCAT  1569 ** EXIT NONE  1570 ** USES ALL  1571 **  1572 **  1574 **  1574 **  1574 **  1575 **  1576 **  1577 **  1578 **  1578 **  1576 **  1577 **  1578 **  1578 **  1579 **  1570 **  1570 **  1570 **  1571 **  1571 **  1572 **  1573 **  1574 **  1576 **  1577 **  1576 **  1577 **  1578 **  1578 **  1579 **  1570 **  15	1562 * 1563 * 1564 * 1565 * 1564 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1566 * 1666 * 16	1566	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE EXIT NONE 1570 * USES ALL 1570 * USES ALL 1571 * WYI A,UO.DOU 1572 ANA B IF NOT TO HANOLE UPDATE 1574 ANY B RADDRESS OF REFCNT 1575 ANA B IF NOT TO HANOLE UPDATE 1576 ANY A,UO.DOU 1577 HVI L,*EDSPROT 1578 HVI L,*EDSPROT 1578 HVI A,UO.DOU 1579 HVI L,*EDSPROT 1570 HVI A,UO.DOU 1571 HVI L,*EDSPROT 1571 HVI A,UO.DOU 1572 HVI L,*EDSPROT 1573 HVI A,UO.DOU 1574 HVI L,*EDSPROT 1575 HVI A,UO.DOU 1576 HVI A,UO.DOU 1577 HVI L,*EDSPROT 1578 HVI A,UO.DOU 1578 HVI A,UO.DOU 1578 HVI L,*EDSPROT 1578 HVI A,UO.DOU 1578 HVI A,UO.DOU 1578 HVI A,UO.DOU 1578 HVI HVI L,*EDSPROT 1578 HVI A,UO.DOU 1578 HVI A,U	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * ENTRY (H,L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 USES ALL 1572 ANA B IF NOT TO HANDLE UPDATE 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B AND ANA B ISTAN 1577 ANA B AND ANA B ISTAN 1577 HOY A,M 1580 RLC 1580 HOY A,M 1580 HOY A,M 1580 HOY A,M 1581 HOY A,M 1582 HOY B,A	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IINES A SECOND. 1567 * ENITRY (H,L) - ADDRESS OF REFCAT 1568 * ENITRY (H,L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 USES ALL 1572 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1577 ANA B ANA ANA ANA ANA ANA ANA ANA ANA A	1570 * USES ALL 1571 1572 1573 002 1574 002 1575 ANA 8 1576 ANA 8 1577 006 1578 AVI L,#DSPROT 1580 RLC 1581 HOV N,A	1570 * USES ALL 1571 1572 1573 002 1574 002 1574 008 1575 006 1578 006 1578 007 1580 008 1580 009 1581 000 009 1581 000 009 009 009 009 009 009 009 009 00	1570 * USES ALL 1572 UFD EQU * 1573 UFD EQU * 1573 UFD RNI A, UO. DDU 1575 ANA B 1576 ANA B 1577 ANI L. EDSPROT 1579 HOV A, H	1570 * USES ALL 1571	1570	1570	1570	1572 UFD 60U * 1572 UFD 60U * 1573 UFD 60U * 1574 MVI A, UG. DDU 1575 ANA B 1576 RNZ 1579 MVI L, #DSPROT 1590 RLC 1581 MOV N, A 1582 MOV 3, A	1571 1572 1573 UFD EQU * 1574 NVI * UG.DDU   1574 1575 ANA B   1576 1577 ANI L.* EDSPROT   1579 1579 ANI A.M   1580 1580 RLC   1581 1581 ADV A.A   1582	1571 1572 UFD EQU * 1574 NVI A,UO.DDU 1574 NVI B 1575 ANA B 1576 ANZ 1577 NVI L, EDSPROT 1579 NOV A,M 1580 RLC 1581 NOV 3,A	1571 1572 UFD EQU * 1574 NVI A,UO.DDU 1574 NVI B 1575 ANA B 1576 ANZ 1577 NVI L, EDSPROT 1579 NOV A,M 1580 RLC 1581 NOV 3,A	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MVI A, UG. DDU 1574 MVI B 1576 MVI B 1577 MVI C, #DSPRDT 1579 MOV A, M 1580 RLC 1581 MOV B, A	1576 1573 UFD EQU * 1574 MVI A, UQ. DDU A 1575 ANA B 1576 ANA B 1577 ANI L, EDSPROT 1578 AVI L, EDSPROT 1580 RLC A, MOV A, MA 1582 ADV B, A	002 1573 UFD EQU + 1574 AND EQU + 1574 AND AND BU 1575 AND B 1577 AND 1577 AND Lybore AND AND AND 1580 RCC 1581 AND BA	002 1573 UFD EQU * 1574 MVI A, UG. DDU 1574 MVI A, UG. DDU 1575 ANA B UG. DDU 1577 MVI L, #DSPROT MVI A, M 1580 RLC 1581 MOV N, A 1582 MOV 3, A	002 1574 MVI 4,00.000 1575 ANA 8 1576 ANZ 1576 RNZ 1577 AVI L, EDSPROT 1580 RLC AM 1581 MOV A,A
1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 **  1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) - ADDRESS OF REFCMT  1569 ** ENTRY (H,L) - ADDRESS OF REFCMT  1570 ** USES ALL  1571 **  1573 UFD EQU **  1574 ANA B IF NOT TO HANOLE UPDATE  1575 ANA B ANA B IF NOT TO HANOLE UPDATE  1576 ANA B ANA B IF NOT TO HANOLE UPDATE  1577 AVI L,*BSPROT  1578 MOY A, M	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 **  1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) = ADDRESS OF REFCMT  1569 ** ENTRY (H,L) = ADDRESS OF REFCMT  1570 ** USES ALL  1571 **  1573 UFD EQU **  1574 ANA B  1575 ANA B  1576 ANA B  1577 AVI L,*DSPROT  1578 MOV A,M  1580 RLC  1580 MOV A,M  1580 RLC  1581 MOV A,M  1580 RLC  1581 MOV A,M  1581 MOV	1562	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) = ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571 * USES ALL  1572 * ANA B  1574 * ANA B  1575 * ANA B  1576 * ANA B  1577 * ANA B  1577 * ANA B  1578 * ANA B  1579 * ANA B  1570 * ANA ANA ANA ANA ANA ANA ANA ANA ANA A	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 * WY A,UQ.DDU 1572 AN B 1574 AN B 1575 ANY B 1576 ANY B 1577 ANY L,*BSPROT 1577 ANY L,*BSPROT 1578 MOY A,M 1580 RLC 1581 MOY A,M 1580 RLC 1581 MOY MAA ROTATE PATTERN	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IIMES A SECOND. 1567 * ENITRY (H.L.) - ADDRESS OF REFCRI 1568 * ENITRY (H.L.) - ADDRESS OF REFCRI 1570 * USES ALL 1571 * USES ALL 1572 UFD EQU * USES 1574 MVI A,UO.DDU 1575 ANA B 1576 ANA B 1577 AVI L, DSPROT 1578 MOY A,M 1580 RLC 1579 MOY A,M 1580 RLC 1581 MOY A,M	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IINES A SECOND. 1567 * ENIRY (H,L) - ADDRESS OF REFCRI 1568 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 UFD EQU * USES ALL 1573 UFD EQU * USES ALL 1574 ANA B 1575 ANA B 1576 ANA ANA B 1577 AVI L, BSPROT 1578 MOY A, H 1580 RLC 1579 HOY A, H 1580 RLC 1581 HOY A, H 1581 HOY A, H 1581 HOY A, H 1581 HOY HAN B 1581 HOY A, H 1581 HOY A, H 1581 HOY A, H 1581 HOY HAN B	1570 * USES ALL 1571 1572 1573 002 1574 002 1575 ANA 8 1576 ANA 8 1577 ANI 1577 ANI 1577 ANI 1580 ANY ANI 1580 ANY ANI	1570 * USES ALL 1571 1572 1573 1574 002 1574 002 1575 ANA 8 1576 ANA 8 1577 ANI 1587 ANI 1580 ANI 1580 ANI 1580 ANI 1581	1570 * USES ALL 1571	1570 * USES ALL 1571	1570 • USES ALL 1571 • USES ALL 1573 UFD EQU • 1573 UFD EQU • 1574 ANA B 1575 ANA B 1577 ANI L.#DSPROT 1579 HOV A.H 1580 RLC 1581 HOV H.A	1570	1572	1572	1571 1572 1573 UFD EQU * 1574 NVI A,UG.DDU 1574 ANA B 1576 RNZ 1577 AVI L,#DSPROT 1579 HOV A,H 1580 RLC 1581 HOV H,A	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MYI A, UG. DDU 1574 MYI B 1576 MNZ B 1577 MYI L, #DSPROT 1579 MOY A, H 1580 RLC 1581 MOY H, A	1571 1572 UFD EQU * 1574 002 1574 NVI A,UG.DDU 1575 ANA B 1576 RNZ 1577 AVI L,#DSPRDT 1579 HOV A,H 1580 RLC 1581 MOV H,A	1571 1572 1573 UFD EQU * 1575 1574 HYI A,UQ.DDU 1575 ANA B 1576 1570 AVI L, BDSPROT 1579 HOV A,H 1580 RLC 1581	1576 EQU * 1574 OCC 1574 OCC 1574 OCC 1576 OCC 1575 OCC 1577 OCC 1578 OCC 1579 OCC 1579 OCC 1580 OCC 1580 OCC 1581 OCC 1581	002 1573 UFD EQU * 1574 MVI A,UO.DDU 1574 MVI A,UO.DDU 1575 ANA B 1577 ANZ 1577 AVI L,#DSPROT 1579 MOV A,M 1580 RLC 1581 MOV H,A	002 1573 UFD EQU * 1574 HVI A, UD. DDU 1575 ANA B 1577 ANZ 1577 AVI L, #DSPROT 1579 HOV A, M 1580 RLC 1581 MOV H, A	002 1574 HVI 4,00.000 1575 ANA B 1575 ANA B 1570 ANI L,#DSPROT 1579 HOY A,H 1580 RLC H,A
1562 ** 1563 ** 1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 ** TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 ** ENTRY (H,L) ** ADORESS OF REFCNT 1569 ** EXIT NONE 1570 ** USES ALL 1570 ** USES ALL 1571 ** ANA B 1573 UFD EQU ** IF NOT TO HANOLE UPDATE 1574 ANA B 1575 ANA B 1576 ANA B 1577 HVI L.*BDSPROT 1578 HOV A.M	1562 #  1563 #  1564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS THE TO UPDATE THE DISPLANTS. CURRENTLY, THIS IS TIME TO UPDATE THE DISPLANT CONTENTS. CURRENTLY, THIS IS THE EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1560 # ENTRY (H,L) # ADDRESS OF REFCAT  1560 # EXIT NONE  1570 # USES ALL  1570 # USES ALL  1571 ANA B  1573 UFD EQU # IF NOT TO HANOLE UPDATE  1574 ANA B  1574 ANA B  1575 ANA B  1576 ANA B  1577 ANA B  1577 ANA B  1578 ANA B  1570 A	1562 *	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) = ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571 * USES ALL  1572 * ANA B  1574 * ANA B  1574 * ANA B  1575 * ANA B  1576 * ANA B  1577 * ANA B  1577 * ANA B  1578 * ANA B  1579 * ANA B  1570 *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) * ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 * HVI A,U0.DDU 1572 ANA B 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANA B 1577 ANA B 1577 ANA B 1578 ANA B 1579 ANA B 1570 ANA B 157	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCRI 1568 * ENTRY (H.L.) - ADDRESS OF REFCRI 1570 * USES ALL 1571 UFD EQU * 1572 UFD EQU * 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANI L.#DSPROT 1579 HOU A, HOULD A, HOULE UPDATE 1579 HOU A, HOULD A, HOULE UPDATE 1570 HANDLE UPDATE 1570 HOUL A, HOULD A, HOUL	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1568 * EXIT NONE 1570 * USES ALL 1571 * HVI A,UO.DDU 1572 * UFD EQU * 1574 * HVI A,UO.DDU 1575 * ANA B 1576 * NA B 1577 * HVI L.#DSPROT 1579 * HOV A,H 1570 * HOV A,H	1570 * USES ALL 1571 1571 1572 1573 002 1574 ANA 8 1575 ANA 8 1576 ANA 1577 ANA 1577 ANI 1580 RLC	1570 * USES ALL 1571 1572 1573 1574 002 1574 1575 ANA 8 1576 1577 1577 1577 1577 1580 RLC	1570 * USES ALL 1571	1570 * USES ALL 1571	1570 • USES ALL 1571	1570	1572	1571 1572 1573 1573 002 1573 1576 1576 1576 1577 1577 1580 1580 1570 1580	1571 1572 1573 UFD EQU * 1574 NVI A,UD.DDU ANA B 1575 ANA B 1576 ANA B 1576 ANZ 1577 AVI L,#DSPROT 1579 AOV A,M	1571 1572 UFD EQU * 1573 UFD EQU * 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANZ 1577 AVI L,#DSPROT 1579 AOV A,M	1571 1572 UFD EQU * 1573 UFD EQU * 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANZ 1577 AVI L,#DSPROT 1579 AOV A,M	1571 1572 1573 UFD EQU * 1574 HVI A,UG,DDU 1575 ANA B 1576 RNZ 1577 HVI L,#DSPROT 1579 HOV A,H	1572 1573 UFD EQU * 1574 MVI A, UG. DDU ANA B 1575 ANA B 1577 ANZ B 1577 ANZ B 1579 AUC A, HOV A, H	002 1573 UFD EQU * 1574 MVI A,UO.DDU 1574 MVI A,UO.DDU 1575 ANA B 1576 RNZ 1577 AVI L,#DSPRUT 1580 RLC	002 1573 UFD EQU * 1574 HVI A, UD. DDU AND B B B B B B B B B B B B B B B B B B B	002 1574 HVI A,UO.DDU 1575 ANA B B B B B B B B B B B B B B B B B B
1562 ** 1563 ** 1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS ** INTER INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1560 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1560 ** EXIT NONE 1560 ** EXIT NONE 1570 ** USES ALL 1571 ** HVI A, UO.DDU 1572 ANA B 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANA CHANDLE UPDATE 1579 ANA ANA 1579 ANA ANA	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) ** ADDRESS OF REFCAT 1569 ** EXIT NONE 1570 ** USES ALL 1571 ** HVI A,UO.DDU 1572 ANA B 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANA B	1562 *	1566	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 USES ALL 1571 AND EQU * IF NOT TO HANDLE UPDATE 1573 UFD EQU * 1574 AND B	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * ENTRY (H,L) - ADDRESS OF REFCNT 1570 * USES ALL 1571 1572 * NA B * USES 1573 UFD EQU * 1574 MVI A,UO.DDU 1575 ANA B IF NOT TO HANGLE UPDATE 1577 ANA B ANA B IF NOT TO HANGLE UPDATE 1578 HOY A,H	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1569 * ENTRY (H.L.) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * NONE * NONE * OOS 1570 * NONE * OOS 1571 * NONE * OOS 1572 * NONE * OOS 1574 * NONE * OOS 1574 * NONE * OOS 1574 * NONE * OOS 1576 * NONE * OOS 1576 * NONE * OOS 1576 * NONE * OOS 1577 * NONE * OOS 1577 * NONE * OOS 1578 * NONE * OOS	1570 • USES ALL 1571 1572 1573 UFD EQU • 1574 HVI A,UG.DDU 1576 ANA B 1576 RNZ 1577 ANI L, EDSPROT 1579 HOY A,H	1570 * USES ALL 1571	1570 * USES ALL 1572 UFD EQU * 1573 UFD EQU * 1573 UFD RAN B 1575 ANA B 1577 ANI L, #DSPROT 006 1578 ANI C, #DSPROT	1570 * USES ALL 1571	1570 • USES ALL 1571 1572	1570	1572 UFD 60U * 1572 UFD 60U * 1573 UFD 60U * 1574 ANA 8 1575 ANA 1570 ANI L, #DSPROT 000 1578 ANI L, #DSPROT ANI A, HOW A, H	1571 1572 UFD EQU * 1573 UFD EQU * 1574 NVI A, UO. DDU 1575 ANA B 1576 RNZ 1577 NVI L, #DSPROT 1579 NOV A, H	1571 1572 1573 UFD EQU * 1574 MVI A, UG. DDU B B B B B B B B B B B B B B B B B B	1571 1572 UFD EQU * 1573 UFD EQU * 1574 AVI A, UG. DDU 1575 ANA B 1576 ANZ BNZ 1577 AVI L, #DSPROT 1579 AVI A, H	1571 1572 UFD EQU * 1573 UFD EQU * 1574 AVI A, UG. DDU 1575 ANA B 1576 ANZ BNZ 1577 AVI L, #DSPROT 1579 AVI A, H	1571 1572 UFD EQU + 10.000 1574 NYI A,UO.00U 1575 ANA B 1576 1577 RNZ B 1577 006 1578 HVI L.#BSPROT 1579 HOY A,M	1572 UFD EQU * 1573 UFD EQU * 1574 HVI A,UG.DDU 1575 ANA B 1576 RNZ 1577 NVI L,#DSPROT 1579 HOY A,H	002 1573 UFD EQU * 1574 HVI A,UD.DDU 1574 HVI A,UD.DDU 1575 ANA B 1577 ANZ 1577 HVI L,#DSPROT 006 1579 HOY A,H	002 1573 UFD EQU * 1574 MVI A,UO.DDU 1575 MNA B 1576 RNZ 1577 OO6 1578 MOY A,H	002 1574 MVI A,UO.DDU 1575 ANA B 1576 RNZ 1577 RNZ 006 1578 MVI L.#DSPROT 1579 MOY A,M
1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENLLY, THIS 1565 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 ** ENTRY (H,L) - ADDRESS OF REFCAT 1569 ** EXIT NONE 1570 ** USES ALL 1571 ** 1572 ** 1573 UFD EQU ** 1574 ANA B IF NOT TO HANOLE UPDATE 1575 ANA B IF NOT TO HANOLE UPDATE 1576 ** 1577 ANI L, #DSPROT	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 **  1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) ** ADDRESS OF REFCAT  1569 ** ENTRY (H,L) ** ADDRESS OF REFCAT  1570 ** USES ALL  1571 **  1572 UFD EQU **  1573 UFD EQU **  1574 AN A B IF NOT TO HANOLE UPDATE  1575 ANA B IF NOT TO HANOLE UPDATE  1576 MAY A.M.O.D.D.	1563	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 * OF EQU * * IS73 OF EQU * * IS74 ANA B 1574 ANA B 1575 ANA B 1576 RNZ IF NOT TO HANOLE UPDATE 1576 RNZ ANA B 1577 OF HANOLE UPDATE 1577 OF TO HANOLE UPDATE 1578 OF TO HANOLE UPDATE 1578 OF TO HANOLE UPDATE 1579 OF TO HANOLE UPDATE 1570 OF TO HANOLE UPDATE	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE ADDRESS OF REFCNT 1570 * USES ALL 1571 * WYI A,UG,DOU 1572 AND B IF NOT TO HANDLE UPDATE 1574 AND B IF NOT TO HANDLE UPDATE 1575 AND A.M. A.M. A.M. A.M. A.M. A.M. A.M. A.M	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENITY (H.L.) - ADDRESS OF REFCAT 1568 * ENITY (H.L.) - ADDRESS OF REFCAT 1570 * USES ALL 1571 USES ALL 1572 UFD EQU * 1573 UFD EQU * 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B ISSPROT 1576 ANA B ISSPROT 1577 ANA CONTENT OF TO HANDLE UPDATE 1578 ANA CONTENT OF TO HANDLE UPDATE 1578 ANA CONTENT OF TO HANDLE UPDATE 1579 ANA CONTENT OF TO HANDLE UPDATE 1579 ANA CONTENT OF TO THANDLE UPDATE 1577 ANA CONTENT OF TO THANDLE UPDATE 1577 ANA CONTENT OF TO THANDLE UPDATE	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IINES A SECOND. 1567 * ENIRY (H.L.) = ADDRESS OF REFCNT 1568 * ENIRY (H.L.) = ADDRESS OF REFCNT 1570 * USES ALL 1571 * USES ALL 1572 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1576 * USES ALL 1577 ANA A.UG.DDU	1570 * USES ALL 1571 1572 1573 002 1574 NVI A,U0.DDU 1575 ANA B 1576 ANA B 1576 ANA B 1576 ANA B 1577 ANA B 1577 ANA B 1577 ANA B 1577 ANA A,U0.DDU	1570 * USES ALL 1571 1571 1572 1573 002 1574 NVI A,UG,DDU 1575 ANA B 1576 RNZ 006 1578 HVI C,#DSPROT	1570 * USES ALL 1571 1572 1573 002 1574 NVI A,UG.DDU 1575 ANA B 1576 RNZ 006 1578 HVI L,#DSPROT	1570 * USES ALL 1571 1573 1574 002 1574 RVI A,UG.DDU 1575 ANA B 1576 RNZ 006 1578 HVI L,#DSPROT	1571 USES ALL 1572 UFD EQU * 1573 UFD EQU * 1574 ANA B 1575 ANA B 1575 ANA B 1575 ANA B 1576 ANA B 1577 ANA B 1578 ANA ANA ANA ANA ANA ANA ANA ANA ANA AN	1571 USES ALL 1571 USES ALL 1573 UFD EQU * 1574 AND AND B 1575 AND B 1575 AND B 1576 AND B 1576 AND B 1577 AND AND B	1572	1571 1572 1573 UFD EQU * 1574 002 1574 HVI #, UO.DDU 1575 ANA B 1576 1576 RNZ NZ 1577 006 1578 HVI L, #DSPROT	1571 1572 1573 UFD EQU * 1574 NVI * 100-DDU 1575 ANA B 1576 RNZ 1577 AVI L,*BSPRDT	1571 1572 UFD EQU + 1573 UFD EQU + 1574 MVI A,UG.DDU 1574 MVI B 1576 ANA B 1576 ANA B 1577 AVI L,#DSPROT	1571 1572 UFD EQU + 1573 UFD EQU + 1574 MVI A,UG.DDU 1574 MVI B 1576 ANA B 1576 ANA B 1577 AVI L,#DSPROT	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MYI A,UG.DOU 1574 MYI B B 1575 ANA B 1576 RNZ B 1577 MYI L,#DSPROT	1572 JFD EQU * 1574 MVI A,UQ.DDU 1574 ANA B 1575 ANA B 1576 ANA B 1577 ANI L,#DSPROT DOG 1579 ANI A.M.	002 1574 UFD EQU + 1575 ANA B 1575 ANA B 1575 ANA B 1576 ANA B 1576 ANZ B 1577 AVI L, #DSPROT OOS 1578 ANY A.M.	002 1573 UFD EQU * 1574 MVI A,U0.0DU 1575 ANA B 1575 ANA B 1576 ANZ 1577 AVI L,#DSPROT 006 1579 ANY A.M.	002 1574 MVI A,UG.DDU 1575 ANA B 1576 RNZ 1576 RNZ 006 1578 MVI L,#DSPRDT
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) * ADORESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 * 1572 * 1573 UFD EQU * 1573 UFD EQU * 1574 ANA B 1574 ANA B 1575 ANA B 1576 ANA B 1577 HANOLE UPDATE 1577 ANA B 1577 HANOLE UPDATE	1562 #  1563 #  1564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT ISOS #  1565 # TIME TO UPDATE THE DISPLAY. CURRENTLY, THIS I ISOS # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 # ENTRY (H,L) # ADORESS OF REFCAT  1569 # EXIT NONE  1570 # USES ALL  1570 # USES ALL  1571 #VI A,UG,DDU  1573 UFD EQU # IF NOT TO HANOLE UPDATE  1574 ANA B IF NOT TO HANOLE UPDATE  1574 ANA B IF NOT TO HANOLE UPDATE	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENIT NONE 1569 * EXIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 * WYI A,UO.DOU 1573 UFD EQU * 1574 WYI A,UO.DOU 1575 ANA B 1575 ANA B 1576 ANA B 1577 ANA B 1577 ANA B 1577 ANA B 1578 ANA B	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 * GOU * * NO.0DU 1574 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE 1576 ANA B IF NOT TO HANDLE UPDATE 1577 ANA B IF NOT TO HANDLE UPDATE	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFCAT 1568 * ENTRY (H,L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * WYI A,UO.DDU 1574 * MYI A,UO.DDU 1575 * ANA B 1576 * ANA B 1577 * ANA B 1577 * ANA B 1578 * ANA B 1578 * ANA B 1578 * ANA B 1579 * ANA B 1570 * ANA B 1570 * ANA B 1571 * ANA B 1571 * ANA B 1571 * ANA B 1572 * ANA B 1573 * ANA B 1574 * ANA B 1575 * ANA B 1576 * ANA B 1577 * ANA B 1577 * ANA B 1577 * ANA B 1578 * ANA B 1578 * ANA B 1578 * ANA B 1579 * ANA B 1570 * ANA B 157	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1568 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 * ANA B 1574 * ANA B 1575 * ANA B 1576 * ANA B 1577 * ANA B 1578 * ANA B 1577 * AN	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 * USES ALL 1571 * WYI A,UO.DOU 1572 ANA B 1574 ANA B 1575 ANA B 1576 ANA B 1576 ANA B 1577 ANA B	1570 * USES ALL 1571 1571 1572 1573 002 1574 NVI A,UG,DDU 1575 ANA 8 1576 ANA 1577 ANA NVI L,#DSPRUT	1570 * USES ALL 1571 1573 1573 002 1574 NVI A,UG,DDU 1575 ANA 8 1576 NVI 1576 NVI 1576 NVI 1578 NVI 1578 NVI 1578	1570 * USES ALL 1571 * USES ALL 1573 UFD EQU * 1573 UFD RVI A,UG.DDU 1575 ANA B 1576 RNZ 1576 ANA B 1576 ANA B 1577 ANA B	1570 * USES ALL 1571	1570	1570	1571 1572 1573 1573 002 1574 ANA A.UG.DDU 1575 ANA B 1576 RNZ 006 1578 AVI L.#DSPRUT	1571 1572 1573 UFD EQU * 1573 UFD EQU * 1574 HVI A,UG.DDU 1575 ANA B 1576 RNZ 006 1578 HVI L,#DSPRUT	1571 1572 1573 UFD EQU * 1573 UFD EQU * 1574 AVI A,UD.DDU 1576 AVI L,#DSPRUT	1571 1572 UFD EQU * 1573 UFD EQU * 1574 NVI A, UD. DDU 1575 ANA B 1576 ANA B 1576 ANA COS 1578 AVI L, #DSPROT	1571 1572 UFD EQU * 1573 UFD EQU * 1574 NVI A, UD. DDU 1575 ANA B 1576 ANA B 1576 ANA COS 1578 AVI L, #DSPROT	1571 1572 UFD EQU * 1573 UFD EQU * 1574 NVI A,UG.DDU 1575 ANA B 1570 RNZ 1577 AVI L,#DSPROT	1572 1573 UFD EQU * 002 1574 MVI A,UQ.DDU 1575 ANA B 1576 RNZ 006 1578 MVI L,#DSPROT	002 1573 UFD EQU * 1574 MVI A,UO.DDU 1574 MVI A,UO.DDU 1575 ANA B 1576 RNZ 1577 AVI L,#DSPRUT	002 1573 UFD EQU * 1574 HVI A,UD.DDU 1575 ANA B 1576 RNZ 1577 006 1578 HVI L,#DSPROT	002 1574 HVI A,UO.DDU 1575 ANA B 1576 RNZ 1577 HVI L,#DSPROT
1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS *  1564 * TIME TO UPDATE THE DISPLAY. CONTENTS. CURRENTLY, THIS IS *  1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * ENTRY (H,L) * ADDRESS OF REFONT IS *  1569 * EXIT NONE IS OF REFONT IS *  1570 * USES ALL IS *  1571 IS *  1572 ANA B IF NOT TO HANOLE UPDATE IS *  1574 ANA B IF NOT TO HANOLE UPDATE	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** TIME TO UPDIE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) ** ADDRESS OF REFONT 1569 ** EXIT NONE 1570 ** USES ALL 1571 ** 1572 ** 1574 ** 1574 ** 1575 ** ANA B 1576 ** 1577	1562 * 1563 * 1564 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1565 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1568 * 1568 * 1568 * 1568 * 1569 * 1571 * 1569 * 1571 * 1571 * 1572 * 1571 * 1572 * 1573 * 1573 * 1574 * 1571 * 1572 * 1574 * 1574 * 1575 * 1575 * 1575 * 1575 * 1575 * 1575 * 1575 * 1575 * 1575 * 1575 * 1577 * 15	1566	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 4 WVI A,U0.DDU 1573 UFD EQU * 1574 MVI A,U0.DDU 1575 ANA B 1576 RNZ IF NOT TO HANOLE UPDATE 1577	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADORESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 EQU * 1574 ANA B 1574 ANA B 1576 ANA B 1577 IF NOT TO HANDLE UPDATE	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 ** USES ALL 1571 ** HVI A,U0.0DU 1573 UFD EQU ** ANA B 1576 ** NA B 1577	1570 * USES ALL 1571 1572 1573 002 1574 MVI A,UG.DDU 1575 ANA B 1576 1577	1570 * USES ALL 1571	1570 * USES ALL 1571	1570 * USES ALL 1572 UFD EQU * 1573 UFD EQU * 1574 ANI A, UG. DDU 1575 ANA B 1577	1570 • USES ALL 1572 UFD EQU • 1573 UFD EQU • 1574 ANI A, UB. DDU 1575 ANA B 1577 ANA B	1570 • USES ALL 1571 1572 1572 EQU • 1573 UFD EQU • 1574 ANI A, UD. DDU 1575 ANA B 1577 RNZ 1577	1571 1572 1573 UFD EQU * 1573 UFD EQU * 1574 ANI 8 UD.DDU ANI 1576 1577	1571 1572 1573 UFD EQU * 1573 UFD EQU * 1574 ANI B UD.DDU ANI B UD.DDU BNZ	1571 1572 UFD EQU * 1573 UFD EQU * 1574 HVI A,UD.DDU B 1575 ANA B 1576 RNZ 1577	1571 1572 UFD EQU * 1573 UFD EQU * 1574 NVI A,UQ.DDU ANA B 1575 ANA B 1570 RNZ 1577	1571 1572 UFD EQU * 1573 UFD EQU * 1574 NVI A,UQ.DDU ANA B 1575 ANA B 1570 RNZ 1577	1571 1572 UFD EQU * 1573 UFD EQU * 1574 AVI A,UQ.DDU 1575 ANA B 1576 RNZ 1577	1572 UFD EQU * 1573 UFD EQU * 1574 MYI A,UD.DDU 1575 ANA B 1576 RNZ 1577	002 1573 UFD EQU * 1574 HVI A,UD.DDU 1574 HVI B 1575 ANA B 1577 RNZ	002 1573 UFD EQU * 1574 HVI A,UD.DDU 1575 ANA B 1576 RNZ 1577	002 1574 HVI A,UO.DDU 1576 ANA B 1576 ANA 1577 1577
1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1566 ** ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H.L.) ** ADDRESS OF REFCAT 1569 ** EXIT NONE 1570 ** USES ALL 1571 1572 ** HVI 1572 ** HVI 1573 UFD EQU ** 1574 ** HVI 1575 ANA B 1576 RNZ	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** IME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IN 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1560 ** ENTRY (H,L) - ADDRESS OF REFCAT 1569 ** EXIT NONE 1570 ** USES ALL 1571 1572 1573 UFD EQU ** 1574 MVI A,UO.DDU 1575 ANA B 1576 RAZ	1562 * 1563 * 1564 * 1565 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1564 * 1565 * 1565 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1566 * 1568 * 1567 * 1568 * 1570 * 1569 * 1570 * 15	1566	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 PROCESSOR WHEN II 1574 AND BOULDOU 1574 AND BOULDOU 1575 AND BOULDOU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * ENTRY (H,L) - ADDRESS OF REFCNT 1570 * USES ALL 1571 1572 ANA B IF NOT TO HANDLE UPDATE 1575 ANA B IF NOT TO HANDLE UPDATE	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L) - ADDRESS OF REFCAT 1569 * ENTRY (H.L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 1572 USES ALL 1573 UFD EQU * AUG.DDU 1575 ANA B IF NOT TO HANGLE UPDATE 1576 RNZ	1570 * USES ALL 1571 1572 1573 002 1574 MVI A,UG.DDU 1575 ANA B B	1570 * USES ALL 1571 1572 1573 002 1574 MVI A,UG.DDU 1575 ANA 8	1570 * USES ALL 1571 1572 1573 002 1574 MVI A,UG,DDU 1575 RNZ RNZ	1570 * USES ALL 1571 1572 1573 002 1574 MVI A,UG,DDU 1576 RNZ	1570	1570 • USES ALL 1571 1572 1572 60U • 1573 UFD 60U • MVI A,UQ.DDU 1575 ANA 8 1576 RNZ	1570 - 0355 ALC 1572 1572 EQU * 002 1574 MYI A,UQ.DDU 1575 ANA B	1571 1572 1573 1573 UFD EQU * 1574 MYI A,UQ.DDU 1576 RNZ	1571 1572 1573 UFD EQU * 1573 UFD EQU * 1574 MYI A,UG.DDU 1575 ANA B 1576	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MYI A,UO.DDU 1574 ANA B 1576 ANA B 1576	1571 1572 UFD EQU * 1573 UFD EQU * 1574 MYI A,UO.DDU 1574 ANA B 1576 ANA B 1576	1571 1572 UFD EQU + 1573 UFD EQU + 1574 HVI A,UO.DDU 1575 ANA B 1576 RNZ	1572 UFD EQU * 1573 UFD EQU * 1574 MYI A,UG.DDU 1575 ANA B 1576 RNZ	002 1573 UFD EQU + 1574 NVI A,UQ.DDU 1575 ANA B 1576 RNZ	002 1573 UFD EQU * 1574 MVI A,UO.DDU 1575 ANA B 1576 RNZ	002 1574 AVI A,UO.DDU 1575 ANA 8 0.000
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1573 UFD EQU * 1574 ANA B	1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT  1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I  1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * ENTRY (H,L) - ADDRESS OF REFORT  1569 * EXIT NONE  1570 * USES ALL  1571  1571  1573  1574  ANA B  1574  ANA B	1562	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 * HVI A,UO.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 * MYI A,UO.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCNT 1568 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * HVI A.UO.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1568 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * HVI A.UO.DDU	1570 * USES ALL 1571 1572 EQU * 1573 UFD EQU * 002 1574 MVI A,UQ.DDU	1570 * USES ALL 1571 1572 1573 002 1574 ANA 8	1570 * USES ALL 1571 1572 1572 EQU * 002 1574 AVI A,UQ.DDU	1570 * USES ALL 1571 * USES ALL 1572 UFD EQU * 1574 MVI A,UG.DDU	1571	1571 - USES ALL 1571 - USES ALL 1573 UFD EQU + UG.DDU 002 1574 ANA B	1572	1572 1572 1572 1572 000	1571 1572 1573 UFD EQU * 1573 UFD RVI *,UO.DDU 1575 ANA B	1571 1572 1573 UFD EQU * 1573 UFD HVI *,UO.DDU 1575 ANA B	1571 1572 1573 UFD EQU * 1573 UFD HVI *,UO.DDU 1575 ANA B	1571 1572 1573 UFD EQU * 1002 1574 HVI A,UG.DDU 1575 ANA B	1572 1573 UFD EQU * 1574 MVI A,U0.DDU B B 1575	002 1573 UFD EQU + 1574 NVI A,UQ.DDU 1575 ANA B	002 1574 UFD EQU 4 002 1574 MVI A,UG.DDU 1575 ANA B	002 1574 AVI A,UO.DDU 1575 ANA B
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 * TIME TO UPDATE THE DISPLAY. CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) * ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 * 1572 * 1574 ** 1574 ** 1574 ** 1574 ** 1575 ** 1577	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) ** ADDRESS OF REFONT 1569 ** EXIT NONE 1570 ** USES ALL 1571 ** 1572 ** 1574 ** 1574 **  1574 **  1574 **  1574 **  1574 **  1563 **  1574 **	1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IN 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IIHES A SECOND.  1568 * ENTRY (H,L) = ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 HWYI A,UG.DDU	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + ENTRY (H.L.) - ADDRESS OF REFCAT 1569 + EXIT NONE 1570 + USES ALL 1571 1572 1573 UFD EQU + 1574 MVI A, UD. DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 1573 UFD EQU * 1574 MYI A, UD.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCHT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 USES ALL 1574 MYI A,UO.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 USES ALL 1574 MYI A,UO.DDU	1570 * USES 1571 1572 1573 UFD EQU 002 1574 HVI	1570 * USES 1571 1573 UFD EQU 1573 UFD RVI	1570 * USES 1571 1573 UFD EQU 1573 UFD EQU	1571 1572 1571 1573 150 EQU 1573 UFD EQU	1571 1571 1573 1573 UFD EQU 1574 MVI	1570 • 03E3 1572 1573 UFD EQU 1573 UFD FWI	1572 1572 1573 1573 002 1574 FVI	1571 1572 1573 1573 002 1574 HVI	1571 1572 1573 UFD EQU 1573 UFD EQU	1571 1572 1573 UFD EQU 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 HVI	1573 UFD EQU 002 1574 HVI	1573 UFD EQU 002 1574 HVI	1573 UFD EQU 002 1574 MVI	002 1574 UFU EQU
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * IMPRINATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 UPD EQU *	1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571 USES  1573 UFD EQU *	1564 * UFD USALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IN 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H+L) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 UFD EQU *	1566	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1573 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCMT 1569 * ENTRY (H,L) - ADDRESS OF REFCMT 1570 * USES ALL 1571 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1569 * ENTRY (H.L.) - ADDRESS OF REFCAT 1570 * USES ALL 1571 UFD EQU *	1570 • USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 • USES 1571 1572 1573 UFD EQU	1570 • USES 1571 1572 1573 UFD EQU	1570 • USES 1571 1572 1573 UFD EQU	1570 • 03E3 1571 1572 1573 UFD EQU	1571 1571 1572 1573 UFD EQU	1571 1572 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	1573 UFD EQU
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) * ADDRESS OF REFORT 1569 * EXIT NONE 1570 * USES ALL 1571 1573 UFD EQU *	1562 ** 1563 ** 1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 ** TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 ** ENTRY (H,L) ** ADORESS OF REFORT 1569 ** EXIT NONE 1570 ** USES ALL 1571 1572 ** USES ALL 1572 ** USES	1562 * 1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IN 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. IS 69 * EXIT NONE EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * USES ALL 1573 UFD EQU **	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + ENTRY (H.L.) = ADDRESS OF REFCMT 1569 + EXIT NONE 1570 + USES ALL 1571 1572 - L573 UFD EQU +	1564	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCHT 1569 * EXIT NONE 1570 * USES ALL 1571 1573 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCHT 1569 * EXIT NONE 1570 * USES ALL 1571 1573 UFD EQU **	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * 03E3 1571 1572 1573 UFD EQU	1570	1570 + 0353 1571 1573 UFD EQU	1571 1572 1573 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1572 1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	15/3 050 E40
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL 571	562 + 563 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFCNT 569 + EXIT NONE 570 + USES ALL 571	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 564 * THE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 550 * EXIT NONE 570 * USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCAT 569 + EXIT NONE 570 + USES ALL 571	20.5. * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 56.5. * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 56.5. * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 56.7. * ENTRY (H,L) - ADDRESS OF REFCNT 55.9. * EXIT NONE 57.0. * USES ALL 57.1.	20.3	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 570 * USES ALL 572	570 * USES 571 572	570 * USES 571 572	570 * USES 571 572	570 * USES 571 572	570 • USES 571 572	570 • USES 571 572	571 572	571 572	571 572	571 572	571 572	571 572				
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL	562 * 563 * 10FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * TIME TO UPDATE THE DISPLAY CONTENTS A SECOND. 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE 570 * USES ALL	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCHT 569 # EXIT NONE 570 * USES ALL	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFCMT 569 + EXIT NONE 571	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * EXIT NONE 559 * EXIT NONE 570 * USES ALL	504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 559 * EXIT NONE 550 * USES ALL 571	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCAT 569 * EXIT NONE 559 * EXIT NONE 570 * USES ALL	570 * USES 571	570 * USES 571	570 * USES 571	570 * USES 571	570 • USES 571	570 • USES 571	570 ÷ 0353	571	571	571	571	571		7/67		
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * THE TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADORESS OF REFORT 569 * EXIT NONE 570 * USES ALL	562 + 563 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFONT 568 + EXIT NONE 570 + USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADORESS OF REFCNT 569 * EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L) = ADDRESS OF REFCNT 568 + EXIT NONE 570 + USES ALL	505.3 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565.4 TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566.4 EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567.4 ENTRY (H,L) - ADDRESS OF REFCHT 569.4 EXIT NONE 570.4 USES ALL	504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 5065 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL	565 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL	570 <b>*</b> USES	570 * USES	570 * USES	570 * USES	570 <b>*</b> USES	570 * USES	370 ÷ 03E3	571	571	571	571	571		1673		3)(7
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) - ADORESS OF REFONT 569 * EXIT NONE 570 * USES	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFONT 569 * EXIT NONE 570 * USES	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, IHIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADDRESS OF REFONT 569 * USES ALL	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 569 + EXIT NONE 569 + EXIT NONE 570 + USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT 569 * EXIT NOME 570 * USES	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT 569 * EXIT NOME 570 * USES	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCRI 569 * EXIT NOME 570 * USES ALL	570 * USES	570 * USES	570 * USES	570 * USES	5/0 • USES	3/0 • USES	. 025	0353	200						6.43	1572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFONT 568 * EXIT NONE	562 # 563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFORT	563 # 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + 568 + ENTRY (H,L) = ADORESS OF REFCNT 569 + EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFORT			֡֡֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜				1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	220 -	570 <b>*</b> 115F5	570 * USES	570 * USES	3/0 • 0353	1571	1571	1571	1571 1572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCHT	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) = ADDRESS OF REFCNT	562 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) - ADDRESS OF REFCHT	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 648 * CATE MANE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT												- VE	570 * USES 571	570 * USES 571	570 * USES 571	570 * USES 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFORT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFORT	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT	<ul> <li>EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.</li> <li>ENTRY (H,L) = ADDRESS OF REFONT</li> </ul>	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II I I I I I I I I I I I I I I I I I	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. # ENTRY (H,L) = ADDRESS OF REFORT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. * EVERY 32 INTERRUPIS, OR ABOUT 34 TIMES A SECOND. * ENTRY (H,L) * ADDRESS OF REFORT	֡											1101	570 • USES 571	570 * USES 571	570 * USES 571 * USES 571	570 • USES 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	+ EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II # IIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT INTER TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	540 * CYIT	569 * FXIT	569 * FXIT	569 * EXIT	569 * EXII	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	# EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN # IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT INTER TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	560 + ENIKT (H)L) = AUUKESS 540 + EVIT MONE	569 # FXIT NONE	569 # FXII NONF	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 # EXIT NONE	569 # EXIT NONE	3569 # ENIKI (H1) = AUUKESS 5569 # EXIT NONE	508 + EXIT (19L) = AUDKESS 570 + EXIT NONE 570 571 + USES ALL	568 + ENIKT (19L) = AUUKESS 569 + EXIT NONE 570 + USES ALL	569 # EXIT NONE 571 # USES ALL	569 # EXIT NONE AUDKESS 570 # USES ALL 571.
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	<ul> <li>EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.</li> </ul>	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	568 * ENTRY (H,L) = ADDRESS	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NOME	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NOWE	568 + ENTRY (H+L) = ADDRESS 569 + EXIT NONE	568 + ENTRY (H+L) = ADDRESS 569 + EXIT NONE	568 + ENTRY (H,L) = ADDRESS 569 + EXIT NONE	568 + ENTRY (H,L) = ADDRESS 569 + EXIT NONE	568 + ENTRY (H,L) = ADDRESS 569 + EXIT NONE	568 + ENTRY (H,L) = ADDRESS 569 + EXIT NONE	568 # ENTRY (H <sub>9</sub> L) = ADDRESS 569 # EXIT NONE	568 # ENTRY (H <sub>9</sub> L) = ADDRESS 569 # EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 + ENTRY (H <sub>2</sub> L) = ADDRESS 569 + EXIT NONE 570 + USES ALL 571	568 + ENTRY (H <sub>3</sub> L) = ADDRESS 569 + EXIT NONE 570 + USES ALL 571	568 + ENTRY (H <sub>2</sub> L) = ADDRESS 569 + EXIT NONE 570 + USES ALL 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 # 563 # 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	C. C	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	567 * ENTRY (H.L.) = ADDRESS	567	567	567	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	567	567 * 568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	567 * 568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	567 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	567 * ENTRY (H <sub>3</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	567 * ENTRY (H <sub>3</sub> L) = ADDRESS 568 * EXIT NONE 570 * USES ALL 571 * USES	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	567 * ENTRY (H <sub>1</sub> L) = ADDRESS 568 * EXIT NONE 570 * USES ALL 571 572
563 * S64 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	563 * SEA TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 # 562 # 563 # 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT		* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY. THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY. THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY. THIS I	567 # ENTRY (H <sub>2</sub> L) = ADDRESS OF	567 # ENTRY (H-L) = ADDRESS OF 568 # EXIT NONE	567 # ENTRY (H-L) = ADDRESS OF 568 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 568 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 * ENTRY (H,L) = ADDRESS OF EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567	567 # ENTRY (H,L) = ADDRESS OF 568 # EXIT NONE 570 # USES ALL 571	567 # ENTRY (H,L) = ADDRESS OF EXIT NONE 570 # USES ALL 571	567 * ENTRY (H,L) = ADDRESS OF EXIT NONE 550 * EXIT NONE 570 * USES ALL 571 * 571 * 571 * 572 * 572 * 573 * 574 * 575 *
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	562 + 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	# TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 648 * CYTT MOME	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT 568 * FXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCAT 568 * FXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-LL) - ADDRESS OF REFONT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-LL) = ADDRESS OF REFONT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	566	566	566	566	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCMT 569 + EXIT NONE 570 + USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H+L) - ADDRESS OF REFCMT 569 + EXIT NONE 570 + USES ALL	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L.) = ADDRESS OF REFCNT 568 + EXIT NONE 570 + USES ALL 571	566
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	562 * 562 * SERVED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	A TIME TO HODATE THE DISOLAY CONTENTS. CHORENTLY, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	566	566	566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 569 # EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 569 # EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	566	566 * EYERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE 570 * USES ALL 571	566
562 <del>+</del> 563 <del>+</del>	562 + 563 +	562 # 563 #	* OFD 13 CALLED BY THE CLUCK INTEXRUPL PRUCESSUR WHEN IT				565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFONT	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFONT 569 * FXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADORESS OF REFONT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) = ADDRESS OF REFCMT 559 * EXIT NONE 570 * USES ALL	565	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCMT 569 * EXIT NONE 570 * USES ALL 571 * USES
562 * 563 *	562 <b>*</b> 563 <b>*</b>	562 * * OTD - OTDALE TAUNI FANCE UL	# UPD IN CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	•		•	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 648 * ENTRY (H,L) = ADDRESS OF REFCNT	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # ENTRY (H,L) = ADDRESS OF REFCNT	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 569 # FXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFONT 569 * EXIT NONE	565 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H-LL) - ADDRESS OF REFONT 568 + EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 550 *	565	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H.L.) = ADDRESS OF REFCHT 569 # EXIT NONE 570 * USES ALL 572
562 *	562 *	501 + OTD - OTDAILE TRUN FANCE DISTERS					564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFONT 569 * FXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * ENTRY ONLY	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL 572
562 *	562 *	562 #	563 *	* 673		4 6734	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT CALL CALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFORT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * ENTRY (H,L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 557 * ENTRY (H,L) = ADDRESS OF REFONT 568 * ENTRY (H)L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 557 * ENTRY (H,L) - ADDRESS OF REFONT 568 * ENTRY (H)L) - ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADORESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADORESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) = ADDRESS OF REFCHT 569 * EXIT NONE 559 * EXIT NONE 559 * EXIT NONE 550 * TIMES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H*L) = ADDRESS OF REFCNT 569 * EXIT NONE 569 * EXIT NONE 571	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 572
		TOTAL TARGET OF THE TARGET OF THE OFFICE OFF	+ 70CT	2001			563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFONT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCNT 568 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFORT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFORT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 5770 * USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UF. TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  567 * ENTRY (H,L) = ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCHT 550 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL
		JOE TE OLD - OFD -	562 *	1207			563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * ITHE TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCHT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) - ADDRESS OF REFONT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,LL) - ADDRESS OF REFONT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TINE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 665 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 665 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * ILME IS UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFONT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TINE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) - ADDRESS OF REFCAT 659 * EXIT NONE 570 * USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * EXIT (H*L) - ADDRESS OF REFCMT 569 * EXIT NONE 570 * USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCNT 559 * EXIT NONE 571 * USES ALL
		TIVILLE TARREST TARRES			- \2 <u>C</u>	<b>1</b> 2007	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 # TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,L) = ADDRESS OF REFCNT 6717 MONE	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,L) = ADDRESS OF REFCHT 569 # FXIT NONE	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,L) = ADDRESS OF REFORT	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,L) = ADDRESS OF REFONT 569 # EXIT NONE	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,LL) = ADDRESS OF REFONT 569 # EXIT NONE	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,L) - ADDRESS OF REFONT	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, IHIS IS 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H+L) = ADDRESS OF REFCHT 568 # EXIT NONE	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 # TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, IHIS IS 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCHT 568 # EXIT NONE	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 # ITHE TO UPDATE THE DISPLAY CONTENTS, CURRENTES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCHT 569 # EXIT NONE	202 * 504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 505 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A 568 * EVERY (H,L) = ADORESS OF REFORT 569 * EXIT NONE	202 * 504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 505 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY 32 INTERRUPTS, OR ABOUT 34 TIMES A 568 * EVERY (H,L) = ADORESS OF REFORT 569 * EXIT NONE	563 * UFD IS CALLED BY THE CLUCK INTERRUPT PROCESSOR WHEN ITS 564 * UFD IS CALLED BY THE CLUCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	503 * 504 * 505 * 505 * 505 * 607 * 607 * 608 * 608 * 608 * 608 * 608 * 609 *	563 # 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H+L) = ADDRESS OF REFCAT 569 # EXIT NONE 570 # USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H.L.) - ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL	256. * 564. * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564. * UFD IS CALLED BY THE ULSPLAY CONTENTS. CURRENTLY, THIS I 565. * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566. * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567. * ENTRY (H,L) = ADDRESS OF REFCNT 569. * EXIT NONE 570. * USES ALL 572.
SAL ## INDIATE FROM PANEL DICPLAYS	561 ** UPDATE FRONT PANEL DISPLAYS	SATIONAL TRACES STANDING TO THE TANDING TH	TOTAL TRANSPORT OF THE PARTY OF	56.2 ±	242 #	562 *	562 * 563 * 10FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) - ADDRESS OF REFONT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERUPT PROCESSOR WHEN IT 564 * IIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * ENTRY (H,L) = ADDRESS OF REFCNT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE ULOCK INTERUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * FXIT NONE	562 # 563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # EXIT NONE	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	562 # 563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # EXIT NONE	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * INTER INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADDRESS OF REFORT	562 # 563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLLY, THIS IN 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) # ADDRESS OF REFORT 568 # EXIT NONE	562 # 563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLLY, THIS IN 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) # ADDRESS OF REFORT 568 # EXIT NONE	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 569 * EXIT NONE	562 * 563 * 100 IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * 11ME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * 12ME TO UPDATE THE DISPLAY CONTENTS A SECOND. 567 * 100 EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * 100 EXIT NONE 5770 * 100 USES ALL	562 *	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * 11ME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCHT 568 * 569 * EXIT NONE 570 * 571 USES ALL	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * EVIR (H.L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 572
SKI AM HEN - HEN - HENNT BANEL DICELLAY	SKI AM HEN - HEN - HENNT BANEL DICELLAY	SVA 19310 LIBOAT CONTRACT CONTRACT OFFI	TOTAL TARGET PARTY	201 - 101 - 0101 - 1011		562 +	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, IHIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCHT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCHT FALLS FOR THE FORT FALLS FALLS FOR THE FORT FALLS	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IN 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,LL) - ADDRESS OF REFONT	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADORESS OF REFCHT 568 * EXIT NONE	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * ENTRY (OR SECOND.	563 # 563 # 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * LIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE	563 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT	562 * 563 * 10FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * THE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFONT 568 * EXIT NONE	562 + 563 + 10FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 + ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + EVERY 32 INTERRUPTS, OF REFCAT 568 + ENTRY (H,L) - ADDRESS OF REFCAT 559 + EXIT NONE 570 + USES ALL	562 # 563 # 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE 570 * USES ALL	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * LIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE 570 * USES ALL	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571
562 *	562 ♦	562 #	• 673	• 673			564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L) = ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * FXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H.L.) - ADDRESS OF REFORT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFORT 568 * EXIT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCHT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY (H,L) = ADDRESS OF REFORT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY (H,L) = ADDRESS OF REFORT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * ENTRY (H,L) - ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE 5770 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCAT 569 * EXIT NONE 550 *	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE 559 * EXIT NONE 571	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL 572
562 #	562 #	562 *	*	•	4 474	+ 6731	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCHT 540 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFORT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) * ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EXIT NONE 569 * EXIT NONE 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 572
562 *	562 *	562 *	*	*		4 6731	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 548 * ENTRY (H,L) = ADDRESS OF REFCHT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFORT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 557 * ENTRY (H,L) = ADDRESS OF REFONT 568 * ENTRY (H,L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 557 * ENTRY (H,L) - ADDRESS OF REFONT 568 * ENTRY (H)L) - ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, IHIS IS66 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADORESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADORESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H*L) = ADDRESS OF REFCNT 569 * EXIT NONE 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L) - ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 572
562 +	562 <b>*</b>	562 * COTO OF	•	•			564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT CALL CALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFORT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 557 * ENTRY (H,L) = ADDRESS OF REFONT 568 * ENTRY (H,L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * ENTRY (H)L)	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * ENTRY (H)L)	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS IS 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS IS 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS IS 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 559 * EXIT NONE 559 * EXIT NONE 550 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) = ADDRESS OF REFCMT 559 * EXIT NONE 559 * EXIT NONE 559 * EXIT NONE 559 * EXIT NONE 550 * E	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE 569 * EXIT NONE 569 * EXIT NONE 569 * EXIT NONE 571	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 569 * EXIT NONE 570 * USES ALL
562 *	562 *	501 + OTD - OTDAILE TRUN FANCE DISTERNIS	•				564 * UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT CALL CALL CALL CALL CALL	564 * UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT	564 * UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 557 * ENTRY (H,L) - ADDRESS OF REFONT 568 * ENTRY (H)L) - ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * ENTRY (H)L) - ADDRESS OF REFCHT 569 * EXIT NONE	564 * UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * IINE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE	564 * UPD IS CALLED BY THE CLÜCK INTERUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLÜCK INTERUPT PROCESSOR WHEN IN 565 * IINE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCMT 568 * ENTRY (H,L) = ADDRESS OF REFCMT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCNT 5569 * EXIT NONE 5569 * EXIT NONE 5570 * USES ALL	964 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 569 * EXIT NONE 569 * EXIT NONE 570 * USES ALL	964 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 569 *	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL
562 <del>*</del> 562 <del>*</del>	562 *	501 ++ OTD - OTDRIE FRONT FANCE DISCLATS 562 #					564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TINE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * EXIT NONE 5770 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TINE TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCAT 568 * EXIT NONE 570 * USES ALL 572
562 * 563 *	562 <b>*</b> 563 <b>*</b>	562 * * OTD - OTDALE TAUNI FANCE UL		•		•	565 # UPD 13 CALLED BT THE CLUCK INTEXNUT! FRUCESSUK MIEN 11 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H-L) = ADDRESS OF REFONT	565 # TIME TO UPDATE THE CLUCK INTEXNUT! FRUCESSUK MIEN 11 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,L) = ADORESS OF REFONT 569 # FXII NONE	565 # TIME TO UPDATE THE CLUCK INTEXNUT! FRUCESSUK MEN IN 566 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 567 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,L) = ADORESS OF REFCNT 569 # FXIT NONE	565 # TIME TO UPDATE THE CLUCK INTEXNUT! FRUCESSUK MEN IN 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # EXIT NONE	565 # THE CLUCK INTERVET FRUCESSUK MEN LI 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # EXIT NONE	565 # THE CLUCK INTERVETY FRUCESSUK MEN LI 565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # EXIT NONE	565 * THE CLUCK INTERVETY THEN LISES OF REFORM THEN LISES OF REFORM THEN LISES OF REFORM THEN LISES OF REFORM 568 * EXIT NONE	564 * UPD 13 CALLED BT THE CLUCK INTEXEUR FRUCESSUK MEN 17 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	564 * UPD 13 CALLED BT THE CLUCK INTEXCUT FRUCESSUK MIEN 17 565 * ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	564 * UPD 13 CALLED BT THE CLUCK INTEXCUT FRUCESSUK MIEN 19 565 * ITHE TO PROATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADDRESS OF REFCNT 569 * EXIT NONE	564 * UPD 13 CALLED BT THE CLUCK INTEXCUT FRUCESSUK MIEN 19 565 * ITHE TO PROATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADDRESS OF REFCNT 569 * EXIT NONE	504 * UPD IS CALLED BY THE CLOCK INTEXCUT FRUCESSOR WIEN IN 566 * TIME TO UDATE THE DISPLAY CONTENTS. CURRENLY, THIS I 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	DOWN TO THE TO UPDATE THE CLOCK INTEXAULT VALUES OF WHEN IN THE SECOND.  SEC. # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  SEC. # ENTRY (H,L) = ADORESS OF REFCHT  SECOND TO USES ALL	204 * UPD 13 CALLED BT THE CLUCK INTEXKUT! FRUCESSUK WHEN 17 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLLY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL	564 * UPD 13 CALLED BT THE CLUCK INTEXKUT! FKUCESSUK WHEN LI 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	Des Tourney of the CLOUK INTERVET FORCESSUR WHEN IN THE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  568 # ENTRY (H,L) = ADDRESS OF REFCAT  569 # EXIT NONE  570 # USES ALL  572
562 <b>*</b> 563 <b>*</b>	562 # 563 *	562 *	TIMEN ACCUMENTATION OF THE COURT POUR PROPERTY OF THE PROPERTY				565 # TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 648 # ENTRY (H,L) = ADDRESS OF REFCNT	565 # TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS 1566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 # ENTRY (H,L) = ADDRESS OF REFCNT 569 # ENTRY	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 366 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCHT 548 # FXII NONE	565 # TIME TO UNDERTE THE DISPLAY CONTENTS. CURRELLLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCHT 569 # EXIT NONE	565 # TIME TO UNDEATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCHT 569 # EXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # EXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # EXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 569 # EXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFORT 568 # EXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFONT 568 # EXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFONT 568 # EXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 569 # EXIT NONE	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCHT 569 # EXIT NONE 5770 * USES ALL	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCAT 569 # EXIT NONE 570 # USES ALL	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCHT 569 # EXIT NONE 570 # USES ALL	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADORESS OF REFCHT 569 # EXIT NONE 570 # USES ALL 572
562 # 563 #	562 <del>+</del> 563 <del>+</del>	562 # 563 #	* UED IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I				565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 668 # ENTRY (H,L) = ADDRESS OF REFCNT 669 # ENTRY (H,L) = ADDRESS OF REFCNT	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * FXIT NONE	565 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L.) - ADDRESS OF REFONT 568 + ENTRY (H.L.) - ADDRESS OF REFONT	565 + TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H-LL) - ADDRESS OF REFONT 568 + EXIT NONE	565 + TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H-LL) - ADDRESS OF REFONT 568 + EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCHT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) * ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H.L.) = ADDRESS OF REFCNT 569 # EXIT NONE 570 # USES ALL 571	565 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H.L) = ADDRESS OF REFCNT 569 # EXIT NONE 570 * USES ALL 572
562 <del>*</del> 563 *	562 #	562 # 563 #	* OFD IS CALLED BY THE CLUCK INTERKUPT PRUCESSUK WHEN IT I		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT 668 * CYLL MONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL 571	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * EXIT NONE 570 * USES ALL	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) - ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL 571	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 572
563 # 11FD TE CALLED BY THE GLOCK THTEODIST BOOKESSOD WIEN IT T	562 # 563 # HEN TE CALLED BY THE CLOPY THECODING BOOMERSON HER T	501 ++ OTO - OTDAIL FRONT FARE UISTEATS. 563 + 563 + 1107 TK CALLED BY THE GLOCK THE BROWERSON HER T	TO 13 CALLED BI THE CLUCK INICARULI FRUCESSUR MICH II	1 11 NORTH 00330000 101000 1N1 A0010 071 A0 031 1V3 31 0311 +	1 11 NONE 00000000 1011000111 ACC 07 1110 01 0011 +	1 A1 MONT 00000000 101000011 A00 0 07 1 A0 07 1 WO 01 WO	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT 568 * CYTE NOWE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADORESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADORESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL 571	565 # TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H+L) = ADDRESS OF REFCMT 569 # EXIT NONE 570 # USES ALL	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571	565 * TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL 571
562 # 563 # HED IS CALLED BY THE CLOCK INTERBURT PROCESSOR WHEN IT I	562 # 563 # - HED IS CALLED BY THE CLOCK INTERBURT PROCESSOR WHEN IT I	301 ++ OFD - OFDRIE FRONT FANCE DISTRATS. 563 + 160 IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I		* UED IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	* HED IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	* UED IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT I	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFONT	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFONT 569 * FXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADORESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCMT 559 * EXIT NONE 570 * USES ALL	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCMT 569 * EXIT NONE 570 * USES ALL 571	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571	565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE 570 * USES ALL 571 * USES
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	562 * 562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	TOTAL ANTHUMEN CANDANCE OF COURT OF CANDANCE OF CASE	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	200	205	205	205	205	200 + LAME TO UPDATE THE DISPLANT CONTENTS. CORRENTLY, 1713 1 266 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCHT 569 + EXIT NONE	200 + LAME TO UTUATE THE DISTRACT CONTENTS. CURRENTLY, 1713 1 266 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 569 + EXIT NONE	200 + IAME TO UPDATE THE UISTRAT CONTENTS. CURRENTLY, 1913 1 566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFCNT 569 + EXIT NONE	505	565 + EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 568 + ENTRY (H,L) = ADDRESS OF REFCNT 569 + EXIT NONE	565 + EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 568 + ENTRY (H,L) = ADDRESS OF REFCNT 569 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 + ENTRY (H,L) - ADDRESS OF REFONT 569 + EXIT NONE	565	505	505	205
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	562 * UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	H TIME TO HEDATH THE SIMPLAY CONTENTS. CHERENTLY. THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 568 # EVIT MONE	566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADORESS OF REFCNT 568 # EXIT NONE	566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 569 # EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * EXIT NONE	566	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCHT 568 + EXIT NONE 570 + USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H-L) - ADDRESS OF REFCNT 569 + EXIT NONE 570 + USES ALL 571
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	562 + 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* TIME TO UPDATE THE DISPLAY CONTENTS. CORRENILT, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H-L) - ADDRESS OF REFONT 640 + CYLT MONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L.) - ADDRESS OF REFCNT 568 + EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + 568 + ENTRY (H,L) - ADDRESS OF REFONT 568 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + 568 + ENTRY (H,L) = ADORESS OF REFONT 569 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + 568 + ENTRY (H,L) = ADORESS OF REFONT 569 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADORESS OF REFCNT 568 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L) = ADDRESS OF REFCNT 568 + EXIT NONE 570 + USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + 568 + ENTRY (H,L) - ADDRESS OF REFCAT 569 + EXIT NONE 570 + USES ALL	566	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADORESS OF REFCNT 568 + EXIT NONE 570 + USES ALL 572
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	562 * 563 * UFD IS CALLED BY THE CLOCK INTERNIPT PROCESSOR WHEN IT	564 * UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN	+ ITHE ID OFDATE THE DISPLAT CONTENTS. CORRENILTS THIS I	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L.) = ADDRESS OF REFCNT 648 + CYLT MONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT 568 * FXII NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFONT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 568 + EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 568 * EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + 568 + ENTRY (H,L) - ADDRESS OF REFONT 569 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * 568 + ENTRY (H,L) = ADORESS OF REFCNT 569 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * 568 + ENTRY (H,L) = ADORESS OF REFCNT 569 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFCNT 569 + EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 569 + EXIT NONE 570 + USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 569 + EXIT NONE 570 + USES ALL	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L.) = ADDRESS OF REFCNT 568 + EXIT NONE 570 + USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L) = ADDRESS OF REFCNT 568 + EXIT NONE 570 + USES ALL 572
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN	562 + 563 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN	562 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	THE TO STATE THE STREAM SOUNDER SOUNDS	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 640 * CYTT MOME	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * FXIT NONE	566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCAT 569 * FXIT NONE	566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT 569 * EXIT NONE	566	566	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566	566	566	566 + EVERY 32 INTERRUPTS, OR ABOJT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 569 + EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H+L) = ADDRESS OF REFCAT 569 + EXIT NONE 570 + USES ALL 571	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L) = ADDRESS OF REFCNT 569 + EXIT NONE 570 + USES ALL	566
563 * S64 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT	563 * SEA TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 # 562 # 563 # 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT		* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY. THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY. THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY. THIS I	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCNT 568 * CYLT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-L) = ADDRESS OF REFONT 568 * FXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-L) - ADDRESS OF REFONT 568 * FXIT NONE	566	566	566	566	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 568 * EXIT NONE	566	566	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	566	566	566	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL 571 - 571
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 * 562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I		# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	567	567 # EVERT 32 INTERRUFISS UK ABL 568 # ENTRY (H)-L) = ADDRESS OF 569 # FXIT NONE	567 # EVERT 32 INTERRUFISS UK ABL 568 # ENTRY (H)-L) = ADDRESS OF 569 # EXIT NONE	567	567	567	567 # ENTRY (H <sub>2</sub> L) = ADDRESS OF 568 # EXIT NONE	567 # EVERT 32 INTERRUFISS UK ABL 568 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567	567 # EVERT 32 INTERRUFIS, UK ABL 568 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 # EVERT 32 INTERRUFIS, UK ABL 568 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 # EVERT 32 INTERRUFIS, UK ABL 568 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE 570 * USES ALL 571	567	567	567 # EVERT 32 INTERRUFIS, UK ABL 568 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE 570 # USES ALL 571
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERNPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I		* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	568 * ENTRY (H <sub>2</sub> L) = ADDRESS OF	567 # ENTRY (H <sub>2</sub> L) = ADDRESS OF 568 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 568 # EXIT NONE	567 * ENTRY (H,L) = ADDRESS OF EXIT NONE	567 * ENTRY (H,L) = ADDRESS OF EXIT NONE	567 * ENTRY (H,L) = ADDRESS OF 569 * EXIT NONE	567 * ENTRY (H,L) = ADDRESS OF 569 * EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF EXIT NONE	567 * ENTRY (H,L) = ADDRESS OF 569 * EXIT NONE	567 * ENTRY (H,L) = ADDRESS OF 569 * EXIT NONE	567 * ENTRY (H.L) = ADDRESS OF EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE 570 * USES ALL	567 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE 570 # USES ALL 571	567 * ENTRY (H,L) = ADDRESS OF EXIT NONE 570 * USES ALL 571	567 # ENTRY (H,L) = ADDRESS OF 568 # EXIT NONE 570 # USES ALL 571 # 571
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 + 563 + 564 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 569 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I		* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	* UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	567 # ENTRY (H <sub>2</sub> L) = ADDRESS OF	567 # 568 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NOWE	567 # 568 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NOWE	567 # ENTRY (H,L) = ADDRESS OF 568 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 568 # EXIT NONE	567 # ENTRY (H,L) = ADDRESS OF 568 # EXIT NONE	567 # ENTRY (H <sub>2</sub> L) = ADDRESS OF 568 # EXIT NONE	567 * ENTRY (H <sub>2</sub> L) = ADDRESS OF 568 * EXIT NONE	567 # 568 # ENTRY (H <sub>2</sub> L) = ADDRESS OF 569 # EXIT NONE	567 # 568 # ENTRY (H <sub>2</sub> L) = ADDRESS OF 569 # EXIT NONE	567 # 568 # ENTRY (H <sub>2</sub> L) = ADDRESS OF 569 # EXIT NONE	567 # 568 # ENTRY (H,L) = ADDRESS OF 569 # EXIT NONE	567 * ENTRY (H,L) - ADDRESS OF 568 * EXIT NONE 570 * USES ALL 571	567 * ENTRY (H,L) = ADDRESS OF 568 * EXIT NONE 570 * USES ALL 571	567 # ENTRY (H,L) = ADDRESS OF 568 # EXIT NONE 559 # EXIT NONE 570 # USES ALL	567 * ENTRY (H,L) = ADDRESS OF 568 * EXIT NONE 570 * USES ALL 571 * 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * THE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I	THE TAKEN 33 THICDRIDIC OD TOURS A CECOMO	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN # IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT INE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN # IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I	567 # ENTRY (H.L.) = ADDRESS OF	567 * 568 * ENTRY (H,L) = ADDRESS OF 569 *	567 * ENTRY (H,L) = ADDRESS OF 569 * EXIT NONE	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS OF 569 * EXIT NONE	567 * ENTRY (H <sub>2</sub> L) = ADDRESS OF 568 * EXIT NONE	567 * ENTRY (H <sub>2</sub> L) = ADDRESS OF 568 * EXIT NONE	567 * 568 * ENTRY (H <sub>3</sub> L) = ADDRESS OF 569 * EXIT NONE	567 * 568 * ENTRY (H <sub>3</sub> L) = ADDRESS OF 569 * EXIT NONE	567	567	567	567 * ENTRY (H,L) = ADDRESS OF 569 * EXIT NONE	567 * ENTRY (H,L) = ADDRESS OF 559 * EXIT NONE 570 * USES ALL 571	567	567 * 568 * ENTRY (H,L) = ADDRESS OF 569 * EXIT NONE 570 * USES ALL 571	567 * ENTRY (H,L) = ADDRESS OF 568 * EXIT NONE 570 * USES ALL 571.
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDISFIED DISPLAY CONTENTS. CURRENLLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLY, THIS I 565 * EVERY 32 INTERRUPTS. OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	+ EVERY 32 INTERRUPTS. OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS. OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS. OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS. OR ABOUT 32 TIMES A SECOND.	568 * ENTRY (H.L.) = ADDRESS 568 * EXIT MOME	567 * ENTRY (H.L) = ADDRESS 568 * EXIT NONE	566 * ENTRY (H.L) = ADDRESS 569 * FXIT NONE	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	567 * 568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	567 * 568 * ENTRY (H.) = ADDRESS 569 * EXIT NONE	567	567	567 * 568 * ENTRY (H <sub>3</sub> L) = ADDRESS 569 * EXIT NONE	567 * 568 * ENTRY (H <sub>3</sub> L) = ADDRESS 569 * EXIT NONE	567	567 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL	567 * ENTRY (H <sub>7</sub> L) = ADDRESS 568 * EXIT NONE 570 * USES ALL 571	567 * ENTRY (H <sub>3</sub> L) = ADDRESS 568 * EXIT NONE 570 * USES ALL 571	567 * ENTRY (H <sub>1</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571 - 572
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	<ul> <li>EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.</li> </ul>	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	568 # ENTRY (H-L) = ADDRESS 640 # EVIT NOME	568 + ENTRY (H,L) = ADDRESS 568 + EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * FXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H,L) - ADDRESS 569 * EXIT NONE	568 * ENTRY (H.L.) - ADDRESS 569 * EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H1L) = ADDRESS 569 * EXIT NONE	568 # ENTRY (H <sub>3</sub> L) = ADDRESS 569 # EXIT NONE	568	568	568 * ENTRY (H.L.) - ADDRESS 569 * EXIT NONE	568 * ENTRY (H+L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H+L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	568 * ENTRY (H <sub>2</sub> L) = ADDRESS	568 * ENTRY (H <sub>9</sub> L) = ADDRESS 569 * EXIT NOME	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NOME	568 + ENTRY (H+L) = ADDRESS 569 + EXIT NONE	568 + ENTRY (H,L) = ADDRESS 569 + EXIT NONE	568 + ENTRY (H+L) = ADDRESS 569 + EXIT NONE	568 + ENTRY (H,L) - ADDRESS 569 + EXIT NONE	568 + ENTRY (H,L) - ADDRESS 569 + EXIT NONE	568 + ENTRY (H,L) = ADDRESS 569 + EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 # ENTRY (H <sub>2</sub> L) = ADDRESS 569 # EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568	568 + ENTRY (H <sub>3</sub> L) = ADDRESS 569 + EXIT NONE 570 + USES ALL 571	568 + ENTRY (H <sub>2</sub> L) = ADDRESS 569 + EXIT NONE 570 + USES ALL 571 - 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, IHIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND.	# EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	<ul> <li>UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT</li> <li>TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I</li> <li>EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.</li> </ul>	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	<ul> <li>UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT</li> <li>TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I</li> <li>EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.</li> </ul>	568 # ENTRY (H <sub>2</sub> L) = ADDRESS	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H-L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 # ENTRY (H,L) = ADDRESS 569 # EXIT NONE	568 # ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 + ENTRY (H,L) = ADDRESS 569 + EXIT NONE	568	568	568 * ENTRY (H <sub>3</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H <sub>2</sub> L) = ADORESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H+L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571 571
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	# EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. *	568 * ENTRY (H <sub>2</sub> L) = ADDRESS	568 # ENTRY (H.L) = ADDRESS	568 # ENTRY (H.L) = ADDRESS 569 # EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568	568	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 • USES ALL	568	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H+L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 56 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	568 * ENTRY (H <sub>9</sub> L) = ADDRESS	568 * ENTRY (H <sub>9</sub> L) = ADDRESS 569 * EXIT NONE	568 + ENTRY (H2L) = ADDRESS 569 + EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENIRY (H,L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE	568 * ENTRY (H <sub>1</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571	568 * ENTRY (H <sub>2</sub> L) = ADDRESS 569 * EXIT NONE 570 * USES ALL 571 572
562 * 563 * 0FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 + 563 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 + TIME TO UPDISPLAY CONTENTS. CURRENLLY, THIS I 565 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 +	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	+ EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II # IIHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. *	560 + CNIKI (H)LJ = AUUKESS	569 # FXIT NONE	569 # FXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 # EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	200 + ENIKI (11)L) = AUUKE33 559 + EXIT NONE	569 # EXIT NONE **********************************	569 # EXIT (1912) = AUDRESS 569 # EXIT NONE 570 # USES ALL 571	569 # EXIT NONE FOURESS 571 # USES ALL	569 # EXIT NONE ADDRESS 570 # USES ALL 572
562 * 563 * 0FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE DISPLAY CONTENTS. CURRENLY, IHIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	+ EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. +	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * IIHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. *	SAGO # CYTT NOWE	569 # FXII NONE	569 # FXII NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 # EXIT NONE	569 # EXIT NONE 570 # USES ALL 571			
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	+ EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. *	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	SAG # CYIT NONE	569 # EXIT NONE	569 # FXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 # EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE	569 * EXIT NONE 570 * USES ALL 571	569 * EXIT NONE 570 * USES ALL 571	569 + EXIT NONE 570 • USES ALL 571	569 # EXIT NONE 570 * USES ALL 571
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 *	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, IHIS I 566 * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. 567 *	* EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. *	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. *	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. *	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT # IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.	540 # CYIT	569 # FXIT	569 * FXIT	569 * EXII	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT 570 * USES 571	569 <b>*</b> EXIT 570 <b>*</b> USES 571	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY	562 * 563 * UPD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE USPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * 568 *	* EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. *	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. # ELLED	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. #	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. # ELETAV	540 # CY17	549 * FXIT	569 * FXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-L) * ADDRESS OF REFCHT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-L) * ADDRESS OF REFCHT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-L) - ADDRESS OF REFCAT	<ul> <li>EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.</li> <li>ENTRY (H-L) = ADDRESS OF REFORT</li> </ul>	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. # ENTRY (H-L) = ADDRESS OF REFCHT	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IT INE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. # ENTRY (H-L) = ADDRESS OF REFORT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II  * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I  * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND.  * ENTRY (H-L) = ADDRESS OF REFCHT	540 * EVIT	11X3 * 095	249 * FXII	269 * EXIT	269 <b>*</b> EXIT	269 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571	569 * EXIT 570 * USES 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFORT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFORT	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT	<ul> <li>EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.</li> <li>ENTRY (H,L) = ADDRESS OF REFONT</li> </ul>	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II I I I I I I I I I I I I I I I I I	# UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. # ENTRY (H,L) = ADDRESS OF REFORT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I * EVERY 32 INTERRUPIS, OR ABOUT 32 TIMES A SECOND. * EVERY 32 INTERRUPIS, OR ABOUT 34 TIMES A SECOND. * ENTRY (H,L) * ADDRESS OF REFORT			* *	269 * EXII	269 * EXII	269 * EXII	269 * EXII	269 * EXII	269 * EXII	269 * EXII	269 * EXII	2007 * EXII	570	570 * USES 571	570 • USES 571	570 • USES 571 • USES 572
562 * 563 * 10FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADORESS OF REFORT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * THE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADORESS OF REFORT	563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * EVERY (H,L) = ADDRESS OF REFORT	* EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  * ENTRY (H,L) = ADDRESS OF REFONT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II  TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I  EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  ENTRY (H,L) = ADDRESS OF REFORT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II  * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I  * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  * ENTRY (H,L) = ADDRESS OF REFORT	* UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II  TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I  EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  * ENTRY (H,L) = ADDRESS OF REFONT						107	1777		1147	1771	1771	1177	570 * USES 571	570 + USES 571	570 * USES 571 * USES 571	570 • USES 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADORESS OF REFORT	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFORT	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 564 * ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADORESS OF REFORT	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + 568 + ENTRY (H,L) - ADORESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT		֡										- CFU	570 * USES 571	570 * USES 571	570 * USES 571	570 + USES 571 572
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT	562 * 563 * 0FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * 1THE TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFONT	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENILY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT													570 * USES 571	570 <b>*</b> USES 571	570 * USES 571	570 • USES 571 572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * 677 * 678 * 678 * 678 * 678 *	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H+L) = ADDRESS OF REFCHT	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFORT	566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-L) = ADDRESS OF REFONT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-L) - ADDRESS OF REFCNT	1777											1 000	570 * USES 571	570 * USES 571	570 * USES 571	570 * USES 571 572
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENLLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	562 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * ENTRY (H,L) - ADDRESS OF REFCAT	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFCNT 568 + EXIT NONE 559 + EXIT NONE	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFORT	564 * UFD IS CALLED BY THE CLUCK INTERRUPT PROCESSOR WHEN IN 565 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFORT	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFORT 1850 I SECOND. 1850 * ENTRY (H.L.) - ADDRESS OF REFORT 1850 I SECOND. 1850 I SECOND.	570 # UCEC	570 # UCEC	220	270 = 11050						777	777	0353	1571	1571	1571	1571 1572
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 568 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * EXIT NONE	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCAT 568 * ENTRY (H,L) = ADDRESS OF REFCAT 569 * USES A LIST	562 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 568 + EXIT NONE 550 + USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT 569 * ENTRY (H.L.) - ADDRESS OF REFCNT 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) - ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES A 1000 1000 1000 1000 1000 1000 1000 1	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCNT 569 * ENTRY (H,L) - ADDRESS OF REFCNT 570 * USES ALL	570 <b>*</b> USES	570 * USES	570 * USES	570 <b>*</b> 075	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	• 076	• 0/6						1571	1571	1571	1571 1572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * LINE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE 570 * USES ALL	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE 570 * USES ALL	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * 11ME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADDRESS OF REFONT 569 * USES ALL	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 569 + EXIT NONE 570 + USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCRI 569 * EXIT NONE 570 * USES	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H.L.) = ADDRESS OF REFCRI 569 * EXIT NONE 570 * USES	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCRI 569 * EXIT NONE 570 * USES ALL	570 * USES	570 * USES	570 * USES	570 * USES	5/0 • USES	3/0 • USES			, , ,	,,,,	,,,,		1571	1571	1,27,1	1572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL	562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL	563 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H+L) = ADDRESS OF REFCHT 569 # EXIT NONE 569 # EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFONT 569 + EXIT NONE 570 + USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H-L) - ADDRESS OF REFORT 569 * EXIT NONE. 570 * USES ALL	564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCNT 569 # EXIT NONE 570 * USES ALL	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFORT 569 * EXIT NONE.	570 <b>*</b> USES	570 * USES	570 * USES	570 * USES	5/0 • USES	370 • USES	270 + 075	210					13/1	1)61	1,572	1572
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE	562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFORT 569 * EXIT NONE 570 * USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * EVERY 32 INTERRUPTS, OR ABOUT 32 ITHES A SECOND. 567 * EVERY 32 INTERRUPTS, OR ABOUT 32 ITHES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H <sub>2</sub> L) = ADORESS OF REFCNT 569 + EXIT NONE 570 + USES ALL	50.5 TO UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 56.5 TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS6.5 TO EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 56.7 EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 56.8 ENTRY (H,L) = ADDRESS OF REFCHT 56.9 EXIT NONE 57.0 USES ALL	504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 569 * EXIT NONE	565 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 550 * USES ALL	570 * USES	570 * USES	570 * USES	570 * USES	570 * USES	570 • 0565	370 + 0353	0353	2222						6431	1572
562 * 563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * LIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADDRESS OF REFCNT 569 * EXIT NONE 570 * USES ALL	562 + 563 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 + TIME TO UNDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFORT 568 + EXIT NONE 570 + USES ALL	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * ITME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) = ADDRESS OF REFORT 569 * EXIT NONE	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCNT 568 + EXIT NONE 570 + USES ALL	5053 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 5654 TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 568 * EXIT NONE 570 * USES ALL	504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 5065 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 568 * EXIT NONE 570 * USES ALL	565 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL	570 <b>*</b> USES	570 * USES	570 * USES	570 * USES	5/0 <b>*</b> USES	570 <b>*</b> USES	671	631	531	231	231	1631			15.72	1572
562 ** 563 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 564 ** TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 ** ENTRY (H,L) ** ADDRESS OF REFONT 550 ** EXIT NONE 570 ** USES ALL 571	562 + 563 + 10FD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 + ENTRY (H,L) - ADDRESS OF REFORT 569 + EXIT NONE 570 + USES ALL 572	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 560 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) * ADDRESS OF REFORT 569 * EXIT NONE 570 * USES ALL 572	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFCNT 568 + EXIT NONE 570 + USES ALL 571	505.3 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 565.4 TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS66.4 EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567.4 ENTRY (H,L) = ADDRESS OF REFCNT 559.4 EXIT NONE 5770.4 USES ALL	20.3	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCNT 559 * EXIT NONE 5770 * USES ALL 572	570 * USES 571 572	570 * USES 571 572	570 * USES 571 572	570 * USES 571 572	570 <b>*</b> USES 571 572	570 • USES 571 572	571 572	571 572	571	571	571	571	7/61			
562 + 563 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFCNT 569 + EXIT NONE 570 - USES ALL 571	562 + 563 + UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 + TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) - ADDRESS OF REFCNT 568 + ENTRY NONE 570 + USES ALL 571	562 #	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H,L) = ADDRESS OF REFCAT 569 + EXIT NOME 570 + USES ALL 571	20.5. * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN III 56. * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 56. * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 56. * ENTRY (H,L) - ADDRESS OF REFCNT 56. * ENTRY ONE 57.0 * USES ALL 57.1	20.3	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) - ADDRESS OF REFCAT 550 * USES ALL 570 * USES ALL 572	570 * USES 571 572	570 * USES 571 572	570 * USES 571 572	570 * USES 571 572	570 • USES 571 572	571 571 572	571 572	571 572	571 572	571 572	571 572	571 572	7/5			
562 * 563 * 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) = ADDRESS OF REFCHT 569 * EXIT NONE 570 * USES ALL 571	562 # 563 # 564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 # TIME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 565 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 # ENTRY (H,L) = ADDRESS OF REFCMT 569 # EXIT NONE 570 # USES ALL 571	562 #	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H+L) - ADDRESS OF REFCAT 569 + EXIT NONE 570 + USES ALL 571	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCAT 5569 * EXIT NONE 5569 * EXIT NONE 5570 * USES ALL 5770	504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 504 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 569 * EXIT NONE ADDRESS OF REFCAT 559 * USES ALL 570 * USES	564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 * ENTRY (H,L) * ADDRESS OF REFCAT 5569 * EXIT NONE 5569 * EXIT NONE 5570 * USES ALL 5770	570 • USES 571 572	570 * USES 571 572	570 * USES 571 572	570 * USES 571 572	571 • USES 571 572 ·	571 571 572 573 usp cour	571 572 573	571	572	572	572	572 572	7/5	100	1102 0311 663	
562 * 563 * 100 IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) - ADDRESS OF REFONT 569 * EXIT NONE 570 * 571 - 573 UFD EQU *	562 ** 563 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN ITS 565 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 ** ENTRY (H,L) ** ADDRESS OF REFORT 569 ** EXIT NONE 570 ** USES ALL 571 573 UFD EQU **	563 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 564 * UFD IS CALLED BY THE DISPLAY CONTENTS. CURRENILY, THIS IS 565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 568 * ENTRY (H,L) - ADDRESS OF REFCAT 569 * EXIT NONE 570 * USES ALL 571 UFD 600 *	566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 567 + ENTRY (H.L) = ADORESS OF REFCNT 569 + EXIT NONE 570 + USES ALL 573 UFD EQU +	505.3	2063	565	570 * USES 571 572 573 UFD EQU	570 * USES 571 572 HD EQU	570 * USES 571 572 573 UFD EQU	570 * USES 571 572 UFD EQU	570 * USES 571 572 UFD EQU	571 571 572 UFD EQU	571 572 573 UFD EQU	571 572 573 UFD EQU	571 572 573 UFD EQU	571 572 573 UFD EQU	571 572 573 UFD EQU	571 572 573 UFD EQU	573 UFD EQU	573 UFD EQU	573 UFD EQU	2/3 UFD EQU
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFONT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 1573 UFD EQU *	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) ** ADDRESS OF REFONT 1569 ** EXIT NONE 1570 ** USES ALL 1571 ** USES ALL 1572 UFD EQU **	1562 *	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 * USES ALL 1572 UFO EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * IIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) = ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 1573 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADORESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 USES ALL 1573 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 UFD EQU **	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFO EQU	1570 * USES 1571 1572 1573 UFO EQU	1570 * USES 1571 1572 1573 UFO EQU	1570 * USES 1571 1572 1573 UFO EQU	1570 + 0353 1571 1572 UFD EQU	1571 1572 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	15/3 UFD EQU
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * IMPRINATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 UPD EQU *	1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571 USES  1573 UFD EQU *	1564 * UFD USALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IN 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H+L) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 UFD EQU *	1566	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1573 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCMT 1569 * ENTRY (H,L) - ADDRESS OF REFCMT 1570 * USES ALL 1571 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1569 * ENTRY (H.L.) - ADDRESS OF REFCAT 1570 * USES ALL 1571 UFD EQU *	1570 • USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 • USES 1571 1572 1573 UFD EQU	1570 • USES 1571 1572 1573 UFD EQU	1570 • USES 1571 1572 1573 UFD EQU	1570 • 03E3 1571 1572 1573 UFD EQU	1571 1571 1572 1573 UFD EQU	1571 1572 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	1573 UFD EQU
1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERNUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERNUPT PROCESSOR WHEN IN 1565 * IMPRINATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERNUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1573 UFD EQU * USES	1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 *  1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) = ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571  1573 UFD EQU *  1573 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H+L) = ADDRESS OF REFCMT 1568 * EXIT NONE 1570 * USES ALL 1571 UFD EQU * USES ALL 1571 UFD EQU * USES	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADORESS OF REFCNT 1569 * EXIT NONE   1570 * USES ALL 1571 * USES ALL 1573 UFD EQU * USES ALL	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1569 * 1570 * USES ALL 1571 UFD EQU * USES OF REFCNT 1573 UFD EQU * USES OF REFCNT 1573 UFD EQU * USES OF REFCNT	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L) - ADDRESS OF REFCAT 1569 * ENTRY (H.L) - ADDRESS OF REFCAT 1570 * USES ALL 1571 UFD EQU * USES ALL 1572 UFD EQU * USES USES	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENITY (H.L.) - ADDRESS OF REFCAT 1569 * ENITY (H.L.) - ADDRESS OF REFCAT 1570 * USES ALL 1571 UFD EQU * USES ALL 1572 UFD EQU * USES USES USES USES USES USES USES U	1570 • USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 • USES 1571 1572 1573 UFD EQU	1570 • 0353 1571 1572 1573 UFD EQU	1571 1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	1573 UFD EQU	1573 UFD EQU
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * LIME TO UPDATE THE DISPLAY CONTENTS, CURRENILY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 THES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCHT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 1573 UFD EQU * 1574	1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 *  1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 *  1567 * ENTRY (H*L) - ADDRESS OF REFCMT 1569 *  1570 * USES ALL 1570 *  1571 USES ALL 1571 *  1572 UFD EQU *  1573 UFD EQU *	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * ITME TO UPDATE THE DISPLAY CONTENTS, CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * EXIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 UFD EQU * USES ALL 1574	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 * OF EQU * * MYT A.UG.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) = ADDRESS OF REFCNT 1569 * EXIT NONE ADDRESS OF REFCNT 1570 * USES ALL 1571 * WAT A LUG DOU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1569 * EXIT NONE ADDRESS OF REFCAT 1570 * USES ALL 1571 UFD EQU * AUG. DOWN	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 UFD EQU * AUG. DOWN	1570 * USES 1571 1572 1573 UFO EQU	1570 * USES 1571 1572 1573 UFO EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * 03E3 1571 1572 1573 UFO EQU	1571 1571 1572 1573 UFD EQU	1571 1572 1572 1573 1574 HVI	1571 1572 1573 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU	1571 1572 1573 UFD EQU 1574 MVT	1573 UFD EQU 1573 UFD EQU 1574 MVT	1573 UFD EQU	1573 UFD EQU	1573 UFD EQU
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1573 UFD EQU * 1574 MVI A,UQ.DDU	1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFORT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1573 UFD EQU * 1574 WVI A,UQ.DDU	1562	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCAT 1568 * EXIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 HVI A,UQ.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 * HVI A,U0.00U	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCNT 1568 * ENTRY (H.L.) - ADDRESS OF REFCNT 1570 * USES ALL 1571 1571 FOUR * A.UO.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1568 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * HVI A.UO.DDU	1570 • USES 1571 1572 1573 UFD EQU 002 1574 HVI	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU	1570 * USES 1571 1572 1573 UFD EQU 002 1574 HVI	1570 • USES 1571 1572 1573 UFD EQU	1570 • 03E3 1571 1572 1573 UFD EQU	1571 1572 1573 1573 1574 1674 1874	1571 1572 1573 1573 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 MVI	1573 UFD EQU 1573 UFD EQU 002 1574 MVI	1573 UFD EQU	1573 UFD EQU	002 1574 HVI
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) * ADORESS OF REFORT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 UPD EQU * 1573 UPD EQU * 1574 WVI A,UG.DDU	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT ISOS **  1565 ** TIME TO UPDATE THE DISPLAY. CURRENTLY, THIS ISOS **  1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) ** ADORESS OF REFCAT  1569 ** EXIT NONE  1570 ** USES ALL  1571 **  1572 **  1573 UFD EQU **  1573 UFD EQU **  1574 WVI A,UG.DDU	1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IN 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1568 * ENIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * USES ALL 1572 * WYI **UG.DOU	1566 + EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 + ENTRY (H+L) - ADDRESS OF REFCAT 1569 + EXIT NONE 1570 + USES ALL 1571 1572 1573 UFD EQU + A,UG.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 UFD EQU * AUG.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCHT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 1573 UFD EQU * 0002 1574 WVI A,UG.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * EVIT NONE 1569 * EXIT NONE 1570 * USES ALL 1571 1572 * USES ALL 1573 UFD EQU * 0002 1574 WVI A,UG.DDU	1570 * USES 1571 1572 1573 UFD EQU 002 1574 HVI	1570 * USES 1571 1572 1573 UFD EQU 002 1574 HVI	1570 • USES 1571 1573 150 EQU 002 1574 HVI	1570 • USES 1571 1572 1573 UFD EQU 002 1574 HVI	1570	1570	1571 1572 1573 1573 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 HVI	1571 1572 1573 UFD EQU 002 1574 MVI	1571 1572 1573 UFD EQU 002 1574 MVI	1571 1572 1573 UFD EQU 002 1574 MVI	1572 1573 UFD EQU 002 1574 HVI	1573 UFD EQU 002 1574 HVI	1573 UFD EQU 002 1574 NVI	002 1574 UPD EQU
1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS *  1564 * TIME TO UPDATE THE DISPLAY. CURRENTLY, THIS IS *  1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) * ADDRESS OF REFONT  1569 * EXIT NONE  1570 * USES ALL  1571 *  1572 *  1574	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) ** ADDRESS OF REFCAT 1569 ** EXIT NONE 1570 ** USES ALL 1571 1571 1572 1574 MVI A,U0.DDU	1563 * UFD USALE BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 IIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 * USES ALL 1572 1574 #VI A,UO.DDU	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 1574 MVI A,UG.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 ** WYI A,UO.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 * HVI A,UO.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 * HVI A,UO.DDU	1570 • USES 1571 1572 1573 UFD EQU 002 1574 MVI	1570 * USES 1571 1573 1573 UFD EQU 1574 HVI	1570 * USES 1571 1573 1573 UFD EQU 1574 HVI	1570 * USES 1571 1572 UFD EQU 1573 UFD EQU 1574 HVI	1570	1570 • 05E5 1572 1572 1573 UFD EQU 1574 HVI	1570 - 0353 1571 1572 1573 0FD EQU 1573 0FD EQU	1571 1572 1573 002 1573 1574 HVI	1571 1572 1573 UFD EQU 1573 UFD EQU	1571 1572 1573 UFD EQU 1573 UFD EQU	1571 1572 1573 UFD EQU 1573 UFD EQU	1571 1572 1573 UFD EQU 002 1574 HVI	1573 UFD EQU 002 1574 MVI	002 1574 UFD EQU	1573 UFD EQU 002 1574 MVI	002 1574 HVI
1562 *  1563 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 *  1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 *  1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 * ENTRY (H,L) - ADDRESS OF REFCAT  1569 * EXIT NONE  1570 * USES ALL  1571  1572  1574  1574  1574  1574  1574  1575  AND B	1562 **  1563 **  1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1564 ** UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 ** ITHE TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 ** EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.  1567 ** ENTRY (H,L) - ADDRESS OF REFCAT 1569 ** EXIT NONE 1570 ** USES ALL 1571 1571 1572 AND EQU ** 1574 AND BAND BAND BAND BAND BAND BAND BAND B	1564 * UFD USALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IN 1565 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H+L) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * WAN B * UG.DDU	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 * WYI * WOODDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE TO USES ALL 1570 * USES ALL 1571 USES ALL 1573 UFD EQU * AND B WAY B AND BOUT STAND	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) - ADDRESS OF REFCAT 1569 * ENTRY (H.L.) - ADDRESS OF REFCAT 1570 * USES ALL 1571 1572 UFD EQU * 1573 UFD EQU * 1573 ANA B	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENITY (H.L.) - ADDRESS OF REFCAT 1569 * ENITY (H.L.) - ADDRESS OF REFCAT 1570 * USES ALL 1571 ** USES ALL 1572 UFD EQU ** 1573 UFD EQU ** 1573 UFD EQU ** 1574 MVI B	1570 * USES 1571 1572 1573 UFD EQU 002 1574 MVI	1570 * USES 1571 1572 1573 UFD EQU 002 1574 MVI	1570 * USES 1571 1572 1573 UFO EQU 002 1574 MVI	1570 * USES 1571 1572 1573 UFD EQU 002 1574 MVI	1570 * 03E3 1571 1572 1573 UFO EQU 002 1574 MVI	1570	1571 1571 1572 1573 002 1574 HVI	1571 1572 1573 1573 002 1574 18VI 1575 ANA	1571 1572 1573 UFD EQU 1574 HVI 1575 ANA	1571 1572 1573 UFD EQU 1574 HVI 1575 ANI	1571 1572 1573 UFD EQU 1574 HVI 1575 ANI	1571 1572 1573 UFD EQU 002 1574 HVI	1574 PWI 002 1574 WVI 1575 ANA	002 1573 UFD EQU 002 1574 HVI 1575 ANA	1573 UFD EQU 002 1574 HVI 1575 ANA	002 1574 HVI 1575 ANA
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCMT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 1573 UFD EQU * 1574 MVI A,UO.DDU	1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1573 UFD EQU * 1574 ANA B	1563	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) = ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1572 * AND B 1574 ANA B	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1564 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) * ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 * MAIN A,UO.DDU	1504 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H.L.) = ADDRESS OF REFCNT 1568 * ENTRY (H.L.) = ADDRESS OF REFCNT 1570 * USES ALL 1571 * USES ALL 1571 * MVI A.UG.DDU 002 1574 * MVI A.UG.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCNT 1569 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * HVI A,UO.DDU	1570 * USES ALL 1571 1572 1573 002 1574 MVI A,U0,DDU 1575 ANA B	1570 * USES ALL 1571 1572 1573 002 1574 ANI 8	1570 * USES ALL 1571 1572 1572 EQU * 002 1574 MVI A,UG.DDU 1575 ANA B	1570 * USES ALL 1571	1571	1571 - USES ALL 1571 - USES ALL 1573 UFD EQU + UG.DDU 002 1574 AVI A,UG.DDU	1572	1572	1571 1572 1573 UFD EQU * 1573 UFD RVI *, UG. DDU A 1574 ANA B	1571 1572 1573 UFD EQU * 1573 UFD RVI *,UG.DDU 1575 ANA B	1571 1572 1573 UFD EQU * 1573 UFD RVI *,UQ.DDU 1575 ANA B	1571 1572 1573 UFD EQU * 1574 HVI *,UG.DDU 1575 ANA B	1572 1573 UFD EQU * 1574 MVI A,UD.DDU ANA B	002 1573 UFD EQU + 1574 NVI A,UQ.DDU 1575 ANA B	002 1574 0FD EQU 4 002 1574 MVI A,UO.DDU 1575 ANA B	002 1574 AVI A,UO.DDU 1575 ANA B
1562 * 1563 * 1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) * ADDRESS OF REFORT 1569 * EXIT NONE 1570 * USES ALL 1571 1572 * 1573 UFD EQU * 1574 HVI A,UO.DDU	1562 #  1563 #  1564 # UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS64 # TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS65 # EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 # ENTRY (H,L) # ADORESS OF REFCNT IS69 # EXIT NONE 1570 # USES ALL 1571 # WI A,UO.DOU 1573 UFD EQU # A,UO.DOU	1562 *	1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCAT 1569 * EXIT NONE 1570 * USES ALL 1571 1571 1573 UFD EQU * * NU.DDU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1568 * ENTRY (H,L) - ADDRESS OF REFONT 1569 * ENTRY (H,L) - ADDRESS OF REFONT 1570 * USES ALL 1571 1572 1573 UFD EQU * UG.DDU 1574 ANA B	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H,L) - ADDRESS OF REFCHT 1568 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1572 * HVI A,UG.DOU	1564 * UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN II 1565 * TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS I 1566 * EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND. 1567 * ENTRY (H+L) - ADDRESS OF REFCAT 1568 * EXIT NONE 1570 * USES ALL 1571 * USES ALL 1571 * WILL 1572 * WAY A,UO.DOU	1570 * USES ALL 1571 1572 1574 EQU * 002 1574 AVI A,UO.DDU	1570 * USES ALL 1571 1573 1573 UFD EQU * 002 1574 HVI A,UO.DDU	1570 * USES ALL 1571 * USES ALL 1573 UFD EQU * UG.DDU 002 1574 HVI A,UG.DDU	1570 * USES ALL 1571 * USES ALL 1573 UFD EQU * 1574 WYI A,UG.DDU 1575 ANA B	1571 USES ALL 1571 EQU + 1573 UFD EQU + 002 1574 HVI A,UG.DDU	1571 USES ALL 1571 USE 1573 UFD EQU * 1574 WYI A,UG.DDU ANA B	1571 1572 1573 1573 002 1574 WVI *,UG.DDU 1575 ANA 8	1572 1572 1573 1573 002 1574 WVI A,UO.DDU	1571 1572 1573 UFD EQU * 1573 UFD #VI A,UG.DDU ANA B	1571 1572 1573 UFD EQU * 1573 UFD HVI A,UQ.DDU 1574 ANA B	1571 1572 1573 UFD EQU * 1573 UFD HVI A,UQ.DDU 1574 ANA B	1571 1572 1573 UFD EQU * 002 1574 HVI A,UO.DDU 1575 ANA B	1572 UFD EQU + 1573 UFD EQU + 1574 NYI A,UG.DDU 1575 ANA B	002 1573 UFD EQU + 00.00U 002 1574 MVI A,UO.00U 1575 ANA B	002 1573 UFD EQU * 002 1574 MVI A,UO.DDU 1575 ANA B	002 1574 NVI A,UG.DDU

003.245	032 312 T						
	312 17	1	1617	LDAX	۵		
003.246		22 003	1618 1619	75	900	IF MEMORY, DECODE BYTE VALUE	
:	:		1620 <b>+</b> 1621	15 REG	IS REGISTER. SET REGISTER NAME.	YEK NAME.	
003.251	341	11	1622	IAN	M,3770	CLEAR DIGIT	
003.254 042 022 040 1624 003.257 311	042 0. 311	22 040	1624 1625	SHLD	0LE05+1		
	:						

	•	RCK 1S	CALLED TO READ	A KEYSTROKE FROM THE CONSOLE KEYSET.
	• •	MHENEVE DCK PFP	EDDAS DERINACTA	AUTO-PEPEAT. A #810# 15
		MHEN A	WHEN A VALUE IS ACCEPTED.	
	* *	2 4 4		
	*			
	* *			
1640			,	
1491	* *		:	
	*	1111 01		
1644	*		•	
1645	* *		•	
1646	• •			
8491	*	10101	•	
1649	*		•	
1650	*			
169T	• •		:	
	* *	0100		
1634	*		:	
	*			
1656	*	ENTRY		
75.7.1. / C9.1	* *	EXII	TO CALLER WHEN	A KEY IS HIT
	• •		•	
1660	* 0		•	
1991	*		• :	
	* *		1 1 5	
1664	*		:•	
166	*		•	
1991	*		,	
1991	* *		•	
1669	*		•	
191	*		•	
167	<b>*</b> * *		•	
101	* *			
191	* +	USES		
191	5			
	o RCK	EOU	•	
345	:	PUSH	I	
305	6	PUSH	95	
9	0 -	I A	C,400/20	HAIT 400 MS
1		5	4 · · · · · · · · · · · · · · · · · · ·	
003.267 333 360 168	3 RCK1	Z	IP.PAD	INPUT PAD VALUE
		>	4 . 4	

003.3172 076 003.274 315 003.374 315 003.300 276 003.304 015 003.304 015 003.304 015 003.313 167	0. 92					A, 20/2	
		012			HVI		
	2	000 5 6			CALL MOV	0LY <b>A.</b> 8	MAIT 20 MS
	. 92		1688		CAP	T	: •
: : :	15	000	1690		DCR		DAYE A CHANGE
• :	~:	67 003	1691	•	٠.	RCK1	MAIT N CYCLES
	:		1694	•	NAVE A	KET VALUE	
	29		1695	RCK2	NO.	M. A.	UPDATE RCKA
	356 3 017	376	1696		XR I	3760	INVERT ALL BUT GROUP O FLAG
	22 3	326 003	:		SNC	RCK3	HIT BANK 0
	17		1699		RRC		
	210		1700		RRC		
	17		1021		RRC		
	117				RRC		
•	322 2	267 003			2	RCK1	NO HIT AT ALL
	107			RCK3	<b>A</b> 0 <b>K</b>	8, A	٠
	920	707	:		IAN	۸,4/2	
	115	140 002	:		CALL	HORN	TAKE BIP
	20				MOV	A.8	
•	941	017	1708		INA	170	
	301		1709		POP		
:	341		1710		POP	Ξ.	
	311		1711		RET		RETURN
	:						
	:						
	:						
	:						
	:						
	:						
	:						
:	:						

1715   1715
1776
1717
1779
1721 • 0 0 1721 • 0 1 1721 • 0 0 1722 • 0 1 1723 • • 1 1724 • 0 1 1725 • • 1 1725 • • 1 1729 • • 1 1729 • • 1 1729 • • 1 1729 • • 1 1729 • • 1 1729 • • 1 1729 • • 1 1729 • • 1 1729 • • 1 1730 05PA 05 0 1731 05PA 05 0 10011001001000 5P 1732 05PA 05 0 1733 05PA 05 0 1734 05PA 05 0 1735 05PA 05 0 1735 05PA 05 0 1736 05PA 05 0 1737 05PA 05 0 1738 • • 1 1739 05PA 05 0 1730 05PA 05PA 05PA 05PA 05PA 05PA 05PA 05P
1722
1722
1723
1724 **   7   1728 **   REGISTER INDEX TO 7-SECHENT PATTERN   1729   1730   1731   1732   1732   1732   1733   1733   1733   1734   1
1728 ** REGISTER INDEX TO 7-SECHENT PATTERN 1729 DSPA DS 0 244 230 1731 DB 0 220 234 1732 DB 1001100100100008 AF 200 215 1733 DB 100110100100100008 AF 200 215 1733 DB 1001101100100108 DE 200 215 1733 DB 100011011000110 DB 0 222 217 1735 DB 10001101100010 DB 0 222 217 1735 DB 10001101100010 DB 0 222 217 1735 DB 1000110110110 DB 0 222 217 1735 DB 100011010001 DB 0 222 217 1735 DB 00000001B DB 0 224 217 1735 DB 001010010 DB 0 225 217 1735 DB 001010010 DB 0 226 217 174 DB 001010010 DB 0 226 217 174 DB 001010010 DB 0 226 217 174 DB 001010010 DB 0 227 217 174 DB 001010010 DB 001001000 DB 0 228 217 174 DB 001010010 DB 001010000 DB 001010010 DB 001010100 DB 001010010 DB DB 00101001
1729 ** REGISTER INDEX TO 7-SEGRENT PATTERN 1729 1730 DSPA DS 0 10011001010000 SP
1730 DSPA DS 0 244 230 1731 DW 10011000100100B SP 262 234 1732 DW 100110101000100B AF 262 234 1733 DW 1001101101010100B BC 232 217 1735 DW 10001101100010B BC 233 316 1735 DW 10001101100010B BC 230 316 1735 DW 1000111110010B PC 230 316 1740 DGDA DS 0 241 1741 DB 00000001B 0 241 1742 DB 01001000B 2 240 1744 DB 01001000B 3 240 1744 DB 01001000B 3 240 1745 DB 01100010B 5 241 1745 DB 01001000B 3 241 1745 DB 01001000B 3 241 1745 DB 01001000B 3 241 1745 DB 00000100B 6 241 1745 DB 01100010B 7 241 1745 DB 00000100B 9 241 1745 DB 00000000B 9 241 1745 DB 00000000B 9 241 1755 THE TABLE MUST CONTINUE TO 3777A FOX PRUPER COPY. 2756 THE TABLE MUST CONTINUE TO 3777A FOX PRUPER COPY. 2757 THE TABLE MUST CONTINUE TO 3777A FOX PRUPER COPY. 2759 THE TABLE MUST CONTINUE TO 3777A FOX PRUPER COPY. 2750 DB 0000000A-7
244 230 1731 DN 10011000101008 SP 220 234 1732 DN 1001110010000108 AF 200 234 1733 DN 100111001010000108 AF 200 234 1733 DN 100011001010000108 BE 222 217 1735 DN 1000111110010108 PC 230 316 1735 DN 1100111100110018 PC 230 316 1735 DN 11001110110110011008 PC 230 316 1740 DDD DD 0 0 01100118 1 DB 01100118 1 DB 01100108 D 0 01100109 DB 001000008 D 0 011000008 D 0 011000108 D 0 011000109 D D 0 011000109 D D 0 011000109 D D 0 011000109 D D D 0 011000109 D D D D D D D D D D
220 234 1732 DW 1001110010010000 AF 206 215 1733 DW 1000110110000110B BC 232 217 1734 DW 100011010101000010B PC 230 316 1735 DW 1000111010011000B PC 230 316 1739 ** OCTAL TO 7-SETMENT PATTERN 1739 ** OCTAL TO 7-SETMENT PATTERN 1739 ** OCTAL TO 7-SETMENT PATTERN 1730 DW 110011B 1 DW 0110011B 1 DW 01110011B 1 DW 01110011B 1 DW 01110011B 1 DW 01110010B 2 DW 01110010B 2 DW 01110010B 3 DW 01110010B 3 DW 01110010B 3 DW 01110010B 5 DW 01110001B 7 DW 01110001B 7 DW 01110001B 7 DW 01111001B
206 215 1733 DW 100011010000106 BC 302 214 1734 DW 100011011001000108 DE 222 217 1735 DW 10001111101001000108 PC 232 217 1735 DW 110011101001100008 PC 230 316 1736 DW 110011101001100008 PC 230 316 1739 BW 000000018 DW 01100118 DW 01100118 DW 011001000 DW 01100118 DW 0110010000 DW 011001000 DW 011001000 DW 011001000 DW 011001000 DW 01100010 DW 01100000 DW
350 302 214 1734 DW 100011010001000 DE 354 230 316 1735 DW 100011111001000100 HC 355 222 If 1735 DW 1000111010010000 PC 356 1738 ** OCTAL TO 7-SETMENT PATTERN 1739 PB 01110011B 1 357 163 1742 DB 01110011B 1 351 1742 DB 01110011B 1 352 001 1743 DB 011100100B 2 353 044 1745 DB 00100100B 3 364 004 1745 DB 00100100B 3 365 000 1749 DB 00100100B 8 365 000 1749 DB 00100100B 8 365 000 1779 DB 00100100B 8 366 000 1779 DB 00100100B 9 367 040 1775 PB 08 00100000B 8 368 000 1779 DB 00100000B 8 368 000 1779 DB 00100000B 8 368 000 1779 DB 00100000B 8 369 000 1779 DB 00100000B 8 360 000 1779 DB 00100000B 8 360 000 1779 DB 00100000B 8 361 040 1775 PB 00100000B 8 362 040 1775 PB 00100000B 8 363 040 1775 PB 00100000B 8 364 001 1775 PB 00100000B 8 365 000 1775 PB 00100000B 8 367 040 1775 PB 00100000B 8 368 0010 1775 PB 00100000B 8 377 040 1775 PB 0010000B 8 377 040 1775 PB 0010000B 8
222 217 1735 DW 10001110100100 HL 230 316 1736 DW 110011010011000 PC 1738 ** OCTAL TO 7-SETMENT PATTERN 1739 T740 DDDA DS 0 0000000 B 1 1742 DB 01100010 B 2 1744 DB 01100000 B 2 1744 DB 01100010 B 2 1744 DB 01100000 B 3 01100000 B 5 1745 DB 000100100 B 5 00100100 B 0 00100000 B 0 001000000 B 0 00100000 B 0 001000000 B 0 00100000 B 0 00100000 B 0 00100000 B 0 00100000
1738   **
1736
1740   DODA   DS   O   O   O   O   O   O   O   O   O
001     1741     D8     000000018     0       163     1742     D8     01100118     1       110     1744     D8     011000008     2       140     1745     D8     0011001008     4       064     1745     D8     001001008     5       004     1746     D8     001001008     6       161     1748     D8     001000018     7       000     1750     D8     001000008     9       040     1750     D8     001000008     9       040     1752     **     10 ROUTINES TO BE COPIED INTO AND USED IN RAM.       1753     *     MUST CONTINUE TO 3777A FUX PRUPER COPY.       1754     *     THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL.       1755     *     THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL.       1756     +     0000-70
116 1742 DB 01110011B 1 116 1743 DB 01001000B 2 140 1744 DB 01100000B 3 044 1745 DB 00100100B 6 044 1746 DB 00100100B 7 100 1749 DB 0010000B 8 040 1750 DB 00100000B 9 040 1755 THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL
110 1743 DB 0100100B 2 140 1744 DB 01100100B 3 062 1745 DB 001100100B 5 044 1746 DB 00100100B 5 004 1749 DB 01110001B 7 161 1749 DB 0110001B 7 000 1750 DB 00100000B 9 040 1755 THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL 1755 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.62 1745 08 001100108 4 0.44 1745 08 00100108 5 0.04 1746 08 00100108 5 0.04 1748 08 00100018 7 0.00 1750 08 001000008 9 0.40 1752 ** IO ROUTINES TO BE COPIED INTO AND USED IN RAM 1755 ** THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL 1755 4 0.00 0.01000008 9
044 1746 DB 00100100B 5 004 1747 DB 00000100B 6 161 1748 DB 01110001B 7 000 1750 DB 00100000B 9 040 1752 ** IO ROUTINES TO BE COPIED INTO AND USED IN RAM 1754 WUST CONTINUE TO 37774 FOX PRUPER COPY. 1755 * THE TABLE MUST ALSO BE BACKMARDS TO THE FINAL 1756 0RG 4000A-7
0004 1747 DB 00000100B 6 16.1 1748 DB 01110001B 7 000 1749 DB 00100000B 8 0040 1750 DB 00100000B 9 1752 ** IO ROUTINES TO BE COPIED INTO AND USED IN RAM 1754 * MUST CONTINUE TO 3777A FOX PRUPER COPY. 1755 * THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL 1756 0RG 4000A-7
16.1 17.48 DB 01110001B 7 000 17.49 DB 00100000B 8 00.0 17.50 DB 00100000B 9 17.52 ** IO ROUTINES TO BE COPIED INTO AND USED IN RAH 17.53 * MUST CONTINUE TO 37.774 FOX PRUPER COPY. 17.55 * THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL 17.56 0RG 4000A-7
040 1750 08 001000008 9 040 1752 ** 10 ROUTINES TO BE COPIED INTO AND USED IN RAM 1753 * HUST CONTINUE TO 3777A FOR PRUPER COPY. 1755 * THE TABLE HUST ALSO BE BACKWARDS TO THE FINAL 1757 0RG 4000A-7
1752 ** IO ROUTINES TO BE COPIED INTO AND USED IN RAM. 1753 * MUST CONTINUE TO 3777A FOX PRUPER COPY. 1755 * THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL. 1756 ORG 4000A-7
1752 ** 10 ROUTINES TO BE COPIED INTO AND USED IN RAN. 1753 * HUST CONTINUE TO 3777A FOX PRUPER COPY. 1755 * THE TABLE MUST ALSO BE BACKMARDS TO THE FINAL 1756 ORG 4000A-7
1753 * MUST CONTINUE TO 37774 FOX PRUPER COPY. 1755 * THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL. 1756 ORG 4000A-7
1754 * MUST CONTINUE TO 37774 FOR PRUPER COPY. 1755 * THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL 1756 0RG 4000A-7
.371 1757 0KG 4000A-7
• 371 1757 0KG 4
1758
3.371 1759 PRSROM EQU *
001 1760
0 80 To/T 000 7/5.5
3.373 000 1762 08

			1772	*	XINIT	1 - SIZE MEMORY	<b>&gt;</b>	/RAMBGO JUNBO/
			1774		THIS IN TH 040000	ROUTINE DIFFER AT IT IS NON-DOA, AND IT HIL	XINITI IS JUMPED TO DURING PAMB'S MEMORY SIZING THIS ROUTINE DIFFERS FROM THE STANDARD PAMB FUNCTION IN THAT IT IS NON-DESTRUCTIVE TO WHAT MAY BE RAM BELOM 0400000A, AND IT WILL NOT WRAP-AROUND IN A 64K RAM SYSTEM	FUNCTION RAM BELOM IK RAM SYSTEM
	:		1778	• •	ENTRY	JUMPE	ROM OLD INITI	
	:		1780	• :•		(HL) = FIRS	T RAM SEARCH LOCATION	
	:		1782	* *	EXIT	(HL) = FIRS	COR ZERO IF RAM THROUGH 64K)	FOUND
	:		1784			(E) = 0 AS	REQUIRED	
:	176		1786	XINITI	MO V	I. 4	GET THE VALUE OF THE	E CURRENT TRIAL ADDRESS
	065 276		1788		S S	E E	COMPARE IT TO ITS OL	LD VALUE
:	:		1789		MOV	<b>4</b> • • • • • • • • • • • • • • • • • • •	RESTORE OLD VALUE	3 4 6
:	312 11 031	000	1621		040	0	INCREMENT THE SEARCH	H ADDRESS
	322 000 303 117	000	1792		CAC	XINITI INITZ	HAS NOT WRAPPED AROUND BACK INTO INLINE CODE	UND
	:							
	:			•				
	:							
	:							

	:								
	:		1797	: .	XINIT -	- EXTENDED INITI	IALIZATION		/Ram8Go 2/
	:		7.0		DEC IDE	OK.	CAPY RAN FRONT PANEL AND HIT RON TO	. AND HIT ROH T	0
			1801	* * 1	JUMP BACK	PROPRIATE TO INLINE	POSITIONS		
			1804		Modified	to only do	one move directly to	to RAM.	/Ram8Go 2/
	:		800		ENTRY	(DE) - RANBGO			/Ram8Go 2/
			1808	• • •	EXIT				
			1810	• •		RAM AT ZERO S	PRSKAM+PRSL-1 ZERO SET UP IF PRESENT		
			1811	•					
030.000			1813	H17ROM	200	03 00 00 A	HIT ROM Address	S.	
			1815			+701+7	) Tu 10	5	
004.016	257 062 0	040 990	1816 1817	XINIX	XRA STA	A CTLFLG2	Initialize the	f ag	
			1818	•		check routine to	¥		
: 2	: 1		1820	:		771			
004-024	7.5	146 004	:		Z	D, XINA			
20.0	4:2	4:	:	I N I X	LXI	H, XINB			
6	167		1825	7	¥0.¥	¥ .			
0.0	023		1826		XXI	<b>0 I</b>			
004.036	015	03.2	1828		DCR	, i			
	<u>;</u>		1830	•	Ξ,				
			:		¥:	101 KAN AL 1810			
004.042		000 000	:		LXI	H,RAMBGO			
• 050					NO.	B, A	original	in B	
004.051		: 8			ORI	OP.CTL2	n on RAM	at Zero	
4.056					J. B	XINB	Serecura gades	FOSS	
4.061		135 004		XIN2	75	SNIX	No change with	decrement	
			1840	*	Copy R	ROM to RAM			
: ,									
490 4 00	001	000 000	•		K	B.RAMBGOL D.RAMBGO	Length to Copy		
4.072	032	:	:	XIN3	LDAX	Q	Move RAMBGO into plac	o place	
4.073	022		1840	•	STAX	<b>a</b> a			
004.075			184		DCX				
4.076			184	-	M0V	A,8			
04-100	302	072 004			2N2	XIN3	Not all moved y	yet	
		:							

004-103 009 009 009 1055 LXI	001 000 010 1855 LXI 8,HI7ROHL 004 000 030 1856 LXI H+HI7ROH 004 1856 LINA 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	001 000 010 1855 041 000 030 1856 032 1857 1858 043 1858 043 1860 013 1861 170 1862 261 11004 1864 072 066 040 1866 36 040 1866 36 040 1868 052 066 040 1868 052 060 040 1868 052 060 040 1868 052 060 040 1868 052 060 040 1868 052 060 040 1868	£ 2 ±		3.5 2.5	Not all moved yet  Turn on Ram at 0  RESTORE NORMAL VALUES I RETURN TO INLINE CODE Select RAM  Check for a change
041 000 030 1856 LXI HHITROH 042 1857 XIN4 LQAX D 1657 1859 HOV D 167 1862 DO	041 000 030 1856 LXI HHITROH 042 1857 XIN4 LDAX D 043 1859 HOV N-A 107 1862 HOV N-B 108 1862 HOV A-B 109 1864 HOV HOT A-B 109 1865 HOV HOT A-B 109 1870 HOV HOT A-B 109 1870 HOV HOT A-B 109 1870 HOV HOW HOV HOV HOV HOV A-B 109 1870 HOV	041 000 030 1856 032 1859 043 1869 043 1869 043 1861 170 1862 261 111 004 1863 302 111 004 1865 072 066 040 1866 360 060 040 1869 062 066 040 1869 062 066 040 1869 062 066 040 1869 062 066 040 1867 062 066 040 1867 062 066 040 1867	£ 92 ; <u>₹</u>		L L L L L L L L L L L L L L L L L L L	Note HIT ROM Into place  Not all moved yet  Turn on Ram at 0  RESTORE NORMAL VALUES  RETURN TO INLINE CODE  Select RAM  Check for a change
167 1859 NOV N'A  043 1860 INX H  1043 1861 OCX B  1043 1864 INX H  1862 ORA CAB  261 1864 JNZ XIN4 NOT AII  302 111 004 1864 JNZ XIN4 NOT AII  042 313 362 1870 OCX H  1870 OCX CAB  051 311 003 1871 XIN5 LXI DAPESTORE  041 012 040 1872 AINA OVY AAB  176 1873 OCX B880 OCX H  176 1873 OCX B880 OCX H  176 1873 OCX B880 OCX H  177 1871 OCX B880 OCX H  1871 OCX B880 OCX H  1872 OCX H  1873 OCX B880 OCX H  1874 OCX CACC  1875 OCX H  1875 OCX H  1877 OCX H  1878 OCX H  1878 OCX H  1879 OCX H  1870 OCX H  1870 OCX H  1870 OCX H  1870 OCX H  1871 OCX H  1871 OCX H  1871 OCX H  1872 OCX H  1873 OCX B880 OCX H  1874 OCX H  1875 OCX H  1875 OCX H  1876 OCX H  1877 OCX H	1.67 1.659 NOV N+A 0.43 1.660 INX H 1.641 0.05 A.9 1.651 1.651 0.05 A.9 2.61 1.652 0.08 A.9 3.02 1.11 0.04 1.665 0.08 CTLEC2 0.02 0.06 0.40 1.665 0.08 CTLEC2 0.02 0.01 0.04 0.04 0.06 CTLEC2 0.02 0.01 0.04 0.04 0.06 CTLEC2 0.02 0.01 0.04 0.06 0.06 0.06 0.06 0.06 0.06 0.06	167 1858 023 1859 043 1860 013 1861 170 1862 261 1862 302 111 004 1864 323 365 040 1866 366 040 1866 36 040 1866 323 362 040 1869 021 371 003 1872 041 012 040 1873	<b>5</b> ₹	INX INX INX INX INX INZ INZ INZ INZ INZ INZ INZ INZ INZ INZ	15 24 -	Not all moved yet  Turn on Ram at 0  RESTORE NORMAL VALUES  AETURN TO INLINE CODE  Select RAM  Check for a change
023 1859 INX D 0013 1860 DOX H 013 1862 DOX H 1862 DRA C 1863 DRA C 302 111 004 1864 JRZ 002 0040 1866 DOX H 1865 DRA CTLELG2 005 006 040 1866 DOX H 1865 DRA CTLELG2 005 006 040 1869 DOX DRA 005 006 040 1869 DOX DRA 001 012 040 1872 LXI H 1873 DOX DOX DRA 005 01874 DAX DOX DRA 006 01874 DAX DOX DRA 007 DRA 008 DRA 009 DRA 00	043 1859 1NX 0 043 1860 00X 8 170 1862 00X 6 261 1863 00X 1863 072 066 040 1866 00X 1867 00X 1867 00X 1867 00X 1877 00X 1878 1877 00X 1878 1878 1878 00X 1878 1878 00X 1878 00	043 1859 043 1860 013 1860 013 1862 261 1863 302 111 004 1864 323 362 040 1865 366 040 1865 362 066 040 1865 323 362 1869 021 371 003 1872 041 012 040 1873	<u>\$</u>	DOUT DOUT DOUT DOUT DOUT COMP KCHG KCHG	•• •• •• •• •• •• •• •• •• •• •• •• ••	Not all moved yet  Turn on Ram at 0  RESTORE NORMAL VALUES  RETURN TO INLINE CODE  Select RAM  Check for a change
170 1862 170 170 1862 170 170 1863 1864 170 1865 170 1865 170 1865 170 1865 170 1865 170 1865 170 1865 170 1866 170 1866 170 1866 170 1866 170 1866 1866 1866 1866 1866 1866 1866 1870 1870 1870 1870 1870 1870 1870 1870	170 1862 170 171 1862 170 1864 1864 1874 7874 1864 1865 1865 1866 1866 1866 1866 1866 1866	043 1661 170 1862 261 1863 302 111 004 1864 1865 072 066 040 1865 365 040 1866 323 362 1869 021 371 003 1872 041 012 040 1873	\$ <del>\$</del>	DOCKA LNZ LNZ LNZ DOUT COCK COCK COCK COCK COCK COCK COCK COC	•• •• •• •• •• •• •• •• •• •• •• •• ••	Not all moved yet  Turn on Ram at 0  RESTORE NORMAL VALUES  AETURN TO INLINE CODE  Select RAM  Check for a change
170 1862 NOV A+8  261 1863 ORA C  261 1864 JNZ XIN4  1865 OAO 1866 OAO 1866  062 066 040 1869 OUT CIFECE  323 362 1871 XIN5 LXI OP-CTL2  1873 000 1874 JNP INIT  1873 362 1873 AHP INIT  1875 XINA OUT OP-CTL2  1876 CHP H  1879 CHP H  1879 CHP  1870 1881 CHP  1881 XINAL EDU +-XINA  1883 XINAL EDU +-XINA	170 1862 NOV C.5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	170 1862 261 1863 302 111 004 1864 072 066 040 1866 366 040 1866 365 066 040 1868 021 371 003 1871 041 012 040 1872 1873	\$ <del>2</del> \$ <del>2</del>	ORA JNZ JNZ JNZ SSTA OUT CKI CCHP KCHG KCHG	••• •••	Not all moved yet  Turn on Ram at 0  RESTORE NORMAL VALUES  AETURN TO INLINE CODE  Select RAM  Check for a change
302 111 004 1864 JNZ XIN4  1865 040 1866 040 1866  326 040 1866 040 1866  323 362 0 1869 001 0P CTLE  323 362 1876 001 0P CTL2  303 073 000 1874 1875 LXI PRSRAM+PASL-1  1876 XINA 001 0P CTL2  1877 CMP NOV A+N  1881 001 0P CTL2  353 1881 001 0P CTL2  353 362 1881 001 0P CTL2  353 1883 PCHL  1883 PCHL  1883 PCHL  1883 PCHL  1883 PCHL  1884 XINAL E0U +-XINA	302 111 004 1864 JNZ XIN4  072 066 040 1866 LDA CTLFLGZ  365 046 040 1866 LDA CTLFLGZ  062 066 040 1866 DUT CR2-CR2  062 066 040 1869 DUT CR2-CR2  041 012 040 1872 LXI DPRSRM  041 012 040 1873 JHP INIT  1873 362 1873 JHP INIT  1874 DUT CR2-CR2  1875 CHP MOV A+M  1881 XINAL EDU +-XINA  1884 XINAL EDU +-XINA	302 111 004 1664 302 111 004 1865 072 066 040 1866 36 040 1868 323 362 1869 021 371 003 1870 041 012 040 1872 303 073 000 1874	Σ <u>₹</u>	JAKA JAKA STA STA STA OUT LKI LKI CCHP CCHP CCHC CCHC CCHC CCHC CCHC CCH	75 × 87 × 80 × 80 × 80 × 80 × 80 × 80 × 80	Turn on Ram at 0  RESTORE NORMAL VALUES  RETURN TO INLINE CODE  Select RAM  Check for a change
1865 072 066 040 1866 1867 081 082 086 062 066 040 1866 082 087 1867 091 371 003 1871 041 012 040 1872 1873 1873 1873 1873 1881 1882 1881 1882 1884 1884 1884 1884	1865 CDA CTELCZ 365 040 1866 CDA CTELCZ 365 040 1869 STA CTELCZ 323 362 1869 STA CTELCZ 323 362 1870 CDA STA CTELCZ 323 362 1871 KIN5 LXI D.PRSRAN-PASE-1 1873 000 1874 JNP INIT 176 1879 HOV A.N 170 1881 DCR N 1881 CMP A.B 1883 SS 1882 FCHL 353 362 1883 FCHL 354 XINAL EDU P-XINA	1865 072 066 040 1866 366 040 1866 366 040 1868 323 362 1868 021 371 003 1871 041 012 040 1872 1873	S2	SORE SOLE SOLE SOLE SOLE SOLE SOLE SOLE SOL		Turn on Ram at 0  RESTORE NORMAL VALUES  I RETURN TO INLINE CODE  Select RAM  Check for a change
075 066 040 1866	075 066 040 1866	1123 072 066 040 1866 1126 366 040 1867 1130 062 066 040 1869 1133 323 362 1869 1135 021 371 003 1871 1140 041 012 040 1872 1143 303 073 000 1874	S ₹	LEDA STRI STRI DUT LEXI LEXI LAP CONP CONP XCHG	- P & S L	Turn on Ram at 0  RESTORE NORMAL VALUES  RETURN TO INLINE CODE  Select RAM  Check for a change
062 066 040 1868 51A CTLELEZ 323 362 1869 00T 0P-CTLE 1879 LXI DP-RSRAH-PASL-1 303 073 000 1874 LMP LXI H-PRSRAH-PASL-1 303 073 000 1875 LM 0UT 0P-CTLE 323 362 1879 CMP H 1881 L882 KCHG 353 1882 KCHG 351 1883 PCHL 1883 KCHG 351 1883 PCHL 1884 XINAL EDU P-XINA	062 066 040 1868 51A CTLELEZ 323 362 1869 00T 0P-CTLEZ 1871 003 1871 1872 LXI H-PRSRAH-PRSL-1 041 012 040 1873	133 062 066 040 1868 133 323 362 1869 135 021 371 003 1871 140 041 012 040 1872 143 303 073 000 1874	S	STA DUIT LKII JHP DUIT COMP HOV KCHG	75 S S S S S S S S S S S S S S S S S S S	Turn on Ram at 0  RESTORE NORMAL VALUES  AETURN TO INLINE CODE  Select RAM  Check for a change
323 362 1869 0UT OP.CTL2  1870 1871 1872 1875 LXI O.P. PRSROM 041 012 040 1874 LXI O.P. PRSRAH+PKSL-1 1873 303 073 000 1874 JAP INIT 1875 XINA OUT OP.CTL2 1876 1879 CHP N. 170 1880 HOV A.B 170 1881 CHP N. 323 362 1881 OUT OP.CTL2 353 1881 PCHL 1882 PCHL 1884 PCHL 1884 XINAL EQUXINA	323 362 1869 0UT OP.CTL2 1870 041 012 040 1872 1873 303 073 000 1874 1875 323 362 1876 1877 1880 170 1880 1880 1880 1880 1880	133 323 362 1869 1870 1871 003 1871 140 041 012 040 1872 1873 193 073 000 1874	S X	LKII LKII LMP OCR CGR OCR COR COR COR CCR CCR CCR CCR CCR CC	1. P & S.L.	Turn on Ram at 0  RESTORE NORMAL VALUES  RETURN TO INLINE CODE  Select RAM  Check for a change
1870 1870 1871 1971 1972 1973 1973 1973 1973 1974 1975 1875 1875 1876 1879 1879 1879 1879 1879 1879 1879 1879	1870 1870 1871 0041 012 040 1872 1873 303 073 000 1874 1875 323 362 1876 1878 007 006 1878 1878 007 007 007 007 007 007 007 007 007	1870 •135 021 371 003 1871 •140 041 012 040 1872 1873 •143 303 073 000 1874	S Z	LXI LXI JHP MOV CMP MOV XCHG	SRAM+PASL	RESTORE NORMAL VALUES  AETURN TO INLINE CODE  Select RAM  Check for a change
303 073 000 1872 LXI H,PRSRAH+PKSL-1 1875 JAP INIT 1875 JAP INIT 1870 DCR H 1881 DUT 1882 XCHG 1883 DCCTL2 1884 XINAL EdU +-XINA	041 012 040 1872	04.140 041 012 040 1872 1873 04.143 303 073 000 1874	<u> </u>	LXI DOUT DOCK DCR OUT XCHG	SRAM+PRSL	Select RAM Check for a change
303 073 000 1874 JAP INIT 323 362 1875 XINA 0UT 0P.CTL2 176 1877 NOV A.B 170 1887 CMP H H H H H H H H H H H H H H H H H H H	303 073 000 1874 JHP INIT 1875 XINA 0UT 0P-CTL2 1876 1877 HOV A-H 176 1879 CHP H 276 1879 CHP H 353 362 1881 OUT 0P-CTL2 353 362 1882 XCHG 351 1882 XCHG 351 1882 YCHG	04.143 303 073 000	<u> </u>	UHP OUT DCR CMP HOV OUT	INIT 0P.CTL2 A.M H H A.B	Select RAM Check for a change
303 073 000 1874 JMP INIT 323 362 1875 XINA 0UT 0P-CTL2 176 1879 CMP M 276 1879 CMP M 273 1882 CMP M 373 1882 CMP M 353 1882 XCHG 353 1883 XCHG 351 1884 XINAL EDU *-XINA	303 073 000 1874 JAP INIT 323 362 1875 XINA 0UT 0P.CTL2 176 1879 CAP H 276 1879 CAP H 373 362 1881 0UT 0P.CTL2 353 362 1882 XCHG 353 1883 PCHL 353 1883 PCHL 351 1883 PCHL 1884 XINAL EQU *-XINA	04.143 303 073 000	<u> </u>	OUT OUT OCR CMP MOV XCHG	INIT OP.CTL2 A.M H A.B	Select RAM Check for a change
323 362 1876 XINA DUT OP.CTL2 Select RAH 176 1878 HOV A,H 276 1879 CDCR H 170 1880 HOV A,B 223 362 1881 OUT OP.CTL2 Select ROH 353 1882 FCHG 351 1883 PCHG 351 1884 XINAL EDUXÍNA	323 362 1876 XINA OUT OP.CTL2 Select RAN 176 1879 NOV A.B CDCR TOT A 180 OUT OP.CTL2 Select RON 323 362 1881 OUT OP.CTL2 Select RON 323 362 1883 P.C.HG 351 1883 P.C.HG 351 1883 P.C.HG 60U PXINA		<b>\$</b>	OUT DCR CCR HOV OUT	0P.CTL2 A.M H A.B	for a
176 1877 NOV A.H. 1976 1878 NOV A.H. 1976 1879 NOV A.H. 1979 NOV A.H. 1983 362 1882 XCHG 1983 PCHL 1984 XINAL EDU *-XINA	176 1877 NOV A,H 065 1877 NOV A,H 176 1887 CMP N,H 177 1881 CMP N,H 1883 1883 PCHL Select ROH 1883 PCHL +-XINA	7681 676 666 77	<b>\$</b>	MOV CMP MOV CUT	8 . E . A	for a
276 1879 DCR H 276 1879 CMP N 373 362 1880 DNU A+B 353 1882 XCHG 353 1883 FCHL 351 1883 FCHL 351 1884 XINAL EQU #-XINA	276 1879 0CR H 276 1879 CHP N 373 1881 001 00-CTL2 Select R0H 351 1883 PCHL 351 1883 PCHL 351 1884 PCHL 351 1884 PCHL 351 1885 PCHL 4-XINA	126 382 1878		OCR CMP MOV OUT	₩ E ◀	for a
276 1879 CMP H Check for a 170 1860 HOV A,8 1 1862 OHC BOWN A,8 1 1863 OHC BOWN A,8 1 1863 OHC BOWN A,8 1 1864 XINAL EDU F-XINA	276 1879 CMP H Check for a 170 1880 A.8 A.8 Select ROM 35.3 36.2 1881 CMT OP.CTL2 Select ROM 35.3 1883 P.CHL EDU *-XINA	065		CNP HOV OUT XCHG	A 9.8	for a
170 1880 HOV A+8 323 362 1881 OUT OP-CTL2 351 1882 XCHC 351 1884 XINAL EOU +-XINA	170 1880 HOV A+8 353 1881 OP-CTL2 Select RDH 353 1883 XCHC 351 1883 PCHL 351 1884 XINAL EJU +-XINA	276		MOV OUT XCHG	A,8	NO C
323 362 1881 00T 0P-CTL2 Select 353 1882 XCHG 351 1883 PCUL #-XINA	323 362 1881 00T 0P-CTL2 Select 353 1882 XCHG 351 1883 PCHL E0U #-XINA	170		OUT XCHG		+00
353 1882 XCHG 351 1883 PCHL 1884 XINAL EDU	353 1882 XCHG 351 1883 PCHL 1884 XINAL EDU	323 362		XCHC	0P.CTL2	200
1884 XINAL EOU	1884 XINAL EOU	353 351		DCHI		
		1884	NAL	Eou	*-XI NA	
			:			

306 004 1899 307 177 004 1891 157 1894 031 1894 031 1894 031 1896 043 1896 1896 041 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1896 1900	100 004 1890 EXTURE ADD A A = 2 * A A A = 2 *	105 004 1880 1891 640 641 640 1891 640 641 640 1892 641 640 1893 641 640 1894 641 641 641 641 641 641 641 641 641 64				1887	*	Extended	Command Table		/kam660 2/
17   1004   1890   EXTCHO ADI   4   1   1   1   1   1   1   1   1   1	306 004 1830 EXICHO ADI 4  207 1831	306 004 1890 EXTCHO AD 4 2 2 A 4 2 2 A 4 2 2 A 4 2 2 A 4 2 2 A 4 2 2 A 4 4 4 0 0.44 0.00 1894 0.00 1894 0.00 1894 0.00 1894 0.00 1894 0.00 1894 0.00 1895 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.	:			1889					
157 004 1892 LXI D.EXTCMOA 1593 MY H-0 1594 MY H-0 1595 MY H-0 1595 MY H-0 1596 MOV H-0 1597 MOV	021 177 004 1892 LXI D.EXTCHOA  046 0000 1894 HVI High  170 1895 HVI High  170 1896 HVI HIGH  170 1897 HVI HIGH  1897 HVI HVI HIGH  1897 HVI HIGH  1897 HVI HIGH  1897 HVI HIGH  1897 HVI HVI HIGH  1897 HVI HIGH  1897 HVI HIGH  1897 HVI HIGH  1897 HVI HVI HIGH  1897 HVI HIGH  1897 HVI HIGH  1897 HVI HIGH  1897 HVI HVI HIGH  1897 HVI HIGH  1897 HVI HIGH  1897 HVI HIGH  1897 HVI H	157 177 004 1892 LXI D.EXTCROA 157 1893 ROY HAN HI - Processor Address 034 1895 ROY HAN HI - Processor Address 035 1895 ROY HAN HI - Processor Address 157 1899 ROY HAN HI - Processor Address 158 1899 ROY HAN HI - Processor Address 158 1899 ROY HAN HI - Processor Address 159 1899 ROY HAN HI - Processor Address 150 1899 ROY HAN BOTTON TO THE FROM TO THE PROCESSOR ADDRESS ROY BOOT TO THE PROCESSOR ADDRESSOR ADDRESS ROY BOOT TO THE PROCESSOR ADDRESS RO				1890	<b>-</b>			<b>*</b> ~ <b>=</b>	
046 000 1894 HVI	046 000 1894 NVI HVI HVI HVI - Processor Address 1914 1895 NAD HVI HVI HVI - Processor Address 1914 1895 NAD HVI HVI HVI HVI - Processor Address 1914 1899 NAV HVI HVI HVI - Processor Address 1915 NAV HVI HVI - Processor Address 1915 NAV HVI HVI - Processor Address 1915 NAV HVI HVI - Primary Boot 1915 NAV HVI HVI - Primary Boot 1915 NAV HVI SECROR 1917 NAV HVI STANDE NAV HVI	1994   1994   1994   1994   1995			8	1892			), EXTCHOA		
130	1895   DAD   D	1995   DAD   D   *(III.) = Processor Address   DAD   D   PAPPER   DAD   D   PAPPER   DAD   D   PAPPER   DAD   DA	:		0	1894			0.		
176   1896	176 1896	176 1896 HOV AND		131		1895				- Processor Addres	
150	157 1999 nov H,N  322 000 1902 EXTCNDA DW ERROR '0' Illegal  322 000 1903 DW ERROR '0' Illegal  323 004 1903 DW ERROR '0' Illegal  325 004 1904 DW ERROR '0' Illegal  326 004 1905 DW ERROR '0' Illegal  327 000 1905 W ERROR '0' Illegal  328 004 1905 W ERROR '0' Illegal  328 006 004 004	154 1999 HOW L+A HL = Processor Address 157 1990 PCHL ERROR Processor Address 157 1900 PCHL ERROR POLITION DW ERROR 10-1 Illegal Boot 1902 EXTCHDA DW ERROR 12-1 Processor Address 153.004 1902 EXTCHDA DW ERROR 12-1 Processor 253.004 1903 DW ERROR 13-1 Illegal Boot 253.004 1905 DW		943		1896			E		
15.7 18.99	157 1999 NOV L,A HL = Processor Address 150 1900 EXTCND DW EARDR 11 -> Primary Boot 22 000 1902 EXTCND DW ERROR 12 -> Secondary Boot 23 004 1904 DW ERROR 12 -> Secondary Boot 23 004 1905 DW ERROR 13 -> Illegal 1907 ** AUTUB Auto Boot 1908 ** AUTUB Performs an auto boot of the primary device. 1910 ** AUTUB Performs an auto boot of the primary device. 1910 ** AUTUB Performs an auto boot of the primary device. 1911 ** EXIT: To PRIBUD 1911 ** EXIT: To PRIBUD 1911 ** USES: ALL 1912 ** AUTUB LXI H, ** WELAG 1914 ** USES: ALL 1915 ** AUTUB LXI H, ** WELAG 1916 ** AUTUB LXI H, ** WELAG 1917 ** AUTUB LXI H, ** WELAG 1918 ** AUTUB LXI H, ** WELAG 1919 ** AUTUB LXI H, ** WELAG 1910 ** AUTUB LXI H, ** WELAG 1911 ** AUTUB LXI H, ** MILLO WELAG-IL-CA CLI-CA CLI-CA SPRII AUTUB LXI H, ** AUTU	157 1899 HOV L'A HL = Processor Address 1900 PCHL   1900	:	•		1898			Ε.	:	
131 1900 PCHL ERROR 10.2 FILEGRAL 1901 1902 EXTCMD DW FREED 1.2 -> PFILMATY BOOT 1902 EXTCMD DW FREED 1.2 -> PFILMATY BOOT 1903 DW FREED 1.2 -> Secondary Boot 1903 DW FREED 1.3 -> Secondary Boot 1903 DW FREED 1.3 -> Secondary Boot 1903 DW FREED 1904 DW FREED 1904 DW FREED 1905 DW F	351 1900 PCHL ERROR 101 1900  322 000 1902 EXTCHOA DW ERROR 101> Primary Goot 256 004 1904 DW ERROR 121> Primary Goot 256 004 1905 DW ERROR 121> Primary Goot 256 004 1905 DW ERROR 131 Illegal Scondary Boot 1907 ** AUTUB Auto Boot 1908 ** Illegal Boot 1908 ** AUTUB Performs an auto boot of the primary device. 1909 ** AUTUB Performs an auto boot of the primary device. 1910 ** AUTUB Performs an auto boot of the primary device. 1910 ** AUTUB Performs an auto boot of the primary device. 1910 ** AUTUB PEXIT: To PRIBUD 1912 ** EXIT: To PRIBUD 1912 ** EXIT: To PRIBUD 1912 ** EXIT: To PRIBUD 1914 ** AUTUB LXI H+.MFLAG 1919 ** AUTUB	351 1900 PCHL ERROR 11991 322 000 1902 EXTCHOA DH PRIBOD 323 0004 1903 OH PRIBOD 325 0004 1903 OH PRIBOD 325 0004 1905 OH PRIBOD 325 0004 1905 OH PRIBOD 325 0004 1905 OH PRIBOD 326 0004 1905 OH PRIBOD 327 0005 1905 OH PRIBOD 328 0004 1905 OH PRIBOD 328 0004 1905 OH PRIBOD 328 0004 1905 OH PRIBOD 329 0004 1905 OH PRIBOD 320 0004 1906 OH PRIBOD 320 0004 1906 OH PRIBOD 320 0004 1900 OH PRIBOD	•	. 57		1899		:	¥ 6-	rocessor Addres	
222 000 1902 EXTCHOA DH ERROR '0' 11 - 9 FILMENTY BOOT 236 004 1904 1905 DH ERROR '1' - 9 FILMENTY BOOT 236 004 1905 DH ERROR '1' - 9 FILMENTY BOOT 236 004 1905 DH ERROR '1' - 9 FILMENTY BOOT 236 004 1905 H AUTOB PETFORMS AN AUTO BOOT OF the primary device. 1908 H AUTOB PETFORMS AN AUTO BOOT OF the primary device. 1910 H AUTOB PETFORMS AN AUTO BOOT OF THE primary device. 1910 H EXIT: To PRIBOD DUDU-NFR 1911 H EXIT: To PRIBOD DUDU-NFR 1915 H AUTOB DISPLAY DODU-DUD-NFR 1915 H AUTOB DISPLAY DODU-DUD-NFR 1912 H AUTOB DISPLAY DODU-DUD-NFR 1920 DISPLAY DATE DOS 371 DISPLAY DATE DISPLAY DATE DOS 371 DISPLAY DATE DISPLAY DATE DISPLAY DATE DOS 371 DISPLAY DATE DATE DATE DATE DATE DATE DATE DATE	252 000 1902 EXTCHOA DW ERROR 11 -> Primary Goot 253 004 1903 DW SECBOO 125 -> Secondary Boot 1904 1905 DW SECBOO 12 -> Secondary Boot 1905 DW SECBOO 12 -> Secondary Boot 1907 ++ AUTUB - Auto Boot 1908 + AUTUB Performs an auto boot of the primary device. 1909 + AUTUB Performs an auto boot of the primary device. 1909 + AUTUB Performs an auto boot of the primary device. 1910 + AUTUB EXIT: To PRIBUD 1911 + ENTRY: NONE 1912 + EXIT: To PRIBUD 1912 + EXIT: To PRIBUD 1913 + EXIT: To PRIBUD 1915 + USES: ALL 1915 + USES	252 000 1902 EXTGNOA DW ERROR '0' 1 linesal 255 004 1904 DW SECBOO '2' -> Secondary Boot 256 004 1904 DW SECBOO '2' -> Secondary Boot 256 004 1904 DW SECBOO '2' -> Secondary Boot 256 004 1904 DW SECBOO '2' -> Secondary Boot 256 004 1905 PW BOO COLOR TO CO				1900					
253 004 1903	233 00,4 1903 00,4 PRIBBO '1' -> Primary Boot 226 000, 1904 1904 00,4 1904 00,4 1905 00,4 ERROR '3' Illegal Boot 226 000, 1905 00,4 ERROR '3' Illegal Boot 1905 00,4 1900 00,4 1	234 004 1903 0N PRIBOD '12 -> Frimary Boot 1904 1905 1905 1905 1905 1905 1905 1905 1905			00	1905	EXTCMDA		RROR		
232 000 1905 0W SECRON '3' Illegal BOOL 1907 ** AUTUB - AUTO BOOL 1908 ** AUTUB Performs an auto boot of the primary device. 1908 ** AUTUB Performs an auto boot of the primary device. 1908 ** AUTUB Performs an auto boot of the primary device. 1908 ** AUTUB Performs an auto boot of the primary device. 1908 ** AUTUB PERFORMS AUTUBLE PERFORMS AUTU	225 000 1905 00	220 004 1905 DN SECRET  1907 ** AUTUB - Auto Boot  1908 * AUTUB PETforms an auto boot of the primary device.  1910 * AUTUB PETforms an auto boot of the primary device.  1911 * EXIT: To PRIBUD  1912 * EXIT: To PRIBUD  1913 * EXIT: To PRIBUD  1914 * EXIT: To PRIBUD  1915 * USES: ALL  1915 * USES: ALL  1916 * USES: ALL  1917 * USES: ALL  1918 * USES: ALL  1919 * ANI  1920 * ANI  1921 * ANI  1921 * ANI  1921 * ANI  1922 * ALL  1931 * ANI  1921 * ANI  1922 * ALL  1931 * ANI  1923 * ANI  1923 * ANI  1924 * ANI  1925 * ANI  1925 * ANI  1927 * ANI  1928 * ANI  1929 * ANI  1920 * ANI  1920 * ANI  1921 * ANI  1921 * ANI  1922 * ANI  1923 * ANI  1923 * ANI  1924 * ANI  1925 * ANI  1925 * ANI  1926 * ANI  1927 * ANI  1928 * ANI  1929 * ANI  1920 * ANI			*	1903			RI 800	=> Primary 60	
1907 ** AUTUB - Auto Boot of the primary device. 1908 * AUTUB performs an auto boot of the primary device. 1909 * AUTUB performs an auto boot of the primary device. 1911 * EXIT: To PRIBUD 1913 * EXIT: To PRIBUD 1914 * USES: ALL 1915 * USES: ALL 1915 * USES: ALL 1916 * USES: ALL 1917 * USES: ALL 1918 * USES: ALL 1919 * USES: ALL 1919 * USES: ALL 1910 * USES: ALL 1910 * USES: ALL 1911 * USES: ALL 1911 * USES: ALL 1912 * USES: ALL 1914 * USES: ALL 1915 * USES: ALL 1916 * USES: ALL 1917 * USES: ALL 1918 * USES: ALL 1919 * USES: ALL 1919 * USES: ALL 1910 * USES: ALL	1907 ** AUTUB - Auto Boot   1908   1909 * AUTUB PETFORMS an auto boot of the primary device.   1909 * AUTUB PETFORMS an auto boot of the primary device.   1909 * AUTUB PETFORMS and auto boot of the primary device.   1910 * AUTUB PETFORMS   1911 * EXIT: To PRIBOD   1912 * EXIT: To PRIBOD   1915 * AUTUB LXI H** HFLAG   1916 * AUTUB LXI H** HFLAG   1916 * AUTUB LXI H** HFLAG   1917 HOV H** HERRY CILELG-NELAG-I ENDOIS DISPIBY Update, and Re-Fr HERRY   1922   1922   1924 HAVI HERRY CILELG-NELAG-I HORD   1923   1924 HAVI   1925   1925   1926 HAVI   1925   1926 HAVI   1926   1927   1	1907 ** AUTUB - Auto Boot   1908 *   1908 *   1908 *   1908 *   1908 *   1908 *   1909 *   1909 *   1909 *   1909 *   1909 *   1900 *   19				1905			EROR	Illegal	
1907 ** AUTUB — Auto Boot  1908 * AUTUB performs an auto boot of the primary device.  1909 * AUTUB performs an auto boot of the primary device.  1910 * AUTUB performs an auto boot of the primary device.  1911 * EXIT: To PRIBUD  1913 * EXIT: To PRIBUD  1914 * USES: ALL  1915 * USES: ALL  1916 * USES: ALL  1917 * USES: ALL  1917 * USES: ALL  1918 * AUTUB LXI  1917 * USES: ALL  1910 * USES: ALL  1910 * USES: ALL  1911 * USES: ALL  1911 * USES: ALL  1912 * USES: ALL  1912 * USES: ALL  1914 * USES: ALL  1915 * USES: ALL  1915 * USES: ALL  1916 * USES: ALL  1917 * USES: ALL  1918 * USES: ALL  1918 * USES: ALL  1920 * USES: ALL  1921 * USES: ALL  1921 * USES: ALL  1922 * USES: ALL  1923 * USES: ALL  1924 * USES: ALL  1925 * USES: ALL  1926 * USES: ALL  1927 * USES: ALL  1928 * USES: ALL  1929 * USES: ALL  1920 * USES:	1907 ** AUTUB - Auto Boot  1908 * AUTUB Performs an auto boot of the primary device.  1909 * AUTUB Performs an auto boot of the primary device.  1910 * EXIT: To PRIBUD  1911 * EXIT: To PRIBUD  1912 * USES: ALL  1915 * USES: ALL  1916 * USES: ALL  1917 * USES: ALL  1918 * USES: ALL  1918 * USES: ALL  1918 * USES: ALL  1919 * USES: ALL  1910 * And  1910	1907 ** AUTUB — Auto Boot of the primary device.  1908 * AUTUB performs an auto boot of the primary device.  1910 * EXIT: To PRIBOD 1911 * EXIT: To PRIBOD 1912 * EXIT: To PRIBOD 1914 * EXIT: To PRIBOD 1915 * USES: ALL 1916 * USES: ALL 1916 * USES: ALL 1917 * USES: ALL 1918 * USES: ALL 1919 * ANI 3770—UO.DU.N.F.R 1910 * ANI 3770—UO.DU.N.F.R 1921 * ANI 3770—UO.DU.N.F.R 1922 * INX H.A. 1923 * INX H.A. 1924 * ANI A1 1925 * ANI A1 1925 * ANI A1 1926 * STA OSPROT 1927 * UNI A1 1928 * UNI A1 1929 * SPRU AIT PRIBOD 1929 * SPRU 1929 * SPRU 1929 * SPRU 1929 * SPRU 1929 * UNI A1 1929 * SPRU 1929 * SPRU 1929 * SPRU 1929 * SPRU 1929 * UNI A1 1929 * SPRU 1929 * SPRU 1929 * SPRU 1929 * SPRU 1920 * UNI A1 192							: :		
1909 ** AUTOB performs an auto boot of 1910 ** 1910 ** 1911 ** EXIT: To PRIBOD 1912 ** EXIT: To PRIBOD 1914 ** 1914 ** USES: ALL 1914 ** USES: ALL 1915 ** USES: ALL 1916 ** USES: ALL 1916 ** USES: ALL 1916 ** USES: ALL 1917 ** USES: ALL 1918 ** USES: AUTO: ALI PER 1918 ** USES: AUTO: AUT	1909 ** AUTOB performs an auto boot of 1910 ** 1910 ** 1911 ** ENTRY: NONE 1911 ** EXIT: To PRIBOD 1913 ** EXIT: To PRIBOD 1914 ** USES: ALL 1915 ** USES: ALL 1915 ** USES: ALL 1916 ** USES: ALL 1917 ** USES: ALL 1918 ** USES: ALL 1919 ** USES: AND	1909 ** AUTOB performs an auto boot of 1910 ** 1910 ** 1911 ** ENTRY: NONE 1911 ** EXIT: To PRIBOD 1913 ** EXIT: To PRIBOD 1914 ** USES: ALL 1915 ** USES: ALL 1916 ** USES: ALL 1918 ** UNIVORUMENTALY 1918 ** UNIVORUMENTALY 1919 ** USES: ALL 1921 ** UNIVORUMENTALY 1921 ** UNIVORUMENTALY 1922 ** UNIVORUMENTALY 1923 ** UNIVORUMENTALY 1923 ** UNIVORUMENTALY 1924 ** UNIVORUMEN				80	* *				/Ram8Go 2/
1910 * ENTRY: NONE 1911 * ENTRY: NONE 1913 * EXIT: To PRIBOD 1914 * USES: ALL 1915 * USES: ALL 1916 * USES: ALL 1916 * USES: ALL 1917 * USES: ALL 1918 * USES: ALL 1919 * USES: ALL 1919 * USES: ALL 1910 * USES: ALL 1910 * USES: ALL 1910 * USES: ALL 1910 * USES: ALL 1911 * USES: ALL 1911 * USES: ALL 1912 * USES: ALL 1913 * USES: ALL 1914 * USES: ALL 1915 * USES: ALL 1916 * USES: ALL 1917 * USES: ALL 1918 * USES: ALL 1918 * USES: ALL 1919 * USES: ALL 1919 * USES: ALL 1910 * USES:	1910 * ENTRY: NONE  1911 * ENTRY: NONE  1913 * EXIT: To PRIBUD  1914 * USES: ALL  1915 * USES: ALL  1916 * USES: ALL  1916 * USES: ALL  1917 * USES: ALL  1918 * USES: ALL  1919 * USES: ALL  1910 * USES: ALL  1910 * USES: ALL  1911 * USES: ALL  1911 * USES: ALL  1911 * USES: ALL  1912 * USES: ALL  1913 * USES: ALL  1914 * USES: ALL  1915 * USES: ALL  1916 * USES: ALL  1917 * USES: ALL  1920 * USES: ALL  1921 * USES: ALL  1921 * USES: ALL  1921 * USES: ALL  1922 * USES: ALL  1923 * USES: ALL  1924 * USES: ALL  1925 * USES: ALL  1926 * USES: ALL  1927 * USES: ALL  1928 * USES: ALL  1929 * USES: ALL  1920 * USES: ALL  19	1910 # ENTRY: NONE  1911 # EXIT: To PRIBOD  1913 # EXIT: To PRIBOD  1914 # USES: ALL  1915 # USES: ALL  1916 # USES: ALL  1916 # USES: ALL  1916 # USES: ALL  1916 # USES: ALL  1917 # USES: ALL  1918 # USES: ALL  1919 # USES: ALL  1919 # USES: ALL  1910 # USES: ALL				5.6			an auto		
1912 # EXIT: TO PRIBOD 1913 # EXIT: TO PRIBOD 1914 # USES: ALL 1915 # USES: ALL 1916 # USES: ALL 1916 # USES: ALL 1916 # USES: ALL 1917	1912 # EXIT: TO PRIBOD 1913 # EXIT: TO PRIBOD 1914 # USES: ALL 1915 # USES: ALL 1916 # USES: ALL 1917	1912 # EXIT: TO PRIBOD 1913 # EXIT: TO PRIBOD 1914 # USES: ALL 1915 # USES: ALL 1916 # USES: ALL 1916 # USES: ALL 1917 # USES: ALL 1918 # USES: ALL 1919 # WILLAG 1919 # WILLAG 1919 # WILLAG 1910 # WILLAG 1920 # WILLAG 1921 # WILLAG 1922 # WILLAG 1923 # WILLAG 1924 # WILLAG 1925 # WILLAG 1925 # WILLAG 1926 # WILLAG 1926 # WILLAG 1927 # WILLAG 1927 # WILLAG 1928 # WILLAG 1928 # WILLAG 1929 # WILLAG 1929 # WILLAG 1920 # WILLAG 1920 # WILLAG 1921 # WILLAG 1922 # WILLAG 1923 # WILLAG 1924 # WILLAG 1925 # WILLAG 1926 # WILLAG 1927 # WILLAG 1928 # WILLAG 1929 # WILLAG 1920 # W		:		1910	* *	- :			
1913 * EXIT: TO PRIBOD 1914 * USES: ALL 1915 * USES: ALL 1916 * USES: ALL 1917 * USES: ALL 1917 * USES: ALL 1918 * USES: ALL 1919 * USES: ALL 1919 * USES: ALL 1910 * USES: ALL	1913 + EXIT: TO PRIBUD 1914 + USES: ALL 1915 + USES: ALL 1915 + USES: ALL 1916 + USES: ALL 1917 + USES: ALL 1918 AUTOB LXI H+.MFLAG 1919 ANI 3770-UO.DOU-UO.NFR 1920 ANI 3770-UO.DOU-UO.NFR 1921 HWY H,A H 1922 ERRNZ CLELGMFLAG-I 1923 ERRNZ H,CBSSI+CB-MTL+CGCLI+ 066 360 1924 HWI A,-I 066 360 1925 ERRNZ H,CBMFLAG-I 1925 HWI A,-I 056 060 060 1926 EI	1913				1911	• •		NO NE		
1915 # USES: ALL 1916 # 1916 # 1916 # 1916 # 1916 # 1917 # 1919   NOV   A,H   LAC   1918   AUTOB   LXI   H, HFLAG   L919   NOV   A,H   A,H   L921   ANI   3770-UG,DOU-UG,NFR   L921   ANI   A,H	1915 # USES: ALL 1916 # USES: ALL 1916 # USES: ALL 1916 # USES: ALL 1919 AUTOB LXI H, "MFLAG 1920 ANI 3770-U0.DDU-U0.MFR 1921 BY ANI 3770-U0.DDU-U0.MFR 1921 ANI 3770-U0.DDU-U0.MFR 1922 ANI 1922 ERRNZ CHLEG-MFLAG-L 1923 AVI A,-1 066 360 1926 BRNZ CHLEG-MFLAG-L 1924 AVI A,-1 062 006 040 1926 STA 0SPROT 373 1929 STA SPROT 373 1929 SPHL 393 253 004 1930 JMP PKIBOO GOOT PR	1915 # USES: ALL 1916 # 1916 # 1916 # 1916 # 1916 # 1917   1918   AUTOB   LXI   H+.MFLAG   1918   AUTOB   LXI   H+.MFLAG   1919   ANI   3770—UG.DOU—UG.NFR   Enable   1921   ANI   AHA   ERRNZ   CTLFLG—NFLAG—I   1921   ANI   AHA				1913	* •	EXIT:	To PR1800		
1916 * 1917 1917 1917 1918 AUTOB LXI H, MFLAG 1919 AUV A, M 346 275 1920 AUI 167 1921 AUX H, A 1922 AUX 1923 ERRNZ CTLFLG-, MFLAG-1 043 1924 AVI A, M 1924 AVI A, M 1925 AVI A, M 1925 AVI A, M 1925 AVI A, M 1925 AVI A, M 1926 AVI A, M 1927 AVI A, M 1928 AVI A, M 1928 AVI A, M 1929 AVI A, M 1929 AVI A, M 1929 AVI A, M 1929 AVI B, M 1929 A	1916 * 1917 1917 1918 1919 1919 1919 1919 1919	1916 * 1917 1917 1918 1918 1919 1919 1919 1919				1915	• •	:	ALL		
041 010 040 1918 AUTOB LXI H, MFLAG 176 1919 AUT A, H 346 275 1920 ANI 3770-U0. DDU-U0. NF X 167 1921 ANI 3770-U0. DDU-U0. NF X 167 1922 INX H 1923 ERRNZ CLELG MFLAG-1 046 340 1924 MYI A, -1 046 340 1925 ANI A, -1 047 004 040 1926 STA DSPROT 373 1927 EI 052 035 040 1928 LHLD REGPTR 371 1929 SPHL	041 010 040 1916 AUTOB LXI H**MFLAG 176 1919 ANI 3770-U0.00U-U0.NFR 1920 ANI 3770-U0.00U-U0.NFR 1921 I922 INX H*A 1922 INX H*A 043 1924 INX H*A 076 340 1924 INX H*CB.SSI*CB.HTL*CB.CLI* 066 340 1924 INVI H*CB.SSI*CB.HTL*CB.CLI* 067 377 1925 INVI A*-1 OSPROT 1929 STA OSPROT 1929 STA OSPROT 1929 SPHL 1929 SPHL 371 1929 SPHL	041 010 040 1918 AUTOB LXI H; MFLAG 176 1919 ANI 3770-U0.D0U-U0.NFR 167 1920 ANI 3770-U0.D0U-U0.NFR 167 1921 HOV H; A 1923 ERRNZ CTLFLG-, MFLAG-1 1924 HVI H; CB.SSI+CB.HTL+C3.CLI+ 062 066 040 1926 STA 0.5PROT Ail Per 1927 EI 0.5PROT Ail Per 1928 LHLD REGFTR 371 1929 SPHL PRIBOD 600t PR				1916	*	:			
176 1919 MOV A,M 346 275 1920 ANI 3770-U0,DOU-U0,NFR 167 1921 MOV H,A 1922 INX H 0043 1922 INX H 1923 ERRNZ CTLFLG-,MFLAG-1 066 360 1924 MVI M,CB,SSI+CB,MTL+CG,CLI+ 076 307 1925 MVI A,-1 052 035 040 1926 SPROT A11 Per 373 1927 E1 052 035 040 1930 JMP PRIBOO GOOT PR	176 1919 MOV A,M 346 275 1920 ANI 3770-U0,DDU-UD,NER 1921 INX H,A 1922 INX H,C 1923 ERRY CTLELG-MELAG-1 066 360 1924 MYI H,CB.SSI+CB.MIL+CG.CLI+ 076 377 1925 MVI A,-1 062 006 040 1926 STA DSPROT AIPPET 373 1927 EI DSPROT AIPPET 373 1929 LHLD REGPTR 371 1929 SPHL AGOOT PR	176 1919 MOV A,M 346 275 1920 ANI 3770-U0,DOU-U0,NFR 167 1921 MOV H,A 043 1922 INX H,C H-LG-,MFLAG-1 066 360 1924 MVI H,CB,SSI+CB,MIL+CG,CLI+ 076 377 1925 MVI A,-1 052 035 040 1926 STA 373 1927 EI 052 035 040 1928 LHLD REGPTR 371 1929 SPHL 373 253 004 1930 JMP PKIBOO GOOT PK			0	1917	•		H MF LAG		
346 275 1920 ANI 3774-00.000-00.NFK 167 1921 MOV H,A 1922 ERRN CTLFLGNFLAG-1 066 360 1924 MVI M,CB.SSI+CB.MIL+CB.CLI+ 076 377 1925 MVI A,-1 067 006 040 1926 STA 0SPROT 373 1927 EI 052 035 040 1928 LHLD REGPTR 371 1929 SPHL	346 277 1920 AN1 3774-00.00.NFK 167 1921 AOV 043 1922 ERRNZ CTLFLGMFLAG-1 066 360 1924 MYI M,CB.SSI+CB.MTL+CG.CLI+ 076 377 1925 MYI A,-1 373 1926 COV 1928 LHLD REGPTR 371 1929 SPHL RIBOO GOOT PR	346 275 1920 AN1 3/74-00.000-00.nrx 167 1921 HOV H,A 1922 INX H 1923 ERRNZ CTLFLGMFLAG-1 066 360 1925 HVI M,CB.SSI+CB.MTL+CB.CLI+ 076 377 1925 HVI A,-1 052 035 040 1926 STA 0SPROT 373 1927 EI 052 035 040 1920 SPHL REGPTR 371 1929 SPHL REGPTR 373 253 004 1930 JMP PKIBOO GOOT PR				1919			H 6 M		
1927 1924 1927 1930 1922 1932 1932 1932 1932 1932 1932 1932	197 1921 INX H 1923 INX H 1923 ERRNZ CTLFLG-,MFLAG-1 1924 MVI M,CB.SSI+CB.MTL+Cd.CLI+ 1925 MVI A,-1 1925 MVI A,-1 1926 STA DSPROT 1927 EI 1929 LHLD REGPTR 1929 SPHL 1929 SPHL	197 1971 INX H EIRDIG 1972 INX H EIRDIG 1973 1973 ERRN CTLFLG-WFLAG-1 1973 ERRN CTLFLG-WFLAG-1 1973 ERRN MVI M,CB.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.SSI+CB.MIL+CG.CLI+CG.CCI+CG.SSI+CB.MIL+CG.CCI+CG.CCI+CG.CCI+CG.SSI+CB.MIL+CG.CCI+CG.	:	0.1	2	1920		:	3 / /u-nna•nna-n		
066 360 1923 ERRNZ CTLFLG-, MFLAG-1 076 377 1925 MVI M, CB, SSI+CB, MTL+Cd, CLI+CB, SP 076 377 1925 MVI A, 1 062 006 040 1926 STA 0SPROT All Periods 373 1927 EI 052 035 040 1928 LHLD REGPTR 371 1929 SPHL REGPTR	066 360 1923 ERRNZ CTLFLG-, MFLAG-1 076 377 1925 MVI A,-1 076 377 1925 HVI A,-1 373 1927 E E E E PEGPTR 371 1929 SPHL REGPTR 371 1929 SPHL PRIBOD GOOT PRIMARY	066 360 1923 ERRNZ CTLFLG-, MFLAG-1 076 377 1925 MVI M, CB, SSI+CB, MTL+Cd, CLI+CB, SP 076 377 1925 STA 0SPROT A11 Periods 373 1927 E1 052 035 040 1928 LHLD REGPTR 371 1929 SPHL REGPTR 373 253 004 1930 JMP PRIBOD GOOT PRIMARY		16/		1921			<b>4</b>	e -	
217 066 360 1924 MYI M,CB.SSI.CB.MTL.CB.CLI.CB.SP.SP. 221 076 377 1925 MYI A,-1. 223 066 040 1925 MYI A,-1. 226 373 1927 EI EI B.C.C.C.C.C.C.C.C.C.C.C.C.C.C.C.C.C.C.C	217 066 360 1924 MVI M,CB.SSI+CB.MTL+Cd.CLI+CB.SP 221 076 377 1925 MVI A1 226 373 192 1926 STA 0SPROT All Periods 226 373 095 040 1928 LHLD REGPTR 227 052 035 040 1928 SPHL PKIBOD doot PKImary	217 066 360 1924 MYI M,CB.SSI+CB.MIL+Cd.CLI+CB.SP -221 076 377 1925 MYI A,-1 -222 042 006 040 1926 STA 0SPROT All Periods -226 373 1927 EI CHLD REGPTR -237 371 1929 SPHL REGPTR -232 371 1929 JMP PKIBOO doot PKImary	:			1923		ERRNZ	CTLFLGMFLAG-1		
221 076 377 1925 NVI A3-1 223 373 062 096 090 1926 STA 05PROT 226 373 1927 EI 227 052 035 090 1928 LMLD REGPTR 232 371 1929 SPHL PRIBUD doot PRImary	221 076 377 1925 NTA 05PROT A11 Periods 223 006 040 1926 E1 CHLD REGPTR 224 373 1929 SPHL REGPTR 232 371 1929 SPHL PRIBOO GOOT PRImary 233 303 253 004 1930 JMP PRIBOO GOOT PRImary	221 076 377 1925 STA 05PRUT A1-1 223 373 1926 E1 E1 CHLD REGPTR 227 052 035 040 1928 LHLD REGPTR 232 371 1929 SPHL PKIBUO doot PKImary 233 303 253 004 1930 JMP PKIBUO	217		60	1924		IAN	M, CB. SSI+CB.MTL	+C3.CLI+C8.SPK	
226 373 1927 EI 227 052 035 040 1928 LHLD REGPTR 232 371 1929 SPHL 233 303 253 004 1930 JMP PRIBOO doot PRImary	226 373 1927 EI 227 052 035 040 1928 LHLD REGPTR 232 371 1929 SPHL PKIBOO GOOT PKImary 233 303 253 004 1930 JMP PKIBOO	226 373 1927 EI 227 052 035 040 1928 LHLD REGPTR 232 371 1929 SPHL 233 303 253 004 1930 JMP PRIBOO doot PRImary	223			1925		NVI	A,-1 OSPROT	All Periods OFF	
.227 052 035 040 1928 LHLD REGPTR .232 371 1929 SPHL PRIBOD GOOT PAIMARY .233 303 253 004 1930 JMP	.227 052 035 040 1928 LHLD REGPTR .232 371 1929 SPHL .233 303 253 004 1930 JMP PKIBOO doot PKImary	.227 052 035 040 1928 LHLD REGPTR .232 371 1929 SPHL .233 303 253 004 1930 JMP PRIBOO GOOT PRIMARY	226			1927		ÉI			
233 303 253 004 1930 JMP PRIBOO GOOT PRIMARY	-233 303 253 004 1930 JMP PRIBOD GOOT PRIMARY	-232 303 253 004 1930 JMP PRIBOO GOOT PRImary	.227	25	35	1928		LHLD	REGPTR		
			233	33	53.00	1930		J. H. P.	PKIBOO	Pkimary	

		10.14				
		1935		PRIBOO	is called to boo	٥
		1936	* *	defi	d in the	guration port IP.CON.
		1938		Secondary	ry boot device.	If the Ca.PRI seitch is one,
		1939	*	then address	170 15	primary device, otherwise,
		1940	* •	address 17	4 is the boo	device. From there, t
		1941	• •	Configurat	TOU SWITCH T	urther determines device type
		1943	•	•		
		1944				
36 257		1945	SEC800	XRA	<b>V</b>	
297 062	140 041	1946	SECBOO.	STA .v.	AIO.OIA	Zero Boot Unit
245 001	2:3	1048		1 Y 1	70.702	
004.247 057		1949		CHA		Invert Primary Flag
250	264 004	1950		JHP	8001	. > • •
:	:	1981				
04.253 257		1952	PR 1800	XRA	<	
004.254 062	061 041	1953	PRIB00.	STA	AIO.CNI	
04.257 001		1954		LXI	B, MSGPRI	
04.262 333	362	1955		Z	I P CON	
;		1956		;		
264 061	200 042	1957	8001	Ľ	SP , STACK	Initialize the stack-pointer
221 232	34.3	1950		1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	7 × × ×	
345	305	1040		1710	1 - CON	
004.274 312	317 004	1961		75	8002	174 is the device to boot
:	:	1965				
		1963	*	Boot De	vice is 170	
,,,	,	1964		1	•	
	120 041	1046		27.4	20 T & C	4000
004.304 361		1967		4 O d	× × ×	1000 5 4
305	014	1968		Z	CN 1 70M	:
307	=	1969		75	BERR	No Device installed at 170
000		1970		ERRNZ	CNO.NDI	F. 89
312 01		1441		RRC		
004.313 017		1972		RC		Device Type converted to index
314 30	327 004	1973		a E T	8003	
		1974	4	•	10 17	
		1976				
317	174	1977	8002	IA	A-1740	
004.321 062	120 041	1978		STA	80 A	Save Boot Device Address
.324		1979		POP	PSH	
325	003	1980		INA	CN.174M	Mask out device type
		1981				
		1982	•	Initial	ize Vectors and	Display
36		1984	8003	PUSH	MSd	> 0 > 0 > 0 > 0 > 0 > 0 > 0 > 0 > 0 > 0
004.330 305		1985		PUSH	:	,
	•	1986				
004.331 076	320	1987		N I	A, CB.SSI+CB.CLI	+C8.SPK
***	0	2		V L	1	Tire of the book to the bods

1989			Interrupt Vectors							Time-Out Counter			2			Initialize Rom/Ram Vectors			Mesory		Enable Clock Int. turn off Display Update							5				Force Boot to return to 8006		Boot Device Flag	•				
1989  1989  1989  1989  1981  1981  1981  1981  1981  1981  1981  1981  1982  1983  1983  1984  1884				M, #EIRET	T	M, EIRET/256	•	8004					1	•	8, BOOTA			H, D. RAM			:	• MFLAG	60	0,FPLE0S	SAOV ASELFN	L,9-MSGLEN			) ) 	enna	PSW				•	0.41	0,800A	A9.8	I
1989           16 007         1989           41 037         040         1990           43 303         1992         80           43 303         1999         80           44 002         1994         1999           45 002         1999         1999           43 100         1999         1999           43 100         1999         1999           43 100         1999         1999           43 100         1999         1999           43 100         1999         1999           43 100         1999         1999           44 100         1999         1999           45 100         1999         1999           47 100         1999         1999           48 100         100         2000           40 100         2010         2010           40 100         2010         2010           41 200         200         2010           42 100         040         2010           45 040         2010         2010           46 040         2010         2010           47 200         040         2020           40 040		LXI	IAE	:			:					:	-			:	:		ر		ORI	STA	<b>6</b> 0 <b>6</b>	ראו			MVI	:	OCR	785	909	PUSH		STA	NOV	IAM	LXI DV0	NOM	XNI
76 007 64 303 65 303 65 002 65 002 65 002 65 002 67 122 04 67 124 04 67 124 04 67 124 04 67 127 04 67 127 04 67 124 04 67 127 04 67 127 04 67 127 04 67 127 04 68 003 69 03 04 60 03 60	1989	1990	1992	1994	1995	1996	1998	-	: ~	۰.۰	~	.~		• ~		:			• . •	:	201	201	2 2	50	200	2025	26 27	28	20	2 2	7	)   	5	7	2038	7	2 50	2042	2043
	: :	007	303	: 6:	£3	0 99	75	02 343 00		62 122	41 031	45 154	41 221	5	132 0	110		240 04	323		003	010 04		013 04	151 00	00	37	)23		9 6	61	25 25		62 121 0 07	. 25	46 000	21 125	176	043

351 2046 PCHL  072 010 040 2047 3040 ANI 3774-U0.DDU Turn on Display Updat 2047 2049 ANI 3774-U0.DDU Turn on Display Updat 346 375 2049 ANI 3774-U0.DDU Turn on Display Updat 052 124 041 2050 ANI 2050 ANI 2050 ANI 2050 ANI 2051 ANI 2050 A	10 040   2046	10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		70			3706		20	•	
2010 040 2046 8006 LDA . HFLAG . HALD . USRCLK . HALD . HALD . HER HALD . HALD	39.1 200 04.0 2046  30.2 00.0 04.0 2056  30.2 00.0 2057  30.3 00.0 2057  30.0 2057	772 115 176 176 2010 COLOR 176 176 176 176 176 176 176 176 176 176			:	:	2042		<b>&gt;</b>	¥.4	HL = Device Frocessor Address
1 072 010 040 2048 8006 LDA .HFLAG 5 042 010 040 2049 AAI	1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	1,000, 0,000,	•	351			2046		L H L		
346 375 2049 ANI 3779-U0.DDU Turn on Display Update 052 124 041 2051 LHD USKEKK 052 212 001 2052 SHLD UNFEC+I Clear Time-Out Vector to just user v 32 232 000 2053 JL CHD USKEKK  32 232 000 042 2053 JL CHD USKEKW  2054	346.375 2049 ANI 3770-U0,DDU Turn on Display Update 0.052 124 0.05	346 3175 049				0.4	2048	9008	LDA	. MFLAG	
042 010 040 2050 STA .#FLAG 042 124 041 2055 LHUD USFC*I Clear Time-Out Vector to Just user v 042 040 040 2053 JHUD UIYEC*I Clear Time-Out Vector to Just user v 332 322 000 2053 JC ERROR Boot Routines return here 2055	056 210 40 4 2050	056 200 40 2050 51AffL.K. Restore of ishial front panel mode 052 124 04 2050 51AffL.K. Square of ishial front panel mode 052 124 04 2052 540 040 2053 540 0176CH Clear Time-cuturn here 052 205 040 2053 540 0176CH Clear Time-cuturn here 052 205 040 205 050 040 040 040 040 040 040 040 040 0			375		2049		INA	3770-00.000	Display Update
042 124 044 2051 LHID UVRCLK 042 040 2052 SHLD UVRCAL 332 322 000 2053 JC ERROR Boot Routines return here 322 000 2053 JC ERROR Boot Routines return here 2054 JAP USERFWA  2058 * Device Processors 2059 * Device Processors 2050 * Device Processors 2051 * Device Processors 2052 * Device Processors 2054 * Device Processors 2054 * Device Processors 2055 * Device Processors 2056 * Device Processors 2057 * Device Processors 2058 * Device Processors 2059 * Device Processors 2050 * Device Processors	201 124 044 2051 LHUD USRCKI Clear Time-Out Vector to Just user v 31 312 000 2053 JC ERROR Boot Routinss return here 2053 JC ERROR Boot Routinss return here 2053 +	303 200 042 2053		062 (	010		2050		STA	.MFLAG	original front panel
042 040 040 2052 SHLD UVEC+1 Clear Time-Dut Vector to Just user v 332 322 000 2053 JC ERRUR Boot Routines return here 2054 JMP USERFWA  2058 * Device Processors 2058 * Device Processors 2058 * Device Processors 2059 * Device Processors 2050	303 200 042 2055 SHUD UIVEC+1 Clear Time-Out Western here 323 322 000 2055 JMP USERHA  303 200 042 2055 JMP USERHA  2055 * Device Processors 2056 * Device Processors 2057 ** Device Processors 2058 * Device Processors 2059 * Device Processors 2050	2055 5HD UIVEC+1 Clear Time-Out Wester to Just user v 2054 10 00 2053 303 200 042 2055 303 200 042 2055 2059		052	7 421		2051		LHLD	USRCLK	
332 322 000 2053 JC ERRUR Boot Routines return here 303 200 042 2055 JMP USERFHA  2057 ** Device Processors 2058 * Device Processors 2059 * Device Processors 2050 * Device	332 20 004 2053 JC ERRRK Boot Routines return here 2053	332 200 042 2053 JC ERRUK BOOK ROUTINES return here 2054 4		045 (	040		202		SHLO	UIVEC+1	r Time-Out Vector to just user v
303 200 042 2055 JMP USERFMA  2058 * Device Processors 2069 Bun * Berr 2065 Bun * Berr 2065 Bun * Berr 2065 Bun Berr 2066 Bun Berr 2066 Bun Berr 2067 Bun Berr 2070 Bun	2055 JAP USERFUA  2057 ** Device Processors  2058 ** Device Processors  2058 ** Device Processors  2050 BUDA EQU ** BERN 1 *-BODAZ-CNO.H17  2060 BUDA EQU ** BUDAZ-CNO.H17  2061 BERN 1 *-BODAZ-CNO.H47  143 005 2069 BW BERN 1   BE	205 042 2055 2056 2058 2058 2058 2058 2058 2058 2058 2058		332	322 (		2053		ر د	ERROR	Routines return her
2057 ** Device Processors 2058 * 2058 * 2050 600A EQU * 2060 2061 ERRNZ +-B0DA/2-CND.HI7 2062 ERRNZ +-B0DA/2-CNU.H47 2064 ERRNZ *-B0DA/2-CNU.H47 2064 ERRNZ *-B0DA/2-CNU.H47 2064 ERRNZ *-B0DA/2-CNU.H47 2065 DW BERR IIIEgal Device 143 005 2069 DW BERR 143 005 2069 DW BERR 143 005 2069 DW BERR 2070 2070 BW BERR 2071 HERRNZ *-HSGPRI 2072 ERRNZ *-HSGPRI 2073 HSGLEN EQU *-HSGSEC-HSGLEN 2074 214 215 2075 HSGSEC BR IOIIO008,1U0011008,1U0011018	2057 ** Device Processors 2058 * 2058 * 2059 2060 BODA EUU * 2060 BODA EUU * 2060 BODA EUU * 2061 BERNZ *-BDDA/2-CND.H17 2062 2063 BH BERNZ *-BDDA/2-CND.H17 2065 DW BERNZ *-BDDA/2-CND.H17 20 20 30 33 33 7 2072 NSGERI DB 100110008;100111108;101111118 2074 SOF NSGERI DB 100110008;100011003;100011018 2074 SOF NSGER EUU *-NSGER HOUTIOUS;100011018	2057 ** Device Processors 2058 * 2059 2060 8004 ERRY +=8004/2-CN0.H17 2062 BRNZ +=8004/2-CN0.H17 2064 BRRZ +=8004/2-CN0.H17 2065 BRRZ +=8004/2-CN0.H17 2065 BRRZ +=8004/2-CN0.H17 2066 BRRZ +=8004/2-CN0.H17 2066 BRRZ +=8004/2-CN0.H17 2066 BRRZ +=8004/2-CN0.H17 2066 BRRZ +=8004/2-CN0.H17 2067 BRRZ +=8004/2-CN0.H17 2073 RSCEN EQU ++RSCPRI 2074 RSCEN EQU ++RSCPRI 2077 RSCEN EQU ++RSCREN EQU ++			0	:	2054	:	d K 7	USERFWA	
2057 ** Device Processors 2058 * 2059 ** 2059 2060 BODA EQU * 2061 CRRN2 *-BODA/2-CND.H17 2062 CRS DN BH17 2064 CRSN2 *-BODA/2-CND.H27 2064 CRSN2 *-BODA/2-CND.H47 2065 CRS DN BERR IIIEgal Device 143 005 2068 DN BERR 143 005 2069 DN BERR 143 005 2069 DN BERR 2070 CO71 230 336 337 2072 MSGPRI DB 1001100008,110111108,110111118 2074 214 215 2075 MSGSEC DB 101001008,100011008,100011018	2057 ** Device Processors 2058 * 2058 * 2050 * Con	2057 ** Device Processors 2058 * 2058 * 2058 * 2059 * 2060 800A EQU * 2061 2064 2064		:			:		:		
2059 2060 2060 2061 2062 2063 2064 2064 2064 2064 2064 2064 2064 2065 2066 2066 2066 2066 2066 2066 2067 2068 2068 2068 2068 2069 2070 2070 2071 230 336 337 2072 2073 2074 214 215 2075 2076 2076 2076 2076 2076 2076 2076 2076	2059 600A EQU + 2061 ERRN +B0DAZ2-CNO.H17 2061 ERRN +BDDAZ2-CNO.H47 2063 ERRN +BDDAZ2-CNO.H47 2064 ERRN +BDDAZ2-CNO.H47 2065 ERRN +BDDAZ2-CNO.H47 2065 ERRN +BDDAZ2-CNO.H47 2065 ERRN +BDDAZ2-CNO.H47 2070 BERR 111egal Device 2070 BERR 1101011110B,11011111B 2073 HSGPRI DB 100100B,1D01110B,11011111B 2074 214 215 2075 HSGSEC DB 11010100B,1D011101B 2077 FRSGSEC DB 1101010B,1D011101B 2077 FRSGSEC DB 1101010B,1D011101B 2077 FRSGSEC DB 1101010B,1D011101B	2059 800A E0U * 2061 ERRN 4-B00A/2-CN0.H17 2062 ERRN 7-B00A/2-CN0.H17 2063 ERRN 7-B00A/2-CN0.H47 2064 ERRN 7-B00A/2-CN0.H47 20 336 337 2076 20 336 337 2077 2074 214 215 2075 #SGSEC BB 1001100B,11011110B,11011111B 224 214 215 2075 #SGSEC ERRN 7-MSGSEC-MSGLEN 2070 ERRN 7-MSGSEC-MSGSEC-MSGLEN 2070 ERRN 7-MSGSEC-MSG		:	•		2057	* •	Device		
2062 ERRNZ #-B00A/2-CN0.HI7 2064 DW BH17 2065 DW BERR 143 005 2066 DW BERR 143 005 2069 DW BERR 143 005 2071 230 336 337 2072 MSGPRI DB 100110008,1101111108,110111118 224 214 215 2075 MSGSEC DB 101001008,100011003,100011018 224 214 215 2075 MSGSEC DB 101001008,100011003,100011018	2062 ERRNZ +-BODA/2-CNO.HI7 2064 ERRNZ +-BODA/2-CNO.HI7 2064 ERRNZ +-BODA/2-CNO.H47 2065 DM BERR 143 005 2068 DM BERR 143 005 2069 DM BERR 2072 MSGRI DB 100110008,1101111118 2073 MSGLEN EQU +-MSGRI DOUTIONS,1100111108,110111118 2074 MSGSEC DB 10101008,110011103,110011018 2075 MSGSEC DB 10101008,110011003,110011018 2076 ERRNZ +-MSGSEC-MSGLEN	2061 ERRNZ +-B0DA/Z-CNO.HIZ 2063 D4 BHIZ 2064 D5 BHIZ 2065 D4 BHIZ 2065 D7 BERR 2073 BERR 2073 BERR 2073 BERR 2074 214 215 2076 MSSSEC D8 2076 MSSSEC D8 2076 MSSSEC D8 2076 MSSSEC D8 2077 ASSSEC D8 2077 ASSSEC D8 2078 MSSSEC D8 2078 MSSSEC D8 2079 MSSSEC D8 2070 MSSSEC D8 207	.12	:			2059		EQU		
032 006 2063 0W 8H17 2064 2065 ERRNZ *-BDD6/72-CNU.H47 2065 DW BH47 143 005 2068 DW BERR IIIIEGAI Device 143 005 2069 DW BERR 2070 2071 230 336 337 2072 MSGPRI DB 100110008,110111108,110111118 2073 MSGLEN EQU *-MSGPRI 2074 214 215 2075 MSGSEC DB 101001008,1000110008,100011018	032.006 2064 BH17 2065 2064 D04 BH17 2065 ERRNZ *-BUGA/Z-CKG.H47 2065 D04 BH47 2065 D04 BERR 11169a1 Device 2069 D04 BERR 100110009,11011111111111111111111111111	032 006 2063 DW BH17 2064 2064 ERRNZ *-BUGA72-CKU.H47 2065 DW BH47 152 006 DW BH47 152 006 DW BERR 1116931 Device 2067 DW BERR 1100110009,1101111118 2073 HSGEEN EUU *-HSGRI 100110003,100011018 2074 2075 HSGSEC OB 1001001009,100011003,100011018 2074 2075 HSGSEC OB 2074 4-HSGSEC-HSGLEN EUU 2075 HSGSEC HSGLEN ERRNZ *-HSGSEC-HSGLEN EUU 2075 HSGSEC HSGLEN EUU 2075 HSGSEC HSGLEN ERRNZ *-HSGSEC-HSGLEN EUU 2075 HSGSEC HSGLEN EUU 2075 HSGSEC HSGLEN ERRNZ *-HSGSEC HSGLEN EUU 2075 HSGSEC HSGSEC HSGSEC HSGLEN EUU 2075 HSGSEC HSGSEC HSGLEN EUU 2075 HSGSEC H	:			:	2061 2062		ERRN2	*-800A/2-CN0.H	H17
152 006 2065 ERRNZ *-BUDA/Z-CND.H47  143 005 2066 DW BERR IIIegal Device 143 005 2069 DW BERR  2070 2071  230 336 337 2072 MSGPRI DB 100110008,1101111108,110111118  2074 214 215 2075 MSGSEC DB 101001008,100011003,100011018  244 214 215 2075 MSGSEC DB 101001008,100011003,100011018	152 006 2065 ERRNZ *-BUDK/Z-CND.H47  143 005 2066 DW BH47  143 005 2069 DW BERR  143 005 2069 DW BERR  2070 2070 2071  230 336 337 2072 MSGFRI DB 10011000B,110111110B,11011111B  2074 MSGSEC DB 1010010B,11011110B,11011111B  2075 MSGSEC DB 1010010B,11001101B  2076 MSGSEC DB 1010010B,11001101B  2076 ERRNZ *-MSGSEC-MSGLEM  2076 ERRNZ *-MSGSEC-MSGLEM	152 006 2065 DW 8H47  152 006 2066 DW 8H47  143 005 2069 DW BERR  230 336 337 2073 HSGPRI DB 100110008,110111108,110111118  244 214 215 2075 HSGSEC DB 1010008,100011018  244 214 215 2075 HSGSEC DB 1010001008,100011018  267	:	035	900	:	2063		#.Q	8H17	
143 005 2068 DW BERR IIIIEgal Device 143 005 2069 DW BERR 2070 2070 230 336 337 2072 MSGPRI DB 100110000B,11011110B,11011111B 2073 MSGLEN EQU +-MSGPRI 2074 214 215 2075 MSGSEC DB 10100100B,10001100B,10001101B	143 005 2068 143 005 2068 163 005 2068 163 005 2070 230 336 337 2073 2073	143 005 2068 143 005 2068 163 005 2068 173 005 2070 230 336 337 2077 2071 2072 2073 2073 2074 2074 2074 2075 400 100110008,110111118 2076 810 10110008,110111118 2076 810 10110008,110111118 2076 810 10110008,1101111118 2076 810 10110008,1101111118 2076 810 10110008,1101111118 2076 810 10110008,1101111118 2076 810 10110008,1101111118 810 10110008,110111118 810 10110008,110111118 810 10110008,1100111118 810 10110008,1100111118 810 10110008,1100111118 810 10110008,1100111118 810 10110008,1100111118 810 10110008,110111118 810 10110008,110111118 810 10110008,110111118 810 10110008,110111118 810 10110008,110111118 810 10110008,110111118 810 10110008,110111118 810 10110008,1100111118 810 10110008,110011118 810 10110008,1100111118 810 10110008,1100111118 810 10110000110008,1100111118 810 10110000110008,1100111118 810 101100001100008,1100111118 810 101100001100011	000.000		: 4		2065		ERRNZ	*-BODA/2-CND.H	H+7
143 005 2069	143 005 2068 DN BERK 1116931 DEVICE 2070 2070 2071 2073 336 337 2073 MSGLEN EQU *-MSGRIT 111111111111111111111111111111111111	143 005 2069 DW BERK 143 005 2069 DW BERK 2070 2070 DW BERK 2071 WGPRI DB 100110006,1101111106,110111118 2073 WGGEN EQU +-MSGPRI 2074 215 2075 WGGSC DB 101001006;100011018 2074			: 3: 6	:	2002		5		
2070 2071 230 336 337 2072 MSGPRI DB 100110000B*11011110B*11011111B 2073 MSGLEN EQU *-MSGPRI 2074 215 2075 MSGSEC DB 10100100B*10001100B*10001101B 2076 ERRNZ *-MSGSEC-MSGLEN	2070 2070 2077 42072 HSGPRI DB 10011000B,11011110B,11011111B 2073 HSGLEN #-MSGPRI 2075 HSGSEC DB 10100100B,10001100B,10001101B 2076 HSGSEC BERNZ +-MSGSEC-MSGLEN	2070 2070 2072 2073 2073 2073 2073 2073	05.131 05.133	4 . 4 J .W	005		5069		E O	BERR	
230 336 337 2072 MSGPRI DB 1001100008,110111108,110111118 2073 MSGLEN EQU +-MSGPRI 2074 2075 MSGSEC DB 101001008,100011008,100011018 2076 ERRNZ +-MSGSEC-MSGLEN	230 336 337 2072 MSGPRI DB 100110008,1101111108,110111118 2073 MSGLEN EGU +-MSGPRI 2074 217 215 2075 MSGSEC DB 101001008,100011003,100011018 2076 ERRNZ +-MSGSEC-MSGLEN	230 336 337 2072 MSGPRI DB 100110008,110111118 2073 MSGLEN EGU *-MSGPRI 2074 214 215 2075 MSGSEC DB 101001008,1000110038,100011018 2076 ERRIZ *-MSGSEC-MSGLEN		:	:		2070				
244 214 215 2075 HSGSEC DB 101001008,100011008,100011018 2076 ERRNZ +-MSGSEC-MSGLEN	2074 2074 2076 ERRNZ +-MSGSEG-HSGLEN 2076 ERRNZ +-MSGSEG-HSGLEN	244 214 215 2075 MSGSEC DB 101001006,100011006,100011018 2076 ERRNZ 4-MSGSEC-MSGLEN 2076 ERRNZ 4-MSGSEC-MSGLEN	05.135	230	336	337	2072	HSGPRI	:	100110008,1101	i de
.140 244 214 215 2075 MSGSEC DB 101001008;100011018 .000 244 214 215 2076 ERRNZ 4-MSGSEC-MSGLEN	1000 244 214 215 2075 HSGSEC D8 101001008;100011018 .000	101001008;100011008 0000 244 215 2076 #SGSEC-HSGLEN  0000 244 215 2076 #SGSEC-HSGLEN  0000 1008;100011018	500				2074	2000			
			05.140	244	214	215	2075	MSGSEC	:	101001008,100C *-MSGSEC-MSGLE	
						: :					
						:					
					:	:	:				
				:	•	:	:				
							:				
				:			:				
				:	:	:					
						:					
				:		:					
				:		:					

2082	2002	147 077 016 040 200	2002   2003   2004   2004   2005	072 010	2080	* *	BERR	- Boot Error	/Ram8Go 2/	
012 010 040 2083 ** 012 010 040 2086 GRR LDA .HFLAG Display Update 2086 GRR LDA .HFLAG Display Update 06.00 040 2086 GRR LDA .HFLAG Display Update 06.00 040 2086 CRR LDA .HFLAG Display Update 06.00 040 2088 CRR LAI D.FPLED STA .HFLAG DISPLAY Update 0.01 0.00 040 2089 CRR LAI D.FPLED STA .HFLAG DISPLAY DISPLAY CREATER BERRAL STAR BERRAL DISPLAY COUNT 2091 CALL STAR BERRAL DISPLAY COUNT 2095 BERRI DISPLAY CREATER BERRAL DONE DISPLAYING MESSAGE 2000 2094 LXI BABERRA DISPLAY CALL DONE DISPLAYING MESSAGE 2100 CPT K.STAR CALL DONE DISPLAYING MESSAGE 2100 CPT K.STAR CALL CALL STAR CALL DISPLAYING MESSAGE 2100 DISPLAYING MESSAGE 2	2083	012 010 040 205 96 8 RRR LDA . HFLAG 062 010 040 205 574	012 010 040 2004 SERR LDA . "MFLAG Display Update 062 2004 OR1 UDDOU DIsable DISPLAY UPDATE DISPLAY UPDAT	072 010	2082	•		s boot	0	
072 010 040 2085 BERR LDA .MFLAG 010 010 040 2085 BERR LDA .MFLAG 002 2086 BERR LDA .MFLAG 002 2086 BERR LDA .MFLAG 002 2086 CX	001 210 040 2085 BERR LDA "HELAG Disable Display Update Of 2010 040 2087 STA "HELAG DOIS DISPLAY Update Of 2010 040 2087 STA "HELAG DOIS DISPLAY Update Of 2010 040 2088 LXI BABERRA STA "HELAG STA "HALAG STA "H	001 200 040 2885 868R LDA *#FLAG 002 010 040 2889 868R LDA *#FLAG 002 010 040 2889 868R LDA *#FLAG 001 210 040 2889 868R LDA *#FLAG 021 013 040 2899 LXI	017 010 005 005 005 007 1 164.00 015 010 010 100 007 007 005 000 007 005 005 007 007 0	012 010	2083	•				
001 210 060 2087 STA . HFLAG 001 210 005 2089 LXI	001 210 040 2087 STA .#FLAG 001 210 005 2089 LXI B.BERRA 001 210 005 2089 LXI B.BERRA 015 151 007 2090 LXI B.BERRA 016 2090 LXI B.BERRA 017 2092 BERRI B.BC = Time-Out Count 018 2095 BERRI B.BERRB B.C = Time-Out Count 019 2095 BERRI B.BC B.BERRB B.C = Time-Out Count 019 2095 BERRI B.BERRA 02 2096 BERRI B.BERRA 03 33 360 2109 B.BERRA 04 2105 BERRA B.BT70-SO-S3-S4-S5-S0 05 2105 BERRA B.BT70-SO-S3-S4-S5-S0 06 2105 B.BT70-SO-S3-S4-S5-S0 07 2105 B.BT70-SO-S3-S4-S5-S0 08 3770-SO-S3-S4-S5-S0 08 3770-SO-S3-S4-S5-S0 08 3770-SO-S3-S4-S5-S0 09 3770	001 210 040 2087 STA .#FLAG 001 210 055 2089 LXI	006 200 40 2087 STA .#FLAG 001 200 50 2089 LXI	366 002	2085	BERR	10 N	.#FLAG U0.00U	isable Display Updat	
001 210 005 2089 LXI	001 210 005 2089 LXI	0.1 210 005 2089 LXI	0.01 20.05 2089 LXI 6.9FERAL 0.21 040 2091 LXI 9.5FERAL 0.22 0202 LXI 5.07 2091 CXIL 5.07 2091 0.01 30.00 2093 ERRI LXI 8.8ERRB BC = Time-Out Count 0.01 2090 LXI 8.8ERRB BC = Time-Out Count 1.00 2090 LXI 8.8ERRB BC = Time-Out Count 1.00 2090 LXI 8.8ERRB BC = Time-Out Count 1.01 2090 LXI 8.8ERRB BC = Time-Out Count 1.02 2099 LXI 8.8ERRB BC = Time-Out Count 1.03 2090 LXI 8.8ERRB BC = Time-Out Count 1.04 200 2099 LXI 8.8ERRB BC = Time-Out Count 1.05 2090 LXI 8.8ERRB BC BC = Time-Out Count 1.06 2090 LXI 8.8ERRB BC BC BC = Time-Out Count 1.07 200 200 LXI 8.8ERR BC	062 010	2087		STA	. MFLAG		
315 151 007 2091 CALL ShOV Set up Error Message in LED* 011 2092 DB BERRAL Set up Error Message in LED* 012 2093 LXI B*BERRB BC = Time-Out Count 013 2095 BERRI DCX B 170 2096 MOV A*B 261 2096 JZ ERROR Done displaying message 262 2099 JZ ERROR Cancel was hit ER*STAR A*B 212 322 000 2099 JZ ERROR Cancel was hit ER*STAR A*B 212 322 000 2102 JZ ERROR BERRI Cancel was hit BERRI JZ 2200 2102 JZ ERROR BERRI CANCEL WAS HIT BERRI DB 3770-50-53-54-55 O 2104 BERRI DB 3770-50-53-54-55 O 2105 BERRA DB 3770-50-53-54-55 O 2106 DB 3770-50-53-54-55 O 2107 DB 3770-50-53-54-55 O 2108 DB 3770-50-53-54-55 O 2109 DB 3770-50-53-54-55 O 2110 DB 3770-50-53-54-55 O 2111 DB 3770-50-53-54-55 O 2112 DB 3770-50-53-54-55 O 2114 BERRA EOU DB 3770-50-55-50 E 2115 BERRA EOU DB 3770-50-55-50 E 2116 BERRA EOU DB 3770-50-50-50 E 2117 BERRA EOU DB 3770-50-50-50 E 2118 BERRA EOU DB 3770-50-50-50 E 2118 BERRA EOU DB 3770-50-50-50 E 2119 BERRA EOU DB 3770-50-50-50 E 2110 BERRA EOU DB 3770-50-50-50 E 2111 BERRA EOU DB 3770-50-50 E 2111 BERRA EOU DB 37	315 151 007 2091 CALL \$MOV Set up Error Message in LED* 011 2092	115 1907 2091 CALL SARA Set up Fror Message in LED1 601 000 000 2093 LXI B FERB B BC = Time-Dut Count 013 2092 COUNT	115 1907 2001 CALL Signy Set Up Fror Message in LED1 2002 2004 ERRAL Set Up Fror Message in LED1 2002 2004 ERRAL Set Up Error Message 2003 2004 ERRAL Set Up Error Message 2004 ERRAL Set Up CX Set Up Error Message 2004 ERRAL Set Up CX Set Up Error Message 2004 ERRAL Set Up ERROR Set Up Error Message 2004 ERRAL Set Up ERROR Set Up Error Message 2004 ERRAL Set Up Set	001 210 021 023	2089		LXI LXI	8.8ERRA D.FPLEDS		
001 000 000 2094 LXI B.BERRB BC = Time-Out Count 013 2095 BERRI DCX B BC	001 000 000 2094  LXI B.BERRB BC = Time-Dut Count 103 2095 BERRI DCX 2095 BERRI DCX 2096 BERRI DCX 2097 CX 2097 CX 312 322 000 2098  333 360 2100  376 157 2101  376 157 CX 317 005 2105  386 2106  386 2106  386 2106  387 0-50-53-54-55-56  387 0-50-53-54-55-56  388 0 3770-50-53-54-55  389 0 3770-50-53-54-55  380 0 3770-50-53-54-55  380 0 3770-50-53-54-55  380 0 3770-50-53-54-55  380 0 3770-50-53-54-55  380 0 3770-50-53-54-55  380 0 3770-50-53-54-55  380 0 3770-50-53-54-55  380 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	001 000 000 2093 013 200 2094 EKRI DCX	001 000 000 2093	315 151	2091		CALL	SHOV	et up Error Message in LED*	
013 2095 BERRI DCX 6	013 2095 BERRI DCX	013 020 0094 BERRI DCX	013 020 0094 8ERR1 0CX		2093		3	OCKA PL		
170 2096 NOV A,B 261 2097 DEKKI NOV A,B 261 2097 DEKKI NOV A,B 312 322 000 2098 JZ ERROR Done displaying message 2099 IN IP.PAD	170 2096 NOV A,B 261 2097 DEKKI ORA C 312 322 000 2096 JZ ERROR Done displaying message 209 JZ ERROR Done displaying message 333 360 2100 IN IP.PAD 312 322 000 2102 JZ ERROR 312 322 000 2102 JZ ERROR 303 170 005 2103 JMP BERRA DB 3770-50-53-54-55 0 306 2106 DB 3770-50-53-54-55 0 306 2106 DB 3770-50-53-54-55 0 307 2109 DB 3770-50-53-54-55 0 317 2109 DB 3770-50-53-54-55 0 318 211 DB 3770-50-53-54-55 0 319 3770-50-53-54-55 0 310 0B 3770-50-53-54-55 0 311 DB 3770-50-55 0 31 DB 3770-50-55 0 31 DB 3770-50-55 0 31 DB	201 202 000 000 000 000 000 000 000 000	170 2097 BERRY DONE displaying message 261 2097 0008	000 000	5000	9.50	CXI	696ERRS	C = Time-Out	
261 2097 ORA C ERROR Done displaying message 312 322 000 2098 JZ ERROR Done displaying message 2099 IN IP-PAD	261 2097 ORA C ERROR Done displaying message 2099 JZ ERROR Done displaying message 2099 LO	2697 0RA C 2097 2097 2097 2099 12 12 322 000 2099 12 2099 12 2099 12 2099 18 1P-PAD 2099 2099 18 2099 18 2099 18 2099 18 2099 18 2099 2099 2099 2099 2099 2099 2099 209	2697	170	2096	0 X	X O K	æ• •		
312 322 000 2098 JZ ERROR Done displaying message 2099 IN IP.PAD	312 322 000 2098 JZ ERROR Done displaying message 2099 IN IP-PAD	312 322 000 2098 JZ ERRUR Done displaying message 2099 2099 IN IP-PAD 4.510 2099 IN IP-PAD 4.5210 JZ I IN IP-PAD 4.5210 JZ I I III IIII IIII IIII IIII IIII II	312 322 000 2098 JZ ERROR Done displaying message 209 209 IN IP-PAD 4 200 2090 IN IP-PAD 4 200 2000 2002 2002 JD 2 2	197	2007		ORA	C		
333 360 2009 376 157 2101 CPI K.STAR * Cancel was hit 303 170 005 2103 JHP BERRI Cancel was hit 303 170 005 2103 JHP BERRI 206 2105 BERRA 0B 3770-50-53-54-55 306 2106 0B 3770-50-53-54-55 306 2107 0B 3770-50-53-54-55 307 2109 0B 3770-50-52-53 377 2109 0B 3770-50-51-54-55-56 377 2110 0B 3770-50-51-54-55-56 377 2110 0B 3770-50-51-54-55-56 336 2112 0B 3770-50-55 336 2113 BERRAL EQU #-BERRA 2115 0B 3770-50-55 316 2115 0B 3770-50-55 3170-50-5	333 340 2009 376 157 2101 CP1 K.STAR * Cancel was hit 2 322 000 2102 JZ ERROR Cancel was hit 2 322 000 2102 JZ ERROR Cancel was hit 2 32 2103 JMP BERRI Cancel was hit 2 304 2104 DB 3770-50-53-54-55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	333 360 2100 IN IP-PAD 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	333 360 2100 IN IP-PAD 4 376 2101 CP1 ERRING Cancel was hit 202 2102 JHP BERRI Cancel was hit 203 170 005 2103 JHP BERRI Cancel was hit 203 170 005 2105 2104 SERRA DB 3770-50-53-55-55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	312 325	2098		75	ERROR	one displaying messag	
376 157 2101 CP1 K.STAR * Cancel was hit 12 322 000 2102 JZ ERROR Cancel was hit 2 303 170 005 2103 JHP BERRI Cancel was hit 206 2103 JHP BERRI CANCEL WAS HIT 2104 DB 3770-S0-S3-S4-S5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	376 157 2101 CP1 K.STAR * Cancel was hit 312 322 000 2102 JZ ERROR Cancel was hit 212 322 000 2102 JAP BERRI Cancel was hit 2104 005 2104 08 3770-S0-S3-S4-S5 0 0 0 08 3770-S0-S3-S4-S5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	312 322 000 2101	316 157 2101 CP1 K.STAR TO THE STAR TO THE	333 360	2100		Z	I P. PAO		
312 322 000 2102 JZ ERROR Cancel was hit 303 170 005 2103 JHP BERRI Cancel was hit 2104 2104 BERRA 08 3770-50-53-54-55 0 0 0 0 3770-50-53-54-55 0 0 0 0 3770-50-53-54-55 0 0 0 0 3770-50-53-54-55 0 0 0 0 3770-50-53-54-55 0 0 0 0 3770-50-53-54-55 0 0 0 0 3770-50-52-53 0 0 0 0 3770-50-52-53 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	312 322 000 2102 JZ ERROR Cancel was hit 303 170 005 2103 JHP BERRI 6ERRI 62104	312 322 000 2102 JZ ERROR Cancel was hit 303 170 005 2103 JHP BERRI Cancel was hit 2104 005 2104 0 B 3770-50-53-54-55 0 C C C C C C C C C C C C C C C C C C	312 322 000 2102 JZ ERROR Cancel was hit 2303 170 005 2104 JHP BERRI Cancel was hit 206 2104 2104 0 B 3770-50-53-54-55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	376	2101		CP I	K.STAR		
303 170 005 2103 JHP BERRI 206 2105 BERRA DB 3770-S0-S3-S4-S5-S6 b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	303 170 005 2103 JHP BERRI 206 2105 BERRA DB 3770-S0-S3-S4-S5-S6 b 306 2106 0B 3770-S0-S3-S4-S5 o 306 2107 0B 3770-S0-S2-S3 t 307 2108 0B 3770-S0-S2-S3 t 377 2109 0B 3770-S0-S2-S3 t 214 2111 0B 3770-S0-S1-S4-S5-S6 E 214 2112 0B 3770-S0-S5 r 2114 BERRAL EQU #-BERRA 2118 0B 3770-S0-S5 r 2115 BERRA EQU U	303 170 005 2103 JMP 6ERR1 2104 2105 8ERRA D8 3770-50-53-54-55-50 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	303 170 005 2103 JMP BERRI 206 2105 BERRA DB 3770-50-53-54-55-56 D 2104 DB 3770-50-53-54-55 D 2 D 2105 DB 3770-50-53-54-55 D 2 D 2105 DB 3770-50-53-54-55 D 2 D 2 D 2 D 2 D 2 D 2 D 2 D 2 D 2 D	312 322	2102		75	ERROR	Mas hi	
206 2105 BERRA 08 3770-S0-53-54-55-56 b 2106 BERRA 08 3770-S0-53-54-55 0 0 0 3770-S0-53-54-55 0 0 0 3770-S0-53-54-55 0 0 0 3770-S0-53-54-55 0 0 0 3770-S0-52-53 0 0 0 3770-S0-52-53 0 0 0 3770-S0-52-53 0 0 0 0 3770-S0-52-53 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	206 2105 BERRA 08 3770-S0-S3-S4-S5-S6 b 306 2106 08 3770-S0-S3-S4-S5 0 0 3770-S0-S3-S4-S5 0 0 3770-S0-S3-S4-S5 0 0 3770-S0-S3-S4-S5 0 0 3770-S0-S2-S3	206 2105 8ERRA 08 3770-50-53-54-55 0 0 370-50-53-54-55 0 0 370-50-53-54-55 0 0 0 370-50-53-54-55 0 0 0 370-50-53-54-55 0 0 0 3770-50-52-53 0 0 0 3770-50-52-53 0 0 0 3770-50-52-53 0 0 0 0 3770-50-52-55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	206 2105 868RA 08 3779-50-53-54-55 0 0 306 2107 08 3779-50-53-54-55 0 0 3170-50-53-54-55 0 0 0 3770-50-53-54-55 0 0 0 3770-50-53-54-55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	303 170	2103		d E 7	BERRI		
306 2106 08 3770-S0-S3-S4-S5 0 0 370-S0-S3-S4-S5 0 0 3 3770-S0-S3-S4-S5 0 0 3 3770-S0-S3-S4-S5 0 0 3 3770-S0-S3-S4-S5 0 0 3 3770-S0-S2-S3	306 2106 08 3774-S0-S3-S4-S5 0 0 3774-S0-S3-S4-S5 0 0 0 3774-S0-S3-S4-S5 0 0 3774-S0-S2-S3 5 0 0 3774-S0-S2-S3 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	306 2106 08 3770-50-53-54-55 0 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0	306 2106 08 3770-50-53-54-55 0 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0	:	2105	BER	9.0	3770-50-53-54		
340 2107 08 3770-50-53-54-55 0 342 2108 08 3770-50-52-53 377 2109 08 3770-50-51-54-55-56 214 2112 08 3770-50-55 336 2113 08 3770-50-55 214 BERRAL EQU +-BERRA 2115 BERRAL EQU 0 Time-Out Count (Full 4r	340 2.107 08 3770-50-53-54-55 0 342 2108 08 3770-50-52-53	340	340		2106		98	3770-50-53-54		
377 2109 08 3770 214 2110 08 3770–50–51-54-55-56 E 214 2112 08 3770–50–55 F 336 2113 08 3770–50–55 F 2114 BERRAL EQU +-BERRA 2115 BERRA EQU 0 Time-Out Count (Full #F	377 2109 08 3770 22 21 21 0 08 3770 22 21 21 0 08 3770 21 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	377 2109 08 3770 5110 08 3770 5110 08 3770 5110 08 3770 5110 08 3770 5110 08 3770 5110 08 3770 5110 08 3770 5110 08 3770 5110 5110 08 3770 5110 5110 5110 5110 5110 5110 5110 5	377 2109 08 3770 2110 08 3770 20-51-54-55-56 E 2110 08 3770-50-55 2110 08 3770-50-55 2110 08 3770-50-55 2110 08 3770-50-55 2110 08 3770-50-55 2110 08 3770-50-55 2110 08 2110 0 Time-Out Count (Full Mr		2107		8 O	3770-50-53-54 3770-50-53-53		
377 2110 08 3770 214 2111 08 3770-50-51-54-55-56 E 336 2113 08 3770-50-55 336 2113 08 3770-50-55 2114 BERRAL EQU *-BERRA 2115 BERRA EQU 0 Time-Out Count (Full Ar	377 2110 08 3770 214 2111 08 3770-50-51-54-55-56 E 336 2112 08 3770-50-55 336 2113 08 3770-50-55 2114 BERRAL EUU +-BERRA 2115 2116 BERRB EQU U	377 2110 08 3770 214 2111 08 3770-50-55 2112 08 3770-50-55 336 2113 08 3770-50-55 2114 BERRAL E0U *-BERRA 2115 BERRA E0U 0 Time-Uut Count (Full #r	377 2110 08 3770 214 2111 08 3770-50-51 336 2113 08 3770-50-55 2114 BERRAL EOU *-BERRA 2115 BERRA EOU 0 Time-Out Count (Full #7	:	2109		0.0	3770		
214 2111 08 3770-50-51-54-55-56 E 334 2112 08 3770-50-55	214 2111 DB 3770-S0-S1-S4-S5-S6 E 336 2112 DB 3770-S0-S5 2112 DB 3770-S0-S5 2114 BERRAL EUU +-BERRA 2115 BERRA 2115 BERRA 62116 BERRB EQU U Time-Uut Count (Full Mr	214 2111 08 3770-S0-S1-S4-S5-S6 E 336 2112 08 3770-S0-S5 4-S5-S6 F r r 2113 08 3770-S0-S5 5-S5-S6 2114 8ERRAL EQU +-8ERRA 2115 315 310 0 Time-Out Count (Full #r 2116 315 316 3170-S0-S5 5-S5-S6 F r r r r r r r r r r r r r r r r r r	214 2111 08 3770-50-51-54-55-56 E 2113 08 3770-50-55	:	2110		90	3770		
336 2113 08 3770-50-55 336 2114 BERRAL EOU *-BERRA 2115 2116 BERRB EOU 0 Time-Out Count (Full Mr	336 2113 08 3770-50-55 F C 2114 BERRAL EUU +-BERRA 2115 2116 BERRB EOU 0 0 Time-Uut Count (Full Mr	336 2113 08 3770-50-55 T T 2114 BERRAL EUU +-BERRA S T 2115 BERRA E EUU 0 BERRA E EUU (Full Mr 2115 BERRA E EU 0 BERRA E EU 0 Time-Dut Count (Full Mr 2115 BERRA E EU 0 BERRA	336 2113 08 3770-50-55 2114 BERRAL EUU +-BERRA 5110 2116 BERRA EUU U Time-Uut Count (Full 47		2111		90	3770-S0-S1-S4	55-56	
2114 BERRAL EQU *-BERRA 2115 2116 BERRB EQU 0 Time-Dut Count (Full Ar	ZII BERRAL EUU +-BERRA ZII SII BERRB EUU U Time-Uut Count (Full Mr	ZII BERRAL EUU +-BERRA ZII BERRA EUU U Time-Uut Count (Full #r	ZII6 BERRB EQU U Time-Dut Count (Full or		2117		8 6	3770-55		
2115 2116 BERRB EOU O Time-Out Count (Full #r	2115 2116 BERRB EQU O Time-Out Count (Full Ar	2115 BERRB EQU O Time-Out Count (Full #r	2115 BERRB EOU O Time-Out Count (Full at		2114	BER	EOU	#-BERRA	-	
.000 Z116 BERRB EQU O Time-Out Count (Full Ar	.000 Z116 BERRB EOU O Time-Out Count (Full dr	ONO Z116 BERRB EOU O Time-Out Count (Full Ar	ONO ZII6 BERRB EOU O Time-Out Count (Full Ar		2115					
				8	2116	BERR	EOU	3	ime-Dut Count (Full Ar	
							:			
:	:									
: :	: :									
	: :									
	: : :									
							:			
							:			

			2120		CLKINT	- Clock Inter	rupt Processor
:			2121		CLKINT	processes the	clock interrupts by:
			2123			tor	abort
			2124	• •		Checking for	Time-Dut
			2126	* *			
:			7717	• *	20 010	ock routine is	s only to be used at boot
:			2129				
	65		2130	CLKINI	PUSH	P S W	
222	333 360		2132		Z	IP.PAD	
:	76 157		2133		CP I	K.STAR	
		000	2134		75	ERROR	Cancel is down, so abort the 800T
_	7.5	033 040	2136		LDA	TICCNT	
234 2	4:		2137		V V V	<b>Y</b>	
_	70		2139		745	CK13	Not time to increment internal timer
:	372 122	041	2140		LDA	TIMEOUT	
<b>~</b> :	074		2141		INR	<	
+ 1	-	041	2142		STA	TIMEOUT	
:	e :		2143		٠. د	30	
-	36 331	200	2145		3	CKIS	Not the end yet
			2146		Time-0	-Out Error	
:	121 210		2148		LDA		A = Boot device flag
005.257	376 000	000	2149		CP I	CND. H17	
	302 305		2150		ZNS	CKII	Not an HI7
	:		2152	•	Abort	Н17	
			2153				
265	25 <i>1</i> 062 243	040	2155		STA	D.DLYMO	
	372 242	040	2156		LOA	D. DVCTL	A - Device Control
:	346 200		2157		INA	DF.WR	Remove all but Ram/Arite
	062 242	040	2158		STA	D. DVCTL	
	363 17	500	2159		100	OF DC	ILFN OF MOTOR
		3	2161			7117	
			2162	*	Abort	H47	
:	100 766	:	2163			¥ 477 (07.0	
307	302 31	000	2165		7 2	CK 12	
:	315 033	007	2166		CALL	080	Reset H47
	002 000		2167		0.8	M.RES,D.STAI	
			2168		ERRNZ	CK I 2-*	
			2170		Restor	e User Clock Vector	ector
:			2171				
317	052 124	6 6 6	2172	CKIZ	LHLD	USRCLK UIVEC+1	Restore User Clock
:							

005-326 3D 143 005 2D 5 005-332 345 2D 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7							
005,333 - 345 - 2176 - 040	:	303	143 005		JHP	BERR	Goot Time-Out Error
005-337 31 21 21 21 21 21 21 21 21 21 21 21 21 21	- :	361			POP	PSH	
005-331 314 2140 XRH  005-331 314 2140 XRH		345	124	2178	PUSH	H USRCLK	
		343 311		2180	XTHL RET		s Clock
		:					
	:	:	:				
		:					
	:						
				:			
		:					
		:					
		:					
		:					
	:						
	:						
	:						
		:					
		:					

	2184	* *	H89 COM/DAT	/DAT		:	:	/Ram8Go	77	:
	2186 2187 2188		H89COM a	nd H890AT HTR-89 and	are provided a RAMBGU.	S COMBON	entry points			:
006.023	2189	•	SET	6023A						
000.063	:		ERRHI OS	* *						
006.023 303 361 006		H890AT	<b>JHP</b>	DAT.	:					:
006.027	2195	•	SET	6027A	:					:
006.026	: :	H89COM	OS	*100						
	:									:
										: · · · · · · · · · · · · · · · · · · ·
										:
	:		:				:	:		:
							:	:		:
	:						:			:
	:				:					:
										. :
	:				:					
	:									:
:										
	:		:	:	: :					:
	:		:							
		:	:				:			

203 * BH17 boc 204 * CALC 205 * CALC 207 * CALC 209 BH17 LXI 211 LXI 212 BH17 LXI 212 BH17 LXI 213 BH17 CALC 222 BH17 CALC 223 BH17 CALC 223 BH17 CALC 223 BH17 CALC 223 BH17 CALC 224 CALC 222 CA	2208 * 2208 2208 2209 2212 2212 2213 2214 2215 2215 2217 2218 2220 2220 2220 2220 2220 2220 2220	8H170 8H171 8H172 8H173 8H174
***************************************	* * * * * * * * * * * * * * * * * * *	# # # # # # # # # # # # # # # # # # #
2003 2003 2003 2003 2003 2003 2003 2003		
, e.e. e.e. e.e. e.e. e.e. e.e. e.e. e.		

000.004 2257 BH17AL	,,63	1741	EQU		 
	1677	1		*-8H17A	
	:				
	:				
	:				
	:				

/Ram8Go 2/	•																						<b>^</b> pe			COBBAD			
	pecified unit of an H-47 disk. The specified in AIO.UNI.	= Unit of H-47 to boot.			Hove in H-47 Hessage		NO errors at boot	Hait 1/2 Second	Errors, so try again						Try again			L - Ready Bits					Specified Unit is not ready		4 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	DECEMBER SECTOR COURT COMME	0.000	111111111111111111111111111111111111111	
- 800t H47	boots the speci	:		B,8H47A	SHOV	BHTCAL	J.	A,500/2	BH471			101	080. W.RES.D.STAT	NON		A	X C	RRDY		AIO.UNI B.A	× ×	8118	L 8H473	, t		DD • L SC	DAT		
8H47	8H47	ENTRY:		ĽŽ	CALL	3	N C	IAW	JAP L	BH472			CALL	CALL	2 3		۲ ۲ ۷	CALL	သ	4 A	XRA	CALL	4 X X Y	Boot	145	08	2	0.8	SC
*.			*	BH47		17444				*	*		84472		•		84473							*					
2260	2262	2265	2267	2269	2271	2273	2275	2277	2279	2281	2282	2284	2286	2288	2290	2292	2294	2295	2296	2298	2300	2301	2302	2304	2306	2308	2309	2311	2312
				900					000				033 007	7 007			134 000	315 134 007		-		304 007	1 006				5 006	}	
		:		01 324	315 15		320	076 372	03 16	:			315 03 002 00	315 16	ੜ ਵ		330	15.13	<u>.</u>	072 061	57	15.30	302 211		15 23	003	330		30
					:		67 3	:							:			:							:		:	:	
				96.1	006-160	7	006.16	006-170	06.1	:			006.200	06.2	). 		006.214	006-215	7 90	006.221	06.2	96.2	006.232			006.240	006.241	06.2	06.2

006-272 002 2313 CALL DAT Transfer count of 2 006-272 002 2314 CALL DAT DE COMPANY 2 006-272 002 2315 CALL DAT DE COMPANY 2 006-272 002 2315 CALL DAT DE CALL DAT	115 355 006 2313 CALL DAT Transfer count of 2 315 107 007 2315 RG 315 107 007 2315 RG 315 107 008 2316 CALL DAT 315 300 008 2316 CALL DAT 315 300 008 2316 CALL DAT 316 300 2316 CALL DAT 317 325 DB 318 300 008 2318 CALL DAT 318 300 008 2319 CALL DAT 319 310 008 2310 CALL DAT 310 008 2310 CALL DAT 311 008 2310 CALL DAT 312 311 008 2310 CALL DAT 313 009 2311 DB 314 008 2312 CALL DAT 315 310 008 2313 CALL DAT 316 009 2313 CALL DAT 317 009 2314 DB 318 008 2315 CALL DAT 319 009 2315 CALL DAT 310 009 2315 CALL DAT 311 008 2315 CALL DAT 312 009 2315 CALL DAT 313 009 2315 CALL DAT 314 008 2315 CALL DAT 315 316 006 2315 CALL DAT 316 009 2315 CALL DAT 317 009 2315 CALL DAT 318 009 2315 CALL DAT 319 009 2315 CALL DAT 310 009 2315 CALL DAT 311 000 2315 CALL DAT 312 000 000 2315 CALL DAT 313 000 000 000 000 000 000 000 000 000	115 555 006 2313 CALL DAT Transfer count of 2  007 2315 CALL DAT  115 167 0		355				DAT 2	•
900 002 231	900 2 2314	902 231			2313	J	ALL	2	er count of
319 67 05 215 RCL 319 147 067 2116 CALL 310 147 067 2116 CALL 310 147 067 2117 CCL 311 060 2119 CALL 312 060 1011 212 CALL 313 060 101 212 CALL 314 060 101 212 CALL 315 061 061 212 CALL 316 061 061 212 CALL 317 212 CALL 318 060 071 212 CALL 319 061 061 212 CALL 310 062 213 CALL 310 062 213 CALL 311 060 213 CALL 312 060 071 213 CALL 313 071 060 213 CALL 314 061 071 212 CALL 315 061 061 213 CALL 316 060 071 213 CALL 317 071 071 071 071 071 071 071 071 071 0	315 167 007 2315 RC	315 67 007 2315 RCL			2314	•			
315 167 007 2316 CALL WON Try solin 315 330 006 2318 CALL COM CALL CALL COM CALL CALL CALL CALL CALL CALL CALL CAL	315 167 007 2316 CALL NON Try again 315 310 006 2319 CALL CON Durput Read Command 316 322 RG CALL CON Durput Read Command 317 222 RG CALL CON Durput Read Command 320 007 2322 RG CALL CON DURPUT Read Command 320 007 2322 RG CALL DAT Track Mumber 320 007 2324 RRC CALL DAT Track Mumber 321 007 2324 RRC CALL DAT Track Mumber 322 RR CALL DAT Track Mumber 323 RR CALL DAT Track Mumber 324 RR CALL DAT Track Mumber 325 RR CALL DAT Track Mumber 326 007 2334 RRC CALL DAT Track Mumber 327 2334 RRC CALL DAT Track Mumber 328 RRC CALL DAT Track Mumber 329 RR CALL DAT Track Mumber 320 007 2334 RRC CALL PAT TRACK CALL PAT TRACK TRACK TRACK TALL PAT TRACK MUMBER 327 2334 RRC CALL PAT TRACK TALL PAT TRACK MUMBER 330 110 006 2334 GH474 CALL PAT TRACK TALL PAT TRACK MUMBER 331 10 006 2334 GH474 CALL PAT TRACK TALL PAT TRACK MUMBER 332 15 006 007 2335 GH474 CALL PAT TRACK TALL PAT TRACK MUMBER 333 11 006 2334 GH474 CALL PAT TRACK TALL PAT TRACK MUMBER 334 11 006 2334 GH477 CALL PAT TRACK TALL PAT TRACK MUMBER 34 11 006 2334 GH477 CALL PAT TRACK MUMBER 35 12 10 10 10 10 10 10 10 10 10 10 10 10 10	315 167 007 2316 CALL NOW Try spain 319 310 006 2318 CALL COM Dutput Read Command 310 006 2321 CALL COM DO. REAG 319 320 006 2322 CALL DAT Track Number 310 325 066 2322 CALL DAT Track Number 311 006 2322 CALL DAT TRACK Number 312 021 022 CALL DAT TRACK Number 313 021 022 CALL DAT TRACK Number 314 022 CALL DAT TRACK NUMBER 315 021 022 CALL DAT TRACK NUMBER 316 021 2320 CALL DAT TRACK NUMBER 317 021 021 021 021 021 021 021 021 021 021			2315	∝	ی		
115 330 006 2518  115 330 006 2518  116 357 066 2522  117 061 011 222  118 357 062 222  118 357 062 222  118 358 062 222  118 358 062 222  119 358 062 222  110 062 223  110 063 063 063 063 063 063 063 063 063 06	115 310 006 2118  217 007 2118  218 006 2119  219 006 2119  210 007 222  221 RC  222 RR  222 CALL  222 RR  222 RR  222 RR  223 RR  224 RC  224 RR  225 RR  227 RR  228 RR  227 RR  228 RR  228 RR  228 RR  229 RR  220 001 200 002 2332  231 006 2332  232 RR  233 RP  233 RP  234 RP  235 RR  234 RP  235 RR  234 RP  235 RR  234 RP  235 RR  237 CALL  237 RR  238 RR  239 RR  231 006 233 RP  231 006 233 RP  232 RR  234 RP  235 RR  234 RP  235 RR  234 RP  237 RR  234 RP  237 RR  234 RR  235 RR  236 RR  237 RR  237 RR  238 RR  248 R	115 330 006 2316  216 007  217 006 2317  218 007  219 007		-	2316	ه د	ALL O	Z	
315 330 006 5319 CALL COM Dutput Read Command 315 355 006 2320 006 322 061 322 062 323 062 323 063 324 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 323 064 325 065 065 323 064 325 065 065 323 064 325 065 065 065 065 325 065 065 065 325 065 065 065 325 065 065 065 325 065 065 065 325 065 065 065 325 065 325 065 065 325 065	315 330 006 5310 CALL COM Dutput Read Command 2007 2332 BG D0.REAB Track Number 000 2323 BG D0.REAB 000 2323 BG D0.REAB 000 2323 BG D0.REAB 000 000 000 000 000 000 000 000 000 0	315 330 006 519 CALL COM Butput Read Command 315 350 006 5210 CALL CAT Track Number 322 CALL CAT			7167	¥:	: د		iry again
13   15   50   52   52   52   52   52   52   5	15   25   25   25   25   25   25   25	330 2221 RC 00.4646 00.0.4646 00.0.4646 00.0.4646 00.0.4646 00.0.4646 00.0.4646 00.0.4646 00.0.4646 00.0.4646 00.0.4646 00.0.222 00.0.222 00.0.41 00.0	: •	330	2310	ن	-	¥00	700
315 355 006 2332 RC CALC DAT Track Number 1  2000 007 2332 006 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	330 2321 RC 0AL 0AT Track Number 000 2322 0BD 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	315 355 506 2222 CALL DAT Track Number 0000 2222 CALL DAT TRACK Number 0000 2222 CALL DAT ALGUMI 017 2225 RRAZ SCENOLILILB STATE SCENOLILILB SCENOLILILB STATE SCENOLILILB STATE SCENOLILILB STATE SCENOLILILB STATE SCENOLILILB STATE SCENOLILILB SCENOLILILILILILILILILILILILILILILILILILILI		) : ) : ) :	2320			DO REAB	
315 555 006 2332 CALL DAT Track Number 2332 CALL DAT 100000 B 130 CALL DAT 10000 B 131	115 355 006 2322	315 555 006 2322 CALL DAT Track Number 2330 DB CALL DAT	•		2321	~	د د		
1900   2323   198	000 2323 DB 0 0 017 2324 RC AIO.UNI 017 2326 RRC RRC C C C C C C C C C C C C C C C	000 2332		355	2322	<u>ي</u>	ALL	DAT	Track Number
330 0 01 2325	930 01 2324 RC AIO.UNI 017 2325 LDA AIO.UNI 017 2325 RRC RRC C 232 RRC C 232 RRC C 232 RRC C 233 C RRC C 233 C RRC C 233 C RRC C 233 C RC C C C C C C C C C C C C C C C C	324   RC			2323	<u> </u>	8	0	
017 2325 LDA A10.UMI 017 2327 RRC 017 2326 RRC 017 2326 RRC 018 RR 2 019 2329 RRC 020 2331 CALL DAT. 0315 060 042 2339 RCC 021 200 042 2339 BH474 CALL DAT. 022 167 007 2339 BH474 CALL DAT. 022 167 007 2339 BH474 CALL DAT. 023 110 060 2331 RKC 023 110 060 2331 RKC 023 110 060 2345 BH474 CCC 03 111 060 077 2345 BH474 CCC 040 110 070 110 070 RCCC 050 110 070 110 070 RCCCC 050 110 070 110 070 RCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	017 2325 LDA A10.UNI 017 2326 RRC 017 2326 RRC 017 2328 RRC 018 2330 BRRX UNI.H-011000008 Start at sector 1 023 ERRX SECH-000111118 023 ERRX SECH-000111118 024 2334 CALL DAT. 025 2334 BAYA CALL PIN Pre-Mature DONE means 025 233 BAYA CALL PIN BAYA BAYA BAYA BAYA BAYA BAYA BAYA BAY	017 2325 RRC 017 2327 RRC 017 2329 RRC 018 2339 BR 2 CH-000111118 340 2331 GRR 2 SC,H-000111118 350 006 2333 RCL 050 2333 RCL 051 260 007 2333 RCL 052 2336 BH474 CAL 053 117 006 2336 BH474 CAL 053 117 006 2336 BH474 BB 3776-52-53-55 GRI another byte 053 117 006 2339 BH474 BB 3776-52-53-55 GRI another byte 053 117 006 2339 BH474 BB 3776-52-53-55 GRI another byte 053 117 006 2349 BH474 EUU +BH474 EUU +BH	:	:	2324		ن		
017 2326 RRC 017 2328 RRC 017 2328 RRC 12329 ERRN UNITH-01100000B 366 001 2331 ERRN SEC, P-00011111B 315 361 006 2332 CALL DAT. 315 365 007 2333 LCL DAT. 315 365 007 2337 LXL DATEFRA 315 066 007 2336 BH474 CALL DAT. 315 066 007 2337 LXL DATEFRA 317 066 2340 BH474 CALL DATEFRA 318 066 007 2336 BH474 CALL DATEFRA 318 066 007 2336 BH474 CALL DATEFRA 318 066 007 2336 BH474 DB BH474 CALL DATEFRA 318 066 007 2336 BH474 DB BH474 CALL DATEFRA 318 066 007 2336 BH474 CALL DATEFRA 319 066 007 2336 BH474 CALL DATEFRA 310 066 2340 BH474 CALL DATEFRA 311 066 2340 BH474 CALL DATEFRA 312 2346 BH474 EUU +-BH477 CALL TATEFRA 314 066 007 2336 BH474 CALL DATEFRA 317 067 0736 CALL DATEFRA 318 066 007 2336 BH474 CALL DATEFRA 318 066 007 233	017 2326 RRC RRC COLT 2326 RRC RRC COLT 2320 RRC RRC COLT 2320 GRIN 1 1 1 2320 GRIN 1 2320 GRIN 1 1 2320 GRIN 2330 GRIN 200 GRIN 2330 GRIN 2330 GRIN 200 GRIN 2330 GRIN 2330 GRIN 200 GRIN 2330 GRIN 200 GRIN 2330 GRIN 2330 GRIN 200 GRIN 2330 GRIN 200 GRIN 2330 GRIN 233	017 2326 RRC 017 2328 RRC 017 2339 CRL 017 2331 CRL 017 2331 CRL 017 2333 CRL 017 017 017 017 017 017 017 017 017 017		190	2325	_	Vα	AIO.UNI	
017 2227 RRC 107 2229 RRNZ UNI, M-011000008 366 001 2330 ERRNZ UNI, M-011000008 360 2331 ERRNZ SEC, M-000111118 SEC, FG CC	017 2327 RRC 346 001 2330 GRR I I -01100000 Start at sector I 232 GRR I 233 GR I -0 111118 GR I -0 12 12 12 12 12 12 12 12 12 12 12 12 12	017 2327 RRC 2329 ERRAZ UNITH-01100000 36 70 2330 ERRAZ SECH-00011111B Start at sector 1 233 ERRAZ LAI DAY.  234 CALL DAY.  235 CALL DAY.  237 CALL DAY.  237 CALL DAY.  238 CALL DAY.  239 BH474 CALL DAY.  231 066 2339 IN D D BH474 Cet another byte  232 233 BH474 DB 3770-82-83-85-86  331 066 2349 DB 3770-82-83-81 177  2347 DB 3770-82-83-81 177  2348 DB 4770-82-83-81 177  2348 DB 4770-84-84-87  351 2346 BH474 EGU 4-BH477	:		2326	• • • • • • • • • • • • • • • • • • •	RC		
115 366 001 2328 RERAZ UNI. P-011000008  366 001 2330 0042  315 366 007 2335 RALL DAY. CALL CALL CALL CALL CALL CALL CALL CAL	115 361 006 2338	315 365 001 2328 RERNZ UNI. H-011000008 366 001 2330 0RT 1			2327		RC		
2329 ERRNZ UNI.M-OILD00000 Year at Sector 1 2330 ERRNZ SEC.M-O00111118 Start at Sector 1 2331 ERRNZ SEC.M-O00111118 Start at Sector 1 2332 ERRNZ SEC.M-O00111118 Start at Sector 1 2334 CALL DAY. 2335 CALL DAY. 2335 CALL DAY. 2336 CALL DAY. 247 CALL DAY. 2534 CALL DAY. 2545 CALL DAY. 2546 CALL DAY. 2547 CALL DAY. 2547 CALL DAY. 2547 CALL DAY. 2547 CALL DAY. 2548 CALL DAY. 2548 CALL DAY. 2548 CALL CALL CALL CALL CALL CALL CALL CAL	346 001 2329 ERRNZ 10.1.100000B Start at sector 1 233	366 001 2339	:		2328		RC		
315 366 001 2330 0011 1118	315 361 006 2332	346 001 2330 001 2330 001 2330 001 2330 001 2330 001 2331 001 2331 001 2333 6.2 CALC DAT.  315 366 007 2336 84474 CALC PIN Pre-Hature DOME means end, error set if 02 2339 1 0 0 0 2339 1 0 0 0 2339 1 0 0 0 2339 1 0 0 0 2339 1 0 0 0 2339 1 0 0 0 2339 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			2329	w	RRNZ	UNT.M-011000008	
115 361 066 2331 ERRAZ SEC.N-000111118 310 062 2332 RCL 333 RCC 333 RCC 333 RCC 334 RATA CALL PIN Pre-Mature DONE means end, error set if post of the	2331 ERRAZ SEC.M-000111118 315 361 006 2332 CALL DAT; 330 2334 CALL DAT; 315 066 007 2335 BH474 CALL PIN Pre-Hature DONE means 32 167 007 2337 JUP BH474 DE 3770-50-55-56 FH7 22 2340 BH474 DB 3770-50-55-56 FH7 22 2342 BH474 DB 3770-50-55-56 FH7 22 2345 BH474 DB 3770-50-52-53-51 FH7 24 25 2346 BH474 EGUU #-BH4774 GT 777	2331 ERRAZ SEC.N-000111118 310 62332 R.C. CALL DAY. 223 2334 R.C. DAY. 312 105 0042 2335 R.C. DAY. 312 105 0042 2335 R.C. DAY. 313 006 0077 2336 R.C. DAY. 322 2339 R.C. DAY. 322 2339 R.C. DAY. 322 2339 R.C. DAY. 334 B.C. DAY. 343 B.C. DAY. 354 B.C. DAY. 354 B.C. DAY. 3770—50—52—55—56 R.C. DAY. 361 2345 B.C. DAY. 362 2345 B.C. DAY. 3770—50—52—53—56 R.C. DAY. 361 2345 B.C. DAY. 362 2346 B.C. DAY. 363 B.C. DAY. 364 B.C. DAY. 365 B.C. DAY. 3770—50—52—53—51 R.C. DAY. 366 B.C. DAY. 3770—50—50—50—50—50—50—50—50—50—50—50—50—50	300	100	2330	0	RI		
115 361 006 233	315 361 006 2332 CALL DAY. 330 2333 RC 2334 RC 2347 LXI D.USERFAA 312 167 007 2335 BH474 LXI D.USERFAA 322 2338 BH474 DB BH474 Get another byte 322 2342 BH47A DB 3770-50-52-55-56 377 2343 DB 3770-50-52-53-55 361 2344 BH47AL EQU #-BH47A 361 2346 BH47AL EQU #-BH47A	1315 361 006 233 CALL DAY. 233 234 LXI DINUSERMA 315 066 007 2336 BH474 CALL PIN Pre-Mature DONE means end, error set if 32 2339 LNX D Pre-Mature DONE means end, error set if 32 2339 LNX D BH474 CALL PIN BH474 CALL P		1			RRNZ	SEC. M-000111118	
315	330 2333 RC 331 0042 2334 LXI D.USERFHA 315 066 007 2335 BH474 CALL PIN 322 167 007 2337 JC WDN 323 167 007 2337 JC WDN 323 167 007 2337 JC WDN 322 2339 JMP Pre-Mature DONE means 324 2339 JMP BH474 DB 3770-50-52-53-56 "H" 377 2345 DB 3770-50-52-53-56 "H" 361 2345 BH474 EQU #-BH474 "T" 3234 BH474 EQU #-BH474 "T"	310 042 233 021 200 042 233 312 066 007 2334 312 066 007 2334 312 167 007 2337 312 167 007 2337 313 167 007 2337 314 006 317 006 318 006 007 008 317 006 334 008 318 00 3770-50-52-55 322 234 008 3770-50-53-56 341 066 311 006 318 008 377 234 008 377 20-53-56 361 2346 8447AL E0U *-8447A.	•	361 006			1 1	DAT	
135 166 004 2335 LXI D,USERFWA  315 066 004 2335 BH474 CALL PIN Pra-Mature DONE means end, error set if 022 2336 BH474 CALL 90	2334 LXI D,USERFWA 312 066 007 2335 BH474 CXLL PIN Pre-Mature DDNE means 32 167 007 2337 BH474 CXLL PIN Pre-Mature DDNE means 32 167 007 2337 BH474 DD C42 2339 INX D C42 2334 BH474 DB 3770-50-52-53-55 FF CAP 234 BH474 DB 3770-50-52-53-55 FF CAP 2345 BH474 DB 3770-50-52-53-51 PF CAP 2345 BH4774 EQU #-BH4774 PF CAP 2345 BH4774 EQU #-BH4774 PF CAP 2345 BH4774 FF CAP 2345 BHA774 FF CAP 2345 BHA	021 200 042 2335 315 066 007 2336 8H474 CALL PIN Pre-Mature DONE means end, error set if 023 15 066 007 2339 10X 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				, a	֡֝֝֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓֓		
021 200 042 2335 LXI D.USERFWA 318 066 007 2336 8H474 LCALL 318 066 007 2337 UC NON 32 2338 IN	021 200 042 2337 LXI D,USERFHA 315 066 007 2336 6H474 CALL 91N Pre-Mature DONE means 022 2338 STAX D 023 2340 SA47 BH474 DB 3770-50-52-53-55-56 377 2343 DB 3770-50-52-53-56 377 2345 BH47AL EDU #-BH47X 361 2346 BH47AL EDU #-BH47X	021 200 042 233 315 066 007 2336 BH474 CALL PIN Pre-Hature DDNE means end, error set if 315 066 007 2337					: : : د		
315 066 007 2335 8H474 CALL FINANCE CALL FOR FIGURE BONE means end, error set if 023 2338 Local CALL FOR FINANCE CALL CALL FOR FINANCE CALL CALL FOR FINANCE CALL CALL CALL CALL CALL CALL CALL CA	315 066 007 2336 8H474 CALL PIN Pre-Mature DONE means 32 167 007 2337 4	312 167 007 2336 8H474 CALL PIN Pre-Mature DDNE means end, error set if 023 167 007 2337 JC WDN Pre-Mature DDNE means end, error set if 023 167 007 2339 LNX D BH74 DB 3770-50-55-56 THT 2343 BH74 DB 3770-50-53-51 TT 006 2344 DB 3770-50-53-51 TT 006 2345 DB 3770-50-50-50-50-50-5		0			,		
32 10 000 UZ 233 BHYYY UNIN Pre-Mature DONE means end, error set if 32 10 000 UZ 233	312 105 007 2330 BH474 UALL FIN Pre-Mature DONE means 022 157 007 2337 UNX D STAX D STAY STAY STAY STAY STAY STAY STAY STAY	312 000 07 233 BH474 CALL FIN Pre-Mature DONE means end, error set if 312 000 07 2339 BH474 D Get andther byte 223 2339 LNX D Get andther byte 234 BH474 DB 3770-52-55-56 TH7	•		6333		1	D.O.S.EKTEA	
32. Lof 007, 2337	332 167 007 2337 JC WDN  303 311 006 2340  222 2342  2342 8H474 DB 3770-50-52-53-55-56 "H"  377 2343 DB 3770-50-52-53-55-56 "H"  362 2345 DB 3770-50-52-53-55-56 "H"  361 2345 BH477L EDU #-BH477  362 2346 BH477L EDU #-BH477	23.1 Lof 007 233		990	2336	<b>*</b>	ALL	Z	
2342 2348 214X D 233 23349 1HX D 234 2 1HX D 234 2 1HX D 234 2 1HX D 234 2 1HY D 234 3 1HY D 234 5 1HY	022 2339 1NX D 2340 2340 2340 2340 2340 2340 2341 222 2342 8H47A DB 3770-50-52-53-55 2342 2345 DB 3770-50-52-53-55 2345 2345 DB 3770-50-52-53-51 2346 BH47AL EQU #-BH47A	232 2336 1HX D 233 1HX D 233 1HX D 234 1HX D 234 2H47A 0B 3770-50-52-53-55-56 377 2344 DB 3770-50-52-53-55 381 2345 BH47AL EDU #-BH47A 1770-50-55-56 381 2346 BH47AL EDU #-BH47A 1770-50-50-50 381 2346 BH47AL EDU #-BH47A 1770-50-50 381 2346 BH47AL EDU #-BH47A 1770-50-50 381 2346 BH47AL EDU #-BH47A 1770-50 381 2346 BH47A	:	167			: : ذ د	Z	DONE means end, error set if
023 303 311 006 2339 100 23342 23342 222 23342 242 252 23343 361 23345 361 23345 361 23346 36474 08 3770-50-52-53-56 361 23346 864774 666 3770-50-52-53-56 361 23346 864774 666	023 11 006 2349 INX D 8 H474 66t 2341 222 2342 8H474 0B 3770-52-53-55-56 341 242 8H474 0B 3770-52-53-55-56 341 2346 8H474L EQU #-8H47A 53-51 65t 2346 8H47AL EQU #-8H47A	023 303 311 006 2342 2342 2342 2342 343 2343 344 361 2344 361 2345 361 2346 364 364 364 364 364 364 364			2338	<b>σ</b> 1	TAX	a	
303 311 006 2340 JMP BH474 Get 2342 2342 BH47A DB 3770 2343 DB 3770 2344 DB 3770-50-52-53-55-56 2345 DB 3770-50-52-53-55 267 2345 DB 3770-50-52-53-51 2345 BH47AL EQU #-BH47A	303 311 006 2340 JMP BH474 Get 2341 222 2342 BH474 0B 3770 2343 DB 3770 2344 DB 3770 2345 DB 3770 50-52-53-56 361 2346 BH474L E0U #-BH477	303 311 006 2340 JMP BH474 Get 2342 2342 BH47A DB 3770 2343 DB 3770 2344 DB 3770 2345 DB 3770 2345 DB 3770 25 - 53 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 23 - 55 - 56 25			2339		×	Q	
222	222	222			2340	•	d.	BH474	Get another byte
222 2342 84474 DB 3770-50-52-53-55-56 3770 2344 DB 3770-50-52-53-51 361 2346 BH474L EQU #-8H474 361 2346 BH4774L EQU #-8H477	222 2342 8H474 DB 3770-50-52-53-55-56 2343 BB 3770-50-52-53-51 2345 BB 3770-50-52-53-51 2345 BB 3770-50-52-53-51 2346 BB 3770-50-50-52-53-51 2346 BB 3770-50-52-53-51 2346 BB 3770-50-50-50-50-50-50-50-50-50-50-50-50-50	222 2342 84474 DB 3770-50-52-53-55-56 3770 3770-50-52-53-56 361 2345 B4474L E0U #-84477 361 2346 B44774L E0U #-84477			2341				
361 2345 08 3770-52-53-51 245 08 3770-52-52-52-52-52 25 25 25 25 25 25 25 25 25 25 25 25 2	361 2343 08 3770-52-53-51 2345 08 3770-52-52-53-51 2345 08 3770-52-52-52-52-52-52-52-52-52-52-52-52-52-	361 2345 08 3770-52-53-51 245 08 3770-52-53-51 245 08 3770-52-53-51 245 08 3770-52-53-51 245 245 245 245 245 245 245 245 245 245			2342	H47A	<b>.</b>	3770-50-52-53-5	-26
361 2345 06 37704-50-53-51 361 2346 BH47AL E0U #-BH47A	361 2345 06 3770-50-55-50 361 2346 BH477AL E0U #-BH477	361 2345 06 3770-52-53-51 361 2346 BH47AL E0U #-BH47A	:		2343	-	2 2	2//2	
2345 BH47AL E0U	2346 BH47AL E0U	2346 BH47AL E0U			2344		9 9	3//4-50-52-53-5	
2346 BH47AL EQU	2346 BH474L EQU	2346 BH47AL EQU			2345		9	3770-52-53-51	.4.
			:		2346	H47AL	20	#-BH47A	
									•••••••••••••••••••••••••••••••••••••••
				:	: : : : : : : : : : : : : : : : : : : :	:	:		
					:		:		

007-001 311 2409 ** EIRET - EI RETURN 007-001 311 2409 ** EIRET - EI RETURN 007-002 373 241  ** EIRET - EI RETURN 007-003 341  ** EIRET - EI RETURN 007-004 341  ** EIRET - EI RETURN 007-007 241  ** EIRET - EIRET - EI RETURN 007-007 241  ** EIRET - EIRE	
1000   063   2406   1NX   SP   1001	ita
2409 ** EIRET   S a 2410 ** EIRET   S a 2412 ** and execute 2413 ** and execute 2414 *	
2410	
2411	/Kamsco /
2412 * and executes a KETurn Instruction 2413 * EIRET EI  003 311 2415 EIRET EI  2418 ** IBD - Input from Boot Device 242 * ENTRY: BDA - Boot Port as save 242 * ENTRY: BDA - Boot Port as save 242 * ENTRY: BDA - Boot Port as save 242 * ENTRY: BDA - Boot Port as save 242 * ENTRY: BDA - Boot Port as save 242 * ENTRY: BDA - Boot Port as save 242 * ENTRY: BDA - Boot Port Index 242 * ENTRY: BDA - Boot BOOT BOOT BOOT BOOT BOOT BOOT BOOT BOO	les Interrupts,
2414 ERET EI  2415 ERET EI  2416 ** EF  2416 ** EF  2418 ** EF  2418 ** EF  2421 ** EF  2421 ** EF  2422 ** EF  2422 ** EF  2424 ** EF  2424 ** EF  2425 ** EF  2426 ** EF  2426 ** EF  2426 ** EF  2427 ** EF  2426 ** EF  2427 ** EF  2428 ** EF  2428 ** EF  2428 ** EF  2429 ** EF  2429 ** EF  2420 ** EF  24	
2418 ** 180 - Input from Boot Device 2420 ** 180 inputs data from the Boot Port as save 2420 ** 180 inputs data from the Boot Port as save 2421 ** ENTRY: BDA ** Boot Device Address 2422 ** ENTRY: BDA ** Boot Device Address 2422 ** ENTRY: BDA ** Boot Device Address 2422 ** ENTRY: BDA ** Boot Device Address 2424 ** ENTRY: BDA ** Boot Device Address 2425 ** ENTRY: BDA ** Boot Device Address 2426 ** ENTRY: BDA ** Boot Device Address 2427 ** ENTRY: BDA ** Boot Device Address 2428 ** ENTRY: BDA ** Boot Device Address 2428 ** ENTRY: BDA ** Boot Device Address 2430 ** BDA ** BOOT DO ** BOOT DEVICE ADDRESS 2431 ** BDA ** BDA ** BOOT DEVICE ADDRESS 2432 ** BDA ** BDA ** BOOT DEVICE ADDRESS 2433 ** ENTRY: BDA ** BOOT DEVICE ADDRESS 2440 ** BDA ** BDA ** BOOT DEVICE ADDRESS 2440 ** BDA *	
2418 ** 180 - Input from Boot Device 2419 ** 180 inputs data from the Boot Port as save 2420 ** 180 inputs data from the Boot Port as save 2421 ** 180 inputs data from the Boot Port as save 2421 ** EMIRY: BDA 2422 ** EMIRY: BDA 2424 ** EXII: A = Data input from port 2425 ** EXII: A = Data input from port 2426 ** USES: PSW 2428 ** USES: PSW 2428 ** USES: PSW 2428 ** USES: PSW 2428 ** USES: PSW 2429 ** USES: PSW 2420 ** USES: PSW 2430 ** USES	
2420	/Ram860 2/
2420 * IBD inputs data from the Boot Port as save 2422 * ENTRY: BDA	
2422	saved at boot time.
2424	
2424	
2425	
2427 # 10MKK destroyed 2428 # USES: PSM 2429 # 2429 # 2429 # 2431 005 345 2431 006 325 2432 PUSH 006 325 2432 PUSH 007 126 2433 PUSH 007 126 2434 PWY By H 011 072 120 041 2436 PWB BDA 011 072 120 041 2436 PWB BDA 012 042 353 2440 PWH PAININ Stuff Instruction 024 353 244	
2428 + USES: PSM 2429 + 2429 + 2429 + 2429 + 2429 + 2429 + 2429 + 2429 + 2431 IBD XTHL 005 345 2432 PUSH PSM 006 325 2432 PUSH D D D D D D D D D D D D D D D D D D D	
004 343 2431 IBD XTHL 005 365 2432 PUSH PSW 006 375 2432 PUSH PSW 007 126 2433 PUSH DD 007 126 2434 IBD XTHL 011 072 120 041 2436 LDA BDA 014 202 2437 ADD D 015 353 2438 XCHG 016 147 202 2440 NVI L,MI.N 021 042 002 040 2441 SHLD IDWRK 024 353 244	
004 343 2431 IBD XTHL 005 365 2432 PUSH PSW 006 325 2433 PUSH DD 007 126 2434 INV DD,M D,M D = Port Index 010 043 2435 LDA BDA 011 072 120 041 2436 LDA BDA 014 202 2437 ADD D 015 353 2430 MVI L,MA M = Actual Output 017 056 333 2440 MVI L,MA M Stuff Instruction 024 353 2442 XCHG 025 321 2443 POP D 025 321 2444 XTHL 030 303 002 040 2446 JMP IDWRK D0 the actual inpu	
005 365 2432 PUSH PSW 006 325 2433 PUSH D PSW 006 325 2433 PUSH D PSW 000 7 126 2433 PUSH D PSW 0 P P P P P P P P P P P P P P P P P P	
000 325 2433 PUSH D	
010 043 2434 NUV D.NK H 010 043 2435 INX H 011 072 120 041 2436 LDA BDA 014 202 2437 ADD D 015 353 2438 XCHG 017 056 333 2440 HVI LIMIK Stuff Instruction 024 353 244 POP D 025 321 2443 POP D 025 331 2444 POP PSW 030 303 002 040 2446 JNP IDWRK Do the actual inpu	
011 072 120 041 2436 LDA BDA 014 202 2437 ADD D 015 353 2438 XCHG 016 147 202 2439 MVI L,MI.IN Stuff Instruction 024 353 2440 MVI L,MI.IN Stuff Instruction 024 353 2442 XCHG 025 321 2442 YCHG 025 321 2443 POP PSW 027 343 2445 XTHL 030 303 002 040 2446 JMP IOWRK Do the actual inpu	
014 202 2437 ADD D 015 353 2438 XCHG 016 147 202 2439 XCHG 017 056 333 2440 MVI LyMi.IN Stuff Instruction 024 353 2442 XCHG 025 321 2443 PDP PSW 025 341 2444 PDP PSW 027 343 002 040 2446 JMP IDWRK Do the actual inpu	
015 353 2438 XCHG 016 147 2439 MOV H,A 017 056 333 2440 MVI L, MI.IN Stuff Instruction 024 353 2442 XCHG IDWRK Stuff Instruction 024 353 2442 XCHG 0 025 321 2443 PDP PSW 025 341 2444 XTHL IDWRK Do the actual inpu	
017 056 333 2440 MVI LyMI.IN Stuff Instruction 024 353 2440 MVI LyMI.IN Stuff Instruction 024 353 2442 XCHG DWRK Stuff Instruction 024 353 2443 POP PSW 2445 XTHL 10WRK Do the actual input	
021 042 002 040 2441 SHLD IDWRK Stuff Instruction 024 353 2442 XCHG DP D 0 025 321 2443 PDP PSW 027 343 2445 XTHL IDWRK D0 the actual input	pur Address
024 353 2442 XCHG 025 321 2443 PDP D 026 361 244 PDP PSW 027 343 2445 XTHL IDWRK Do the actual inpu	on and Port
026 361 2444 PUP U 026 361 2444 PUP PSW 027 343 2445 XTML IDWRK Do the	
027 343 022 040 2446 JMP IOWRK Do the	
.030 303 002 040 2446 JMP IDWRK Do the	
	nput

343 176 043 343 343 345 325 126 072 120 042 120 042 120 042 120 041 135 135 135 135 135 135 135 135 135 13	2 2 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5	08D	Ť :	S the data to S the data to MET+1) = Port MET+1) = Port MET+1) = Port MET+1) = Port MET+10 = Port ME	ot Device  o the indexed Boot Device.  Device Address Address Index  the port  A = data byte to output  A = Boot Device Address  H = Actual Device address Stuff Instruction and address
21 61 43 03 002 040	2478 2479 2480 2481		POP YTHL JMP	PSW I OWRK	Do the actual 1/0
	2483 2484 2485 2486 2486	* • • • •	PIN PIN inp	N - Port In N inputs a byte of d data-transfer-ready	data from the H-47 with handshake.
	2488 2489 2490	* * *	ENTRY:	NONE - C.	Set if ERROR
	24912 2493 2494		uses:	• <b>•</b>	ata
315 004 007 000 346 240		Z H	CALL	180 0.STAI S.OTR+S.DON	

346 040 067 300 315 004		**********		:		
300		2502		ANI	S.DON	
315 004		2504		RNZ	Error because done befor	ore DTR
	700	2505		CALL	180	
100	;	2507		0.8	D.DATI	
007.107 247 007.110 311		2508 2509		ANA Ret		
: :						
		\$ 1152	•	R. 50P	- Set-Up Device Parameters	/Ram8Go 2/
		2512 *				
		2513 4		R. SDP s	sets up arguments forthe specific unit.	
		2515			- Motor ON	
		2516 #			D.TRKPI - Address of device track number	
	•	2517		Modifie	d to access drive 3, or sy2:.	
		5519		:		
		2520		ENTRY:	AIO.CNI - Colt Nesber	
		2522		EXIT:	HL = D.TRKPT	
		2523				
		4767		USES		
		2526				
007.111 076 012	: ;	12527	8.50P	NVI		
790	2 5	2528		¥ 70	COCHINE Set the max error country	at for the operation
365	•	2530		PUSH	PSH	
:		2531		CPI	I+I	
433	750.640	2532	:	,,,	1 30 0 4 600	
		2534		ERRNZ	5	
000.00	:	2535	:	ERRNZ	06.051-4	
:	:	2536				
7.127 076 003		2537		NVI FOOM	A,3 DE_DS2_8	
007.131 303 073	036	2539	:	JHP	K.SOP.	
	:					
		36.41	:		: 0	/ C - C - C - C - C - C - C - C - C - C
	:	2542		2		77 000 100 100 100 100 100 100 100 100 1
		2543		RRDY C	hecks to see if the drive specified in	
	:	2544		AIO.UNI is	ready.	
		2545		- VOTA		
		2547			10.014	
		2548		EXITE	L = Ready Bits	
		2549	•			
		2550	*	USES:	1,220	

		жамом						/Ram8Go 2/																			/Ram8Go 2/		1 30 10 th	any co				an error or time-out		
			L = Ready Bits					Hove	SMOV performs a short (<256) byte move)	•		= dest_nateon				JE, L				L - Byte Count				V 6 4 2 C 6 2 C 5			for Done	4	THON Waits for the done bit to be set. A time	that the comman				here is	clear if no error	
	COM DO.RRDY	NId	L, A	Z Q				- Short Move	rforms a		) u			RET+1	100	PSM, BC, DE, L		\ \ \	T	L, A	90			SMOV	•		I Xait		ts for t	לין מכא סיי מים		NONE		PSE		PSWAR
	CALL	CALL	AOV	CALL				SMOV	SMOV pe		ENTRY:			EXIT		USES:		XTHL	XX	XTHL MOV	LDAX	STAX	XXI	DCR	7	RET	ZQ		S S S S S S S S S S S S S S S S S S S	re-triede		ENTRY:		EXIT		USES:
*	RRDY							* *		•	* *	• •	• •	*	* *	* *		SHOV			SMOV.						*	• •	* *	• •	*	*	*	*	<b>*</b> 1	
55	55	25	5.0	2558	:			2561	2563	2564	2565	2567	2568	5569	2570	2572	2573	2574	2576	2577 2578	2579	2581	2583	2584	2586	2587		10 11	Ω.κ	<b>`</b>	·.r	vo.		<b>O</b> .	2598	n
:	330 006	200 990	:	167 007																				156 007												
		15	157	12		:	:		:	:		:			:		:	43	2	343	12	220	23.5	5.5	,	=	: :		:		:	:		:		:

007.117 013 2603 MDM LXI 8.WDMA 007.172 013 2605 MDM1 DCX 8 007.173 217 2605 MDM1 DCX 8 007.174 217 2606 MDM CX 6 007.175 310 2609 RZ CCC 007.175 310 2609 RZ CCCC 007.175 310 2609 RZ CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
013 2605 WDN1 DCX B 170 2606 WDN1 DCX B 261 2607 DRA C 310 2609 RZ 2610 CALL IBD 000 2612 ANI S.DON 315 004 007 2613 ANI S.DON 315 172 007 2613 ANI S.DON 315 172 007 2615 DB 000 2613 ANI S.ERR 000 2617 BD 000 2617 BD 000 2617 ANI S.ERR 000 2618 ANI S.ERR 000 2617 ANI S.ERR 000 2618 ANI S.ERR 000 2619 ANI S.ERR 000 2619 ANI S.ERR 000 2619 ANI S.ERR 000 2619 ANI WIR Waits for a transfer 000 2620 WIR Waits for a transfer 000 2630 WIR Waits for a transfer 000 000 000 000 000 000 000 000 000 00
261 2607 0RA C C 2608 067 2608 077 C C C C C 2608 077 C C C C C C C C C C C C C C C C C C
067 2608 STC 310 2609 RZ 310 2611 CALL IBD 000 2612 ANI S.DON 312 172 007 2614 JZ WDNI 312 172 007 2614 JZ WDNI 315 004 007 2615 CALL IBD 000 2617 BB D.STAI 346 001 2619 STC 360 Z620 RNZ 311 262 RET 2620 RNZ 311 262 HONA EQU 32000 2627 ** WTR waits for a transfer 2629 ** WTR waits for a transfer 2630 ** WITR waits for a transfer 2631 ** WITR Waits for a transfer 2631 ** WITR Waits for a transfer 2632 ** WITR Waits for a transfer 2633 ** WITR Waits for a transfer
315 004 007 2610  5612  5612  5613  5614  5615  5614  5615  5615  5615  5615  5616  5616  5616  5617  5617  5617  5618  5628  5628  5628  5628  5638  6638
202 000 2612 08 0.57AI 203 346 040 2613 ANI S.DON 205 312 172 007 2616 213 000 2617
205 312 172 007 2614 JZ WDN1 2615 210 315 004 007 2616 DB D. STAI 213 000 213 000 2619 STC 2619 STC 2621 STC 2622 RNZ 2621 STC 2623 RET 2625 WDNA EOU 32000 2625 WDNA EOU 32000 2628 WTTR Walts for a transference of time-out wait 2632 WITH also time-out wait 2632 WITH AND MAIN AND AND AND AND AND AND AND AND AND AN
210 315 004 007 2616 CALL 180 213 000 2617 08 0.5TAI 214 346 001 2619 STC 215 300 2620 RNZ 220 247 2620 RNZ 221 311 2623 RET 2625 HONA EQU 32000 2625 HONA EQU 32000 2627 ** WTR waits for a transfer 2629 * WITR waits for a transfer 2630 * WITR WAITS
214 346 001 2618 ANI SERR 216 067 2629 STC 220 247 2623 RET 221 311 2624 RET 2625 WONA EQU 32000 2627 ** WTR - Halt for Tra 2629 * WTR Walts for a transf 2630 * first, and if it is fo 2632 * WITH AND TRANSF 2631 * WITH AND THE FOR THE FOR TRANSF 2631 * WITH AND THE FOR
2620 RNZ 2620 RNZ 2621 311 2623 RET 2623 RET 2624 RET 2625 WONA EQU 32000 2627 ** WTR - Halt for Tra 2629 * WTR waits for a transf 2630 * first, and if it is fo 2631 * WITH A STORE 2631
2620 KNZ 2621 311 2622 ANA A 221 311 2624 WONA EQU 32000 2625 WONA EQU 32000 2627 ** WTR - Walt for Tra 2628 * WTR waits for a transf 2630 * first, and if it is fo 2631 * Will also time-out waits
.220 247 2623
2625 WONA EQU 32000 2627 ** WTR - Halt for Tra 2628 * WTR waits for a transf 2639 * first, and if it is fo 2631 * Will also time-out waits
2627 ** WTR - Hait for Tra 2628 * WTR waits for a transf 2630 * first, and if it is fo 2631 * Will also time-out wait
first, and if it is for wall also time-out wall
• •
# ENTRY: NONE
2635 # EXIT: PSW = "C" set 17 ERROR 2636 # 10 error
* USES: PSM.BC
**
007.222 001 000 175 2641 WTR LXI B:WTRA BC = time-out count
225 315 004 007 Z643 WTRI CALL
234 300
2648
NOK NOK
7 261 2651
1 310

Subroutines	Subroutines	HONITOR	#01.02.00°	•00•		Unix H8ASH V1.4.1 VTR 16:53:59	1 5-Jul-80 11-SEP-80	Page 64
007.242 007.245 007.246 007.250	315 004 007 000 346 200 312 225 007 311	2654 7 2655 2656 2657 7 2658 2659 2660		CALL DB ANI JZ RET	IBD D.STAI S.DTR MTR1	No DTR yet		
175.000		2661	HTRA	Eou	32000	Time-Out count		

2669   PATCH   PRINCE   CONTROL	2659 (16) The UNIT with code which also checks for Auto-Boot.  2670 (16) The UNIT with code which also checks for Auto-Boot.  2671 (16) The UNIT with code which also checks for Auto-Boot.  2672 (16) The UNIT with code which also checks for Auto-Boot.  2673 (16) The UNIT with address is already on the Stock if I also Check for Will address is already on the Stock if I also Check for Auto-Boot.  2673 (16) The UNIT with address is already on the Stock if I also Check for Auto-Boot.  2674 (17) The UNIT with address is already on the Stock in I also Check for Auto-Boot.  2675 (17) The UNIT with address is already on the Stock in I also Check for Auto-Boot.  2675 (17) The UNIT with address is already on the Stock in I also Check for Auto-Boot.  2675 (17) The UNIT with a sold of the Unit with Check for Auto-Boot.  2675 (18) The UNIT with a sold of the Unit with Check for Auto-Boot.  2675 (18) The UNIT with a sold of the Unit with Check for Check for Auto-Boot.  2675 (18) The UNIT with a sold of the Unit with Check for	2669 * 2669 * 2669 * 2670 * 2671 * 2671 * 2672 * 2674 * 2674 * 2674 * 2674 * 2677 * 2677 * 2677 * 2677 * 2681 * 26	replaces code which initially only initialized be ukk with code which also checks for Auto-Boot.  Hu feturn address is to AUTOB instead of ERROR.  NDME  HL = INIT exit address Tape UART Initialized  PSW.BC  *  Iize LOAD/DUMP Uart  A.UMI.1B+UMI.1B+UMI.1CB+UMI.1CA  OP.TPC  SET B BIT, NO PARITY, I STOP, XIC  H.ERROR  OP.CTL2  CN.ABO  No Auto-Boot  H.AUTOB
271	2617 2617 2617 2618 2619 2619 2619 2619 2619 2619 2619 2619	2671 2672 2673 2673 2673 2674 2675 2677 2680 2680 2680 2681 2683 2684 2684 2685 2684 2687 2689 371 2689 371 2689 371 2689 371 2689 371 2689 372 373 374 275 375 375 375 375 375 375 375 375 375 3	NOME  HL = INIT exits to AUTOB instead of ERROR.  NOME  HL = INIT exit address Tape UART Initialized  PSW.BC  A,UMI.18+UMI.L8+UMI.16X  OP.TPC  A,UMI.18+UMI.L8+UMI.16X  OP.TPC  Tor Auto-Boot  H,ERROR  NO AUTO-BOOT
2673	2677 • EXITY: NOME  2677 • USES: PSW-8C  2681 • Unitablize LDAD/DUMP Uart  2682 • AICH A-WHILLS-WHI	2673	
2677 • FXIT: HL = INIT exit address 2677 • Tape UNRT Initialized 2678 • USES: PSW-BC 2678 • USES: PSW-BC 2679 • USES: PSW-BC 2680 • USES: PSW-BC 2	227 • EXIT: H INIT extt address 277 • USES: PSW-BC 278 • USES: PSW-BC 279 • USES: PSW-BC 289 • USES: USE	2675	
2676 • EXII: HL - INIT exit address 2677 • USES: PSW-BC 2678 • USES: PSW-BC 2680 • USE	277 • KITT HH = INIT exit saddess 2677 • KITT HH = INIT exit saddess 2677 • KITT HH = INIT exit saddess 2677 • Level •	2676	
2678 * 2678 * 2679 * 2679 * 2679 * 2679 * 2679 * 2689 * 26	2678	2679	
2677 * USES: PSH.8C  2681 2682 2683 2683 2684 2684 2685 401 1 200 200 1 0 0 0 0 0 0 0 0 0 0 0 0 0	2677 * USES: PSW-BC 2667 * USES: PSW-BC 2680 * Initialize LDAD/DUMP Uart 2681 * Initialize LDAD/DUMP Uart 2683 * Initialize LDAD/DUMP Uart 2684 * Initialize LDAD/DUMP Uart 2685 * UNIT DATE SET 8 BIT; NO PARITY, I SYDP, XIB 2689 * Check for Auto-Baot 333 362 2690 * PATCH2 333 362 2690 * PATCH2 310 000 2690 * PATCH2 2700 * P	2679 * USES: 2681	
2681 FATCHI EQU * * 2682 PATCHI EQU * * 2683 * Initialize LOAD/DOHP Wart 2683 * Initialize LOAD/DOHP Wart 2683 * Initialize LOAD/DOHP Wart 2685	2681 FATCH1 EQU	2681 PATCH1 EQU 2681 PATCH1 EQU 2682 PATCH1 EQU 2683 PATCH1 EQU 2685 PATCH1 PATCH1 PATCH1 PATCH1 PATCH1 PATCH1 PATCH1 PATCH1 P	
2684 * Initialize LOAD/DUMP Wart 2683 * Initialize LOAD/DUMP Wart 2684 * Initialize LOAD/DUMP Wart 2685	2.683	2682 PAICHL EUU 2683 2684	<b>: : : : : :</b>
2.664	116   2685   10	2684 TINITI 2685 TINITI 2685 TINITI 323 371 2687 DUT 2689 Check 2689 Check 2689 Check 2689 Check 2690 Check 2690 Check 2691 LXI 333 362 2692 IN 346 200 2691 LXI 2693 ANI 2694 RZ 2695 CXI 310 2695 CXI 311 207 004 2696	
1.0   2.685	076 116 2685 323 371 2685 001 007 106.1PC SET 8 BIT, NO PARITY, I STOP, XI6 2687 2689	2685 2686 323 371 2686 2687 2689 4 2690 2690 2690 333 362 2692 346 200 2691 340 200 2694 2695 2696 2696 2697 2697 2697 2697 2697 2697 2697 2697 2697 2698 26	•
0/6 115	0.6 110 2686 NVI A: NVIL B: UNIL B: UNIL B: UNIL B: VIL B & 2688	0.76 116 2686 323 371 2687 2689 4 2690 041 322 000 2691 346 200 2692 346 200 2693 310 2695 041 207 004 2696 311 207 004 2696	
2688 Check for Auto-Boot 2699 Check for Auto-Boot 2699 Check for Auto-Boot 2690 Check for Auto-Boot 2690 Check for Auto-Boot 2690 Check for Auto-Boot 2693 Ani CN-ABO No Auto-Boot 2693 Ani CN-ABO No Auto-Boot 2694 Ani CN-ABO Check Chec	2688 Check for Auto-Boot 2689 Check for Auto-Boot 2690 LLXI Higher 333 362 2692 LXI Higher 331 22 2692 LXI Higher 331 22 2694 RZ LXI Higher 310 2699 Children 32 2699 Children 33 32 26 26 26 26 26 26 26 26 26 26 26 26 26	2688 2689 4 2689 6 2690 333 362 2691 346 200 2693 310 2694 041 207 004 2696 311 207 004 2696	OF Auto-Boot H, ERROR OP. CIL2 CN. ABO No Auto-Boo
2690 2691 LXI H-ERROR 346 2000 2691 LXI H-ERROR 346 2000 2691 IN OP-CTL2 346 2000 2693 ANI CN-ABO 041 207 004 2695 RET H-AUTOB 310 2699 ** PATCH2 RET H-AUTOB 2699 ** PATCH2 RET H-AUTOB 2700 * PATCH2 RET H-AUTOB 2700 * PATCH2 RET H-AUTOB 2701 * PATCH2 ROVES the NTR6 code out of its original place 2702 * PATCH2 RET H-AUTOB 2703 * FOUTHDRE NOT ALLOWED TO ALTER STACKPOINTER 2704 * RESTOR YALUE AND CARRY FLAG 312 322 000 2705 PATCH2 JZ ERROR NOT ALLOWED TO ALTER STACKPOINTER 2705 PATCH2 JZ RET H-AUTOB 361 2708 JMP IOA INPUT OCTAL ADDRESS	2650 CHECK TO ALLO DOLL  2650 LXI HERROR  310 2659 AMI CN-A6CD  310 2659 AMI CN-A6CD  311 2659 AMI CN-A6CD  311 2659 AMI CN-A6CD  312 2659 AMI CN-A6CD  313 AMI CN-A6CD  314 207 004 2659 AMI CN-A6CD  315 AMI CN-A6CD  316 AMI CN-A6CD  317 AMI CN-A6CD  318 AMI CN-A6CD  319 AMI CN-A6CD  310 2659 AMI CN-A6CD  311 2659 AMI CN-A6CD  312 3650 AMI CN-A6CD  313 362 CO 3 2706 AMI CN-A6CD  314 AMI CN-A6CD  315 AMI CN-A6CD  316 AMI CN-A6CD  317 AMI CN-A6CD  318 AMI CN-A6CD  319 AMI CN-A6CD  310 AMI CN-A6CD  310 AMI CN-ACCD  311 AMI CN-A6CD  312 AMI CN-A6CD  313 AMI CN-A6CD  314 AMI CN-A6CD  315 AMI CN-A6CD  316 AMI CN-A6CD  317 AMI CN-A6CD  318 AMI CN-A6CD  319 AMI CN-ACCD  310 AMI CN-ACCD	2690 2690 3490 348 200 2691 346 200 2693 310 2694 2694 2694 2696 311 207 004 2696	H, ERROR OP.CTL2 CN.ABO NO AUTO-BOO
041 322 000 2691 LXI H,ERROR 333 362 2692 IN 00-CTL2 340 200 2694 RAI CN-ABO NO Auto-Boot 2694 RAI HAUTOB 241 2695 RET HAUTOB 211 2699 ** PATCH2 moves the MTR6 code out of its original place 2700 * PATCH2 moves the MTR6 code out of its original place 2701 * PATCH2 moves the MTR6 code out of its original place 2702 * In the ROM to permit providing for HB9/HB common PlN 2703 * routine- 2704 * 2704 * RESTORE VALUE AND CARRY FLAG 2705 * IN HR RESTORE VALUE AND CARRY FLAG 303 062 003 2709 JMP IDA IDA IDA INPUT OCTAL ADDRESS	041 322 000 2691 LXI H,ERROR 346 200 2692 IN OP.CTL2 346 200 2694 RX CN.A800 NO AUTO-Boot 2694 RX CN.A800 NO AUTO-Boot 310 2699 ** PATCH2 2699 ** PATCH2 2700 * PATCH2 Moves the NTR6 code out of its original place 2701 * PATCH2 Moves the NTR6 code out of its original place 2702 * In the KON to permit providing for H89/H8 common PIN 2703 * routine. 2705 * TOO TOO TOO TOO TOO TOO TOO TOO TOO T	041 322 000 2691 333 362 2692 346 200 2693 310 2694 2695 041 207 004 2696 311	NO AUTO-BOO
333 362 2692 IN OP.CTL2 346 200 2693 ANI CN.ABO NO AUTO-BOOT 2694 RZ N.ABO NO AUTO-BOOT 2695 LXI H,AUTOB 311 2 2699 ** PATCH2 2699 ** PATCH2 2700 ** PATCH2 2700 ** FOUTOH PATCH2 2701 ** FOUTOH PATCH2 2702 ** FOUTOH PATCH2 2703 ** FOUTOH PATCH2 2704 ** FOUTOH PATCH2 2705 ** FOUTOH PATCH2 2706 ** FOUTOH PATCH2 2707 ** FOUTOH PATCH2 2708 ** FOUTOH PATCH2 2709 ** FOUTOH PATCH2 2709 ** FOUTOH PATCH2 2709 ** FOUTOH PATCH2 2709 ** FOUTOH PATCH2 2700 ** FOUTOH PATCH	333 362 2693 IN OP.CTL2 346 200 2693 ANI CN.A800 No Auto-Boot 2694 ANI CN.A800 No Auto-Boot 2695 RET H.AUTOB 311 2699 ** PATCH2 2699 ** PATCH2 2700 * PATCH2 2700 * FOLUTION 2701 * FOLUTION 2702 * FOLUTION 2703 * FOLUTION 2704 * FOLUTION 2704 * FOLUTION 2705 * FOLUTION 2706 * FOLUTION 2707 * FOLUTION 2706 * FOLUTION 2707 * FOLUTION 2708 * FOLUTION 2709 * FOLUTION 2	333 362 2692 346 200 2693 310 2694 041 207 004 2695 311 207 004 2696	NO AUTO-BOO
346 200 2693 ANI CN.ABD NO AUTO-Boot 2694 R1 4.4UTOB 2695 LXI H,AUTOB 2695 LXI H,AUTOB 2695 LXI H,AUTOB 2699 LXI H,AUTOB 2699 ++ PATCH2 2699 ++ PATCH2 2700 + FATCH2 2700 + FATCH2 2700 + FOUTING 2700 +	346 200 2693 ANI CN.ABO NO AUTO-Boot 2694 E.X.	346 200 2693 310 2694 041 207 004 2695 311 207 004 2697	No Auto-Boo
2699 ** PATCH2 2699 ** PATCH2 2699 ** PATCH2 2700 * FATCH2 moves the MTR6 code out of its original place 2702 * In the RDM to permit providing for H89/H8 common PIN 2703 * routine. 2704 * 2705 312 322 000 2706 PATCH2 JZ ERROR NOT ALLOWED TO ALTER STACKPOINTER 043 2708	311 2695 KET H,AUTOB  041 207 004 2696 LXI H,AUTOB  311 2699 ** PATCH2  2699 ** PATCH2  2700 * PATCH2  2700 * Touthe ROW to permit providing for H89/H8 common PIN  2703 * Touthe ROW to permit providing for H89/H8 common PIN  2703 * Touthe ROW to permit providing for H89/H8 common PIN  2704 * Touthe ROW to permit providing for H89/H8 common PIN  2705 * Touthe RROW NOT ALLOWED TO ALTER STACKPOINTER  943  361 2706 PATCH2 JZ ERROW H RESTORE VALUE AND CARRY FLAG  361 2707 JUNX H RESTORE VALUE AND CARRY FLAG  363 062 003 2709 JUNP IDA INAUT GGTAL ADDRESS	2695 041 207 004 2696 311 207 004 2697	000000000000000000000000000000000000000
041 207 004 2696 LXI H,AUTOB 311 2699 ** PATCH2 2699 ** PATCH2 2700 * PATCH2 moves the MTR6 code out of its original place 2701 * PATCH2 moves the MTR6 code out of its original place 2702 * in the RDM to permit providing for H89/H8 common PIN 2703 * routine. 2704 * 2704 * 2705 312 322 000 2706 PATCH2 12 ERROR NOT ALLOWED TO ALTER STACKPOINTER 2705 ATCH2 12 ERROR NOT ALLOWED TO ALTER STACKPOINTER 361 2708 AMP IDA INPUT OCTAL ADDRESS	041 207 004 2696 LXI H,AUTOB 311 2697 RET	041 207 004 2696 311 207 004 2697	H,AUTOB
311 2697 RET 2699 ** PATCH2 2700 * PATCH2 2701 * PATCH2 moves the MTR6 code out of its original place 2702 * in the RDM to permit providing for H89/H8 common PIN 2703 * routine. 2704 * 2705 * 2706 * 2706 * 2706 * 2707 INX H 312 322 000 2706 PATCH2 JZ ERROR NOT ALLOWED TO ALTER STACKPOINTER 043 2707 INX H 361 2708 ADP PSW RESTORE VALUE AND CARRY FLAG 363 062 003 2709 JMP IDA IDA INPUT OCTAL ADDRESS	311 2697 RET  2699 ** PATCH2 2700 * PATCH2 2701 * In the RDM to permit providing for H89/H8 common PIN 2702 * In the RDM to permit providing for H89/H8 common PIN 2703 * routine. 2704 * 2705 312 322 000 2706 PATCH2 JZ ERROR NOT ALLOWED TO ALTER STACKPOINTER 043 2707 INX H 361 2708 PDP PSW RESTORE VALUE AND CARRY FLAG 362 003 2709 JMP IDA INPUT OCTAL ADDRESS	311 2697	
2699 ** PATCH2 2700 * PATCH2 moves the MTR6 code out of its original place 2701 * In the ROW to permit providing for H89/H8 common PIN 2703 * routine. 2704 * 2704 * 2705 * TOUTINE. 2706 * TOUTINE. 2707 * TOUTINE. 2708 * TOUTINE. 2709 ** TOUTINE. 361 ** TOUTINE. 362 ** TOUTINE. 363 ** TOUTINE. 364 ** TOUTINE. 365 ** TOUTINE. 366 ** TOUTINE. 367 ** TOUTINE. 368 ** TOUTINE. 369 ** TOUTINE. 369 ** TOUTINE. 360 ** TOUTINE. 360 ** TOUTINE. 361 ** TOUTINE. 362 ** TOUTINE. 363 ** TOUTINE. 364 ** TOUTINE. 365 ** TOUTINE. 366 ** TOUTINE. 367 ** TOUTINE. 368 ** TOUTINE. 369 ** TOUTINE. 369 ** TOUTINE. 360	2699 ** PATCH2 2700 * FATCH2 moves the MTR6 code out of its original place 2701 * PATCH2 moves the MTR6 code out of its original place 2703 * Inv. H. A.		
2699 ** PATCH2	2699 ** PATCH2 2700 * FATCH2 moves the MTR6 code out of its original place 2701 * in the ROW to permit providing for H89/H8 common PiN 2703 * routine. 2704 * 2704 * 2705 * INX H RESTORE VALUE AND CARRY FLACE 312 322 000 2706 PATCH2 JZ ERROR 043 2707 INX H RESTORE VALUE AND CARRY FLACE 361 2708 JMP IDA INPUT OCTAL ADDRESS		
2700 # PATCH2 moves the MTR6 code out of its original place 2701 # PATCH2 moves the MTR6 code out of its original place 2702 # routine. 2703 # routine. 2704 # 2705 # FRENCH NOT ALLOWED TO ALTER STACKPOINTER 2705	2700 # PATCH2 moves the MTR6 code out of its original place 2701 # for the RDM to permit providing for H897H8 common PIN 2703 # routine. 2704 # routine. 2705 # FOULT IN H RESTORE VALUE AND CARRY FLAG 361 2708 PDP PSW RESTORE VALUE AND CARRY FLAG 303 062 003 2709 JHP IDA INPUT OCTAL ADDRESS	*	/Ram8Go 2
2702 * in the KOM to permit pro 2703 * routine. 2704 * 2704 * 312 322 000 2706 PATCH2 JZ ERROR 043 2707 INX H 361 2708 POP PSW 303 062 003 2709 JMP IDA	2702 * in the KOM to permit pro 2703 * routine. 2704 * 2705 312 322 000 2706 PATCH2 JZ ERROR 043 2707 INX H 361 2707 PDP PSW 303 062 003 2709 JMP IDA	* *	the MTR6 code out of its original place
2703 * routine. 2704 * 2704 * 2705 312 322 000 2706 PATCH2 JZ ERROR 043 2707 INX H 361 2708 PDP PSW 303 062 003 2709 JMP IDA	2703 * routine. 2704 * 2705 2706 PATCH2 JZ ERROR 043 2707 INX H 361 2708 PDP PSW 303 062 003 2709 JMP IDA		Dermit providing for H89/H8 common PIN
2704 * 2705 2705 2706 PATCH2 JZ ERROR 043 2707 PATCH2 JZ H 361 2708 PDP PSW 303 062 003 2709 JMP IDA	2704 * 2705 2705 2705 2705 2706 PATCH2 JZ ERROR 043 2707 INX H H H H H B B B B B B B B B B B B B B	*	
322 000 2706 PATCH2 JZ ERRUR 2707 INX H 2708 PDP PSW 062 003 2709 JMP IDA	312 322 000 2706 PATCH2 JZ ERROR 043 2707 INX H 361 2708 PDP PSH 303 062 003 2709 JMP IDA	*	
043 2707 INX H 361 2708 PDP PSW 303 062 003 2709 JMP IDA	043 2707 INX H 361 2708 POP PSW 303 062 003 2709 JHP IDA	312 322 000 2706 PATCH2	
361 2708 PDP PSM 303 062 003 2709 JMP IOA	361 2708 POP PSN 303 062 003 2709 JMP IDA	043 2707	
		361 2708 303 062 003 2709	

007.304			2712		XTEXT	8118		/Ram8Go 2/	
			731.4X		RITE	T45 T14 -			
			2715X				200 miles (100 miles (		
			2717X		8118	IS THE SPEC	IFIED BIL IN THE ACCUMULATUR.		
			2718X 2719X	* *	ENTRY:	< ∞	* ORIGINAL A	.0-1-04	
			2720X	* *	EX I T:		A AITH BIT(B) SET	• :	
			2722X 2723X	* *	uses:				
			2724X			•			
007.304	305		2726X	8118	PUSH	<b>35</b>			
007.305	365		2728X		HSDA	M S d			
007.306	076 2	00	2729X		MVI	00000	9		
007.310	400		2730X		XX .	<b>***</b>			
007.312	900		2732X	10110	OCR.	8			
007.313		311 007	2733X		727	81151			
007.316	117		2734X		<b>&gt;</b> 0 <b>K</b>	<b>C.A</b>			
007.317	•••		2736X		<b>P</b> 0P	P S H			
007.320	261		2737X		ORA	J			
32	301		2738X 2739X		PUP	вс			
007.322	311		2740X		RET				
35	:		2741		XTEXT	ZERO		/Ram8Go 2/	
	:								
			2743X 2744X	* *	SZERO	SZERO - ZERO HENDRY	:		
			2745X	* *	\$2ER0	SZERO ZEROS A BLOCK OF	K OF MEHORY.		
			2747X		ENTRY		ADDRESS		
			2748X	* *	TIXI	(8) - COUNT			
			2750x		USES	Ť,			
			2751X 2752X			:		:	
007.323	152		2753X	73	XXX	<b>*</b>			
007.324	167		2754X	*:	) ) ) )	۲,۴		:	:
007.326			2756X		DCR	Ľα			
007.327	302	324 007	2757X 2758X	:	JNZ RET	ZR U1	IF HOKE		

	2781					
	2782	* *	HONITOR	LOWING ARE CO	FOLLOWING ARE CONTROL CELLS AND FLAGS USED BY THE KEYSET Tor.	YSET
000.040	2784		ORG	400004		
040-000	2786	START	SO	7	DUMP STARTING ADDRESS	
400-040	7872	Z Z Z Z	200	7	TA CK CCI LANIKOCI LON	76 790 700
040-040	2789	PRSRAM	E 6		FOLLOWING CELLS INITIALIZED FROM	
040.004	2790		SO	1		
040.005	2612	REGI	SO	1	INDEX OF REGISTER UNDER DISPLAY	
040-006	2793	DSPROT	So		PERIOD FLAG BYTE	
	2795	024400	ŝ	-	DISPLAY MUDE	
040.010	2796	HFLAG	0.8	1	FLAG OPTIONS	
	2798	•			SEE #UU.XXX# BITS DESCRIBED AT F	TRONT
040.011	2799	CTLFLG	0.5	1	FRONT PANEL CONTROL BITS	
040.012	2800	REFIND	0.5		H INDEX (0 TO 7)	
200.000	2801	PRSL	EQU	#-PRSRAM	END OF AREA INITIALIZED FROM ROM	
040.013	2803	FPLEDS	Fou	*	PANT PANEL LED PATTERNA	
040-013	2804	AL FOS	0.5	-	י אמריי ורביי סי	
040.014	2805		os Os			
040.015	2806		0.8	-	ADDR 2	
	2807					
040.016	2808		S0	<b>-</b>		
040-020	2810		200		* 4004	
	2811		3	•		
040.021	2815	DLEDS	SO	-	DATA 0	
040.022	2813		Sa	-	DATA 1	
040.023	2814		ŝ	4		
040.024	2816	ABUSS	0.5	2	ADDRESS BUSS	
040.026	2817	RCKA	0.5	-	RCK SAVE AREA	
040.027	2818	CRCSUM	So	7	6 CHECKSUM	
040-031	6107	XXX	2	7	CAPE ERRUR EXIT ADDRESS	
	2821		ŝ	<b>y</b>		
040.035	2822	REGP TR	0.5	2	REGISETR CONTENTS POINTER	
	2823					
040-037	2824	OI VEC	200	<b>o</b> "	INTERRUP	
040.042	2826		0.5	3	JUMP TO SINGLE STEP PROCESSOR	
040.045	2827		SO	m	T0 I/0 3	
50	2828		SO	<b>.</b>		
040.033	6787		2.0	m (	JUMP TO 1/0 5	
3 6	2630		200	<b>n</b> c		
3:	2832		3	•	₹:	
040.064	2833	NAIRET	0.5	~	Used by H-88/H-89	
•	2834	CTLFL62		-	Control byte for OP2.CTL	/Ram8Go 2/
	2000					

041.120	2837		ORG	41120A			
041.120	2838	<b>A</b> O <b>A</b>	90	-		7007	
041.121		806			Boot Device Flag		
041.122		TIMEDUT		-	į	/Ram860 2/	
041.123		3	So	٦,		/Ram8Go 2/	
	2043	USKLER	2	7	Secondary User Clock for	Soot /Ram8Go	
_	2845		END				
ly complete							
2845 statements							
Verrors detected 26126 bytes free							
			:				
		:					
	:						
		:	:				
	:						
			:				
	:	:	:				
	: : : : : : : : : : : : : : : : : : : :						
		:	:	:			
			:				
			:				

\$ZERO	007323	2015	2753L										
•	007364	5785	7025	8528	8718	9328	9445	9528	9675	9885	10035	10575	21905
4	0.000	2191	2192	21955	2196	2197	27645	2765	766	7106	0100	3040	0300
	010010	7085	2087	77961	0	/10	0.59	0747	1763	1107	6107	0107	0000
A.STX	00000	1136	1151	1341									
A.SYN	000056	1126	1146	1339									
A B OR T	001147	887	975L					:					
ABUSS	040024	848	912	1008	1112	1159	1183	1264	1587	2816L			
NO.014	1 04 1061	151E	1212	1278	0477	8677	7352	7.					
ALFOS	040013	1611	28041	271									
A S . 000	:	409F											
AS. 100		410F											
	00000	411E											
AS.SLM		412E											
AUTOB	004507	19161	5696										
8DA		1966	1978	2436	2471	2839L							
80F	041121	2036	2148	2840L									
BERR	005143	1969	2068	5069	2085L	2175							
BERRI	005170	2095L	2103										
BERRA	005210	2089	2105L	2114									
BERRAL	000011	2002	2114E										
BERRB	000000	2094	2116E										
8H17	006032	2063	2210L										
BH170	006055	2221L	2236										
BH171	000027	2222L	2224	5223			:						
BH172	190900	2225L	2227										
BH173	006103	2231L											
8H174	006126	2222	2225	2240L									
8H17A	006146	2210	2253L	2257									
BH17AL	♦00000	2213	2257E										
BH47	006152	5066	76927										
BH471	006164	2274L	2279										
8H472	008500	2274	2286L										
84473	006211	22.93L	2303				:						
BH474	006311	2336L	2340	, , , ,									
4 / 4 LQ	000924	6977	2345L	7340									
BHTIAL	00000	7177	2340E	17666									
0110	10000	11676	1067	707/7									
15110	0002000	1005	6613										
1000	99790	1050	10571										
A002	004217	1961	1977										
200	004327	1072	1001										
8004	004343	19921	1999										
8005	005046	2027L	2030							:			
8008	005101	2033	2048L										
800A	005125	2040	2060E	2902	2065								
BOOTA	037132	152E	2008										
BOOTAL	000130	1536	2011										
C.DSYN	000375	365E											
CB.CLI	:	119E	215	617	822	1924	1987						
CB.MTL	:	1186	689	773	822	1059	1924						
CB.SPK	0002000	120€	617	922	1214	1924	1987						
10000		•						,	,				
10000	00000	11/6	/19	689	8.22	9401	1022	476T	7961				

1003300   1134   1344   1334		001000	•								
00011	100 000		٥.٠								
005317 21.06 21.65 21.06 21.05 21.06 21.05 21.06 21.05 21.06 21.05 21.06 21.05 21.06 21.05 21.06 21.05 21.06 21.05 21.06 21.05	CKII	005305		2164L							
000231 2.134 2.171 000231 751 000221 752 000221 756 000221 263 000201 134 000201 134 00021 134	CK 12	005317	2160	2165	2168	21721					
0000234 775 7534 0000231 776 7766 000031 776 7766 0000031 776 7766 000003 1316 1950 000003 1316 1950 000003 1316 1950 000003 1316 1950 000003 1316 1950 000003 1316 1950 000003 1316 1950 0000000 1316 1950 000000 1316 1950 000000 1316 1950 000000 1316 1950 000000 1316 1950 0000000 1316 1950 0000000 1316 1950 0000000 1316 1950 0000000 1316 1950 0000000 1316 1950 0000000 1316 1950 00000000000000000000000000000000000	CK I 3	005331	2138	2144	2177L						
000000	CLK2	000234	151	75.5E							
000001 139 139 1311 000001 139 139 130 1311 000001 139 139 139 130 130 130 130 130 130 130 130 130 130		000313		7965							
136   136	CLKINT	005221		21311							
000000 138E 1989 0000001 138E 1989 000000 138E 2893 138E 2893 000000 138E 2893 138E 2893 138E 2893 138E 2893 138E 2893 000000 138E 2893 138E	CLOCK	00000		564	727						
000000 141 2693 000100 142 2693 000100 142 2693 000100 144 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 205 2149 000000 146 206 2149 000000 146 206 2149 000000 146 200 200 200 200 200 200 200 200 200 20	CN-170	1 0000 H	1396	1968							
0001000 1.42E 2.693 0001000 1.42E 2.695 2.149 000000 1.45E 2.062 2.149 0.00000 1.45E 2.062 2.149 0.00000 1.45E 2.149 0.000000 1.45E 2.149 0.00000 1.45E 2.149 0.000000 1.45E 2.149 0.00000 1.45E 2.149 0.000000 1.45E 2.149 0.00000 1.45E 2.149 0.000000 1.45E 2.149 0.000000 1.45E 2.149 0.00000 1	CN-174	M 000003	138E	1980							
0000100 14,12 000020 14,12 000000 14,12 130 130 140 140 140 140 140 140 140 14	CN.ABO	000500	٠,	2693							
000000 1475 2062 2149 000000 1475 2062 2149 000000 1475 2062 2149 000000 1475 2062 2149 000000 1475 2062 2149 000000 1475 2062 2164 1475 2062 2164 1475 2062 2164 1475 2062 2164 1475 2062 2164 1475 2062 2170 1475 2180	CN.BAU		1426								
000000 145E 2065 2164 000001 147E 2065 2164 000012 147E 2061 1465 000001 147E 2065 1467 1405 1427 2618L 000001 147E 2065 1467 1405 1427 2618L 000012 1406 1406 1406 1406 1406 1406 1406 1406	E 10 0		<b>⊣</b> ∵								
000001 147E 2065 2164 0000330 2169 2307 2362L 2553 0005330 2307 2307 000537 2198 2376L 000537 2198 2376L 000537 2198 2376L 000537 2109 2376L 000537 2109 2376L 00050 1119 122 1247 1347 1405 1427 2818L 000304 1410 1224 1866 2834L 00011 152 124 1866 2834L 00011 153 1246 2156 00010 231 201 201 200 00010 231 201 201 200 00010 231 201 201 200 00010 231 201 201 200 00010 231 201 201 201 200 00010 231 201 201 201 201 201 201 201 201 201 20	7		1405	1958	3140						
000000 146			1475	2002	2177						
006339 2307 2319 2362L 2553 006334 2188 236L 006335 1199 124C 006325 140C 040006 1417 1424C 04001 1378 1866 1868 2834L 04001 1378 1866 1868 1868 1868 1868 1868 1868 18	CND		1466	1970	1017						
006334         2198         2367           006335         2199         2367           006337         2169         248           003036         14,0         1421           003037         14,0         1424           003036         14,0         1424           003037         14,0         1424           003037         1194         1244           111         1246         1848         2834           00217         1194         1246         1848         1045           00217         1194         1846         1844         149         1104           00217         1194         1848         2834         1045         1057           00016         187         1866         1868         2834         1045         1046           00016         187         187         1866         188         2834         1046         1057           00016         180         231         2507         2507         2507         2507         2507         2507         2507         2507         2507         2507         2507         2507         2507         2507         2507         2507         2507 </td <td>COM</td> <td></td> <td></td> <td>2319</td> <td>2362L</td> <td>55</td> <td></td> <td></td> <td></td> <td></td> <td></td>	COM			2319	2362L	55					
005357 2369 2376L 005357 1492L 005357 1492L 005356 1406L 147 147 147 147 147 1405 1427 2518L 005356 1406L 1157 1157 137 147 147 1405 1405 10048 1057 005012 1119 1246L 005012 1119 1246L 005015 1938 279L 005015 1938 279L 005016 1939 196E 2332 2401 2507 005016 1939 196E 2332 2401 2507 005017 330E 2332 2401 2507 005017 330E 2312 2401 2507 005018 300L 005019 231 200 00502 250L	COM		2198	23671		`					
902337 1492L 1443 902356 1470 1476 903004 147 1476 904017 1154 1476 904017 1154 1476 904017 1154 1476 904017 1154 1486 904017 1154 1486 904018 14	COMI	•	2369	2376L							
002235 1406L 1426L 1427 1347 1405 1427 2818L 140027 1417 1424L 1405 1427 2818L 1427 1347 1405 1427 2818L 100027 1119 1424L 1405 1427 2818L 100027 1119 1424L 1405 1427 2818L 100021 1417 1424L 1405 1427 2818L 100021 1417 1424L 1405 1407 1407 1407 1407 1407 1407 1407 1407	CRC		1402L	1463							
000027 1154 14241 1405 1427 2018L 004027 1155 1154 1405 1247 1347 1405 1427 2018L 1541 1541 1541 1541 1541 1541 1541 154	CRC1		1406L	1426							
040027 1154 1192 1247 1405 1427 2818L 040011 578 686 744 749 771 776 823 1048 1057 040011 578 686 744 749 771 776 823 1048 1057 040012 1823 1834 1866 1868 2834L 04010 2031 2331 2401 2531 04010 3034 2331 2401 2507 04010 3034 2331 2401 2507 04023 2341 258 2158 04024 2581 2881 04025 2781 2881 04025 2781 2881 04027 2881 2881 04027 2881 2881 2881 2881 2881 2881 2881 28	CRC2		1417	14241							
040011 578 686 774 749 771 776 823 1045 1057 119 1246L 1466 2634L 1578 823 1045 1048 1057 119 1248L 1578 823 1045 1048 1057 119 1923 1988 2799L 1866 2634L 1578 119 1923 1988 2799L 1866 1868 2634L 1900010 2331 2009 2331 2000 235 231 2000 235 231 2000 235 2401 2597 2401 2401 2401 2401 2401 2401 2401 2401	CRCSUM		1154	1192	1247	1347	1405	1427	2818L		
040011 578 686 774 749 771 776 823 1045 1057 1040011 578 686 774 749 771 776 823 1045 1057 1057 1057 1057 1057 1057 1057 105	CTC		6111	12461				:			
040066 1817 1834 1866 1868 000066 1817 1834 1866 1868 0000165 7031 799 1866 1868 0000165 7031 799 1866 1868 10001415 7991 2231 799 1866 1868 10001405 7031 2009 2331 2009 2331 2009 2332 2401 2507 2601 20024 2631 2631 2631 2631 2631 2631 2631 2631	CTLFLG		578	989	7.4	749	771	176			1219 1229
040066 1817 1834 1866 1868 040145 7931 799 040141 3041 2231 040145 3091 2231 040140 3091 2331 2009 040241 2331 2009 040243 2821 2155 040251 2681 2155 040252 2681 2156 040252 2681 2156 040262 2781 2156 040272 2801 2156 040273 2811 2811 040251 2781 040271 2811 040251 2781 040271 2811 040271 2811 040271 2811 040271 3131 040131 3131	- :	• •	1923	1988	2799L						:
040141 304L 2231 040141 304L 2231 040140 399L 2372 040235 324L 2009 040235 324L 2155 040244 262L 2155 040244 262L 2155 04024 262L 2156 04024 262L 2156 04024 262L 2156 04025 260L 2156 04026 272L 281L 04027 280L 281L 04027 281L 281L 04027 281L 281L 04027 283L 04027 333L 04027 333L 04027 333L 04027 333L	CILFLG		1817	1834	1866	1868	2834L				
040140 309L 2231 040140 309L 2331 040240 339E 2372 2401 040244 2631 040244 2631 040244 2631 040244 2631 040251 268L 040262 260L 2156 040267 290L 040262 280L 040262 280L 040272 280L 040265 270L 040273 283L 040273 283L 040273 283L 040273 283L 040273 283L 040273 333L 040273 333L 040273 333L			7031	66.6							
040110 233L 2009 000001 380E 2372 2401 04024 253 234 04024 263L 2155 04025 23 260L 2156 04025 23 260L 2156 04025 23 200L 2156 04027 290L 270L 04027 290L 270L 04027 290L 270L 04027 290L 270L 04027 293L 281L 04027 283L 04027 333L 04027 333L 04027 333L 04027 333L 04027 333L	D.COF		304	1627							
040243 040235 040244 040244 040244 040251 040252 040252 040262 040262 040262 040262 040262 040262 040272 040272 040272 040273 040274 040177 040174	D.CON		2331	2009							
040235 324L 040244 263L 040253 262L 2155 260L 040263 260L 040242 260L 040264 270L 040272 280L 040272 281L 040273 283L 040273 283L 040273 281L 040273 281L 040273 281L 040273 281L 040273 283L 040273 281L 040273 281L 040273 281L 040273 281L 040273 281L 040273 281L 040274 272L 040273 270L 040133	D.DATI		380E	2372	2401	2507					
040244 2631 2155 040243 2621 2155 040163 3101 2156 040257 2601 2156 040265 2781 2801 040270 2801 2801 040271 2801 2801 040271 2811 2811 040271 2811 2811 040271 2811 2811 040271 2811 2811 040271 2811 2811 040271 3131 3131	D.0LY		3241								
040243 262L 2155 040251 268L 040163 310L 040267 260L 040267 279L 040270 280L 040265 278L 040271 281L 040271 281L 040273 281L 040271 281L 040271 281L 040273 281L 040273 383L 040273 383L 040273 383L 040273 383L 040273 383L 040273 383L 040274 313L	D.OLYH		763L								
040251 268L 040163 310L 040163 310L 040242 260L 040252 280L 040270 280L 040272 281L 040273 281L 040273 283L 040273 283L 040271 281L 040273 283L 040273 313L 040177 313L 040177 313L	D.DLYM		729Z	2155							
040163 310L 040242 260L 2156 040267 279L 040265 277L 040265 277L 040272 281L 040271 281L 040273 281L 040273 281L 040273 281L 040273 281L 040273 313L 040177 314L 040177 314L	D.ORVI		268L								
040242 260L 2156 040267 279L 040267 278L 040265 277L 040270 280L 040271 281L 040271 281L 040273 283L 040273 283L 040273 270L 040177 314L 040177 313L	0.018		310L								
040267 279L 040270 280L 040265 278L 040272 282L 040273 283L 040273 283L 040273 283L 040273 283L 040273 283L 040273 283L 040273 283L 040273 283L 040177 314L 040177 313L 040174 313L	D.DVCT		709Z	2156	2158						
040270 280L 040266 277L 040265 278L 040273 283L 040273 283L 040273 283L 040273 313L 040177 314L 040177 313L 040177 313L	D.E.CH		279L								
040265 2771 040265 2771 040273 2831 040273 2831 040273 2831 040273 3131 040177 3141 040177 3121 040171 3121	0.E.HC		280L								
040272 2831 040272 2831 040273 2831 040273 2831 040273 2831 040177 3141 040177 3131 040177 3131 040137 3131	U.E. H3		אואל ל								
040271 281L 040273 283L 040273 283L 040261 276L 040177 314L 040177 313L 040174 302L	7		200								
040265 276L 040273 283L 040281 273L 040261 270L 040177 314L 040174 313L 040174 313L			2811								
040273 283L 040232 323L 040261 270L 040177 314L 040174 313L 040174 313L	O FRP		2261								
040232 323L 040261 270L 040177 314L 040171 313L 040133 302L	D.ERRL		2831								
040261 270L 040177 314L 040171 312L 040174 313L 040133 302L	D.ERRT		32.3L								
040177 314L 040171 312L 040174 313L 040133 302L	D.HECN		270L								
040171 312L 040174 313L 040133 302L	D.LPS	04017	314L								
040174 313L 040133 302L 040264 2721	D.MAI	<b>-</b> :	312L								
040133 302L 040264 2721	D.MAO	0	3131								
OFCNT 040264 2721	DON O	0	302L								
פרכעו פוסנס	•OEC	040	1212	2528							

040273 287L 040275 288L 040247 236L 040247 236L 040203 315L 040205 316L 040205 316L 040215 306L 040215 306L 040216 311L 040213 306L 040213 311L 040214 306L 040227 326L 040215 310L 040216 310L 040217 258L 040218 310L 040218 310L 040219 310L 04021 3	2287 2498 2612 2617 2644 2656
140275 288L 140240 236L 2014 140202 315L 2235 140162 311L 2235 140163 311L 2219 140202 310L 2219 140202 317E 380 2167 140210 317E 380 2167 140210 317E 380 2167 140210 317E 380 2167 140210 317E 299 140210 317E 299 140210 317E 299 140210 317E 299 140221 320L 299 140221 430L 299 14	2498 2612 2617 2644
140240 236L 2014 402037 236L 2013 40167 305L 2235 40152 315L 2235 40160 317L 306L 310L 402013 307L 2235 401000 379E 380 2167 402013 307L 2239 402013 316L 2239 402013 317L 308L 402013 301L 2398 402013 302L 299 402013 302L 299 402013 302L 299 402013 456L 299 402013 456L 290 4000014 430L 2308 400001 440L 430L 2308 400001 440L 430L 2308 4000001 440L 430L 2308 400001 440L 430L 230R 400001 440L 430L 230R 400001 440L 430L 230R 400001 440L 430L 230R 4000001 440L 230R 400001 440L 230R 4000001 440L 230R 40000001 440L 230R 4000001 440L 230R	2498 2612 2617 2644
140202 140102 140102 140102 140102 140102 140102 140102 140102 140102 140102 140103 14	2498 2612 2617 2644
740147 3054 2235 740152 3071 2235 740152 3071 2219 740165 3111 2219 740165 3111 2219 740262 3796 380 2167 740213 3111 299 740213 3111 299 740213 3111 299 740213 3111 299 740213 3111 299 740213 313 2322 740214 3034 2405 740215 310 2313 2322 740216 3111 299 740216 3111 299 740216 3111 299 740216 3111 299 740216 3111 299 740216 3111 2308 740216 3311 2322 740216 3311 232	2498 2612 2617 2644
140152 3021 2219 140152 3161 2219 140165 31161 2219 140165 31161 2219 140165 31161 3181 3181 140213 3011 3011 140245 2651 299 140241 2571 2391 2322 140241 2571 2391 2322 140241 2571 3201 2313 2322 140241 3021 3201 2313 2322 140241 3021 3201 3201 3201 3201 3201 3201 320	2498 2612 2617 2644
1000000	2498 2612 2617 2644
140166 1711 140262 1712 1700000 1779 1771 1770 1770 1770 1770 1	2498 2612 2617 2644
7402 62 2711 7402 62 379E 380 2167 7402 13 313E 380 7402 13 311E 380 7402 13 313E 280 7402 45 265E 299 7402 46 315E 299 7402 47 266E 299 7402 47 27 27 27 27 27 27 27 27 27 27 27 27 27	2498 2612 2617 2644
1000000   379E   380   2167   380210   3171   3181   380212   3181   3	2498 2612 2617 2644
3.77	<b>****</b>
11.1.  140.2.13  13.11.  140.2.45  140.2.45  140.2.47  1	
3481, 460133 3181, 460130 3181, 460130 3181, 460130 3011, 46014, 4	
740245 740245 740245 740245 740246	
2581 40241 2581 40241 2512 2512 2513 2513 2514 2514 2514 2514 2515 2515 2516 2516 2516 2517 2517 2517 2518 2518 2518 2519 2519 2519 2519 2519 2519 2519 2519 2510	
7,902.41	
740240 2571 740216 3151 740216 3151 740217 2561 740227 3221 740221 3221 740221 3221 740221 3221 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 3231 740221 7431 740221	
7,0216 319L 299 2,021 235L 299 2,022 2 322L 299 2,022 3 322L 320L 2,022 322L 320L 2,022 322L 323L 2,022 322L 333L 2,022 322L 323L 2,022 322L 333L 2,022 322L 333L 2,022 322L 333L 2,022 322L 323L 2,022 322L 333L 2,022 4,22L 230B 2,022 4,22L 230B 2,022 4,22L 230B 2,022 4,22L 232L 2,022 4,22L 232L 2,022 4,22L 232L 2,022 4,22L 232L 2,022 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
74,0130 2351 299 74,0130 2351 299 74,0127 3661 299 74,0155 3081 2313 2322 23062 74,0136 3031 2332 23962 74,0136 3031 2332 23962 74,0136 3031 2332 23962 74,0136 3031 2332 23962 74,0136 377 2398 24051 75,000013 4531 2308 75,000014 4301 431 2308 75,000015 4541 2308 75,000016 4511 2308 75,000017 431 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 4191 2554 75,000001 75,000001 75,000	
240247 246L 240227 322L 240155 320L 240221 320L 240221 320L 240221 320L 240221 321Z 24022 233Z 240213 233Z 2402L 2403L 2504 2600C 2504 2600C	
340227 340227 340221 340221 340221 340221 340224 340224 340222 34032 340325 340	
240155 308L 240221 320L 34024 341L 340136 303L 340136 303L 303L 303L 303L 303L 303L 303L 30	
240221 240221 240224 240144 23031 240144 23031 24031 2	
0.0000	
740144 3051 240144 3051 240144 3051 240136 3051 3051 3051 3051 3051 3051 3051 3051	
2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2313 2322 2310 2310	
006355 2313 2322 006355 2313 2322 006355 2333 2322 006377 2398 2405L 0060013 453L 0060014 430L 0060015 453L 0060017 431L 0060017 421L 0060018 452L 0060019 452L 0060019 452L 0060010 452L	
006355 2310 2312 2322 006355 2310 2313 2322 23964 2006377 2398 24054 2000013 4294 2322 2308 2000014 4314 2308 2000014 4314 2308 2000015 4524 2320 2320 2320 2320 2320 2320 2320 23	
0006361 2193 2332 006377 2398 2405L 0000013 429L 0000014 430L 0000015 431L 0000016 432L 0000017 431L 0000017 431L 0000018 431L 0000019 431L 0000019 431L 0000010 431L 0000010 431L 0000010 431L 0000010 431L 0000010 431L 0000001 4419L 00000201 4419L 00000201 4411	2391L
0000377 2398 0000000 4291 0000014 4291 0000014 4301 0000016 4301 0000016 4301 0000016 4301 00000016 4301 00000017 4211 000000001 4211 00000001 4211 0000001 4211 0000001 4211 0000001 4211 0000001 4211 0000001 4211 000001 4211 000001 4211 000001 4211 000001 4211 000001 4211 000001 4211	
000020	
9000013 459L 9000202 453L 900014 430L 900015 431L 9000017 432L 9000003 452L 9000003 450L 9000004 450L 9000007 450L 9000001 419L 9000001 419L 9000001 440L 900002 440L	
0000202 000014 431L 0000015 431L 0000015 431L 0000003 421L 0000002 421L 0000002 421L 0000002 421L 0000002 421L 0000007 421L 0000001 431L 0000001 431L 0000001 431L 0000001 431L 0000001 421L 000002 4401L	
000014 0000015 0000016 0000016 0000003 0000002 0000002 0000002 0000007 0000007 0000000 0000000 0000000 0000000 000000	
000015 431L 000016 432L 0000017 421L 0000003 421L 0000002 422L 000002 454L 00002 05 454L 0000007 425L 0000001 454L 000001 451L 000001 451L 000002 434L 000001 451L	
000016	
0000017 +33L 000003 +21L 0000004 +22L 00000004 +22L 0000203 +56L 0000203 +55L 0000007 +25L 0000001 +25L 000001 +25L 000020 +42L 000020 +42L 000022 +42L	
000003 +21L 0000004 +22L 0000002 +56L 0000003 +56L 0000003 +56L 0000003 +56L 0000001 +25L 0000001 +36L 0000001 +36L 0000001 +40L 000002 +40L	
000004 422L 000002 420L 000002 456L 0000203 456L 0000007 423L 0000001 431L 0000001 451L 000002 440L 00002 440L	
000002 +20L 000203 +50L 0000203 +54L 0000007 +55L 0000001 +19L 0000200 +40L 000020 +40L	
000205 000203 0000203 0000007 0000007 000020 000020 000020 000020 000020 000022 000022 000022 000022	
0000203 454L 000007 423L 0000020 423L 0000001 419L 000020 440L 000020 441L 000022 441L	
000022 431 000000 434 0000001 434 0000001 434 0000001 4511 000020 4401	
000007 425L 000020 434L 000001 419L 000020 440L 000021 441L 000022 442L	
000000 434 000000 4194 000000 4401 000000 4401 000000 4401	
000001 419L 000000 451L 000020 440L 000021 441L	
000020	
000021	
000021	
000021	
000022 44	
000003	
-	
. ~	
•	
000210	
•	
000011	

4551. 45	
000000	
000000	
0000001 334E 2245 2535 000001 334E 2247 2538 000001 334E 2247 2538 0000001 334E 2247 2538 0000001 334E 2247 2538 0000002 334E 2248 2248 0000002 334E 2248 2248 0000002 334E 2248 2248 0000002 334E 2157 0000000002 334E 2157 000000000002 334E 2157 000000000000000000000000000000000000	
00000040 331E 2245 2534 0000002 339E 22245 2538 0000010 339E 2223 2226 0000010 334E 2223 2226 0000002 334E 2223 2226 0000002 334E 2223 2226 0000003 334E 2154 2812 0000004 333E 2154 2812 0000007 334E 2157 2812 0000001 334E 2157 2812 0000002 334E 2157 2812 0000002 334E 2157 2812 0000003 342E 2157 2812 0000003 336E 2248 2258 0000003 336E 3812 00000003 336E 3812 0000003 336E 3812 00000003 336E 3812 000000	
000002 337E 2245 2535	
0000004 338E 2246 2535 0000100 334E 2227 2226 0000100 334E 2248 0000100 342E 2288 0000100 342E 2288 0000100 342E 2288 0000100 342E 2288 0000100 342E 2288 0000100 342E 2288 0000100 343E 2157 0000000 133E 1763 0000000 133E 1763 0000000 134E 1356 2415 1584 1926 0000000 134E 1856 341 1295 1511 1902 000000 1813E 1856 0000000 1813E 1856 000000 251E 250 000000 251E 250 000000 251E 250 000000 251E 250 000000 251E 253 000000 251E 253 000000 251E 253 000000 251E 253 000000 251E 253 0000000 251E 253 000000000000000000000000000000000000	
0000010 3391 2224 2226 0000010 3346 2246 2256 0000010 3346 2246 2256 0000010 3346 2246 2256 0000010 3346 2246 2256 0000010 3346 2246 2256 0000010 3346 2246 2274 0000010 3346 2246 2274 0000001 124 28121 163 1615 1618 000120 1346 1356 1549 2250 000001 1346 1356 1549 2250 000121 124 1396 24151 1295 1511 1902 000122 1336 1336 1336 1336 1336 1336 1336 1	
0000010 343E 2246 22246 2000010 344E 2246 22246 2000010 344E 22246 22240	
0000100 3340E 2278 0000100 334E 2157 0000001 335E 2812L 0000001 335E 2812L 0000001 335E 2812L 0000001 335E 2812L 0000002 132E 2812L 0000002 132E 1763 0000002 134E 1763 0000002 135E 1763 000002 135E 1763 0000002 135E 1763 00000002 135E 1763 00000002 135E 1763 0000000002 135E 1764 000000002 135E 1764 000000000000000000000000000000000000	
000000 342E 000000 342E 0000000 343E 000000 342E 0000000 343E 000000 342E 0000000 343E 000000 342E 0000000 343E 000000 343E 000000 343E 000000 343E 000000 343E 0000000 343E 000000 343E 000000 343E 000000 343E 000000 343E 0000000 343E 000000 343E 0000000 343E 000000 343E 0000	
000000 332E	
000001 335E 000000 333E 0000001 343E 2157 040021 1624 2812L 000002 133E 1763 000002 133E 1763 000002 133E 1763 000002 134E 1613 1615 1618 000122 1535L 1613 1615 1618 000127 1538L 1543 1740L 0003342 1538L 1543 1740L 000312 1538L 1544 1956 000312 1538L 1613 1615 1618 000177 329E 1594 1730L 1594 1905 000177 329E 1730L 2216 2249 2250 000177 329E 1730L 1730L 1996 000177 1996 2415L 2270 2803E 000100 1114L 1992 2211 2270 2803E 001222 2096L 1994 2415 1295 1511 1902 001222 1994 1996 2415L 2270 2803E 001222 1994 1996 2415L 2270 2803E 001222 1994 1995 2415L 2270 2803E 001222 2706 001222 1994 1995 2415L 2270 2803E 001222 2706 001222 2706 001222 2706 001222 2706 001222 2706 001222 2706 001222 2706 001223 2706 001223 2706 001274 880 1914E 1855 001274 01000 001177 880 1018L 2497 2506 2611 2616 2643 000000 25140 2531 2531 2531 2500 2611 2616 2643 0000107 2531 2531 2531 2531 2500 2611 2616 2643 0000107 2531 2531 2530 2611 2616 2643 0000107 2531 2531 2530 2611 2616 2643	
0000004 0000004 0000000000000000000000	
0000200 343E 2157 0000201 1624 2812L 000002 131E 1763 0000002 131E 1763 0000002 133E 1763 000002 134E 1763 000122 1536 1763 0003122 1536 1536 1540 0003127 1536 1549 2250 000312 1536 1536 1549 2250 000312 1536 1536 1549 1956 000312 1536 1536 1536 1536 1569 000312 1536 1536 1536 1536 1569 000312 1536 1536 1536 1536 1569 000312 1536 1536 1536 1536 1569 000312 1536 1536 1540 1556 000312 1536 1536 1540 1556 000312 1536 1536 1540 1556 000000 843 934 1005 1578 1584 1926 0000002 1546 1556 941 1295 1511 1902 00122 1594 1902L 00122 1594 1902L 00122 1594 1037 1037 1037 1037 1037 1037 1037 1037	
040021 1624 2812L 0000053 669L 1686 2278 0000002 131E 1763 000002 131E 1763 000122 1535L 1613 1615 1618 003122 1535L 1540 2250 003122 1536 1543 1740L 003127 1536 1544 1926 003127 1536 1544 1926 003342 1536 1546 1546 1584 1926 000322 1934 1005 1578 1584 1926 0002002 1944 1996 2415L 000202 1944 1996 2415L 000202 1944 1996 2415L 000202 1944 1900 2511 2270 2803E 000202 1813E 1856 000202 1813E 1856 000122 193L 1037 000122 193L 1216L 1037 000003 1813E 1856 000003 1813E 1856 0000043 2131 1216L 2549 2250 000122 1944 1996 2415L 2270 2803E 000202 193L 1037 000003 1813E 1856 0000043 2131L 1216L 2616 2643 0000063 1813E 1856 0000073 253 554 636L 640 1874	
000053         609L         1686         2278           000000         131E         1763           000000         131E         1763           000000         131E         1763           000000         134E         1763           000000         134E         156           000000         134E         156           000000         134E         1536           000000         134E         1543           000000         134E         1543           00000         134E         1740L           00000         134E         1740L           0000         134         1536           1536         1256         2216           1536         1750         1546           1700         1144         1996           1700         1144         1996           1700         1144         1996           1700         1144         1996           1700         1144         1996           1700         1144         1996           1700         1144         1996           1700         1144         1996           1700         114	
000000         131E           0000001         132E           0000002         133E           0000003         134E           0000003         134E           0000003         134E           0003127         153BL           153BL         1556           003342         153BL           153BL         1556           153BL         1570           153BL         1576           153BL         1576           153BL         1740L           154BL         1756           155A         1576           154BL         1996           154BL         1937           155A         151           155A         1706           157BL         1524           157BL         1524           157BL         1524           157BL         1524           150000         1814           152BL         17	
0000001         132E         1763           000002         133E         1763           000002         134E         1763           003122         1536         1615         1618           003122         1536         1536         1543         1740L           003135         1536         1536         1536         1536           000177         329         1539         2250         2249         2250           000177         329         1730L         2249         2250         934         1956           0001707         839         852         871         1958         1968         1968         1968         1968         1976 <td></td>	
000002     133E     1763       000003     134E     1053       0003127     1538L     1613     1618       003127     1538L     1554     1618       003127     1538L     1554     1740L       003356     1536     1549     2259       0000177     329E     2216     2249     2250       0000332     1144L     1996     2415L     1926       0000002     1144L     1996     2415L     1926       0000002     1144L     1996     2415L     1926       0000002     1144L     1996     2415L     1902       000002     1144L     1996     2415L     1902       0001222     1144L     1890L     2415L     1902       0001222     1896     2211     2270     2803E       001222     187     1037     1037     1037       0000023     2198L     1855     1706       100000     1814E     1855     1706       100000     1814E     1855     1706       100000     1814E     1855     1706       1001067     2198L     1218L     2497     2506     2611     2616       002140     1221 <t< td=""><td></td></t<>	
0000003     134E     1613     1618     1618       003122     1535L     1613     1618       003127     1538L     1556     2249     2250       000177     329E     2159     2216     2249     2250       0000177     329E     2159     3216     2249     2250       0000177     329E     2159     3216     2249     2250       0002002     843     852     871     934     952       007002     1144L     1996     2415L     1926       007002     1144L     1996     2415L     1926       1 007002     1994     1996     2415L     1906       1 184E     2527     2109     2211     2270     2803E       1 000012     270     2803E     2803E     2803E       2 000012     2807     2211     2270     2803E       3 00012     2807     2191     1037     1037       0 00006     1814E     1855     1706     1816       4 0006     1817     2497     2506     2611     2616     2643       0 002     1231     2497     2506     2611     2616     2643       0 000     551     554	
003122         1535L         1613         1615         1618           003127         1538L         1556         1556         2249         2250           003127         1538L         1556         2216         2249         2250           000177         329E         1730L         2216         2249         2250           000017         839         852         871         932         934         1956           0000000         839         852         871         1958         1956<	
003127         1538L         1556           0003356         1536         1549         2250           000177         1536         2159         2250           000177         1534         2216         2250           0003342         1534         1005         1578         1584           040007         843         934         1005         1578         1584           040006         843         934         1005         1578         1584         1926           040006         843         934         1005         1578         1584         1926           040006         843         934         1005         1578         1584         1926           040012         1344         1996         2415L         2415L         2415L         1902           04012         1344         1932         941         1295         1511         1902           04012         2705         879         1037         1037         1037         1037         1037         1037         1038         1038         1038         1048         1048         1048         1048         1048         1048         1048         1048         1048	
000177 3296 2159 2216 2249 2250 003342 1594 1730L 040007 843 934 1005 1578 1584 1926 040006 843 934 1005 1578 1584 1926 002002 1144L 1996 2415L 000012 1994 1996 2415L 000012 792 816E 941 1295 1511 1902 0004160 861 1890L 0004177 1892 2000 2211 2270 2803E 0004177 1892 1037L 000013 2879 1037L 000003 2193L 001000 1813E 1855 010000 1813E 1855 010000 23 2193L 001067 2431L 1228 007004 2431L 2249 2506 2611 2616 2643 000013 553 554 636L 640 1874	
0000177         329F         2159         2216         2249         2250           0403342         1594         1730L         871         932         934         952           040006         843         934         1005         1578         1584         1956           040006         843         934         1005         1578         1584         1956           0002002         1144L         1996         2415L         2587         934         1956           0007002         1994         1996         2415L         2415L         1966         2415L         1966           000012         1894         1996         2415L         1996         2415L         1992           000417         1892         1902L         2090         2211         2270         2803E           000417         1892         1037         1037         1037         1037         1037         1037         1036         1196         2497         2506         2511         2516         2643         2643         2643         2643         2643         2643         2643         2643         2643         2643         2643         2643         2643         2643	
000342 1594 1730L 671 932 934 952 040006 839 934 1005 1578 1584 1926 002006 1843 934 1005 1578 1584 1926 002006 1944 1996 2415L 2527 2415L 2527 2415L 2527 2415L 2625 200012 2415L 2625 2090 2211 2270 2803E 240013 2022 2090 2211 2270 2803E 24000063 617L 1037 2403 2403 2403 2403 2403 2403 2403 2403	
040007 839 852 871 932 934 952 040006 843 934 1005 1578 1584 1926 002002 1944 1996 2415L 000012 154E 2527 816E 941 1295 1511 1902 000322 770 816 941 1295 1511 1902 000322 770 861 1890L 004160 861 1892 2090 2211 2270 2803E 040013 2022 2090 2211 2270 2803E 0400107 2431L 2497 2506 2611 2616 2643 0400107 880 1018L 0400107 880 1018L 0400107 649L	
004006 843 934 1005 1578 1584 1926 007002 1144L 996 2415L 1995 1511 1902 000012 154E 2527 816E 941 1295 1511 1902 000122 2706 861 1890L 2270 2803E 2001222 879 1037L 2270 2803E 2001222 879 1037L 2270 2803E 200003 617L 1037 1855 1511 1902 2001222 879 1037L 1285 1108L 2616 2613 2616 2613 2616 2613 2616 2613 2616 2613 2616 2613 2616 2613 2616 2613 2616 2613 2616 2613 2616 2613 2616 2611 2616 2613 2616 2611 2616 2613 2610 2611 2616 2613 2611 2616 2613 2611 2616 2613 2611 2616 2613 2611 2616 2613 2611 2616 2613 2610 2611 2616 2613 2613	:
0002002 1144L 1996 2415L   0007002 1944 1996 2415L   000322 792 816E 941 1295 1511 1902   000322 2706 861 1890L   004160 861 1890L   004177 2022 2090 2211 2270 2803E   001222 879 1037L   000063 617L 1037   000063 617L 1037   000063 617L 1037   000063 2193L   001067 916L   001067 2431L 1228   007004 2431L 2497 2506 2611 2616 2643   000073 553 554 636L 640 1874   000107 649L   000007 649L   0000007 649L   000007 649L   000007 649L   000007 649L   000007 649L   0	2793L
0000012 1546 2527 2111 1902 000012 792 8166 941 1295 1511 1902 0004160 861 1890L 0004177 1892 1902L 0004177 1892 1902L 000122 2090 2211 2270 2803E 001022 879 1037L 001022 879 1037L 001000 1813E 1855 010000 2193L 001067 2193L 001067 2431L 1224 1706 002140 1227L 1228 007004 2431L 2497 2506 2611 2616 2643 0000073 553 554 636L 640 1874	
000032         797         816E         941         1295         1511         1902           0004160         861         1890L	
2706 004160 861 1890L 0004177 1892 1902L 040013 2022 2090 000063 617L 1037 000063 617L 1037 000063 1813E 1855 0006023 01814E 1855 0006023 001000 1814E 1855 0006023 001000 2431L 1228 007004 2431L 2497 2506 2611 2616 2643 00107 649L 000107 649L	1005 2052 2008 2102 2124
0004160 861 1890L 004177 1892 1902L 0054177 1892 2022 2090 2211 2270 2803E 001022 8679 1037 000063 6174 1037 030000 1813E 1855 006027 2198L 1855 006027 2198L 1524 1706 002140 1213L 12497 2506 2611 2616 2643 007004 2431L 2497 2506 2611 2616 2643 001077 880 1018L 640 1874	•
0000122 2090 2211 2270 2803E 040013 2022 2090 2211 2270 2803E 0500122 879 1037 050005 1813E 1856 010000 1814E 1855 006023 2193L 001067 2193L 002140 1213L 1524 1706 002140 1227L 1228 007004 2431L 2497 2506 2611 2616 2643 001077 880 1018L 000000 551L 000107 649L	
000122 879 1037 000122 879 1037 000122 879 1037 000063 617L 1037 030000 1813E 1856 000602 2193L 1554 0001067 2193L 1524 1706 002140 1213L 1524 1706 002140 1227L 2497 2506 2611 2616 2643 00177 880 1018L 640 1874	
0000522 879 1037L 000063 617L 1037 030000 1813E 1856 006027 2198L 1855 006027 2198L 1855 006023 2193L 1524 1706 002140 1213L 1216L 2606 2611 2616 2643 007004 2431L 2497 2506 2611 2616 2643 007004 2431L 2497 2506 2611 2616 2643 00177 880 1018L 640 1874	
0000063 617L 1037 030000 1813E 1856 010000 1814E 1855 006027 2198L 006023 2193L 1524 1706 002140 1213L 1224 1706 002140 1227L 1228 007004 2431L 2497 2506 2611 2616 2643 0000073 553 554 636L 640 1874	
030000 1813E 1856 010000 1814E 1855 006023 2193L 006023 2193L 001067 916L 002140 1213L 1524 1706 002160 1227L 1228 007004 2431L 2497 2506 2611 2616 2643 001077 880 1018L 0000073 553 554 636L 640 1874	
010000 1814E 1855 006027 2198L 006023 2193L 002140 1213L 1524 1706 002140 1227L 1226 002140 1227L 1226 00177 880 1018L 2497 2506 2611 2616 2643 000073 553 554 636L 640 1874 0000107 649L	
006027 2198L 006023 2193L 001067 2193L 1524 1706 002143 611 1216L 1706 002160 1227L 1228 001077 880 1018L 2497 2506 2611 2616 2643 001177 880 1018L 640 1874 000000 551L 649L	
000003 2193L 001067 916L 1524 1706 002143 611 1216L 002160 1227L 1228 007004 2431L 2497 2506 2611 2616 2643 00177 880 1018L 001077 553 554 636L 640 1874 000000 551L	
002100/ 002143 1211 1524 1706 002140 1227L 1228 007004 2431L 2497 2506 2611 2616 2643 001177 880 1018L 001070 553 554 636L 640 1874 000000 649L	
002140 1211 1224 1700 002143 1211 1228 007004 24311 2497 2506 2611 2616 2643 000007 553 554 6361 640 1874 000000 5511 6491	
002145 011 1228 007004 2431L 2497 2506 2611 2616 2643 007004 2431L 2497 2506 2611 2616 2643 000177 880 1018L 640 1874 2643 0000107 649L	
007204 2431L 2497 2506 2611 2616 2643 001177 880 1018L 640 1874 0000073 553 554 636L 640 1874	
001107	37.4
NIT 000073 553 554 636L 640 NITO 000000 551L NITI 000107 649L	
NITO 000000 551L NITI 000107 649L	
1649 201000	
000117 656	
000010 558E	
76	
3 000030	

INT	0.0000	595L						
INTS	00000	7009						
INT6	090000	9I4I						
INT /	02000	951L	1	1				
INTXIT	000172	7101		1050				
TOA	290800	6001	1492					
907	003066	506	7647	12011				
1001	003070	12001	2767	1776	2776	34.74	15.46	
LOWKK	200040	1024	6701	1447	0467	0/47	1047	7,017
1 P - CUN	000362	107	0.00	1955	1300	5071		
Ir.rad	000360	7,70	1311	1600	1307	6007	770	7617
14.14.	000371	1035	1161					
17.170	000370	105	1381					
K.DIVD	000117	174E						
K.DOT	000017	176E						
K.RINC	000217	172E						
K.NUMB	000021	175E						
K.PLUS	000257	171E						
K.STAR	000157	173E	2101	2133				
LAST	001100	988	983L					
LOVO	001272	1084	1127					
LOA1	001342	1110	1117					
1.040	001267	10826						
1 8 4	003047	919	14771	1592				
¥ 0 .	003052	7.85	1101	1171	14781			
L C T 2	003022	500						
7	242355	3775						
M. OHTI	243355	374F						
NT NT	001165	9.80	1007					
177	000346	207F	1220					
	0000	2000	701					
	000700	2035	8101	2440				
		2002	2001					
	000303	204E	7661					
	•	2007						
HI-LXID		2005	1019	37.76				
	000323	200E	0701	6413				
MI-KEI	000311	202E	9971		, , ,			
MSGLEN	000003	<b>\$02</b>	2025	2073E	9/07			
MSGPRI	005135	1954	2072L	2073				
M S C S E C	041600	7,44	2072L	9707				
¥ (	000344	834E	1000					
HIKI	000345	837	837					
* X * X	00100	000	980					
ALK?	160100	825	89.9L					
A K	2/0100	106	9186					
4 × ×	00103	100	0 0 0	:			: : : : : : : : : : : : : : : : : : : :	
MMIDET	040064	2832	1966					
	40000	23.71	0046	17776				
080	60700	2166	2286	24611				
) a C	045000	• :-	1047	1056				
		1000	1836	0001	1876	1881	2692	
מו יייי		0.01	0007	1001	0	1001	7.07	
910.10		1016	120					
01.00 00	:	102			700		0 , 7 ,	60.76
08-18C	175000	• •	6411	1203	1671	1377	0047	/997
2	•	1000	7041					

								•				
PATCHI	007254	099	2682E									
ATCH2	007274	921	2706L									
Z I d	990200	916	2336	2497L	2500	2555						
PRI 800	00425	1903	1930	1952L								
PRIBUU.		1953L 1872	28015									
PRSRAM	040004	1872	2789E	2801								
RSROM	003371	1759E	1871									
R S H	001126	886	9531									
. SDP	007111	2218	2527L									
R. SOP.		155E	2533	2539								
RAMBGO	000000	548E	551	1833	1844	2776						
LAMBGOL		1843	2776E									
¥	003260	₩.	938	1508	1677E							
RCK1	003267	9	1691	1703								
CK2	003310	1689	1695									
CK3	003326	1698	1704									
KCKA	040056	1681	2817L									
CEFIND	040012	642	2800L									
KEG I	040002	870	***	196	988	1477	72627					
	001100	890	9311									
אוכר	040032	693	825	1480	1928	2822L						
RMEM	001261	683	1066L									
RNB	002331	1110	1337	1362	1376L							
× 181	002335	1378L	1380									
RNP		1096	1106	1246	1348	1362L						
ROMBOOT		228E										
ROMCLK	034031	156E	2003									
RDY	007134	2293	2295	2553L								
RT.80	00000	185E										
19.1	200000	182E										
7	50000	1036										
	10000	2707	1003	6611								
01	10000	1040										
•	00000	183E	2400	2502	2613	2446						
SOTE	000200	385	2400	2657								
ER	00000	382E	2618									
S.GRTO	02 40 00	224E										
•:	:	225E										
S.GRT2		226E										
S. IEN	000100	384€										
INT.	040343	238L										
•	961140	7047	747									
0 8 2 8 0	70000	38/E										
CHO.		2000										
5.5 KB	07000	3005										
7	040277	227										
	00000	160E	2105	2106	2107	2108	2111	2112	2113	2253	2342	7364
SI	00000	1616	2111	2256	2345							
25	♦00000	162E	2108	2253	2255	2256	2342	2344	2345			
	010000	163E	2105	2106	2107	2108	2253	2255	2256	2342	2344	2345
S.4	00000	164E	2105	2106	2107	2111	)					
\$5	00000	1656	2105	5106	2107	2111	2112	2113	2253	2342		
26	0001000		2105	2111	2253	2342	2344					

0.001063   9121   976   9784   1028   1000001   946   9784   1028   9784   1028   9784   1028   9784   1028   9784   1028   9784   97	
000001 4046 000000 000000	
0000001 +004E 0000002 +003E 0000002 +003E 0000002 +003E 0000037 +000E 0000037 +000E 0000037 +000E 0000037 +000E 0000000 +74E 0000000 +74E 000000 +74E 0000000 +74E 000000 +74E 0000000 +74E 000000 +74E 000000 +74E 000000 +74E 000000 +74E 000000 +74E 000000 +74E 0000000 +74E 00000000 +74E 0000000 +74E 00000000 +74E 0000000000 +74E 000000000000000000000000000000000000	
0000000 3997E 0000000 4000000 4000000 400000 400000 400000 400000 400000 400000 400000 400000 400000 400000 400000 400000 4000000	
000000	
0000004 402E 0000004 402E 0000100 391E 0000100 473E 00001000 473E 0000100 473E 00001000 473E 0000100 473E 0000100 473E 0000100001000000000000000000000000000	
0000200 397E	
0000000 397E 000372 490E 000372 1904 1945L 000372 1904 1945L 000372 1904 1945L 000372 1904 1945L 000300 475E 000200 475E 000000 475E 0000000 475E 000000 475E 0000000 475E 0000000 475E 0000000 475E 000000 475E 000000 475E 000000 475E 000000 475E 0000000 475E	
0000100 000372 901E 000430 000433 1994 000433 1994 000430 473E 473E 473E 475E 000000 473E 475E 000000 475E 0004000 475E 0004000 475E 0004000 475E 0004000 475E 0004000 475E 0004000 475E 00040000 475E 0004000 475E 000600 475E 0004000 475E 000600 475E 000600 475E 000600 475E 0006000 475E 000600 475E	
0000372 901E 2331 004237 1994 004237 1994 004237 1994 000000 473E 476 000000 473E 476 000000 473E 476 000000 473E 644 000000 473E 1945 001249 2580 001225 1004 1333E 1048E 1285 001225 1004 1333E 1048E 1285 001225 1004 1333E 1048E 1285 000000 645 1104 1293 000124 579 000125 1066 000125 1066 000127 1141 1293 00027 1146 000000 652E 1144 1459	
000000	
000220	
000000 473E 476 000000 473E 476 000000 473E 476 000000 473E 476 000000 474E 644 000000 645E 644 000000 645E 644 000000 645E 645 000000 645E 64	
0002000 473E 476 000200 476E 476 000200 476E 476 000151 2010 2023 2091 2212 2271 007151 2010 2023 2091 2212 2271 007154 2580L 2585 00225 1334L 1342 1346 002271 484E 1048L 1342 002271 484E 1048L 1342 002271 484E 1048L 1048L 1054E 1055E 1156 1054E 1055E 1156 1	
000220	
000200	
000155	
007151 2010 2023 2091 2212 2271 007156 2589 20022 2289 2002265 1394 1333	
0002265 1084 1333E 1346 0002265 1084 1333E 1346 0002265 1334L 1342 1346 0002265 1334L 1349 1340 0002265 1334L 1349 1340 000221 1340 000225 1048L 000225 1048L 000225 1048L 000225 1048L 000225 1048L 000225 1054E 1285 1054E 1285 1054E 1285 1054E 1285 1054E 1285 1054E 1285 10522 1264 1285 10523 1286 1286 1286 1286 1286 1286 1286 1286	2574L
002265         1084         1331           002265         1334         1342         1346           002271         1337         1340         1346           001235         618         1048         1048           001235         618         1048         1048           001225         648         1048         1058           001225         648         1048         1058           001226         24E         1048         1058           001246         579         1054E         1058           001247         579         1054E         1054         1258           001249         579         1054E         1054         1258           000220         1278         1278         1266         1267         1266           0002215         1271         1271         1266         1266         1266         1266           002215         1271         1271         1272         1274         1273         1274         1456         1273         1274         1456         10023         1456         10023         1456         10023         1456         10023         1456         10023         1456         10023 <td></td>	
0002265 1334L 1342 1346 0002271 1337L 1340 001235 618 1048L 001235 648 1048L 001225 648 1048L 001225 648 1048L 001225 648 1057 001225 648 1057 001225 648 1057 001225 648 1057 001225 648 1057 001225 648 1057 001225 648 1057 001225 648 1057 0000370 1946 1285 0000370 1946 1286 0002215 127L 1285 000370 1956 1264L 1293L 0002215 127L 1285 000000 524 1067 1142 1313 2819L 00022000 5286 1376 1459 000000 5286 1144 1459 000000 3556 000000 3556 000000 3556 000000 3556	
0002271 1337L 1340 004000 484E 1048L 001235 618 1043E 042200 244E 1957 040125 882 1043E 0402200 242E 1957 0401032 242E 1108 1157 2786L 0401032 242E 1108 1157 2786L 0401037 195E 1285 040130 234E 1285 040130 195E 1285 040130 195E 1285 040130 195E 1285 040131 1200 140 040132 1126 1264 040133 727 729 764 1223 1 127L 1285 040133 727 729 764 127L 1285 040133 727 729 1406 040132 1126 1264 040132 1166 1141 1293L 0402252 1278 1309L 1378 1456 040000 522E 1376 1459 040000 352E 1376 1459 0400000 352E 1376 040000 352E 1376 040000 352E 1376 040000 352E 1376	
001235 618 1048L 001235 618 1048L 040220 244E 1957 0401000 244E 1285 0401000 244E 1285 0401000 244E 1285 0401000 244E 1285 0401000 244E 1285 0401000 244E 1285 0401000 244E 1284L 1067 1141 1293L 1068 1141 1293L 1069 1141 1293L 1069 1141 1293L 1060 1141 1499 0000000 252E 1376 0000000 358E 1144 1459 0000000 358E 1144 1459 000000 358E 1144 1459	
001225 618 1048L 001225 882 1043E 0040200 244E 1957 001032 245E 1957 0011244 579 0001244 579 0001244 579 0002222 124E 1285 0002230 1278L 1285 0002370 195E 1285 0002370 195E 1285 0002370 195E 1285 000224 1066 1141 1293L 1286 000224 1067 1142 1313 2819L 000224 1067 1142 1313 2819L 000225 1264L 195E 1369L 1378 1456 0000371 197E 1376 1459 000000 524E 1376 1459 000000 358E 1144 1459 000000 358E 1376 1459 000000 358E 1376 1459 000000 358E 1376 1459	
042200 244E 1957 001225 682 1043E 0042200 244E 1957 001032 242 1054E 0040130 1054E 0040130 1054E 0040130 1054E 0040130 1054E 0040130 1054E 0040130 1054E 1054E 1283 1056 106033 1057 1057 1142 1313 2819L 002254 1066 1141 1293L 1293L 1067 1142 1313 2819L 002254 1066 1141 1293L 1395E 1376 106000 252E 1376 1376 1459 000000 252E 1376 1144 1459 000000 252E 1376 1144 1459 000000 252E 1376 00	
042200 244E 1957 001032 242E 108 1157 2786L 001040 645 1054E 1157 2786L 0040130 194E 1956 000370 194E 1285 002215 1271L 1285 0040133 2002 2140 2142 2841L 0040133 2002 2140 2142 2841L 0040133 20224 1066 1141 1293L 002244 1066 1141 1293L 002245 1067 1142 1313 2819L 002265 1067 1142 1313 2819L 00026 525E 1376 1459 000000 524E 1376 000000 524E 1376 000000 355E 1144 1459 000000 355E 1144 1459	
001032 242E 1108 1157 2786L 001030 645 1108 1157 2786L 001344 500130 234E 000370 195E 000370 195E 000213 1126 1202L 1266 000213 1126 1202L 1266 1223 12022 12022 1202 1264 1223 12022 12022 1202 1264 1223 12022 12022 1202 1264 12023 12022 1202 1264 12023 12022 1202 1202 1202 1202 1202 12	
040000 645 1108 1157 2786L 001244 579 1054E 1054E 0003130 234E 0000370 195E 1285 1285 002215 1278L 1282 1265 1278L 1282 12841L 1293L 1293L 1278L 1293L 1293	
040130 234E 0000370 1054E 000370 1054E 000370 195E 1285 0002220 1278L 1285 1260 002215 1278L 1282 1278L 1282 1278L 1282 1278L 1282 1278L 1282 1278L 1282 1284L 1282 1284L 1282 1284L 1282 1284LL 1282 1284L 1284L 1282 1284L 12824L	
040130 234E 000370 194E 0002230 128E 002215 1278L 1285 002215 1278L 1285 002215 1278L 1285 002215 1278L 1282 004033 727 729 764 1223 1 040031 1067 1141 1293L 006224 1066 1141 1293L 006224 1067 1142 1313 2819L 006225 1066 1167 1378 1456 0000371 197E 1376 1459 000000 525E 1376 1459 000000 526E 1376 1459 000000 356E 1144 1459 000000 356E 1144 1459	
000000 0000370 000215 002215 040033 040033 1126 040033 1126 1202 1202 1202 1203 1203 1203 1204 1203 120	
0002150 0022150 0022151 0022151 0022151 0041122 0041122 0041122 0041122 0041122 0041122 0041122 0041122 0041122 0041122 0041122 0052054 1065 1166 1176 1166 1177 117	
000215 1271L 1285 002215 1271L 1286 002213 727 729 764 1223 040033 727 729 764 1223 041122 2002 2140 2142 2841L 00224 1006 1142 1313 2819L 00225 1006 1142 1313 2819L 00020 1006 1142 1378 1456 000010 525E 1376 1459 000000 526E 1376 1459 000000 526E 1376 1459 000000 358E 1376 1459 000000 358E 1376 1459 000000 358E 1376 1459	
00000133 1126 1202 1266 1223 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1203 1 1260 1260 1260 1260 1260 1260 1260 12	
040033 727 729 764 1223 1 041122 2002 2140 2142 2841L 002244 1066 1141 1293L 040031 1067 1142 1313 2819L 000371 196E 1309L 1378 1456 0000100 523E 1376 1459 000000 524E 1376 1459 000000 526E 1376 1459 000000 358E 1376 1459 000000 358E 1376 1459 000000 358E 1376 1459	
001122 002244 002244 1066 1141 1293L 002244 1066 1141 1293L 0022031 1067 1142 1313 2819L 0000371 196E 1376 1456 0000100 521E 1376 1459 000000 524E 1376 1459 000000 524E 1376 1459 000000 526E 1144 1459 000000 356E 1376 1459 000000 356E 1376 1459	3 2136 2820L
0002244 1066 1141 1293L 002205 1095 1264L 04031 1067 1142 1313 000371 197E 1309L 1378 000371 197E 1376 000020 523E 1376 1459 000000 524E 1376 000000 526E 1376 000000 356E 1376 0000100 356E 1376 0000100 356E 1376 0000100 356E 1376	) ) !
000205 1095 1264L 1313 040031 1067 1142 1313 000255 1279 1309L 1378 000371 197E 1378 000020 523E 1376 1459 000000 524E 1376 000000 52E 1376 000000 52E 1376 000000 35E 1376 000000 35E 1376 000000 35E 1376 000000 35E 1376	
040031 1067 1142 1313 002252 1279 1309L 1378 000371 196E 1376 1459 0000020 525E 1376 1459 00000040 524E 1376 0000001 525E 1376 0000001 526E 1376 0000001 526E 1376 0000001 526E 1376 0000001 526E 1376 0000001 526E 1376 0000001 526E 1376	
0002552 1279 1309L 1378 000371 196E 1376 0000371 197E 1459 000002 525E 1376 1459 000000 522E 1376 000000 52E 1376 000000 358E 1144 1459 000000 358E	
000371 196E 000371 197E 1376 1459 0000020 523E 1376 1459 000100 524E 1376 000000 522E 1376 000000 522E 1376 000000 358E 1144 1459 000100 358E 000001 000001 355E	
0000371 1976 1376 0000020 5236 1376 0000100 5216 1376 0000000 5226 1376 000000 5226 1376 000000 3586 000000 3556 000000 3556 000000 3576	
000020 523E 1376 0000002 523E 1376 0000000 521E 1376 000000 522E 1376 000000 526E 1144 000000 358E 1144 000000 358E 356	
*IE 000002 525E *IR 000100 521E 1376 *R0 000000 524E 1376 *TE 000000 526E 1376 *TE 000000 526E 1144 *TE 000000 526E 1144 *TE 000100 358E *TE 000100 355E *TE 000100 355E	
.IR 000100 521E 1376 .RE 0000040 524E 1376 .TE 0000001 526E 1144 .TE 0000001 556E 1144 .TE 000100 358E RDA 000001 355E RDA 000001 355E RDA 000001 355E	
**RE 000004 524E 1376  **R0 000040 522E 1376  **TE 0000001 526E 1144  FCT 000000 498E  RDA 000001 358E  RDA 000001 356E  RPE 000004 357E	
*R0 000040 522E 1376  *TE 000001 526E 1144  000000 498E  FCT 000100 358E  RDA 000001 355E  RPE 000004 357E	
FCT 000000 498E 11144 000000 498E 11144 FCT 000100 358E RDA 000001 355E RPE 000004 357E	
000000 RCT 000100 RDA 000001 RDE 000002	
RDA 000001	
- ROA 000001	
*RPE 000004 357	
100 000 101 000 100 101 000 101 000 101 000 101 000 101 000 101 000 100 100 100 000 10	
0 003161 767	

UIVEC UMI.16X	003227	1588	16091									
UMI-16X		290	595	900	614	621	902	1001	1661	2006	2052 2173	28241
AL THE	:	516E	2686							:	:	:
	0001000	506E	2686									
UMI 1X	00000	515E										
UNI 64X	00000	200E 517E										
UNITHB		507F										
UMI.LS	000000	5116										
UMI.L6	♦00000	512E										
UHI-L7	010000	513E										
UMI .L8	♦10000	514E	5686									
CHI.PA	00000	510E										
UMI.PE	00000	509E										
•	000000	464E	469									
UNI.	00000	465E	694									
•	001000	400E	694									
UNI - 5	041000	40/E										
F - 1 - 2	0000	40%E	6363									
000-CLR	10000	2/1/2	5	8107								
000.00	00000	216E	819	1574	1920	2018	5049	5086				
	000000	3417	0,7	***************************************	*****							
UU.NFK	00000	212	747	819	1920							
UP.0P	000174	349E										
UP.FC	000175	350E										
UP.SC	0001 76	352E										
UP.SR	921000	353E										
UP.ST		351E										
USERFHA		245E	2055	2232	2335							
USR	00000	499E										
USR.FE	040000	530E										
USK .UE	00000	231E							:			
USK.PE		236E										
USK-KAK ICCO: TVE	700000	7345	1379									
0.21 0.21 0.21		2325	1463									
CINDELLE		2004	2061		21.70	12786						
M. RES		392E	2167	2287		1010						
NOS	007167	2288	2416	7337	2164	2558	26031					
NON	007172	26051	2614		900	2000	7007					
MONA	175000	2603	7625E									
MME 1	002012	11481	1150									
WNE2	002104	11811	1188									
MEM	001374	884	1140E									
KNB	003024	1148	1152	1182	1442	1455						
NNB I	003025	1456L	1458									
Z Z	003017	1156	1167	1176	1179	1193	1194	14411				
MTR	007222	2397	26411									
WTRI	007225	2643L	2658									
MTRA	175000	2641	2662E									
INIX	004032	18241	1829									
XINC	190400	1837	18396									
2 1 1 X	200400	10421	1821									
Y THE	111400	1021 1021	1001									
CNIX	004135	<b>~</b> ~	1871	1 6 6 4								
XINX	P :-	1821	1010	1001	:	:						
VINAL	71000	v	1001									

Cross Seterate 1161							
100 000 182 2734 2784 2784 2884 2884 2884 2884 2884 288	RAMBGO -	HB FRONT P	ANEL MONITOR	#01.02.0	•0	XREF V1.2.1 Page 78	
2014 - 2714 1785 1784 1785 2894 2894 2894 2894 2894 2894 2894 2894	XINB	040004	1823	1838 1816L	.2788E		
	XINITI	004000		1786L 2757	2621		
	30454 by	;					
	:						
	:						
				:			

## **INDEX**

Addressing Port Pairs, 1-21 Advanced Control, 1-22 Alter Key, 1-9, 1-12 Altering a Memory Location, 1-12 ff Altering a Selected Register, 1-15 Audio Alarm, 1-9

Binary to Octal Conversion, 1-7 Breakpoint Interrupts, 1-24 Breakpointing, 1-16

Cancel, 1-9, 1-19 Checksum Errors, 1-20 Clock, 1-5 Clock Interrupts, 1-5, 1-24 Copying a Tape, 1-20

Decrements Memory, 1-9
Disable Interrupt, 1-5
Displays, 1-7, 1-23
Dump Routines, 1-18 ff
Dumping, 1-18

Execution Control, 1-16 ff Entry Point, 1-19 Ending Address, 1-19

FPLEDS, 1-23

GO Key, 1-16

Halt Instruction, 1-16

I/O, 1-21 I/O Interrupts, 1-24 IP.PAD, 1-22 Increment Memory, 1-9 Initialization, 1-5 Inputting, 1-21 Interrupts, 1-24 Interrupting a Program, 1-17 Keypad, 1-9, 1-22

Load Routines, 1-18 ff Loading, 1-18

MEM Key, 1-9 MFLAG, 1-23 Manual Display, 1-23 Manual Updating, 1-23 Master Clear, 1-5 Monitor Mode, 1-6

Offset Octal, 1-8 Outputting, 1-21

Port I/O, 1-21 Power-Up, 1-5

RAM High Limit, 1-5 RCK, 1-22 REG Key, 1-9 RST, 1-5 ff, 1-9 RTM, 1-5 ff, 1-9, 1-17 Record Errors, 1-20 Refreshing, 1-23 Repeats, 1-9

Single Instruction, 1-17
Single Instruction Interrupts, 1-24
Specifying a Memory Address 1-10 ff
Specifying a Register for Display, 1-14
Stack Space, 1-5
Stepping Through Memory, 1-13
Stepping Through the Registers, 1-15

TICCNT, 1-22 Tape Errors, 1-20 Tick Counter, 1-22

User Mode, 1-6 User Option Bits, 1-23