XEROX®

TECHNICAL REFERENCE MANUAL

Xerox Professional Computer

610P72384

WARNING: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operating with non-certified peripherals is likely to result in interference to radio and TV reception.

WARNING: This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

Reorient the receiving antenna.
Relocate the computer with respect to the receiver.
Move the computer away from the receiver.
Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful. This booklet is available from the U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, D.C. 20402, STOCK NO. 004-000-00345-4.

"HOW TO IDENTIFY AND RESOLVE RADIO-TV PROBLEMS"

Xerox Corporation reserves the right to make improvements to products without incurring any obligation to incorporate such improvements in products previously sold.

THIS PAGE INTENTIONALLY BLANK.

Table of Contents

Introduction

Hardware

Hardware continued

Theory of Operation

Hardware continued

Software

Device Initialization

Operating System Interface

Software continued

Peripherals

Peripherals continued

Appendices

Introduction

The purpose of this manual is to provide technical reference material for the Xerox 820-II and 16/8 Professional Computers for programmers and engineers involved in hardware, software, and interface design. It is also intended for interested persons who have a desire to know how the Xerox 820-II and 16/8 operate and how to access their many features.

A list of the abbreviations and naming conventions used in this manual can be found in Appendix N.

SYSTEM OVERVIEW

The modular design of the 820-II and 16/8 systems enhance the flexibility provided by the operating systems. The combination of operations provided by the system gives it a flexibility that allows it to be tailored to the needs of each user.

Appendices continued

Overview of Xerox Personal Computers

Xerox Personal Computers are comprised of four components: display/processor, disk drives, keyboard, and optional printers. Both the 820-II and 16/8 use the CP/M®-80 2.2 Operating System from Digital Research, Inc. Additionally, the 16/8 PC includes CP/M-86® 1.1and MS™-DOS Version 2.0 as standard operating systems.

HARDWARE DESCRIPTION

System Board

The system board uses a Zilog Z80-A®-based microprocessor operating on a 4 megahertz clock with 64k RAM and 8k ROM. It is a single-board computer and uses a daughter board to interface with the disk drives. The 820-II is equipped with three user-accessible I/O ports. Two of the ports are located on the back of the display/processor; the third is located on the CPU board. On the back are the printer and the communications ports (both RS232C). The port inside the display is a dual parallel port (most printers and other devices that follow a standard Centronics 36-pin interface can be successfully attached).

The 16/8 has all of the above features as well as an Intel 8086®-based microprocessor operating with a 4.772 megahertz clock. The 8086 is equipped with 128k of RAM which is expandable to 256k total by adding a 128k daughter board to the 8086 board.

The 820-II and 16/8 are capable of having up to 8k of read only memory (four 2k ROMs): the 820-II has 6k of this 8k occupied; the 16/8 uses the full 8K. The last 2k on the 16/8 is used for decoding the position-encoded Low Profile Keyboard. The firmware contained in the ROM is capable of doing such things as executing a one-sector loader from disk; i.e., loading CP/M, emulating a terminal, operating in typewriter mode, etc. The monitor also has other commands that are useful for debugging hardware and software. The mother board also contains a speaker as well as an expansion slot (used by the 16/8 for the 8086 board). There are two types of daughter boards: one interfaces the display/processor to floppy disks and the other interfaces the display/processor to a rigid disk controller.

Display

The display/processor houses the video display, the CPU mother board, the disk drive daughter board, and the 8086 processor board if so configured. The video display is a standard 24 line by 80 characters. It uses a 7 x 10 dot matrix for each character in all text modes and displays white characters on a black screen. For graphics characters, it uses a 4 x 4 pixel resolution. The display attributes can be changed to display either in blink, highlight/lowlight, inverse, or graphics characters.

820-II Display/processor for floppy disks	Product Code #U03
820-II Display/processor for rigid disk	Product Code #U05
16/8 Display/processor for floppy disks	Product Code #H69
16/8 Display/processor for rigid disk	Product Code #H70

Keyboards

The 820-II and 16/8 use either a standard 96-character ASCII or Low Profile keyboards. Both keyboards include additional keys to the right of the keyboard, a 10-key numeric key pad and a set of keys for software control of the cursor. The low profile keyboard also includes 12 function keys that can be software-enabled and other keys such as Accept, Delete, Next, Previous, Home, and Undo.

ASCII Keyboard	Product Code #X928
Low Profile Keyboard	Product Code #G25

Disk Drives

Five disk drive options are offered for the 820-II:

Dual 5¼" single-sided floppy disk drives	Product Code #X929
Dual 5¼" double-sided floppy disk drives	Product Code #T66
Dual 8" single-sided floppy disk drives	Product Code #X973
Dual 8" double-sided floppy disk drives	Product Code #F10
One 10mb rigid disk drive with an 8" double-sided disk drive	Product Code #U07

Three disk drive options are offered for the 16/8:

Dual 8" single sided floppy disk drives	Product Code #X973
Dual 8" double sided floppy disk drives	Product Code #F10
One 10mb rigid disk drive with an 8" double sided disk drive	Product Code #U07

Printers

40 CPS Printer and 20 CPS Printer

As their names imply, the printers have a printing speed of 20 and 40 characters per second (CPS) respectfully. Both printers have a wide range of print styles available. The 20 CPS Printer supports 10, 12, and 15 pitch as well as Proportional Spacing (PS), while the 40 CPS Printer supports either metal or plastic printwheels in 10, 12, 15, and PS. More detailed information on these printers can be found in the Printer section under Peripherals.

The standard RS232C printer connector and dual parallel port are available to interface with many types of serial and parallel printers.

40 CPS Printer	Product Code # D80
20 CPS Printer	Product Code # U01

FUNCTIONAL DESCRIPTION

The 820-II and 16/8 systems are a collection of four components working in unison -- the display, keyboard, disk drives, and printer. The computer itself is housed in the display.

System Monitor - ROM

The system monitor contained within the 8k ROM controls the essential functions of initializing and controlling all system input/output resources, and also provides a number of monitor commands that can be used to assist in programming.

Ports

Three ports are standard on the 820-II and 16/8: two serial ports located at the back of the display unit and an additional dual parallel port inside the display unit. These allow printers, communication devices, and other peripheral equipment to be interfaced with the system.

Operating Systems

The 820-II uses Digital Research's 2.2 CP/M-80 Operating System. The 16/8 can use Digital Research's 2.2 CP/M Operating System, as well as their CP/M-86 1.1 Operating System and Microsoft's MS-DOS 2.0 Operating System. These operating systems provide the user with a general environment for program construction, storage, and editing, along with assembly and program checkout facilities.

CP/M-80 operating system software as implemented on the 820-II and the 16/8 is logically divided into four parts:

ROSR	ROM Operating System Routines (hardware dependent)
BIOS	Basic I/O System (hardware dependent)*
BDOS	Basic Disk Operating System*
CCP	Console Command Processor*

*Disk resident portions of CP/M-80

ROSR provides code in ROM that can be executed without the presence of the CP/M system disk and provides the primitive operations necessary to access the disk drives and to interface with peripherals.

BIOS provides the interface between BDOS and ROSR.

BDOS provides disk management by controlling one or more disk drives containing independent file directories.

CCP provides symbolic interface between the user's console and the remainder of the CP/M system.

HARDWARE INTERFACE

The 820-II and 16/8 are equipped with six input/output connectors. Four are on the back of the display unit and two are inside the display.

Disk Drive

Used for connection of either the 8" or the 5¼" Dual Floppy Drives, or the 8" Rigid Disk Drive. This is determined by the type of disk daughter board installed in the display processor.

Keyboard

Used for connection of either the ASCII or Low Profile keyboard.

Printer

A serial printer can be attached to this RS-232-C connector.

COMM

COMM is a second RS-232-C connector and can be used for a modem.

Parallel Port

A dual parallel port inside the display cabinet is also provided.

Expansion Slot

The expansion slot inside the display cabinet provides all of the Z80-A microprocessor control signals for connection to custom devices for future expansion. This slot is used for the 8086 co-processor if you have a 16/8.

CP/M-80

The CP/M-80 2.2-C disk for the 820-II contains the standard Digital Research software development and checkout programs. Xerox issues additional utility programs that are unique to the 820-II. A description of each program is listed below:

Digital Research Files

ASM.COM	The Assembler allows you to create a program which can be read and executed by the 820-II.
DDT.COM	The Dynamic Debugging Tool is used to debug a Z80-A assembly language program.
DUMP.COM	Allows binary command files that are not displayed on screen to be displayed showing the hexadecimal value for each byte.
ED.COM	A line-oriented screen editor.
L80.COM	Reads an .REL file created with the Macro-80 Assembler Program and outputs a command file.
LOAD.COM	Reads a .HEX file and creates a command file.
M80.COM	Converts a program written in Assembly Language to a relocatable (.REL) file and (optionally) a printer listing file (.PRN).
MOVCPM.COM	Lets you modify and move the CP/M system image to allocate a specific lesser memory size.
PIP.COM	Allows you to selectively copy a file or files from one disk to another or on the same disk.
STAT.COM	The status utility is a frequently-used transient command for all system housekeeping; i.e., checking the amount of space available on a disk.
SUBMIT.COM	Used to submit a file of commands for batch processing.

SYSGEN.COM — Used to generate a CP/M-80 system image and copy the operating system to another disk.

XSUB.COM — Same as Submit.com, but has the facility to include line input to programs as well as the console command processor.

Xerox Files

BACKUP.COM — A multi-option utility that allows you to archive and retrieve files, delete files, list directories of any drive, and to verify data integrity of a floppy or rigid disk.

CONFIGUR.COM — Using Configur.com, you can select seven different options:

1. Record Restart Command - lets you enter a one-line command which will automatically load a program. For example, you could enter DIR as the restart command and every time you boot the system, it will automatically display the directory for you. Or you could enter the name of your application software package and it would automatically load that application package for you. This command is recorded on the disk and you can have a different one for each disk.
2. Select Printer Port Options - allows you to determine printer protocol. This option allows configuration for alternate printers without modifying the BIOS.
3. Select Communications Port Options - a convenient method for setting up the communications port on the 820-II or 16/8; that is, baud rate, protocol, stop bits, etc.
4. Select I/O Device Assignments - lets you select alternative input/output device assignments; i.e., set up the system so that everything displayed on the screen automatically prints on the printer.
5. Select Keyboard Data Format - lets you choose 7-bit or 8-bit mode for the keyboard.
6. Select Screen Attributes - includes blink, inverse video, highlight/lowlight, and graphics modes.

7. If you have a floppy disk system, Select Floppy Disk Head Step Rate will appear as selection 7. If you have a rigid disk system, Configure Rigid Disk will appear (program must be loaded from floppy or the first partition of the rigid).
   a. Select Floppy Disk Head Step Rate - lets you adjust the floppy head step rate for optimum performance.
   b. Configure Rigid Disk - lets you divide the eight megabyte rigid disk into sections (e.g., 4 Mb, 2 Mb, 1 Mb, 1 Mb).

COPY.COM	Makes an exact copy of a disk, track for track.
FMT.COM	Allows you to format (initialize) a rigid disk. Verification of the rigid disk is performed using the Backup.com utility.
HELP.COM	A guide for CP/M-80 users that contains basic information about CP/M-80 commands; also cross-references to additional information in the CP/M-80 reference manual, Reorder #9R80448.
INIT.COM	Prepares new (or used) disks for storing information. It will also alert the user to any flawed sectors on the disk.
KILLESC.COM	Turns off the <CTRL> + <ESC> feature to enable use of <CTRL> + <ESC> for other purposes; for example, setting margins and tabs on a 40 CPS printer uses a <CTRL> + <ESC> sequence.
SET.COM	A convenient method to temporarily change communication and printer port options in RAM.
SWAP.COM	A utility that allows the user to swap drive names. For example, "A" and "E" for a rigid disk drive. By designating an alternate drive as the "A" drive, you can load software directly from that drive. Many CP/M-80 application packages have been written to be executed from the "A" disk drive only. Using Swap.com allows you to place your application software on any disk drive and load.
TiME.COM	Displays the time and date on screen. Since there is no battery backup, however, you must re-enter the time and date each time you reload the system.
WHATSA.COM	This utility lists the logical and physical names for each disk drive, as well as the density, number of

sides, and types of disks logged into the system, (e.g., double density, single-sided 8" floppy).

CP/M-86

The CP/M-80 2.2 and CP/M-86 1.1-F disks for the 16/8 contain the standard Digital Research software development and checkout programs. These disks contain the same files as described in the CP/M-80 section as well as the following files.

Digital Research Files

ASM86.CMD	The Assembler allows you to create a program which can be read and executed by the 8086.
DDT86.CMD	The Dynamic Debugging Tool is used to debug a 8086 assembly language program.
ED.CMD	A line-oriented screen editor.
GENCMD.CMD	Uses the hex output of ASM-86 and other language processors to produce a .CMD file.
GENCMD.COM	Uses the hex output of ASM-86 and other language processors to produce a .COM file.
GENDEF.CMD	Reads a 16-bit file containing the disk definition statements, and produces a 16-bit output file containing assembly language statements which define the tables necessary to support a particular drive configuration.
GENDEF.COM	Reads a 16-bit file containing the disk definition statements, and produces an 8-bit output file containing assembly language statements which define the tables necessary to support a particular drive configuration.
HELP.CMD	Provides summarized information for all of the CP/M-86 commands described in the Digital Research Users manual.
LMCMD.CMD	Operates in exactly the same manner as Gencmd.cmd, except Lmcmd also accepts an Intel L-module file as input.
LMCMD.COM	Operates in exactly the same manner as Gencmd.com except Lmcmd also accepts an Intel L-module file as input.
PIP.CMD	Allows you to selectively copy a file or files from one disk to another or on the same disk.

STAT.CMD	The status utility is a frequently-used transient command for all system housekeeping, i.e., checking the amount of space available on a disk.
SUBMIT.CMD	Used to submit a file of commands for batch processing.
TOD.CMD	Time of day.

Xerox Files

CPM86.COM	Used by Load86.com to boot the 8086.
86CON.COM	Switches from Z80-A console to the 8086 console.
GOBACK.CMD	Switches from 8086 console to the Z80-A console.
LOAD86.COM	Loads the 8086 for concurrent processing.
REBOOT.COM	From the concurrent mode, reboots the system as a Z80-A standalone.
SOFTKEYS.COM	Used to set up the 10-key pad with programmable functions (<CTRL> + one of the 10-key pad keys).

MS-DOS

The MS-DOS 2.0 disk for the 16/8 contains the standard Microsoft software development and checkout programs.

Microsoft Files

ANSI.SYS	Allows programs that use the standard ANSI driver to be executed.
COMMAND.COM	This is the MS-DOS command processor. It is recommended that this file be placed on every application program disk.
CONFIG.SYS	Configures system at boot.
CHKDSK.COM	Checks disk.
CREF.EXE	Assists in debugging assembly language programs.
DEBUG.COM	Debugger supplied with MS-DOS.
DISKCOPY.COM	Copies a disk.
EDLIN.COM	Line-oriented screen editor.
EXE2BIN.EXE	Converts .EXE files to binary format.
FC.EXE	Compares two files for similarity.
FIND.EXE	Finds a string in a list of files or standard input.
FORMAT.COM	Formats an 8" floppy or a rigid disk.
LINK.EXE	Linker.
MORE.COM	Used to display text in 23-line segments.

MASM.EXE	Macro Assembler for MS-DOS.
PRINT.COM	Print spooler.
RDCPM.COM	Reads a CP/M-80 file and converts data to MS-DOS-readable file.
RECOVER.COM	Recovers bad or damaged disks.
SORT.EXE	Used to sort text.

Xerox Files

SAMPLE.TXT	Provided to assist going through MS-DOS Handbook.

Notes

Specifications

This section details the following specifications: dimensions, electrical requirements, operating environment, and disk drive capacities of the Xerox 820-II and 16/8 PCs.

Dimensions

Equipment	Height	Depth	Width	Weight
820-II-16/8 Display	12.20"	14.75"	15.00"	30 lbs.
ASCII keyboard	3.75"	9.50"	20.00"	10 lbs.
Low profile keyboard	1.60"	8.25"	19.90"	5 lbs.
5¼" Floppy disk drives	7.00"	10.20"	7.00"	10 lbs.
8" Floppy disk drives	10.50"	17.50"	14.50"	48 lbs.
8" Rigid disk drive	10.50"	17.50"	15.50"	54 lbs.
40 CPS printer	10.00"	17.50"	15.50"	56 lbs.
20 CPS printer	9.25"	17.50"	24.00"	45 lbs.

Electrical Requirements

All Xerox products listed below require voltage of 115 VAC, a frequency of 60 Hz, and a two-pole, three-wire grounded duplex receptical.

Equipment	Current
820-II-16/8 Display	1.1 Amps
8" Floppy disk drives	2.0 Amps
8" Rigid disk drive	2.2 Amps
40 CPS printer	2.0 Amps
20 CPS printer	1.0 Amp

Operating Environment

All Xerox equipment is tested to perform between 50 and 90 degrees Fahrenheit with a relative humidity factor between 20% and 80%.

Disk Drive Storage	Unformatted	Formatted	Usable
5¼" SS/SD	125 k	90 k	81 k
5¼" SS/DD	250 k	168 k	155 k
5¼" DS/SD	250 k	180 k	172 k
5¼" DS/DD	500 k	338 k	322 k
8" SS/SD	400 k	250 k	241 k
8" SS/DD	800 k	497 k	482 k
8" DS/SD	800 k	500 k	490 k
8" DS/DD	1.6 Mb	997 k	980 k
8" Rigid (DS/DD)	10.67 Mb	8.4 Mb	8.192 Mb

Disk Drive Format				
Equipment	Tracks	Sectors	Bytes per Sector	Number of Heads
5¼" SS/SD	40	18	128	1
5¼" SS/DD*	40	17	256	1
5¼" DS/SD	80	18	128	2
5¼" DS/DD*	80	17	256	2
8" SS/SS	77	26	128	1
8" SS/DD**	77	26	256	1
8" DS/SD	154	26	128	2
8" DS/DD**	154	26	256	2
8" Rigid DS/DD	1,024	32	256	4

*Track 0 of 5¼" double density disks has 18 sectors of 128 bytes.
**Track 0 of 8" double density disks has 26 sectors of 128 bytes.

For more specific information on disk formats, see the Disk Drive Specifications section.

DISPLAY SPECIFICATION

SIZE:	12 inch, landscape mode
TYPE:	Aluminized P4 Fluorescence White (W) Phosphorescence White (W) Persistence Short
RESOLUTION:	• 240 active line raster adjusted to 8.5 x 5.3 inch usable area • Brightness level 30 (± 2) foot-lamberts • Resolution at centers (within 1" diameter circle) -100 lines/inch minimum
CHARACTER CELL:	7x10
BUSINESS GRAPHICS:	4x4 Pixel Resolution
CHARACTER SET:	4 sets of 128: (1 U.S. font, 1 Graphics font) (1 U.S. font, Inverse Video font)
CHARACTER LINES:	24
CHARACTERS/LINE:	80
VOLTAGE:	+ 12 (± 5.0%) VDC at 2.0 A DC maximum
RIPPLE:	50 MV P-P synchronous or nonsynchronous with refresh or power frequency.
VIDEO BIT RATE:	10.694 MBPS (93.51 nanoseconds)
BITS/HORZ LINE:	560
HORZ SYNC PULSE:	126 (11.78 microseconds)
TOTAL BITS/LINE:	686
HORZ RATE:	15.59 KHz (64.14 microseconds)
LINES/FIELD:	240
VERT BLANKING LINES:	20
VERT SYNC PULSE:	20(1.28 milliseconds)
VERT RETRACE (lines):	8 TYP
TOTAL LINES/FIELD:	260
FIELD RATE:	59.95 Hz (16.68 milliseconds)
REFRESH RATE:	61 Hz
VIDEO RATE:	15 MHz

FUNCTIONAL DESCRIPTION, XEROX DISPLAY

The display has the following functional characteristics:

- 24 line display
- 80 characters per line
- 7x10 dot matrix per character
- White characters on black
- Software-selectible character attributes
  - Inverse Video
  - Blink
  - Low Intensity
  - Graphics with 4 x 4 pixel resolution
- Brightness adjust

DISPLAY CONTROLLER

The Display Controller is based on displaying characters within a 7x10 cell (7 dots horizontally by 10 scan lines vertically). To guarantee spaces between characters, one dot on each side of the cell is blanked by hardware. Also, to guarantee spaces between character lines, the top two scan lines are blanked by hardware. This gives an actual active character size of 5 dots horizontally by 8 scan lines vertically.

For Business Graphics, the hardware is configured to eliminate the automatic blanking and allow continuous lines both horizontally and vertically. However, the Display Controller is still based on displaying a character within a 7 x 10 cell. The controller design and available refresh memory allows one byte per character. The maximum number of unique characters that can be defined by any 8 bits is 256. Since the standard text font set contains 128 characters, the limit on unique characters for graphics that can be displayed together with text is 128.

The character set for Business Graphics divides the cell into blocks of 4 dots horizontally by 4 scan lines vertically. Since the total number of scan lines per character is 10, the character set actually consists of two sub-sets of 4-4-2 and 2-4-4.

Each subset divides the cell into 6 parts requiring 64 possible combinations or unique characters. Therefore, the total number of unique characters for the complete graphics set is 128. With this

character set, any combination of adjacent 4 x 4 blocks can be chosen. Also, at the character cell boundary, the 4 x 4 blocks can be set vertically by 2 scan lines. Since the total number of horizontal dots per cell is 7, there will be an overlap of one horizontal dot in the center of the cell for diagonal blocks within the cell.

It should be also noted that for the standard text font containing 128 unique characters defined by 7 bits, the eighth bit is used to set the attribute function. For Business Graphics, since both text characters and graphic characters can be displayed simultaneously, it requires all 8 bits to define the character. Consequently, display attributes are not available in graphics mode.

SYSTEM BUS EXPANSION SLOT

ELECTRICAL

The DC system power available at the expansion slot is as follows:

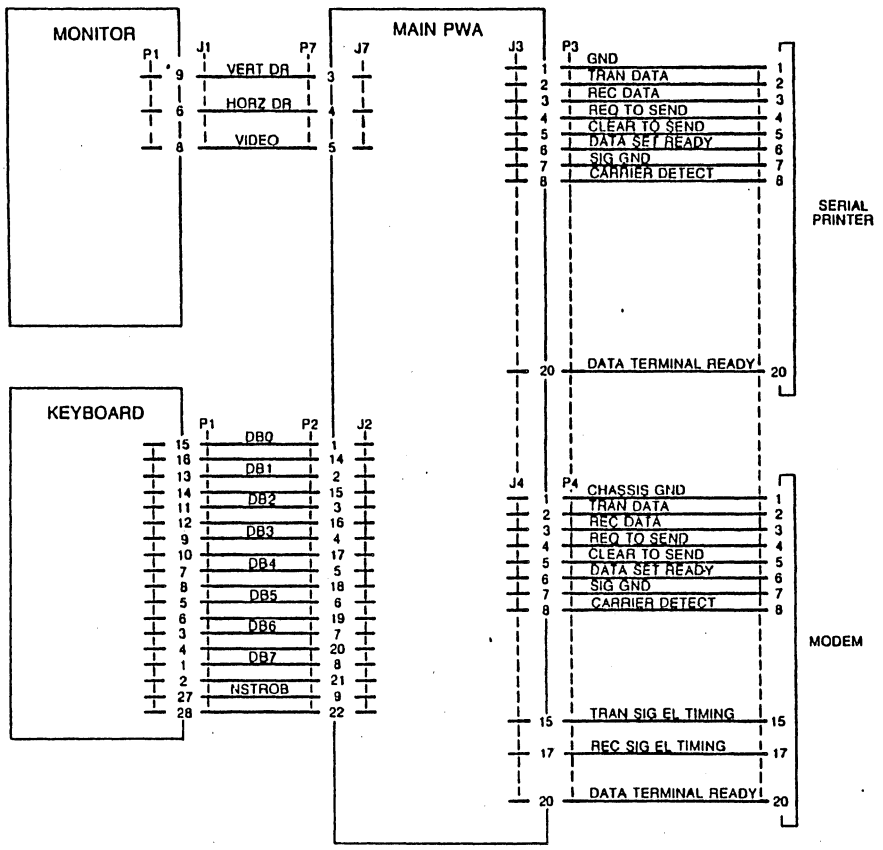
		5¼" system	8" or Rigid system
PIN 50	+ 5V DC	1.2A	2.1 A
PIN 45	+ 12V DC #1	0.3A	1.75 A

ENVIRONMENTAL

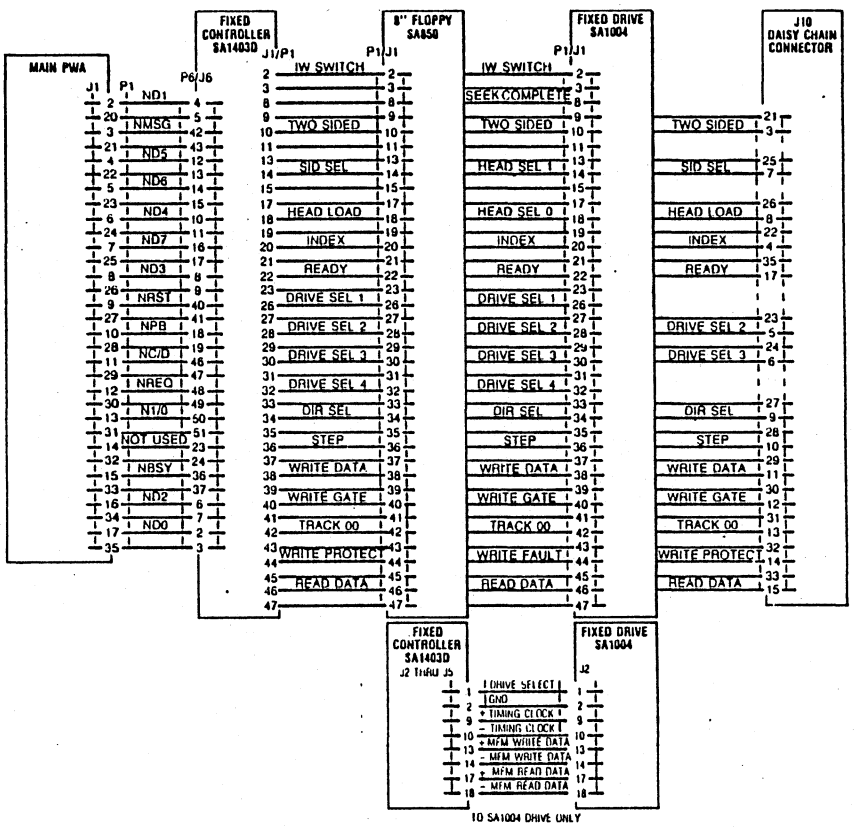
The following temperature, humidity and altitude environmental requirements are specified:

	Temp. (°Celsius)	Rel.Hum. (%)	Altitude (miles)
Operating	10 to 32	20 to 80	1830
Non-operating	-77 to 66	15 to 90	7620

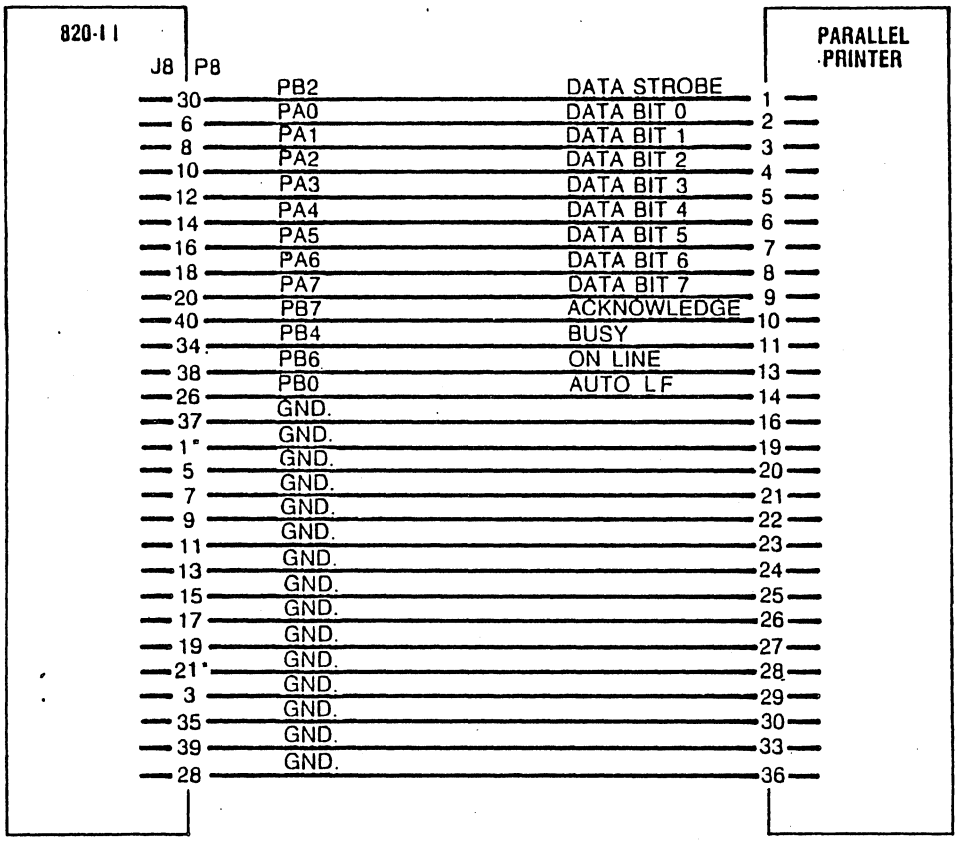
Any optional or additional electronic assembly using the expansion slot must be capable of performing to design specification when the host is subjected to the environmental range, above. Furthermore, the presence of such an assembly in the expansion slot must not degrade performance with regard to the above environmental requirements.

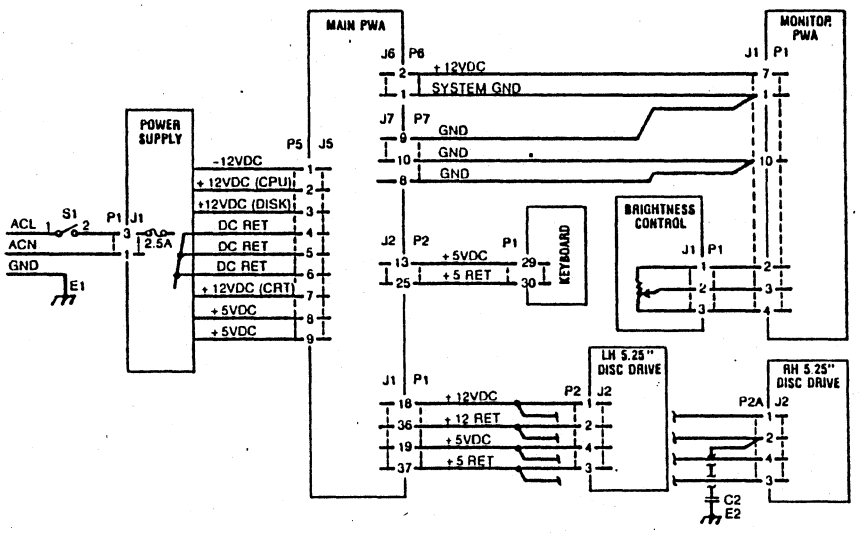

MONITOR
P1
9
6
8
J1
P7
VERT DR
HORZ DR
VIDEO
3
4
5
J7
MAIN PWA
KEYBOARD
P1
P2
J2
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
NSTROB
15
16
13
14
11
12
9
10
7
8
5
6
3
4
1
2
27
28
1
14
2
15
3
16
4
17
5
18
6
19
7
20
8
21
9
22
J3
P3
GND
TRAN DATA
REC DATA
REQ TO SEND
CLEAR TO SEND
DATA SET READY
SIG GND
CARRIER DETECT
DATA TERMINAL READY
SERIAL PRINTER
J4
P4
CHASSIS GND
TRAN DATA
REC DATA
REQ TO SEND
CLEAR TO SEND
DATA SET READY
SIG GND
CARRIER DETECT
TRAN SIG EL TIMING
REC SIG EL TIMING
DATA TERMINAL READY
MODEM

J1 illustration

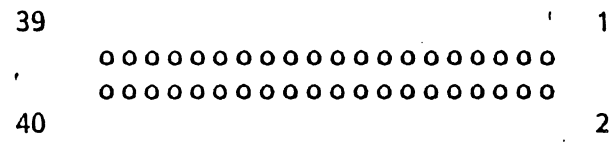


J8 illustration




MAIN PWA
MONITOR PWA
POWER SUPPLY
ACL
ACN
GND
S1
P1
J1
2.5A
E1
P5
J5
-12VDC
+12VDC (CPU)
+12VDC (DISK)
DC RET
DC RET
DC RET
+12VDC (CRT)
+5VDC
+5VDC
J6
P6
+12VDC
SYSTEM GND
J7
P7
GND
GND
GND
J2
P2
+5VDC
+5 RET
P1
KEYBOARD
BRIGHTNESS CONTROL
J1
P1
LH 5.25" DISC DRIVE
RH 5.25" DISC DRIVE
J1
P1
+12VDC
+12 RET
+5VDC
+5 RET
P2
J2
P2A
J2
C2
E2

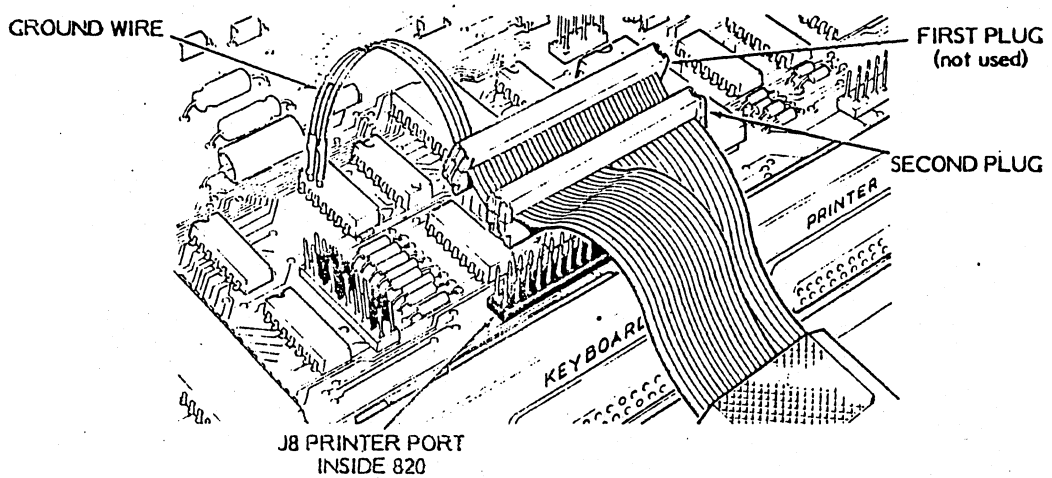
Parallel Port Connector (J8)



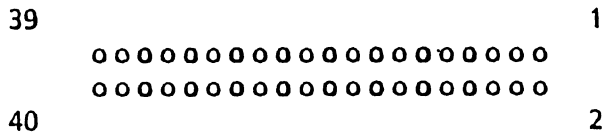
J8

Pin	Value
2	Port A Strobe
4	Port A Ready
6	Port A Bit 0
8	Port A Bit 1
10	Port A Bit 2
12	Port A Bit 3
14	Port A Bit 4
16	Port A Bit 5
18	Port A Bit 6
20	Port A Bit 7
22	Port B Ready
24	Port B Strobe
26	Port B Bit 0
28	Port B Bit 1
30	Port B Bit 2
32	Port B Bit 3
34	Port B Bit 4
36	Port B Bit 5
38	Port B Bit 6
40	Port B Bit 7
Odd # Pins	Ground

Parallel connector picture



COMM Port Strapping Options



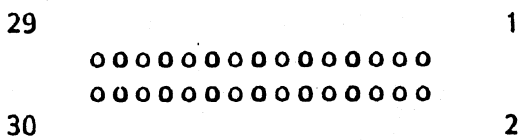
J9

Pin	Value
5--6	(M) TXD to Pin 3
7--8	(T) TXD to Pin 2
9--10	(M) RXD from Pin 2
11--12	(T) RXD from Pin 3
13--14	(M) RTS to Pin 5
15--16	(T) RTS to Pin 4
17--18	(M) CTS from Pin 4
19--20	(T) CTS from Pin 5
21--22	(M) DTR to Pin 8
23--24	(T) DTR to Pin 20
25--26	(M) DCD from Pin 20
27--28	(T) DCD from Pin 8
29--30	Clock supplied to Modem as RX Clock
31--32	Clock supplied to SIO with RX Clock
33--34	Modem supplies SIO with RX Clock
35--36	Clock supplied to SIO with TX Clock
37--38	Modem supplies SIO with TX Clock
39--40	Clock supplied to Modem with TX Clock

The filled-in pins indicate the options as they are jumpered on an 820-II or 16/8.

Note: (M) indicates modem (data communications equipment) function. (T) indicates terminal data equipment) function. For instance, exercising the (M) strap option will allow communication with a modem; exercising the (T) strap option will allow communication with a terminal. The above shows factory settings for (T).

J9 (Etch 2 CPU)



Pin	Value
3--4	
5--6	(M) TXD to Pin 3
7--8	(T) TXD to Pin 2
9--10	(M) RXD from Pin 2
11--12	(T) RXD from Pin 3
13--14	(M) RTS to Pin 5
15--16	(T) RTS to Pin 4
17--18	(M) CTS from Pin 4
19--20	(T) CTS from Pin 5
21--22	(M) DTR to Pin 8
23--24	(T) DTR to Pin 20
25--26	(M) DCD from Pin 20
27--28	(T) DCD from Pin 8
29--30	Clock supplied to Modem as RX Clock

The filled-in pins indicate the options as they are jumpered on an 820-II or 16/8.

Note: To change from ASYNC to SYNC on the Etch 2 CPU requires a modification to the operating system rather than moving jumpers.

Note: (M) indicates modem (data communications equipment) function. (T) indicates terminal data equipment) function. For instance, exercising the (M) strap option will allow communication with a modem; exercising the (T) strap option will allow communication with a terminal. The above shows factory settings for (T).

System Bus Connector

J13

D1	1	2	/RD
D0	3	4	/MEMRQ
D7	5	6	/IORQ
D2	7	8	/WR
D6	9	10	/REFRESH
D5	11	12	/M1
D3	13	14	A0
D4	15	16	A1
SYSRESET	17	18	A2
A4	19	20	A3
A6	21	22	A5
A15	23	24	A7
A13	25	26	A14
A12	27	28	A10
A9	29	30	A11
A8	31	32	/BUSRQ
WAITRQ	33	34	/BUSAK
PCI	35	36	
/INTRQ	37	38	
/HALT	39	40	/CLOCK
SPKR	41	42	/MEM8
	43	44	/MEM4
+ 12V	45	46	
+ 12V	47	48	GND
GND	49	50	+ 5V

Symbol	Pin #	Pin Name	Meaning
D0	3	Data bus	Data Bus (Tri-state, input/output, active high) constitutes an 8-bit bi-directional data exchange with memory and I/O devices.
D1	1	Data bus	
D2	7	Data bus	
D3	13	Data bus	
D4	15	Data bus	
D5	11	Data bus	
D6	9	Data bus	
D7	5	Data bus	
A0	14	Address bus	Address Bus (Tri-state, output, active high) makes up a 16-bit address for up to 65k bytes of memory for I/O devices data exchange. I/O addressing uses the lower 8 bits for direct selection of up to 256 output ports. A0 is the least significant address bit. During refresh time, the lower 7 bits contain a valid refresh address for dynamic memories.
A1	16	Address bus	
A2	18	Address bus	
A3	20	Address bus	
A4	19	Address bus	
A5	22	Address bus	
A6	21	Address bus	
A7	24	Address bus	
A8	31	Address bus	
A9	29	Address bus	
A10	28	Address bus	
A11	30	Address bus	
A12	27	Address bus	
A13	25	Address bus	
A14	26	Address bus	
A15	23	Address bus	
/WR	8	Write	Write (Tri-state, output, active low) indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.
/RD	2	Read	Read (Tri-state, output, active high) indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.
/IORQ	6	I/O Request	Input/Output Request (Tri-state, output, active low) signal indicates that the lower half of

			the address bus holds a valid I/O address for an I/O read or write operation. This signal is also generated with a "/M1" signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus. Interrupt Acknowledge operations occur during "/M1" time, while I/O operations never occur during "/M1" time.
/HALT	39	Halt	Halt (Output, active low) signal indicates that the CPU has executed a Halt Software instruction and is awaiting either a non-maskable or maskable interrupt before operation can resume.
/MEMRQ	4	Memory Request	Memory Request (Tri-state, output, active low) signal indicates that the address bus holds a valid address for a memory read or memory write operation.
/REFRESH	10	Refresh	Refresh (Tri-state, output, active low) indicates that the lower 7 bits of the address contain a refresh address for dynamic memories and the "/MEMRQ" signal should be used to perform a refresh cycle for all dynamic RAMs in the system. During the refresh cycle "A7" is a logic zero and the upper 8 bits of the address bus contain the "I" register.
/M1	12	Machine Cycle One	Machine Cycle One (Tri-state, output, active low) indicates that the current machine cycle is in the op-code fetch cycle of an instruction. Note

			that during the execution of two-byte op-codes, "/M1" will be generated as each op-code is fetched. These two-byte op-codes always begin with a CB, DD, ED, or FD. "/M1" also occurs with "/IORQ" to indicate an interrupt acknowledge cycle.
/BUSAK	34	Bus Acknowledge	Bus Acknowledge (Output, active low) is used to indicate to the requesting device that the CPU address bus, data bus, and control bus signals have been set to their high impedance states and the external device can now control the bus.
/BUSRQ	32	Bus Request	Bus Request (Input, active low) signal is used to request the CPU address bus, data bus, and control signal bus to go to a high impedance state so that other devices can control those buses. When "/BUSRQ" is activated, the CPU will set these buses to a high impedance state as soon as the current CPU machine cycle is finished and the "/BUSAK" signal is activated.
/INTRQ	37	Interrupt Request	Interrupt Request (Input, active low) signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip flop (IFF) is enabled and if the "/BUSRQ" signal is not active.
/WAITRQ	33	Wait Request	Wait Request (Input, active low) indicates to the CPU that the addressed memory or I/O device is not ready for a data transfer. The CPU continues to enter wait states

			for as long as this signal is active. This signal allows memory or I/O devices of any speed to be synchronized to the CPU. Use of this signal postpones refresh as long as it is held active.
/SYSREST	17	System Reset	System Reset (Output, active low) indicates that a reset has been generated either from push button reset or the power on reset circuit. The system reset will occur only once per reset and will be approximately 10 microseconds in duration.
/CLOCK	40	Processor Clock	Processor Clock (Output, active low) is a single-phase system clock of 4 MHz.
PCI	35	Priority Chain In	Priority Chain In (Input, active high) is used to form a priority-interrupt daisy chain when more than one interrupt-driven device is being used. A high level on this pin indicates that no other devices of higher priority are being serviced by a CPU interrupt service routine.
/MEM4	44	Memory Expansion	Memory Expansion (Output, active low) signal is low during "/MEMRQ" for a block of addresses from "4000 thru 7FFF" if the Bank Switch is set for the ROM side of memory.
/MEM8	42	Memory Expansion	Memory Expansion (Output, active low) signal is low during "/MEMRQ" for a block of addresses from "8000 thru BFFF" if the Bank Switch is set for the ROM side of memory.
/SPKR	41	Speaker	Speaker pin provides access to the speaker on the CPU Board. This pin is connected to the open collector output of the speaker

			driver (75451). This output is normally connected thru the speaker and parallel 100 ohm resistor to a + 12 VDC, but can be disconnected by jumper option.
+ 5VDC	50	DC Power	+ 5VDC system power.
GND	49,48	Ground	Ground-System is signal ground and DC return.
+ 12VDC	47,45	DC Power	+ 12VDC system power.
	36	Not Used	
	38	Not Used	
	43	Not Used	
	46	Not Used	

Disk Access Connector

J12

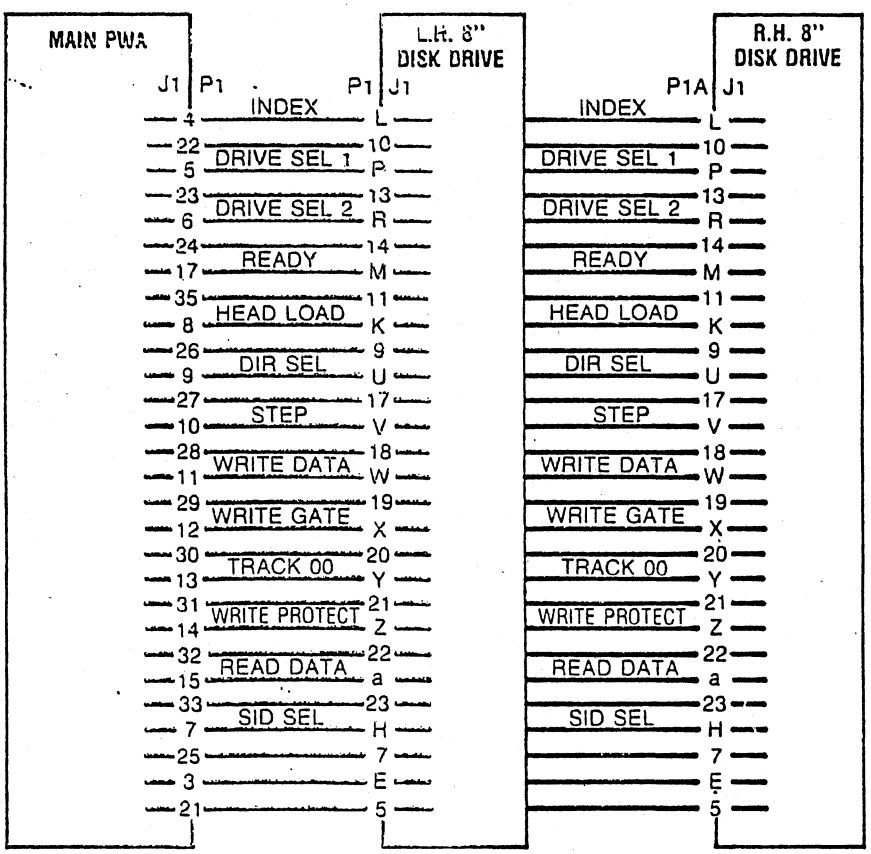
D1	1	2	/RD
D0	3	4	/MREQ
D7	5	6	/IORQ
D2	7	8	/WR
D6	9	10	/BUSAK
D5	11	12	/M1
D3	13	14	A0
D4	15	16	A1
RST	17	18	A2
A4	19	20	A3
A6	21	22	A5
A15	23	24	A7
A13	25	26	A14
A12	27	28	A10
A9	29	30	A11
A8	31	32	/BUSRQ
16 MHz Clock	33	34	/BUSAK1
/HALT	35	36	/1797CS
INT	37	38	PRIO
PP5	39	40	4 MHz Clock
PP2	41	42	PP4
PP1	43	44	PP0
+ 12V	45	46	NM1
GND	47	48	GND
DSKWAT	49	50	+ 5V
	51	52	+ 5V
J1-19	53	54	J1-10
J1-18	55	56	J1-9
J1-17	57	58	J1-8
J1-16	59	60	J1-2
J1-15	61	62	J1-3
J1-14	63	64	J1-4
J1-13	65	66	J1-5
J1-12	67	68	J1-6
J1-11	69	70	J1-7
	71	72	

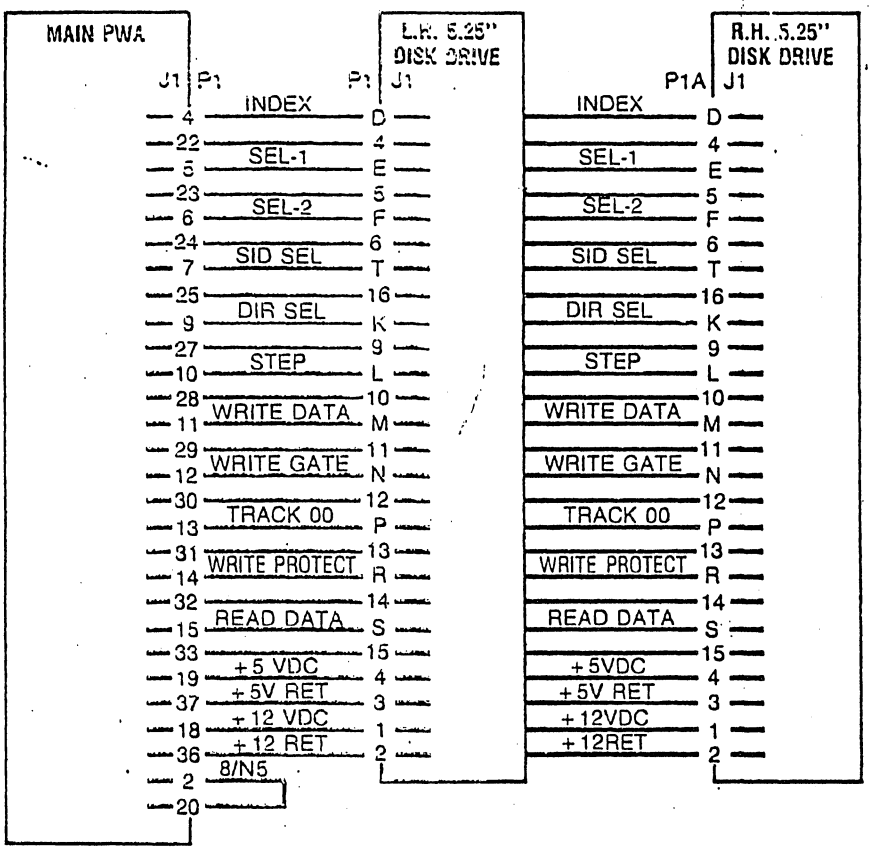
Symbol	Pin #	Pin Name	Meaning
D0	3	Data bus	Data Bus (Tri-state, input/output, active high) constitutes an 8-bit bi-directional data exchange with memory and I/O devices.
D1	1	Data bus	
D2	7	Data bus	
D3	13	Data bus	
D4	15	Data bus	
D5	11	Data bus	
D6	9	Data bus	
D7	5	Data bus	
A0	14	Address Bus	Address bus A0-A15 provides addresses for 65k bytes of memory. Bit A0 and A1 while under /RD and /WR control select the register to receive transfer of data on D0-D7: A1 A0 /RD /WR 0 0 Status REG Command REG 0 1 Track REG Track REG 1 0 Sector REG Sector REG 1 1 Data REG Data REG
A1	16	Address Bus	
A2	18	Address Bus	
A3	20	Address Bus	
A4	19	Address Bus	
A5	22	Address Bus	
A6	21	Address Bus	
A7	24	Address Bus	
A8	31	Address Bus	
A9	29	Address Bus	
2A10	8	Address Bus	
A11	30	Address Bus	
2A12	7	Address Bus	A5 while under /RD and /WR Control Select Density: 0 = double density 1 = single density
A13	25	Address Bus	
A14	26	Address Bus	
A15	23	Address Bus	
PP0	44	SYS-PIO Port A	Port A bit 0
PP1	43	SYS-PIO Port A	Port A bit 1
PP2	41	SYS-PIO Port A	Port A bit 2
PP4	42	SYS-PIO Port A	Port A bit 4
PP5	39	SYS-PIO Port A	Port A bit 5
PRIO	38	SYS-PIO	
/DSKWAT	49	Disk Wait	Generates Wait signal to CPU.
/RD	2	Read	Controls input on the data registers D0-D7.
/MREQ	4	Memory Request	/MREQ indicates that the address bus holds a valid address for a memory read or memory write operation.

/IORQ	6	I/O Request	/IORQ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. /IORQ is also generated concurrently with /M1 during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.
/WR	8	Write	Controls output on the data registers D0-D7.
/BUSACK	10	Bus Acknowledge	/BUSACk indicates to the requesting device that the CPU address bus, data bus, and control signals /MREQ, /IORQ, /RD, /WR have entered the high impedance states. The external circuitry can now control these lines.
/M1	12	Machine Cycle 1	/M1, together with /MREQ indicates that the current machine cycle is the op-code fetch cycle of an instruction execution. /M1, together with /IORQ, indicates an interrupt acknowledge cycle.
/BUSREQ	32	Bus Request	/BUSREQ has the highest priority and is always recognized at the end of the current machine cycle. /BUSREQ forces the CPU address bus, data bus, and control signals /MREQ, /IORQ, /RD, and /WR to go to a high impedance state so that other devices can control these lines.
/BUSAK1	34	Bus Acknowledge	/BUSAK1 is daisy-chained Bus Acknowledge output which indicates to the requesting device that the CPU address bus, data bus, and control signals /MREQ, /IORQ, /RD, /WR have entered the high impedance states. The

			external circuitry can now control these lines.
/1797CS	36	Chip Select	/1797CS logic low selects the Floppy Disk Controller chip and enables computer communication with the device.
INT	37	Interrupt Request	INT is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled.
NMI	46	Non-Maskable Interrupt	NMI is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop and automatically forces the CPU to restart at location 0066h.
/HALT	35	Halt	/HALT indicates that the CPU has executed a Halt instruction and is awaiting either a non-maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOPs to maintain memory refresh.
16MHz	33	Clock	16 MHz clock.
CLK	40	Clock	4 MHz clock.
J1-2	60	Device I/O Interface	All interface lines use negative logic.
J1-3	62	Device I/O Interface	
J1-4	64	Device I/O Interface	
J1-5	66	Device I/O Interface	
J1-6	68	Device I/O Interface	
J1-7	70	Device I/O Interface	
J1-8	58	Device I/O Interface	
J1-9	56	Device I/O Interface	
J1-10	54	Device I/O Interface	
J1-11	69	Device I/O Interface	
J1-12	67	Device I/O Interface	
J1-13	65	Device I/O Interface	
J1-14	63	Device I/O Interface	

J1-15	61	Device I/O Interface	
J1-16	59	Device I/O Interface	
J1-17	57	Device I/O Interface	
J1-18	55	Device I/O Interface	
J1-19	53	Device I/O Interface	
+ 12VDC	45	DC Voltage	+ 12 Volts DC
GND	47	Ground	Ground and DC Return
GND	48	Ground	Ground and DC Return
+ 5VDC	50	DC Voltage	+ 5 Volts DC
+ 5VDC	52	DC Voltage	+ 5 Volts DC
RST	17	Reset	Reset indicates that a System Reset has been generated either from push button reset or power on reset.
	51	Not Used	
	71	Not Used	
	72	Not Used	


MAIN PWA
J1
P1
P1
J1
L.H. 8" DISK DRIVE
P1A
J1
R.H. 8" DISK DRIVE
INDEX
DRIVE SEL 1
DRIVE SEL 2
READY
HEAD LOAD
DIR SEL
STEP
WRITE DATA
WRITE GATE
TRACK 00
WRITE PROTECT
READ DATA
SID SEL


MAIN PWA
J1
P1
L.H. 5.25" DISK DRIVE
P1
J1
R.H. 5.25" DISK DRIVE
P1A
J1
INDEX
SEL-1
SEL-2
SID SEL
DIR SEL
STEP
WRITE DATA
WRITE GATE
TRACK 00
WRITE PROTECT
READ DATA
+5 VDC
+5V RET
+12 VDC
+12 RET
8/N5
+5VDC
+12VDC
+12RET

Notes

Theory of Operation

The display processor houses the system board, disk drive daughter board, the CRT, the power supply, and one bus expansion slot.

The system board has the following:

- Central Processing Unit (CPU)
- 6 to 8k of Read Only Memory (ROM)
- 64k of Random Access (Read/Write) Memory (RAM)
- Counter Timer Circuit (CTC)
- Serial Input/Output Controller (SIO)
- Parallel Input/Output Controller (PIO)
- Two RS-232-C Serial I/O Ports
- Dual 8-bit Parallel Ports
- CRT Controller and CRT Refresh Memory
- Speaker
- Disk Drive Daughter Board Connector
- Bus Expansion Connector
  - 8086 Co-processor (16/8 system)
- Parallel Keyboard Interface

CPU

The CPU is a Zilog Z80-A operating with a clock rate of 4 Mhz. It is initialized to use Interrupt Mode 2 by the ROSR monitor at power on. The Z80-A also provides refresh to the 64k of dynamic memory on the system board. Therefore, the I and R registers should not be altered by an application program.

ROM and RAM Memory

The System Board has two banks of memory. Bank 1 has 64k of RAM. Bank 0 has up to 8K of ROM.

When power is turned on or RESET is pressed, the Monitor, ROM/CRT RAM (Bank 0), is enabled by the hardware and the contents of the monitor ROM are moved by the CPU to the program memory starting at location F000H. When the move is complete, the CPU transfers control to

location F000H and RAM (Bank 1) is enabled. Bank 0 is also enabled when a character is sent to the screen.

6-8k ROM

The CPU board has provisions for 4-2k x 8 Read Only Memory devices. The first 3 (U33, U34 & U35) store the firmware for the ROSR monitor. The fourth (U36) provides translation tables and related firmware for the position-encoded low profile keyboard.

64k RAM

The 64k byte (65536 x 8) RAM provides space for a portion of the ROSR monitor (upper 4k F000h - FFFFh), and 60k (0000h - EFFFh) is free for programs to execute in such as an operating system and an application program. This RAM is dynamic and refresh is provided by the Z80-A CPU.

Counter Timer Circuit (CTC)

The CTC has four independently-programmable counter/timer channels, each with a readable downcounter and a selectable 16 or 256 prescaler. Downcounters are reloaded automatically at zero. Each channel is programmed with two bytes. Once started, the CTC counts down, reloads its time constant automatically, and resumes counting. Internally, the CTC generates a unique vector for each channel.

Serial Input/Output Controller (SIO)

The Serial I/O Controller has two independent, full-duplex channels with separate control and status lines for modems or other devices. Data rates are from 50 to 19,200 bits/second. Channel A (modem) supports both Asynchronous and Synchronous protocols. Channel B (printer) is dedicated to Asynchronous. The receiver is quadruple-buffered and the transmitter is double-buffered. The controller also supports daisy-chain interrupt vectoring for interrupts without external logic.

Serial I/O Ports

The 820-II CPU board contains a Z80-A SIO that provides two user-accessible serial ports to the 25-pin printer and modem connectors on the rear of the display processor. The Communications port is capable of operating in synchronous or asynschronous modes, while the Printer port is only capable of operating asynchronously. On an Etch 2 CPU, there is a 30-pin connector. Selection of synchronous or asynchronous mode is under program control as opposed to the Etch 1 CPU (with a 40-pin

connector) where a physical change is required to make the sync or async selection.

Parallel Input/Output Controller (PIO)

There is a System and a General Purpose Parallel I/O Controller which provides direct interface between the CPU and the peripheral devices. Each controller has two 8-bit I/O ports. The System PIO is dedicated for keyboard input, memory bank and CRT font selection, and floppy disk drive and side selection. The General Purpose PIO provides the user with a dual 8-bit parallel I/O port for interfacing with peripherals.

Parallel Port

The Z80-A General Purpose PIO is accessible on the main CPU board on connector J8. This PIO is programmed by the ROSR monitor at power-on to provide a parallel Centronics-compatible interface for a parallel printer. A transceiver is physically located between the Z80-A PIO and the J8 connector. Jumpers must be installed on option connector J11 to select whether the transceiver will transmit or receive data. See also page 24.

CRT Controller

The CPU board contains the 2k of refresh RAM where the characters that are to be displayed on the screen are stored. It also has the necessary electronics to provide the control signals (sync and video) to the CRT monitor. The CPU board has two character font ROMs; each font ROM contains two character sets.

U57	Normal white on black font
	Reverse video font
U58	Normal white on black font
	Graphic character font

The CRT driver in the ROSR monitor translates character-level escape sequences into commands as to which of the font ROMs to select and which of the two fonts inside the selected font ROM to select. Basically, characters that are stored in the CRT's refresh memory address the selected font ROM; the font ROM provides dot information to the video input of the CRT so the character can be displayed.

The characters on the CRT can have one of the following attributes:

Blink
Inverse video
Graphics
Low intensity

The most significant bit of the character stored in the CRT's refresh memory determines if the character is to be displayed with its attribute enabled.

The ROSR monitor provides a character-oriented command format for controlling the screen and font ROM selection. It is recommended that programs use this method to control the CRT and its attributes.

CRT RAM

Memory Allocation

The CRT RAM occupies 3000H - 3FFFH in bank 0 (System Bank). Each 80-character line on the CRT is allocated 128 bytes in the CRT RAM. Listed below are the starting and ending addresses for each of the 24 rows in the CRT RAM. The example (at the bottom) shows some character locations in CRT memory. (Assumes scroll register = 23)

Row	Starting Address	Ending Address
0	3000H	304FH
1	3080H	30CFH
2	3100H	314FH
3	3180H	31CFH
4	3200H	324FH
5	3280H	32CFH
6	3300H	334FH
7	3380H	33CFH
8	3400H	344FH
9	3480H	34CFH
10	3500H	354FH
11	3580H	35CFH
12	3600H	364FH
13	3680H	36CFH
14	3700H	374FH
15	3780H	37CFH
16	3800H	384FH
17	3880H	38CFH
18	3900H	394FH
19	3980H	39CFH
20	3A00H	3A4FH
21	3A80H	3ACFH
22	3B00H	3B4FH
23	3B80H	3BCFH

Row	Column	CRT Memory Address
0	0	3000H
0	79	304FH
1	1	3081H
1	5	3085H
23	0	3B80H
23	1	3B81H
23	79	3BCFH

Scroll Register

To eliminate the delay associated with software scrolling, hardware scrolling is employed. Writing into the scroll register (Port 14h) adds an offset to the line address developed by the line counter. For instance, with an offset of zero (scroll register = 0), the data at location 3000H (in the CRT refresh memory) will be displayed on the bottom row of the display. If the offset is 23, the data at location 3000H will be displayed on the top row of the screen. The scroll register is loaded from A8 to A15 rather than D0 to D7. Therefore, the scroll value must be in the B register if an indirect OUT instruction is used.

Scroll Register	Row 0, Column 0	Row 23, Column 0
23	3000H	3B80H
22	3080H	3B00H
21	3100H	3A80H
20	3180H	3A00H
19	3200H	3980H
18	3280H	3900H
17	3300H	3880H
16	3380H	3800H
15	3400H	3780H
14	3480H	3700H
13	3500H	3680H
12	3580H	3600H
11	3600H	3580H
10	3680H	3500H
9	3700H	3480H
8	3780H	3400H
7	3800H	3380H
6	3880H	3300H
5	3900H	3280H
4	3980H	3200H
3	3A00H	3180H
2	3A80H	3100H
1	3B00H	3080H
0	3B80H	3000H

Speaker

The 820-II and 16/8 have an audio speaker connected to two I/O ports (28h and 29h). Outputting to one I/O port causes the speaker cone to be pushed out; outputting to the other I/O port pulls in the speaker cone. The actual value output to these ports has no significance. To generate a beep, the application program can simply send an ASCII Bell character to the CRT. To generate a tone other than the standard bell character, the program must move the speaker cone in and out at the desired frequency.

Disk Drive Daughter Board

The disk drive connector on the rear is a "dual personality" connector, depending on which disk drive daughter board is installed on the mother board. Presently, there are two types of disk interface:

Shugart SASI interface controller suitable for interfacing to a SA1403D Rigid Disk Controller.

Floppy-only interface suitable for interfacing to Shugart SA800/SA400L/SA850/SA450 dual daisy-chained disk configurations.

The ROSR monitor detects which daughter board is installed at power-on and selects the appropriate physical disk driver firmware to process physical disk drive requests.

Caution:

If a rigid disk drive unit (U07, U08) is connected to a floppy display/processor (U03/H69, U04), the rigid controller PWB will be destroyed when power is switched on. The rigid disk drive unit must be connected only to a rigid display/processor (U05/H70, U06). Connecting a floppy disk drive unit (929/T66/973/F10, E41/E44/E42/E89) to a rigid display/processor (U05/H70, U06) may cause the processor PWB to fail. Before connecting any disk drive unit to a display/processor, check that the configuration of the display/processor is compatible with the disk drive unit. The configuration can be determined in one of two ways. (1) Check the product code of the display/processor. The product code is the first three digits of the serial number, located on the underside of the display processor. (2) Verify that the proper drive interface PWB is installed by checking the part number.

System Bus

The System Bus contains an 8-bit Data Bus (Tri-state, Input/Output) bi-directional Data exchange with memory and I/O devices. It has a 16-bit Address Bus to address up to 64k of memory for I/O devices data exchange.

Keyboard Interface

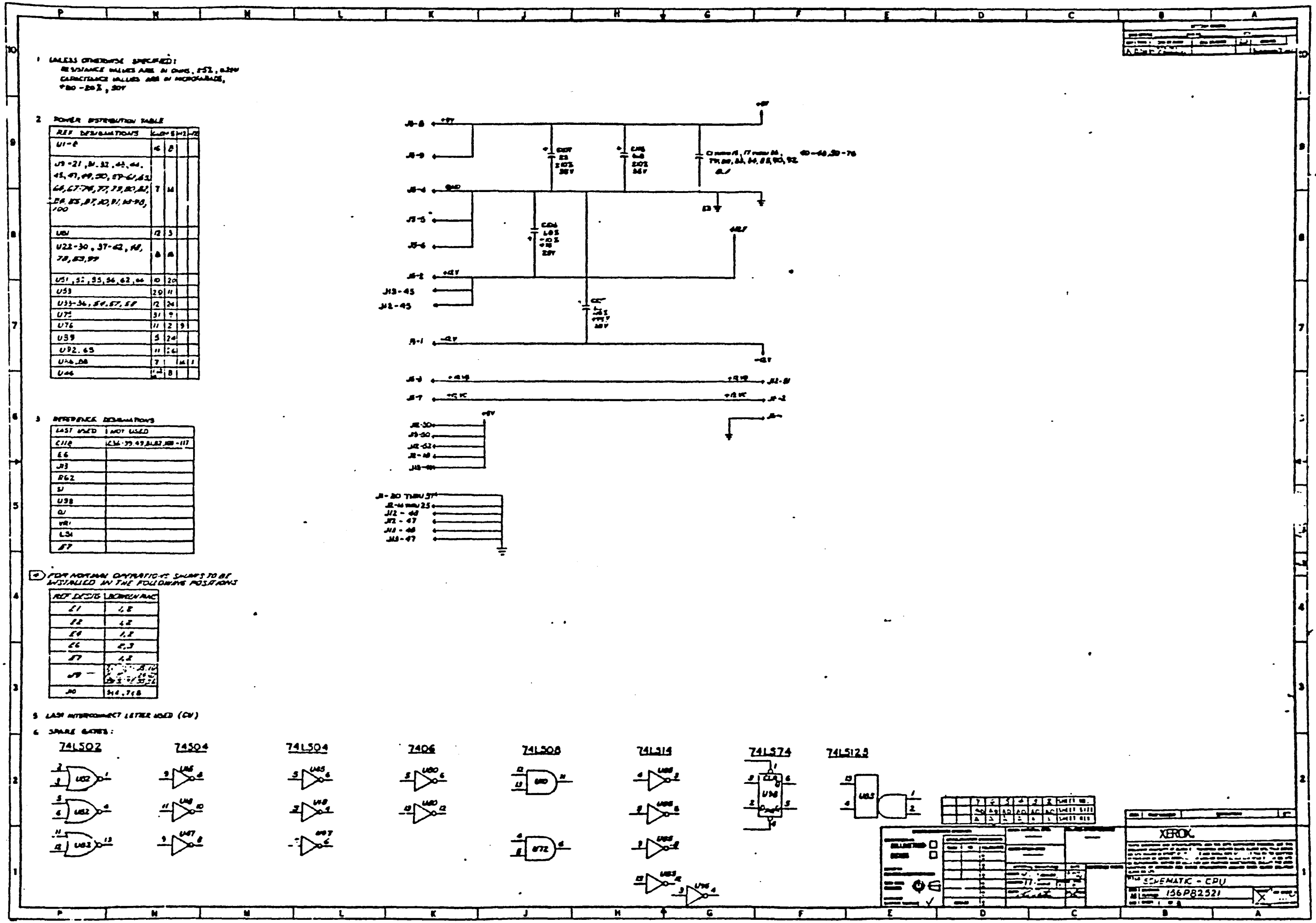
The keyboard FIFO (Etch 2 CPU only) has space for 16 (decimal) entries. Associated with the keyboard FIFO are input and output position pointers and a count of the number of entries currently in the FIFO.

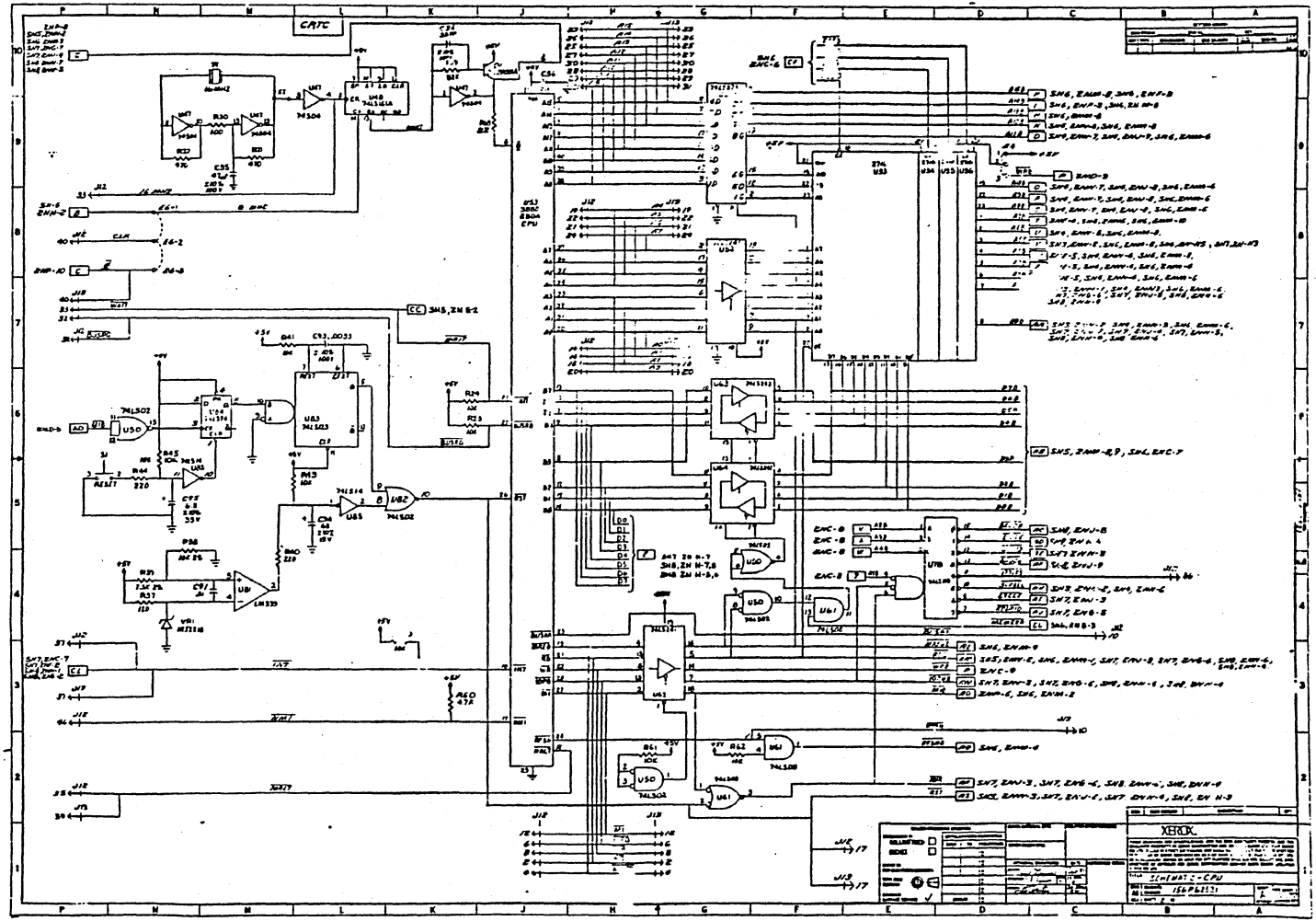
The available memory pointers provide the addresses bounding the available unused RAM in the memory reserved for system use. Although these pointers are a supported feature, there is no guaranteed available memory size.

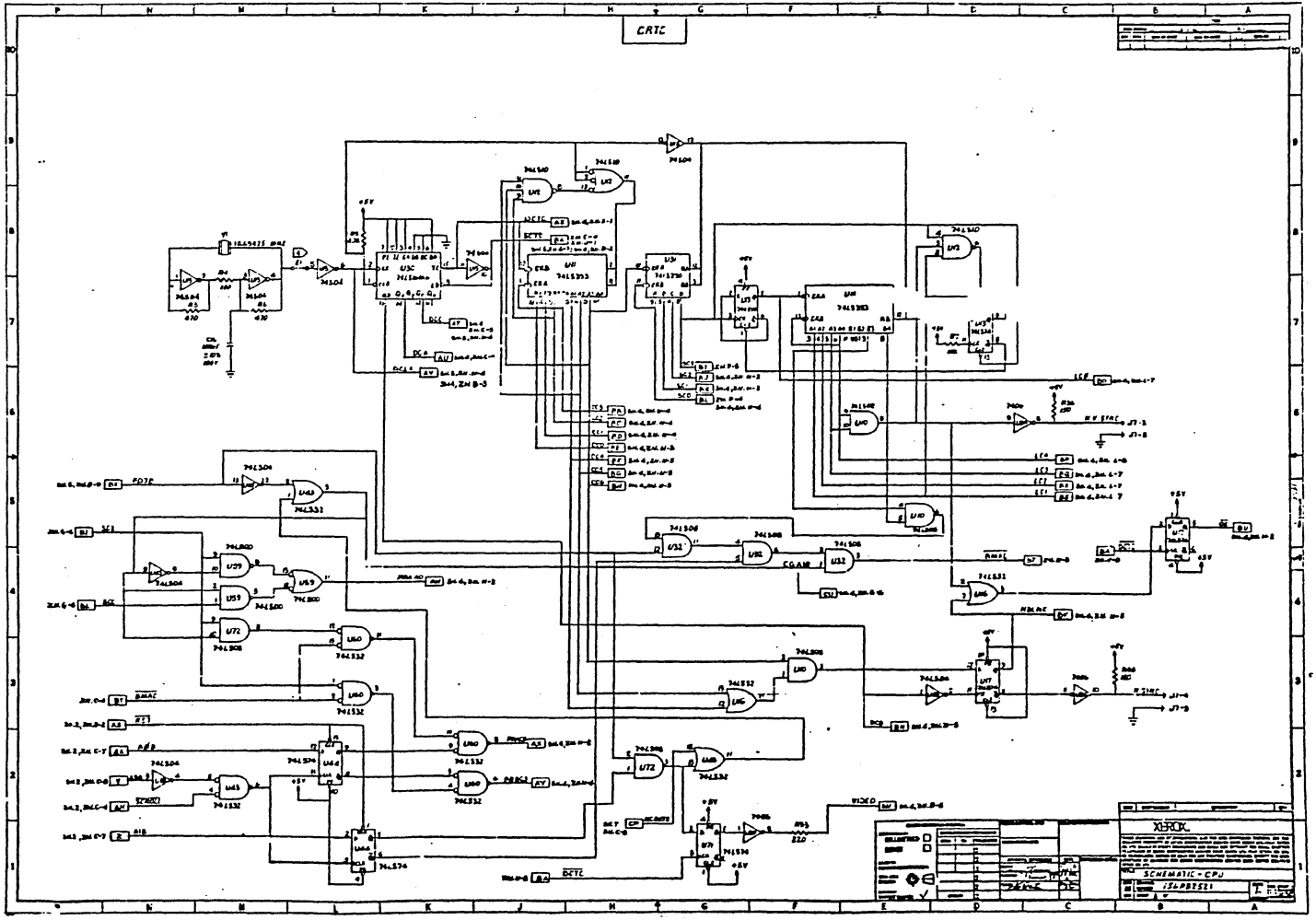
There are two tables used to disk map a logical disk to its physical driver. The first table, Seltab, associates a logical disk number with a physical disk number. The second table, Drvtab, identifies which physical disk driver is appropriate to use with the selected physical disk.

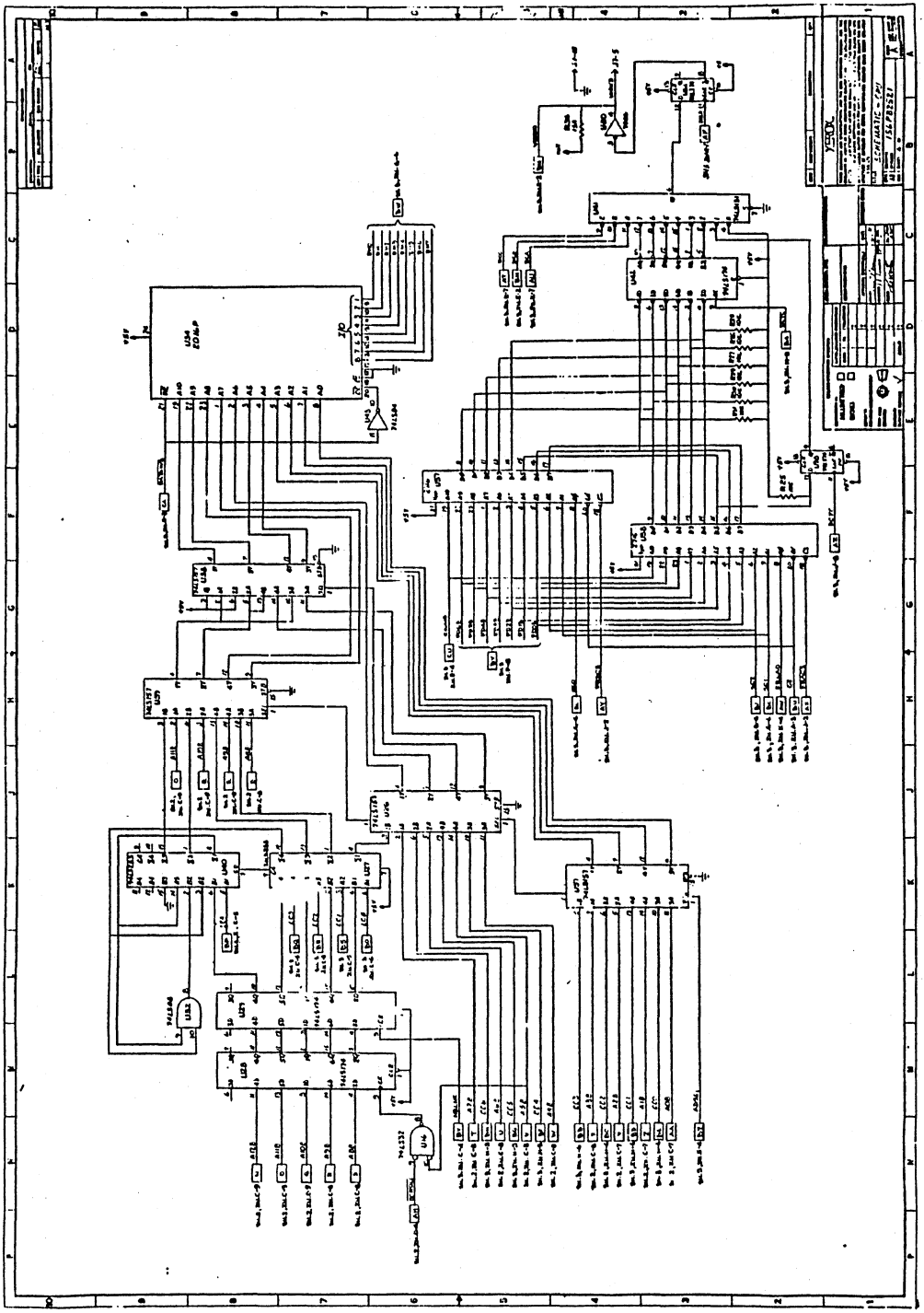
The physical driver command block is a collection of all information necessary for the disk system to perform the requested disk activity.

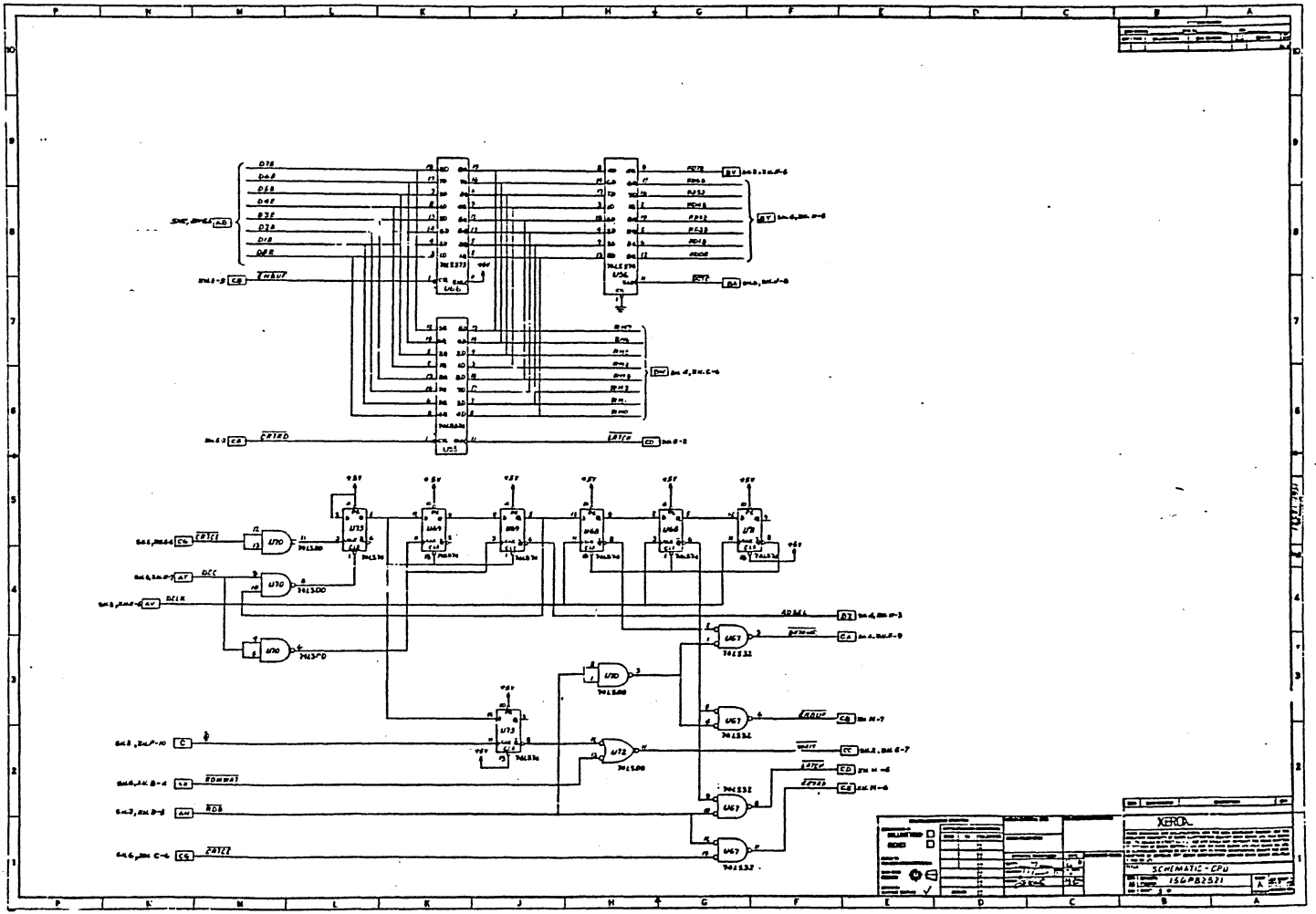
The timer and clock variables are a collection of locations used for maintaining the one second timer and the time-of-day clock and calendar. The console command line buffer immediately follows these variables.

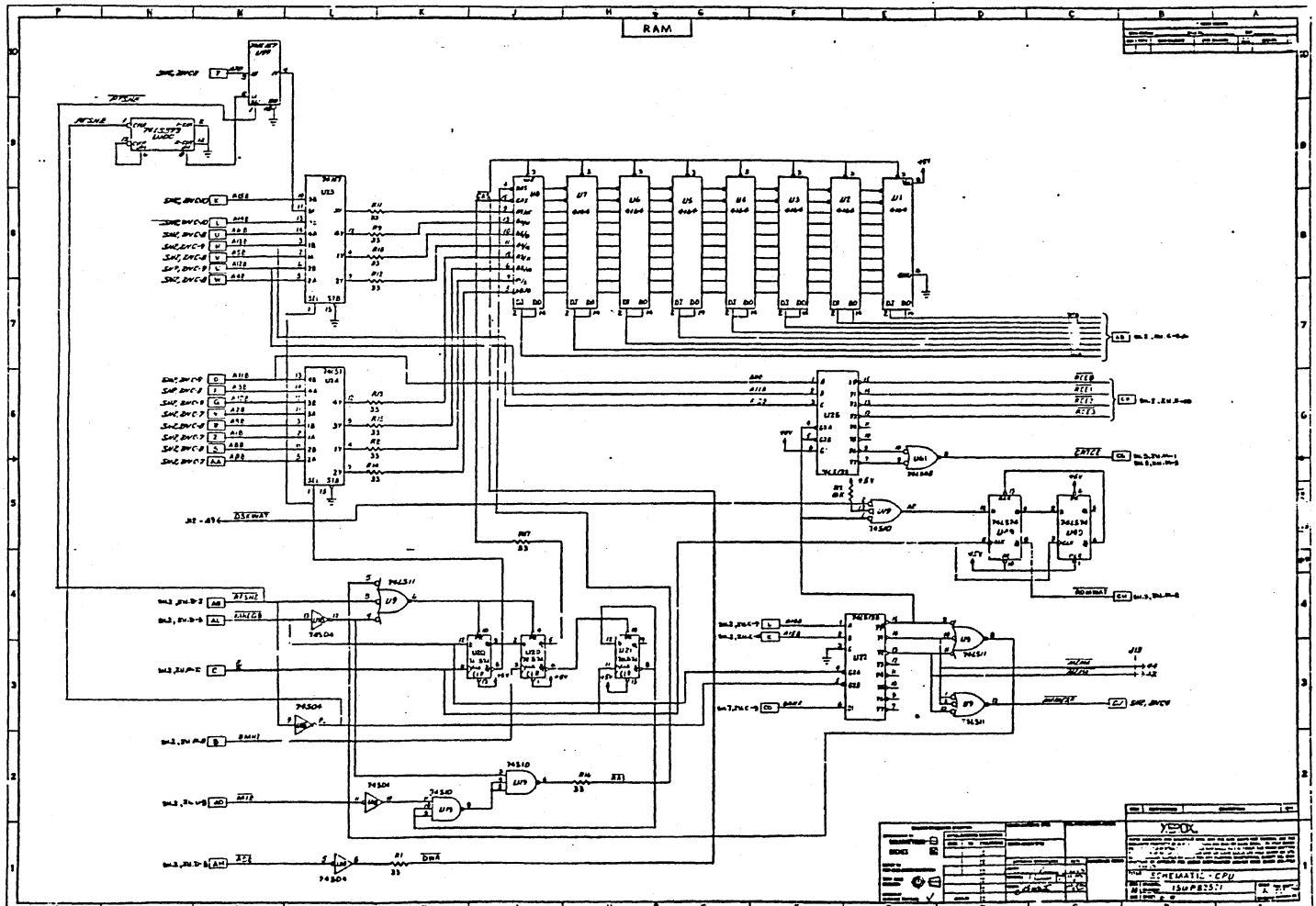

1 UNLESS OTHERWISE SPECIFIED:
RESISTANCE VALUES ARE IN OHMS, ±5%, 0.25W
CAPACITANCE VALUES ARE IN MICROFARADS, +80 -20%, 50V
2 POWER DISTRIBUTION TABLE
3 REFERENCE DESIGNATIONS
LAST USED
NOT USED
4 FOR NORMAL OPERATION SHUNTS TO BE INSTALLED IN THE FOLLOWING POSITIONS
5 LAST INTERCONNECT LETTER USED (CV)
6 SPARE GATES:
74LS02
74S04
74LS04
7406
74LS08
74LS14
74LS74
74LS125
XEROX
SCHEMATIC - CPU
156P82521

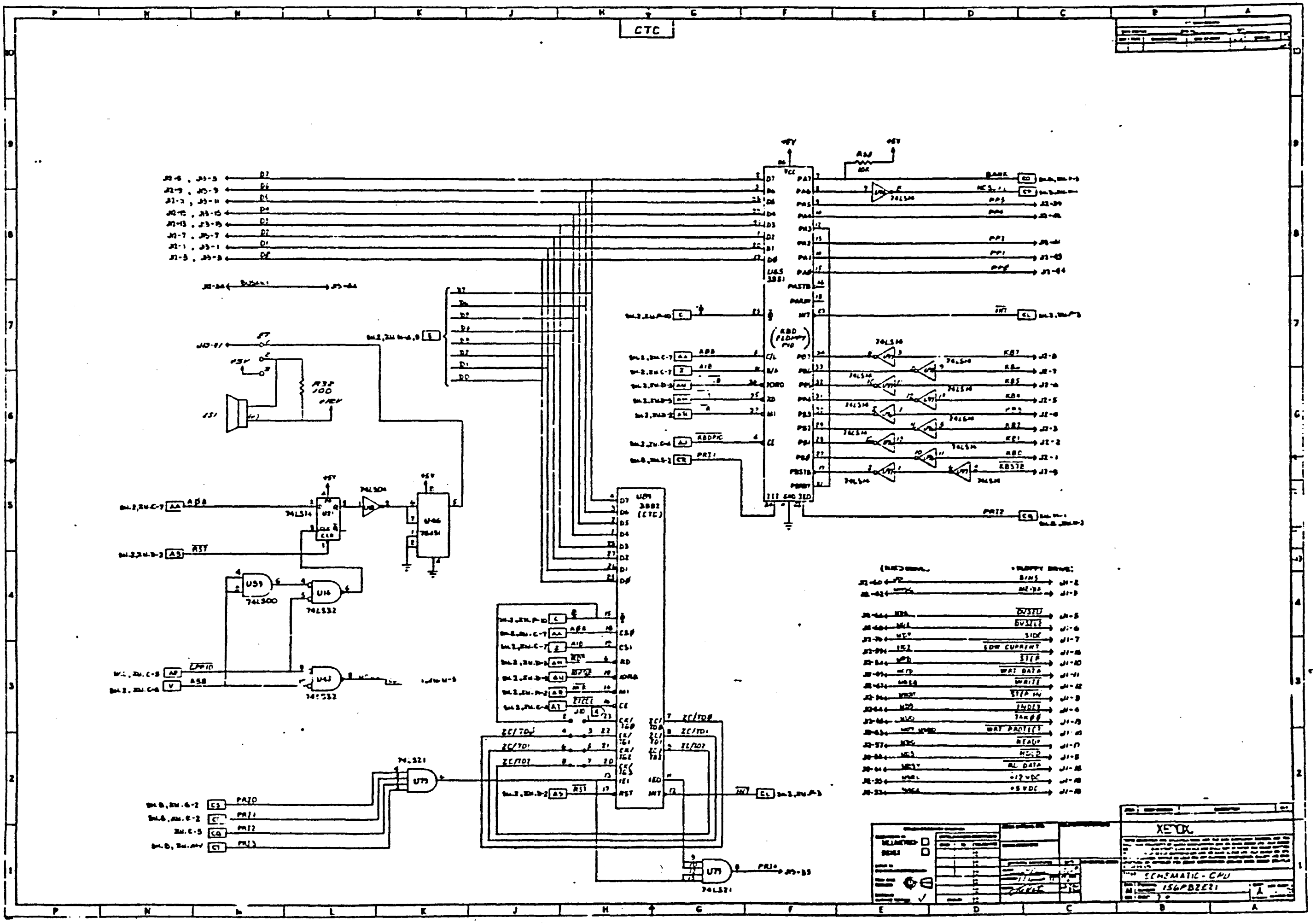

CRTC
XEROX

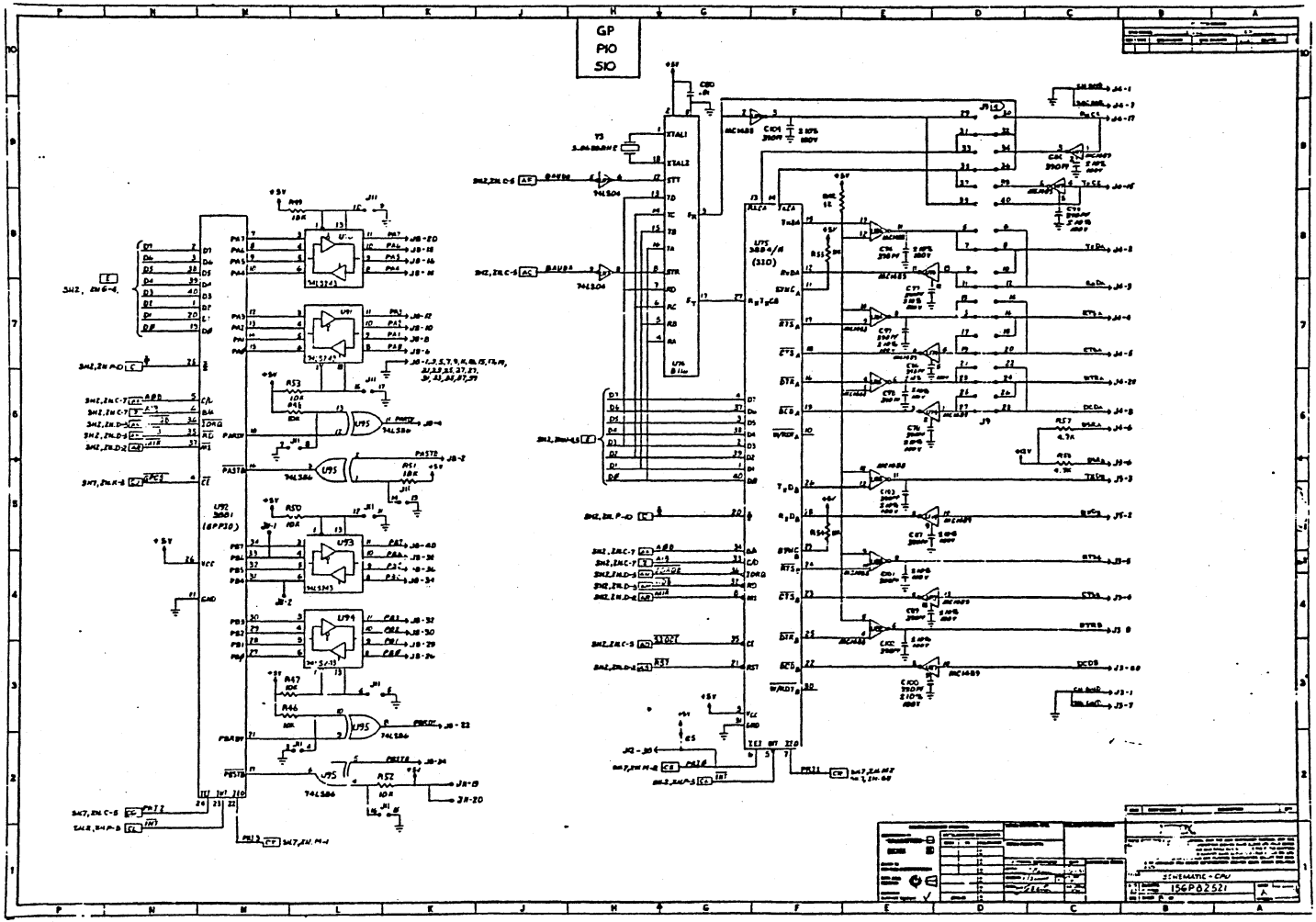

CRTC
SCHEMATIC - CPU

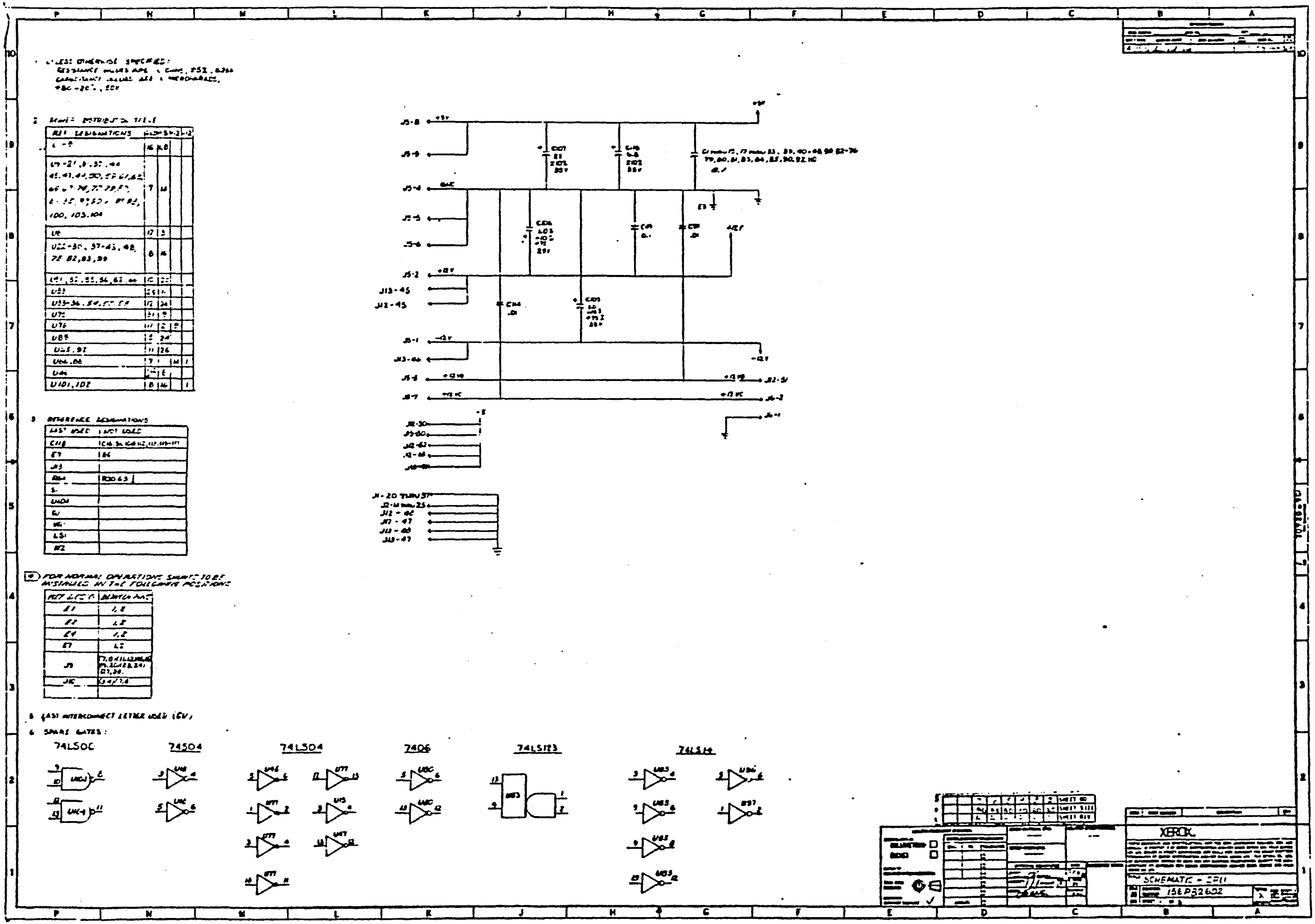


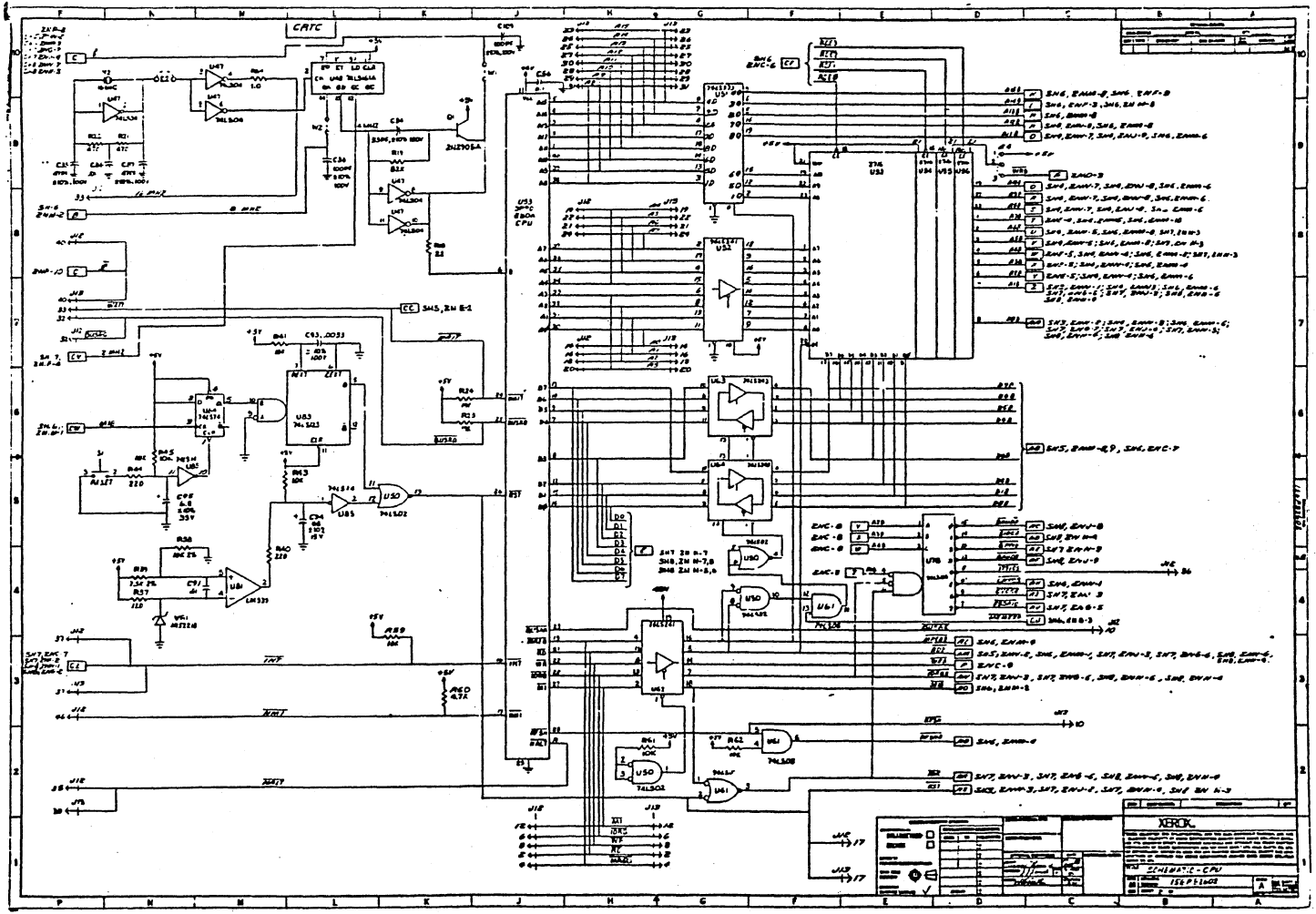

SCHEMATIC - CPU
XEROX

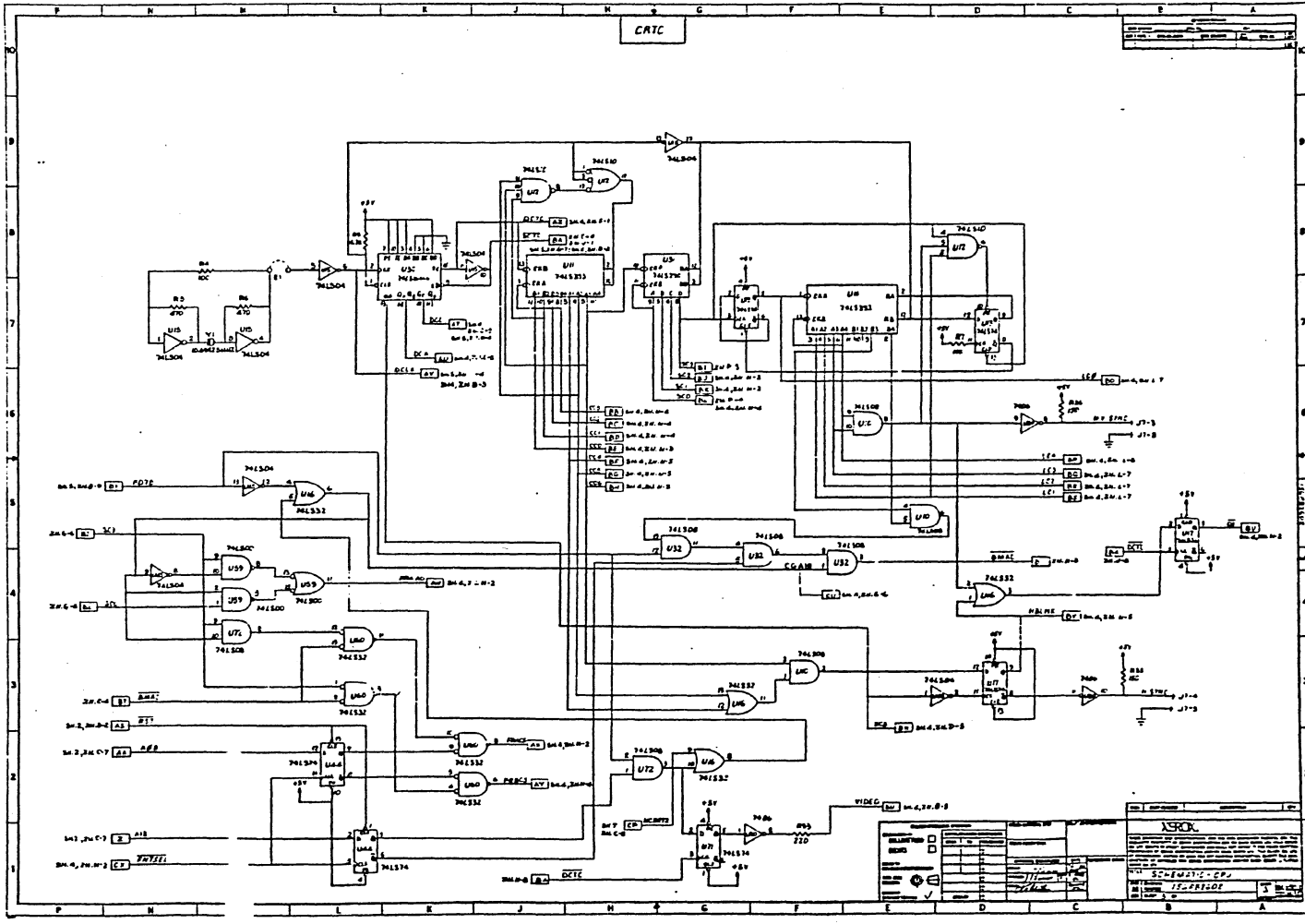

RAM
SCHEMATIC - CPU

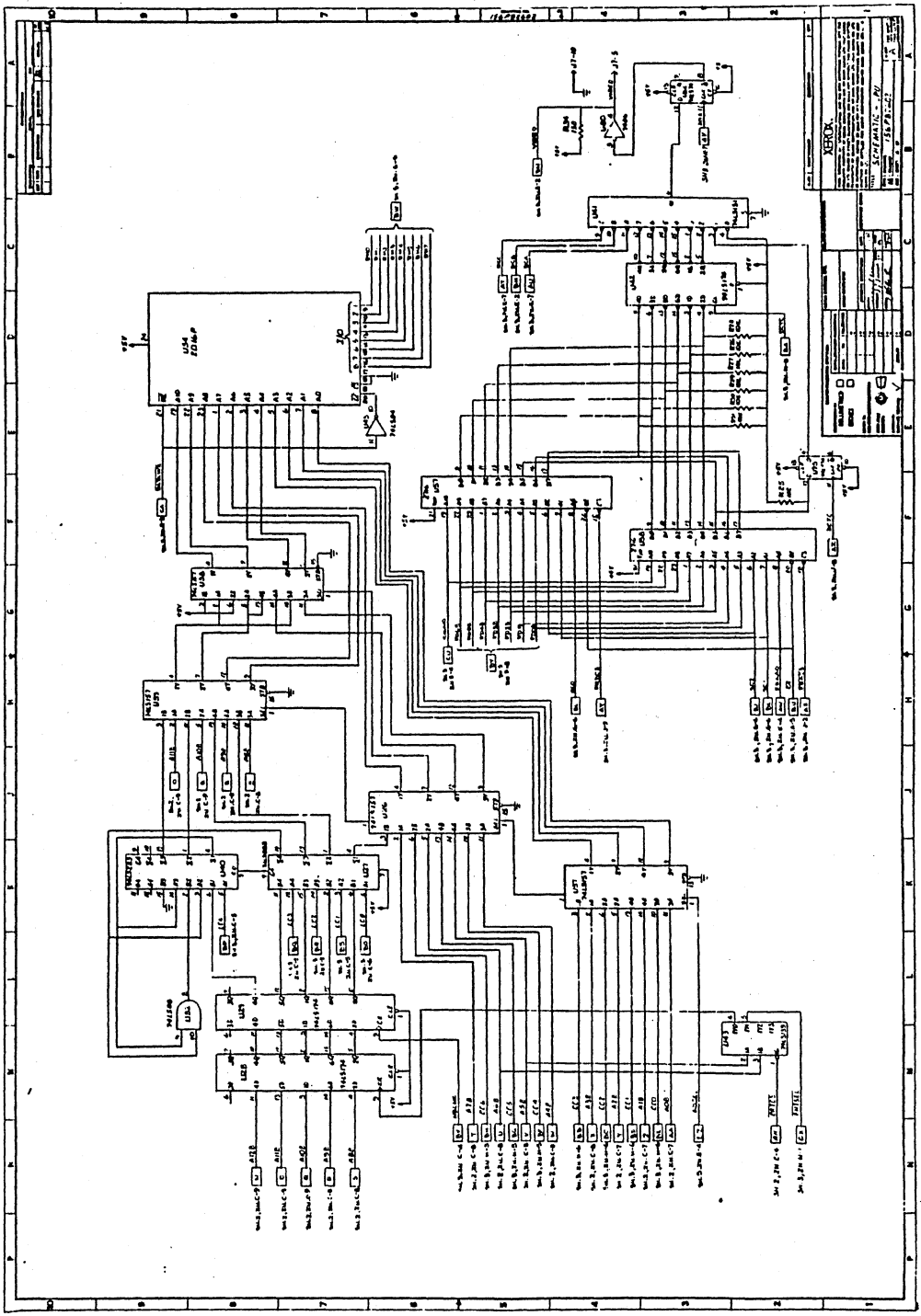

CTC
U65 3881
U59 3882 (CTC)
74LS14
74LS00
74LS32
74LS21
XEROX
SCHEMATIC - CPU


GP
PIO
SIO
SCHEMATIC - CPU
156P02521


SPARE GATES:
74LS04
74LS04
7406
74LS123
74LS14
XEROX
SCHEMATIC - CPU




CRTC
74LS04
74LS32
74LS08
74LS00
74LS74
74LS157
74LS153
74LS283
74LS374
74LS86
74LS240
74LS244
74LS273
74LS377
74LS367
74LS368
74LS373
74LS175
74LS139
74LS138
74LS257
74LS158
74LS109
74LS51
74LS54
74LS20
74LS30
74LS10
74LS11
74LS12
74LS14
74LS21
74LS27
74LS42
74LS85
74LS93
74LS95
74LS107
74LS112
74LS123
74LS125
74LS126
74LS132
74LS133
74LS136
74LS145
74LS147
74LS148
74LS151
74LS152
74LS154
74LS155
74LS156
74LS160
74LS161
74LS162
74LS163
74LS164
74LS165
74LS166
74LS168
74LS169
74LS170
74LS173
74LS174
74LS175
74LS181
74LS182
74LS190
74LS191
74LS192
74LS193
74LS194
74LS195
74LS196
74LS197
74LS221
74LS240
74LS241
74LS242
74LS243
74LS244
74LS245
74LS247
74LS248
74LS249
74LS251
74LS253
74LS257
74LS258
74LS259
74LS260
74LS261
74LS266
74LS273
74LS279
74LS280
74LS283
74LS290
74LS293
74LS295
74LS298
74LS299
74LS322
74LS323
74LS324
74LS347
74LS348
74LS352
74LS353
74LS354
74LS355
74LS356
74LS357
74LS365
74LS366
74LS367
74LS368
74LS373
74LS374
74LS375
74LS377
74LS378
74LS379
74LS381
74LS382
74LS384
74LS385
74LS386
74LS390
74LS393
74LS395
74LS396
74LS398
74LS399
74LS424
74LS425
74LS426
74LS427
74LS428
74LS429
74LS430
74LS431
74LS432
74LS433
74LS434
74LS435
74LS436
74LS437
74LS438
74LS439
74LS440
74LS441
74LS442
74LS443
74LS444
74LS445
74LS446
74LS447
74LS448
74LS449
74LS450
74LS451
74LS452
74LS453
74LS454
74LS455
74LS456
74LS457
74LS458
74LS459
74LS460
74LS461
74LS462
74LS463
74LS464
74LS465
74LS466
74LS467
74LS468
74LS469
74LS470
74LS471
74LS472
74LS473
74LS474
74LS475
74LS476
74LS477
74LS478
74LS479
74LS480
74LS481
74LS482
74LS483
74LS484
74LS485
74LS486
74LS487
74LS488
74LS489
74LS490
74LS491
74LS492
74LS493
74LS494
74LS495
74LS496
74LS497
74LS498
74LS499
74LS500
74LS501
74LS502
74LS503
74LS504
74LS505
74LS506
74LS507
74LS508
74LS509
74LS510
74LS511
74LS512
74LS513
74LS514
74LS515
74LS516
74LS517
74LS518
74LS519
74LS520
74LS521
74LS522
74LS523
74LS524
74LS525
74LS526
74LS527
74LS528
74LS529
74LS530
74LS531
74LS532
74LS533
74LS534
74LS535
74LS536
74LS537
74LS538
74LS539
74LS540
74LS541
74LS542
74LS543
74LS544
74LS545
74LS546
74LS547
74LS548
74LS549
74LS550
74LS551
74LS552
74LS553
74LS554
74LS555
74LS556
74LS557
74LS558
74LS559
74LS560
74LS561
74LS562
74LS563
74LS564
74LS565
74LS566
74LS567
74LS568
74LS569
74LS570
74LS571
74LS572
74LS573
74LS574
74LS575
74LS576
74LS577
74LS578
74LS579
74LS580
74LS581
74LS582
74LS583
74LS584
74LS585
74LS586
74LS587
74LS588
74LS589
74LS590
74LS591
74LS592
74LS593
74LS594
74LS595
74LS596
74LS597
74LS598
74LS599
74LS600
74LS601
74LS602
74LS603
74LS604
74LS605
74LS606
74LS607
74LS608
74LS609
74LS610
74LS611
74LS612
74LS613
74LS614
74LS615
74LS616
74LS617
74LS618
74LS619
74LS620
74LS621
74LS622
74LS623
74LS624
74LS625
74LS626
74LS627
74LS628
74LS629
74LS630
74LS631
74LS632
74LS633
74LS634
74LS635
74LS636
74LS637
74LS638
74LS639
74LS640
74LS641
74LS642
74LS643
74LS644
74LS645
74LS646
74LS647
74LS648
74LS649
74LS650
74LS651
74LS652
74LS653
74LS654
74LS655
74LS656
74LS657
74LS658
74LS659
74LS660
74LS661
74LS662
74LS663
74LS664
74LS665
74LS666
74LS667
74LS668
74LS669
74LS670
74LS671
74LS672
74LS673
74LS674
74LS675
74LS676
74LS677
74LS678
74LS679
74LS680
74LS681
74LS682
74LS683
74LS684
74LS685
74LS686
74LS687
74LS688
74LS689
74LS690
74LS691
74LS692
74LS693
74LS694
74LS695
74LS696
74LS697
74LS698
74LS699
74LS700
74LS701
74LS702
74LS703
74LS704
74LS705
74LS706
74LS707
74LS708
74LS709
74LS710
74LS711
74LS712
74LS713
74LS714
74LS715
74LS716
74LS717
74LS718
74LS719
74LS720
74LS721
74LS722
74LS723
74LS724
74LS725
74LS726
74LS727
74LS728
74LS729
74LS730
74LS731
74LS732
74LS733
74LS734
74LS735
74LS736
74LS737
74LS738
74LS739
74LS740
74LS741
74LS742
74LS743
74LS744
74LS745
74LS746
74LS747
74LS748
74LS749
74LS750
74LS751
74LS752
74LS753
74LS754
74LS755
74LS756
74LS757
74LS758
74LS759
74LS760
74LS761
74LS762
74LS763
74LS764
74LS765
74LS766
74LS767
74LS768
74LS769
74LS770
74LS771
74LS772
74LS773
74LS774
74LS775
74LS776
74LS777
74LS778
74LS779
74LS780
74LS781
74LS782
74LS783
74LS784
74LS785
74LS786
74LS787
74LS788
74LS789
74LS790
74LS791
74LS792
74LS793
74LS794
74LS795
74LS796
74LS797
74LS798
74LS799
74LS800
74LS801
74LS802
74LS803
74LS804
74LS805
74LS806
74LS807
74LS808
74LS809
74LS810
74LS811
74LS812
74LS813
74LS814
74LS815
74LS816
74LS817
74LS818
74LS819
74LS820
74LS821
74LS822
74LS823
74LS824
74LS825
74LS826
74LS827
74LS828
74LS829
74LS830
74LS831
74LS832
74LS833
74LS834
74LS835
74LS836
74LS837
74LS838
74LS839
74LS840
74LS841
74LS842
74LS843
74LS844
74LS845
74LS846
74LS847
74LS848
74LS849
74LS850
74LS851
74LS852
74LS853
74LS854
74LS855
74LS856
74LS857
74LS858
74LS859
74LS860
74LS861
74LS862
74LS863
74LS864
74LS865
74LS866
74LS867
74LS868
74LS869
74LS870
74LS871
74LS872
74LS873
74LS874
74LS875
74LS876
74LS877
74LS878
74LS879
74LS880
74LS881
74LS882
74LS883
74LS884
74LS885
74LS886
74LS887
74LS888
74LS889
74LS890
74LS891
74LS892
74LS893
74LS894
74LS895
74LS896
74LS897
74LS898
74LS899
74LS900
74LS901
74LS902
74LS903
74LS904
74LS905
74LS906
74LS907
74LS908
74LS909
74LS910
74LS911
74LS912
74LS913
74LS914
74LS915
74LS916
74LS917
74LS918
74LS919
74LS920
74LS921
74LS922
74LS923
74LS924
74LS925
74LS926
74LS927
74LS928
74LS929
74LS930
74LS931
74LS932
74LS933
74LS934
74LS935
74LS936
74LS937
74LS938
74LS939
74LS940
74LS941
74LS942
74LS943
74LS944
74LS945
74LS946
74LS947
74LS948
74LS949
74LS950
74LS951
74LS952
74LS953
74LS954
74LS955
74LS956
74LS957
74LS958
74LS959
74LS960
74LS961
74LS962
74LS963
74LS964
74LS965
74LS966
74LS967
74LS968
74LS969
74LS970
74LS971
74LS972
74LS973
74LS974
74LS975
74LS976
74LS977
74LS978
74LS979
74LS980
74LS981
74LS982
74LS983
74LS984
74LS985
74LS986
74LS987
74LS988
74LS989
74LS990
74LS991
74LS992
74LS993
74LS994
74LS995
74LS996
74LS997
74LS998
74LS999
XEROX



Specifications

Print Speed:	Up to 40 characters per second with metalized printwheels.
Character Set:	88, 92, or 96 printable characters per printwheel. Switch-selectable program support for APL and all English language printwheels.
Printwheels:	88, 92, 96 character Xerox - Metal 96 character Diablo - Plastic
Character Spacing:	10-pitch = 10 characters/inch (3.94 ch/cm) 12-pitch = 12 characters/inch (4.72 ch/cm) 15-pitch = 15 characters/inch (5.91 ch/cm) Proportional Space (PS) - see HMI, page 241.
Column Spacing:	1/120 inch (.21mm) minimum.
Print Line:	13.2 inches (335.3mm) 132 columns 10-pitch 158 columns 12-pitch 198 columns 15-pitch
Print Buffer:	2688 bytes.
Paper Width:	16.53 inches (419.9mm) maximum - friction feed without Top Paper Out switch. 16.00 inches (406.4mm) maximum - friction feed with Top Paper Out switch. 15.25 inches (387.4mm) maximum - full width with optional forms tractor (14.75 inches/-374.7mm between holes). 3.25 inches (82.55mm) minimum with forms tractor (2.75 inches/69.85mm between holes).
Carriage Speed:	400 msec maximum for 13.1 inches (332.77mm) of motion.
Tabulation:	Left or right.
Line Spacing:	1/48 inch (.53mm) minimum.
Paper Feed:	Bidirectional, except with unidirectional forms tractor and unidirectional pin feed platen.

Paper Feed Speed:	4 inches (101.6mm) per second plus 40 msec (typical) settling delay time.
Paper Thickness:	.000 - .010 inch (.254mm) at low setting (1-3 part forms) .010 - .027 inch (.254 - .686mm) at high setting (4-6 part forms).
Sensors:	End of ribbon, paper out, and cover open.
Other Features:	Self test; host program control through escape sequences; data receive/transmit speed selection.
Power Requirements:	Strappable for operation from nominal 100, 120, 220, or 240 volt (+ 10%/-15%) AC inputs, 49-61 Hz. 350W maximum power consumption. Factory preset for 120 VAC. Check your printer's serial plate for proper input power.

Cabling Requirements

A standard RS-232-C interface cable is required for connection between the screen and the printer. This cable must be equipped with DB-25P connectors with the following pins connected:

PIN NO.	CCITT DESIG.	TELCO DESIG.	DESCRIPTION
1	101	AA	Protective Ground
2	103	BA	Transmitted Data
3	104	BB	Received Data
4	105	CA	Request To Send
6	107	CC	Data Set Ready *
7	102	AB	Signal Ground
20	108	CD	Data terminal Ready

* Pin 6 must be HI to receive or transmit data.

1.0 INTRODUCTION

The SA1403D Controller consists of a micrprocessor based controller with on-board data separator logic and is able to control a maximum of four drives. The drives can be any combination of Shugart SA1000 fixed disk drives, SA800 floppy disk drives, or SA850 floppy disk drives. The floppy disk track formats are compatible with IBM 1D/2D track formats. The SA1403D can be mounted on the SA1000 drive.

Commands are issued to the controller over a bidirectional bus connected to the host computer. The data separator/"serdes" logic serializes bytes and converts to FM/MFM data, and deserializes FM/MFM data into 8-bit bytes.

Due to the microprogrammed approach utilized in the controller, limited diagnostic capabilities are implemented. This methodology increases fault isolation efficiency and reduces system down time. Error detection and correction will tolerate media imperfections up to 4-bit burst errors.

NOTE: This device utilizes negative logic (i.e., 0V = logical 1)

1.1 SA1403D CONTROLLER FEATURES

OVERLAPPED SEEK	In multiple drive configurations the host can issue seeks to different drives without waiting for the first drive to complete its seek.
AUTOMATIC SEEK AND VERIFY	A seek command is implied in every data transfer command (READ, WRITE CHECK, etc.). If the heads are not positioned over the correct cylinder, a seek is initiated and a cylinder verification is performed after the seek completes.
FAULT DETECTION	Three classes of fault detection are provided for fault diagnosis: 1) Disk related faults. 2) Controller related faults. 3) Host command or I/O timing faults. Fault detection is available from the interface as a status message and is also visibly displayed on a row of status LED's on the controller PCB.
AUTOMATIC HEAD AND CYLINDER SWITCHING	If during a multi-block data transfer the end of a track is reached, the controller automatically switches to the next track. If the end of a cylinder is reached, the controller issues a seek and resumes the transfer.
DATA ERROR SENSING AND CORRECTION	If a data error is detected during a disk data transfer, the controller indicates whether or not it is correctable. If correctable, it can be automatically corrected. (This applies to the SA1000 only. CRC error detection is used on floppy disc drives.)
LOGICAL TO PHYSICAL DRIVE CORRELATION	Logical Unit Number (LUN's) are independent of physical port numbers. All accesses specify LUN's.
ON BOARD SECTOR BUFFER	A sector buffer is provided on the controller to eliminate the possibility of data overruns during a data transfer.
EFFICIENT HOST INTERFACE PROTOCOL	A bidirectional bus between the controller and host provides a simple, yet efficient communication path. In addition, a high level command set permits effective command initiation.
SECTOR INTERLEAVE	Sector interleaving is programmable with up to a 16 way interleave.
ODD PARITY	The 8 data bits on the interface bus can have odd parity. Depending on user preference, parity can be disabled.
FIXED SECTOR SIZE	The sector size is fixed at 256 bytes of data for the SA1000.

NUMBER OF DRIVES — The controller will connect to a maximum of four (4) drives. The drives can be any combination of SA1000's and/or SA850's and/or SA800's

1.1.1 OPTIONAL FEATURES

MICRO DIAGNOSTICS — A set of diagnostic PROM's are available to allow stand alone diagnostic testing of both drive and controller. Reference Appendix A.

1.1.2 SYSTEM CONFIGURATION

The controller and data separator comprise a single PCB that can be mounted onto the SA1000 drive. A maximum of four (4) drives may be connected as shown in Figure 2.

1.2 TRACK FORMATS AND CAPACITY

A) 32 sectors of 256 bytes per sector (SA1000only).
C) 26 sectors of 256 bytes per sector (Floppy only).
D) 26 sectors of 128 bytes per sector (Floppy only.)

IBM 1D/2D TRACK FORMAT — Track format for Floppy Disk drives can be selected under program control in real time. The track formats are:

1) Single density, single sided
2) Single density, double sided
3) Double density, single sided
4) Double density, double sided

	26 SECTOR	32 SECTOR
SA800	2001	N/A
SA850	4003	N/A
SA1002	N/A	16383
SA1004	N/A	32767

TABLE I.

Format/Capacity Relationship
Maximum Logical Sector Address Shown

2.0 SPECIFICATION SUMMARY

2.1 ENVIRONMENTAL LIMITS

	Operating	Storage
Temperature F/C	32°/0° to 131°/55°	-40°/-40° to 167°/75°
Max. Wet Bulb	85°F	non condensing
Relative Humidity	10% to 95%	10% to 95%
Altitude	Sea level to 10,000 ft	Sea level to 15,000 ft

2.2 POWER REQUIREMENTS

Three power supply voltages are required for the SA1400 series controllers. The maximum current requirements are as follows:

+5VDC ± 5% at 4.6 Amps
-5VDC ± 5% at 0.5 Amps
+24VDC ± 10% at 0.1 Amps

Power is applied to the SA1400 series controller via J10 which is a 6 pin AMP Mate-N-Lok connector (P/N 1-380999-0) mounted on the component side of the board. The recommended mating connector, P10, is an AMP P/N 1-480270-0 utilizing AMP pins P/N 60619-1. The J10 pins are labeled on the connector. Figure 1 shows the pin assignments.

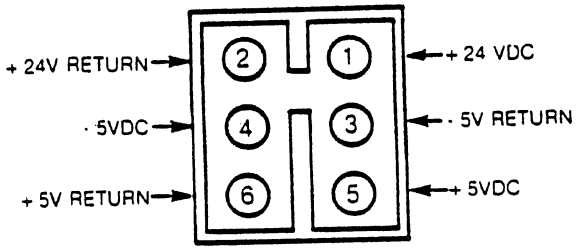


FIGURE 1. J10 DC POWER CONNECTOR

2.3 PHYSICAL PARAMETERS

Length:	13.7 inches (34.8cm) ± .030" (.076 cm)
Width:	8.25 inches (21cm) ± .010" (.025 cm)
Height:	0.5 inches (1.3cm) ± .030" (.076 cm)
Weight:	1.12 lbs (0.5Kg) ± .010 lbs (0.25 g)

3.0 SA1403D DISK DRIVE INTERFACE

Shugart SA1000 and SA800/850 disk drives are interfaced to the controller via J1, J2, J3, J4 and J5. Refer to Figure 2 for connection block diagram.

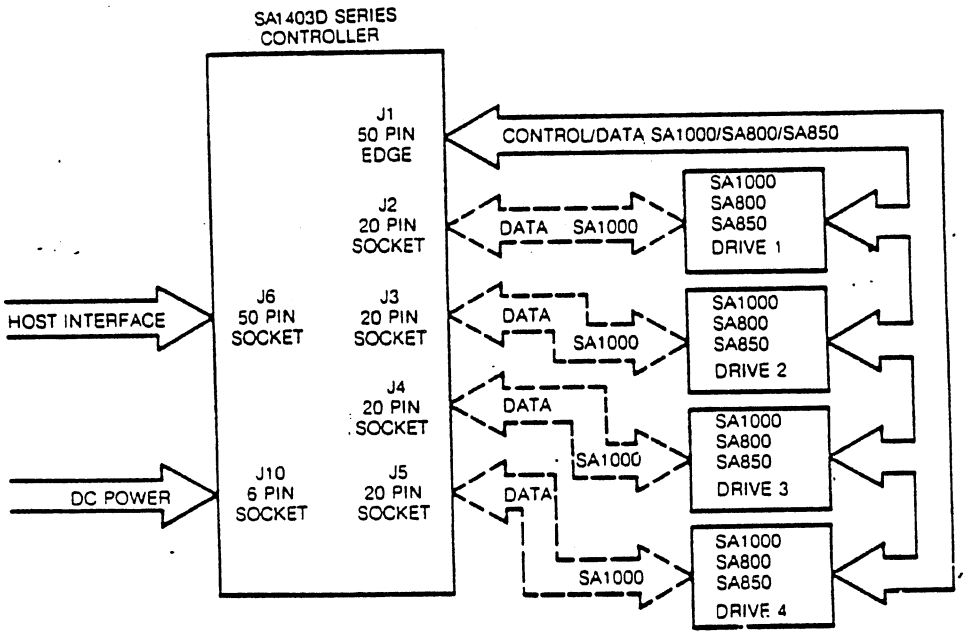


FIGURE 2. SA1403D INTERCONNECT DIAGRAM

NOTE: The last physical device on the control cable (drive to be terminated) must be an SA1000.

J1 is a 50 pin edge type connector which connects all drives in a daisy chain configuration. This connector carries control and data information for the floppy disk drives and control information only for the SA1000 disk drive. Maximum cable length should not exceed 20 feet (6 meters).

The recommended mating connector for J1 is a 3M Scotchflex ribbon connector P/N 3415-0001.

J2 through J5 are 20 pin socket type connectors used to radially connect the SA1000 data lines to the controller. Maximum cable length should not exceed 20 feet (6 meters).

The recommended mating connector for J2 through J5 is a 3M Scotchflex P/N 3421-3000. Figure 3 shows the pinouts for J1 and J2 through J5.

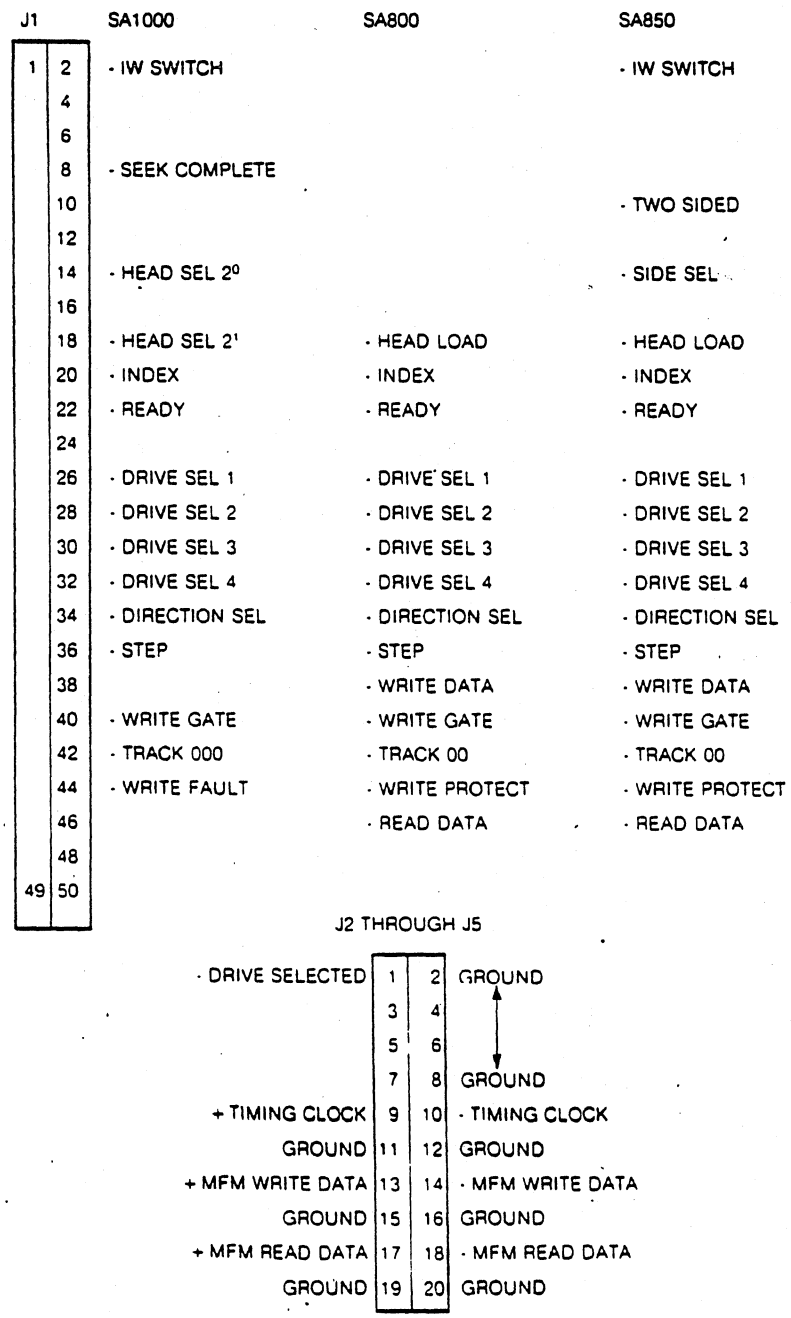


FIGURE 3. SA1403D DRIVE CONNECTOR PINOUTS

3.1 CABLE TERMINATION

The last physical drive at the end of J1 (50 pin) cable must be properly terminated. Termination networks are provided on the drives (refer to SA1000, SA800 or SA850 OEM manuals for location of termination networks). Termination networks must be removed from all drives except the last drive on the cable to avoid multiple termination.

NOTE: If a combination of fixed and floppy drive are used, the last drive at the end of the control cable must be an SA1000.

4.0 HOST CPU INTERFACE

The SA1400 series controller interface is a general purpose 8 bit parallel DMA.

The Host CPU is interfaced to the controller via connector J6, J6 is a 50 pin socket type connector. The recommended mating connector for J6 is a 3M Scotchflex ribbon connector P/N 3425-3000. The J6 interface cable should not exceed 20 feet (6 meters).

4.1 HOST CPU ELECTRICAL INTERFACE

All Host CPU interface signals are negative true. The signals are "Asserted" at 0 VDC to 0.4 VDC. The signals are "Deasserted" or inactive at 2.5 VDC to 5.25 VDC.

4.1.1 HOST CPU INTERFACE TERMINATION

All Host CPU interface timing lines are terminated with a 220/330 ohm network. The Host CPU adapter should be terminated in a similar fashion (see Figure 4).

The devices driving the controller inputs should be open collector devices capable of sinking at least 48 milliamps to a voltage level of less than 0.5 VDC (7438 or equivalent).

The devices receiving the controller outputs should be of the SCHMITT trigger type to improve the noise margin (74LS240, 74LS14, or equivalent). The Host adaptor should not load the bus with more than 1 standard TTL input load per line.

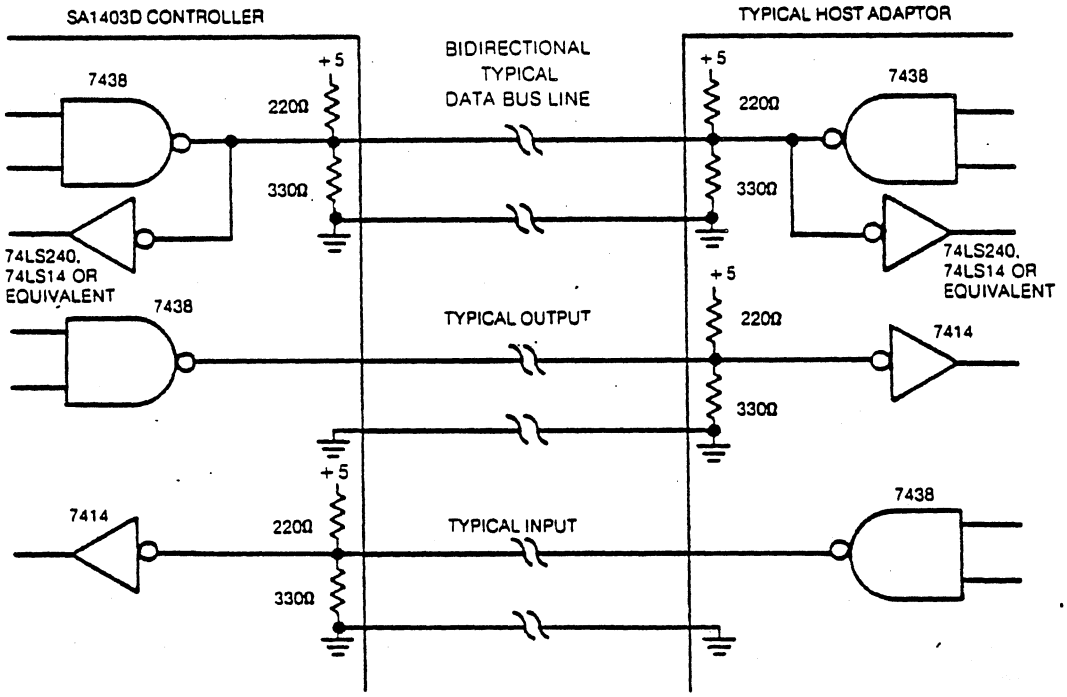
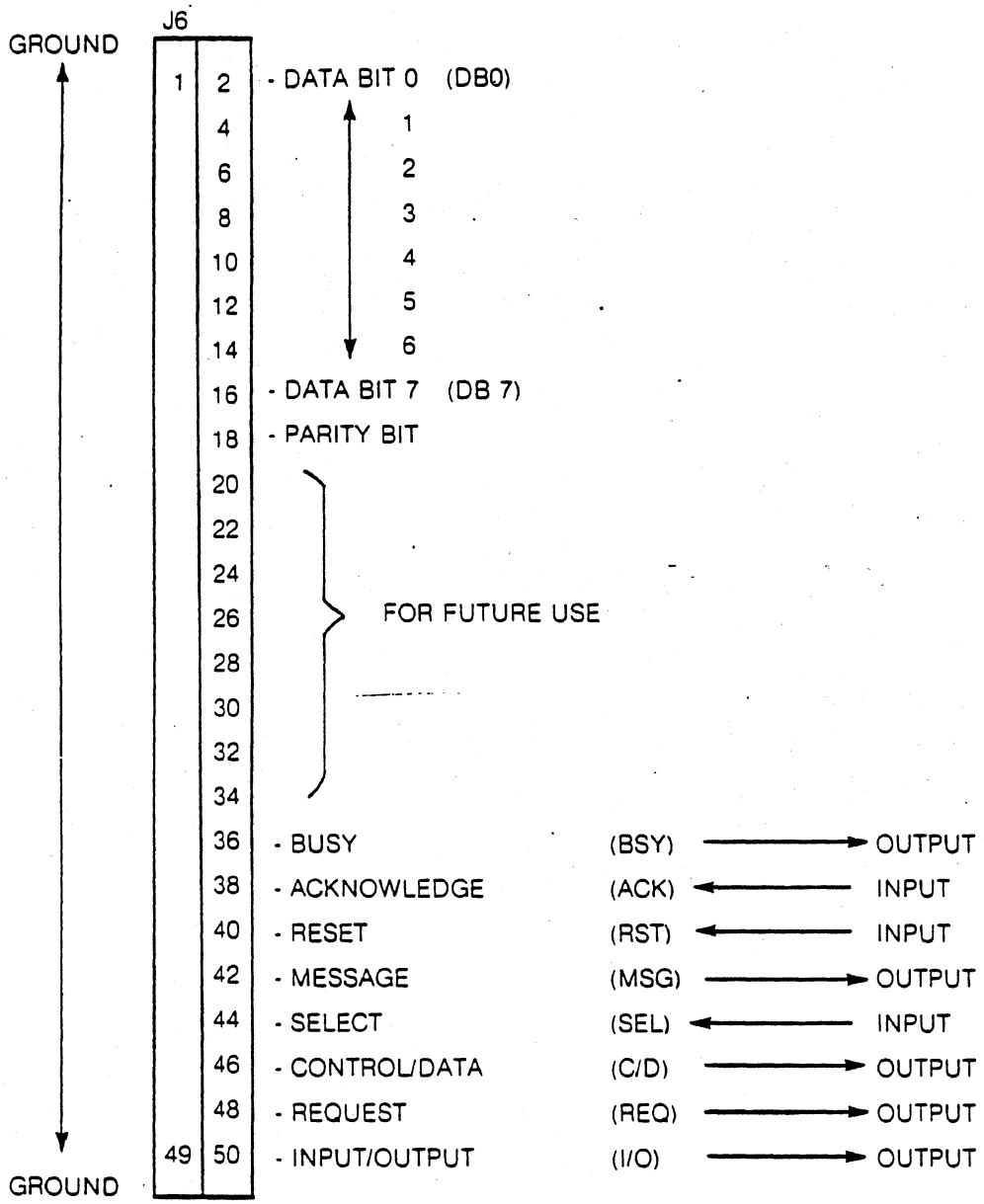


FIGURE 4. HOST ADAPTOR BUS TERMINATION

4.1.2 HOST CPU SIGNAL INTERFACE

The Host CPU signals are interfaced via J6. See figure 5 for J6 pinouts.



NOTE: ALL SIGNALS ARE TTL NEGATIVE TRUE

FIGURE 5. J6 HOST INTERFACE CONNECTOR PINOUT

4.2 SA1403D HOST BUS

4.2.1 THEORY OF OPERATIONS

Disk commands are issued to the SA1403D via the host bus following a defined protocol. The host initiates a command sequence by selecting the controller on the bus. If the controller is not busy, it requests command bytes from the host for task execution. (Command structure is described in 4.5). Depending on the type of command, the controller will request either 6 or 10 bytes. Upon reception of the last command byte, the controller begins execution of the command.

For the data transfer commands, a check is performed on the disk address and status flagged if it exceeds the drive limits. The data is stored in a sector buffer before transfer to the host or disk drive. This buffer eliminates any possibility of data overruns between the host and the disk.

Upon completion of the command, the controller will send completion status to the host. Further delineation of the completion status may be requested by issuing the appropriate sense commands.

Odd parity is generated by the SA1403D for all information that it puts on the I/O bus. If enabled, the SA1403D checks all information that it receives for odd parity.

4.3 SIGNAL DEFINITION

Unidirectional Signals Driven By Controller

I/O — **Input/Output.** When asserted, the data on the bus is driven by the controller; when deasserted, the data on the bus is driven by the host adapter. The host adapter will use this line to enable its drivers onto the data bus.

C/D — **Control/Data.** When asserted the data transmitted across the bus will be the command or status bytes; when deasserted the data will be the disk data bytes.

BUSY — This bit is asserted as a response to the SEL line from the host adapter and to indicate that the host bus is currently in use.

MSG — **Message.** When asserted indicates that the command is completed and status has been transferred. The assertion of this bit is always followed with the assertion of I/O, and the assertion of REQ, to cause a message byte transfer.

REQ — **Request.** This bit operates in conjunction with I/O, C/D, & MSG. When asserted and I/O is asserted, REQ will mean that the data on the host bus is driven by the controller. When asserted and I/O is deasserted, REQ will mean that the data is driven by the host adaptor (H/A).

I/O	C/D	MSG	Meaning
d	a	d	Get command from H/A
d	d	d	Get data from H/A
a	d	d	Send data to H/A
a	a	d	Send status byte to H/A
a	a	a	Command done to H/A

TABLE 2.

a = asserted, d = deasserted, H/A = host adaptor

4.4 UNIDIRECTIONAL SIGNALS DRIVEN BY HOST ADAPTOR

ACK — **Acknowledge.** This bit is asserted as a response to REQ from the controller. The timing requirements on this signal with respect to the data is described in REQuest section. ACK must be returned for each REQ assertion

RST **Reset.** Assertion by the Host causes the controller to cease all operations and return to an idle condition. This signal is normally used during a power up sequence. A reset during a write operation would cause incorrect data to be written on the selected disk. The controller may take a maximum of 2 seconds to respond to the select sequence following deassertion of the RESET line.

SEL **Select.** When asserted indicates the beginning of the command transaction. The H/A asserts SEL to gain the attention of the controller. Data bit zero on the host bus must also be asserted during SEL time to select the controller address. The controller will return BUSY within approximately 1μs.

4.4.1 DATA BUS BITS 0-7 (DB)

These bidirectional data lines are used to transfer 8 bit parallel data to/from the Host adaptor. Bit 7 is most signifant bit. NOTE: All I/F lines utilize negative logic.

4.4.2 PARITY BIT

This bit is asserted to maintain odd parity on all data and status information transfered to the Host. If enabled, the controller will test for odd parity on all command and data information transfered to the controller (see section 9.1).

4.5 HOST INTERFACE PROTOCOL

There are 4 sequences required to initiate and complete a command to the SA1403D series controller:

1) Controller Selection Sequence

2) Command Transfer Squence

3) Data Transfer Sequence

4) Status and Message Transfer Sequence

4.5.1 CONTROLLER SELECTION SEQUENCE

In order to gain the attention of the controller it is necessary to perform a selection sequence. Refer also to Figure 6.

The Host must first test BSY to determine if the controller is available. If BSY is deasserted, the Host will assert data bit 0 (controller ID) and then assert SEL. The controller will then respond by asserting BSY. At this point the Host must deassert SEL and data bit 0. I/O will remain deasserted throughout the selection sequence.

4.5.2 COMMAND TRANSFER SEQUENCE

Following the selection sequence the controller will assert REQ (see Figure 6). The Host will then place the first byte of the command descriptor block (see section 5.0) on the data bus. The Host will then assert ACK (if ACK is not asserted within 256 microseconds after the assertion of REQ, the controller will abort the command transfer sequence and attempt to transfer a status byte). The controller will respond by reading the byte on the data bus and then deasserting REQ. The Host then must deassert ACK to begin the next REQ/ACK handshake. This handshake must be completed to assure that all command and data bytes are transferred.

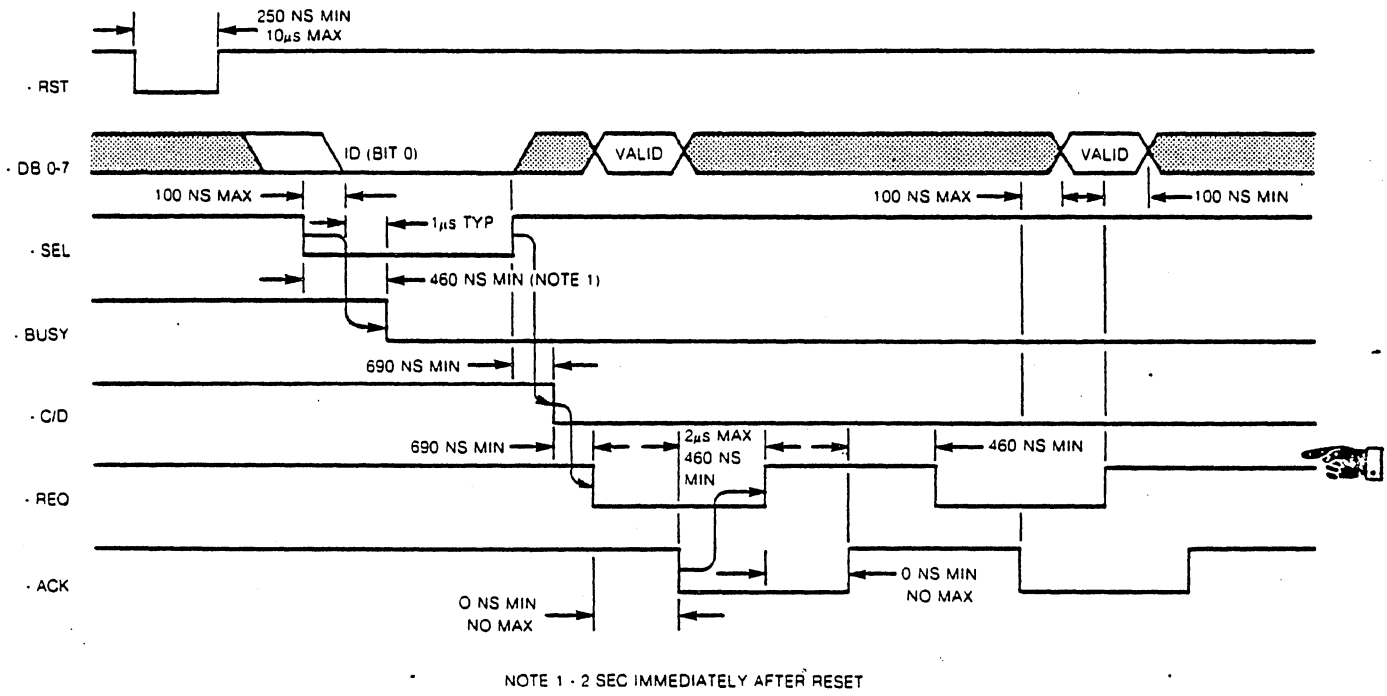


FIGURE 6. SELECT SEQUENCE TIMING

4.5.3 DATA TRANSFER SEQUENCE

Following the command transfer sequence, the controller will respond on one of four ways:

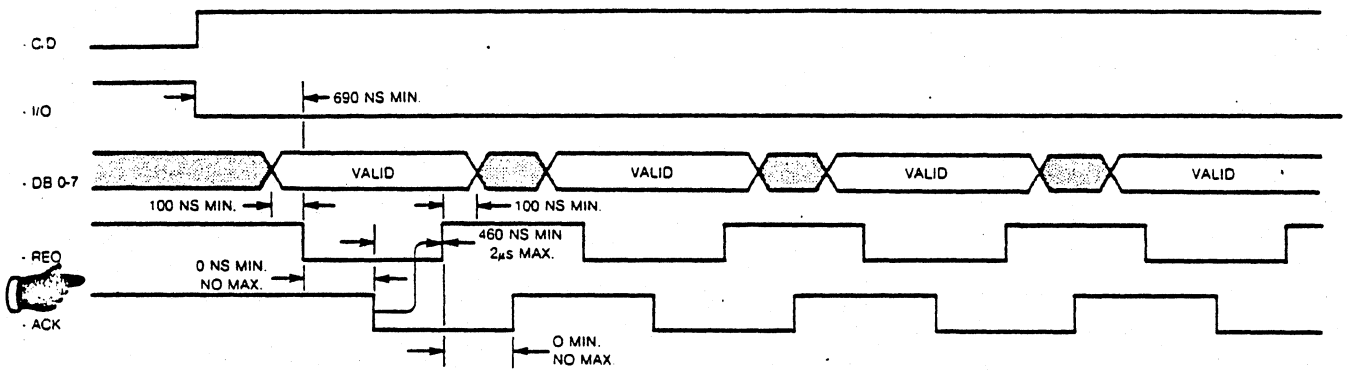
1) Begin seeking the drive.

2) Begin accepting write data from the Host.

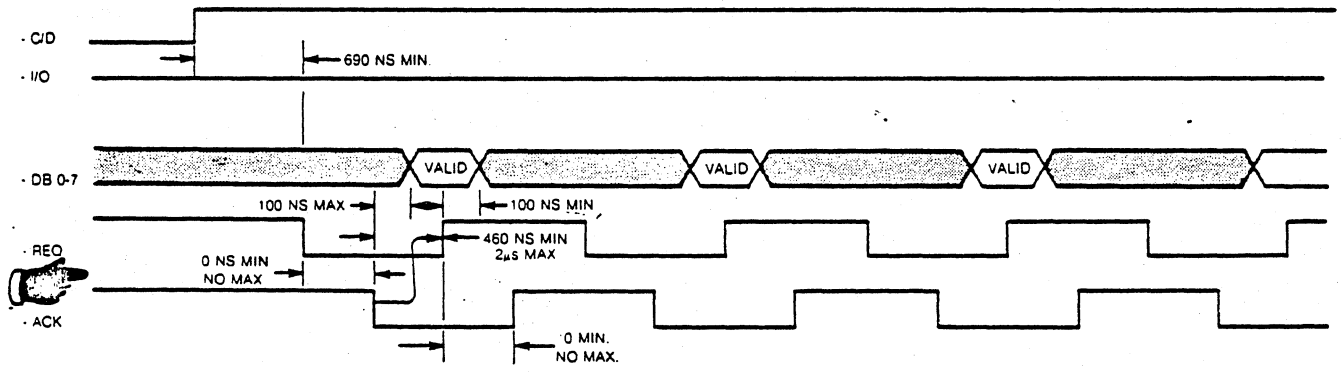
3) Begin transferring read data to the Host.

4) Return status to the Host.

If the command sent to the controller involves a data transfer (see Figure 7), the controller will deassert the C/D line to indicate a data transfer. If the data transfer is from the Host to the controller (write data) the I/O line will be deasserted. If the data transfer is from the controller to the Host (read data) the I/O line will be asserted. The controller will then set the REQ line to request a byte transfer. The Host will respond by transferring a byte across the data bus and then asserting ACK (if ACK is not asserted within 256 microseconds after the assertion of REQ, the controller will abort the data transfer sequence and attempt to transfer a status byte - see section 4.5.4). The Host will then deassert ACK and wait for the next assertion of REQ. This handshake continues until all data has been transferred



READ DATA TRANSFER SEQUENCE (CONTROLLER TO HOST)



WRITE DATA TRANSFER SEQUENCE (HOST TO CONTROLLER)

FIGURE 7. DATA TRANSFER SEQUENCE TIMING

4.5.4 STATUS AND MESSAGE TRANSFER SEQUENCE

Following a command transfer or data transfer, the controller will initiate a status byte and completion message transfer.

When a status byte transfer is required, the controller will assert C/D and I/O (see Figure 8). The controller will then assert REQ. The Host must then read the status byte on the data bus and then assert ACK (if ACK is not asserted within 256 microseconds after the assertion of REQ, REQ will be deasserted. REQ will then be asserted again).The controller will then deassert REQ. The host will then deassert ACK.

Following the status byte transfer, a completion message byte of all zero's will be transfered to indicate operation complete. The controller will assert the MSG line (along with I/O and C/D) and then assert REQ. The Host may read the completion message byte on the data bus and assert ACK (if ACK is not asserted within 256 microseconds, the controller will deassert the MSG line and attempt to transfer a status byte). The controller will respond by deasserting REQ. The Host will then deassert ACK. At this point BSY and all other controller I/O lines will be deasserted and the controller will return to an IDLE LOOP awaiting the next selection sequence.

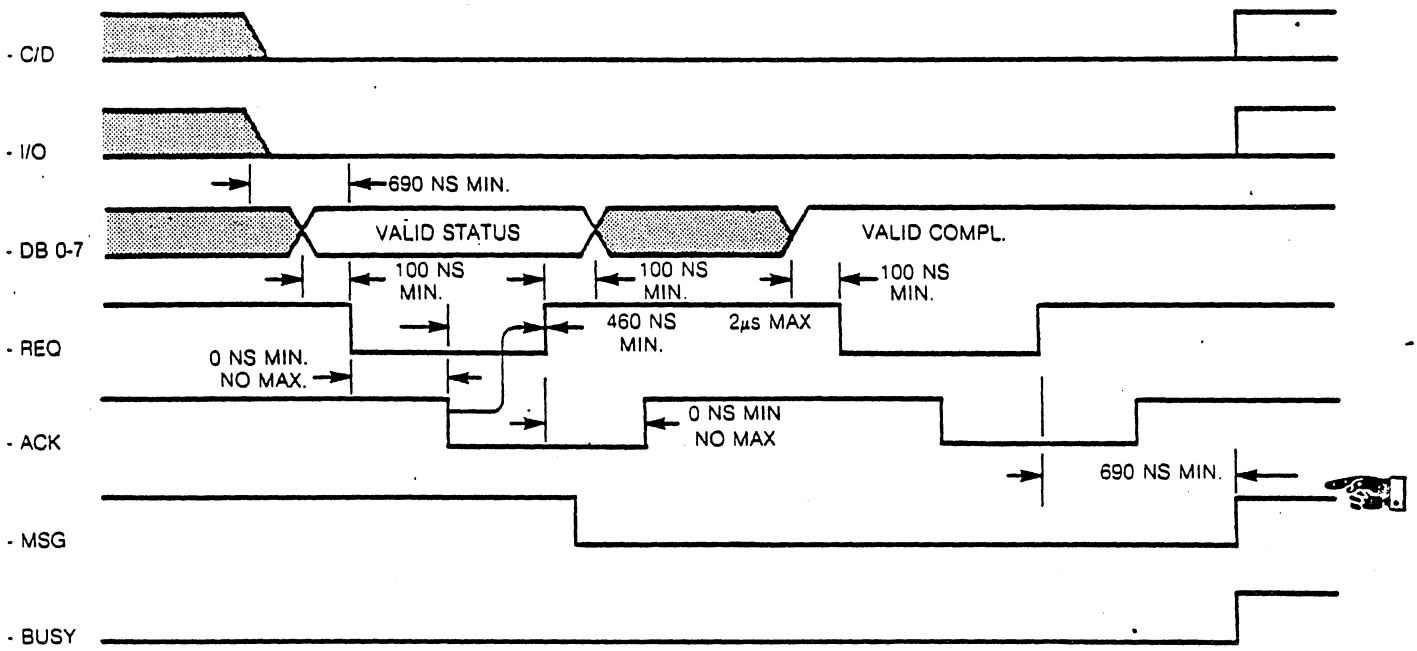


FIGURE 8. STATUS AND COMPLETION SEQUENCE TIMING

5.0 CONTROLLER COMMAND DESCRIPTOR BLOCK

Following the controller selection sequence the controller will request a command descriptor block (CDB) which, depending on the class of command, may be either 6 or 10 bytes in length. The first byte of the CDB contains the command class and the command operation code. The remaining bytes specify the drive logical unit number (LUN), logical sector address, number of sectors to be transfered or a destination device (Copy Command), and a control field byte.

Commands are categorized into four classes as indicated:

Class 0 - Utility, Data Transfer and Status Commands
Class 1 - Disk Copy Commands
Class 2-5,7 - Reserved
Class 6 - Floppy Disk Track Format Selection

The command descriptor blocks in Command Class 0 and 6 are 6 bytes long, and those in Class 1 are 10 bytes long.

The controller will check all incoming command descriptor blocks for validity and will also check (if enabled) all CDB's and data for odd parity (see section 9.1). A parity error will cause an immediate halt of the command or data transfer. This will not cause incorrect data to be written because the write does not occur until the sector buffer has been filled. An error in the command structure will cause a status byte transfer to occur upon completion of the CDB transfer.

5.1 COMMAND DESCRIPTION (CLASS 0)

WARNING!
Commands READ and WRITE require that the floppy diskette used be formatted. If unformatted, the controller will appear to "hang" - i.e., continue waiting for a data address mark. (Reset to clear this condition if it should occur).

Opcode (Hex)	Description
00	Test drive ready - Selects the drive and verifies drive ready. The ready condition is indicated by the status byte. A not-ready drive will cause bit 1 of the status byte to be set.
01	Recalibrate. Positions the R/W of selected drive arm to Track 00, clears error status in the drive.
02	Request Syndrome - returns two bytes of error offset and syndrom to the Host System for Host error correction capability (see Table 3). The first byte is offset in the data field of the error location. The most significant 3 bits of the second byte point to the beginning of the error location. The least significant 4 bits of the second byte are the syndrome which is a data correction mark to be exclusive or'ed with the faulty data. This command is only valid of the automatic data correction has been disabled.

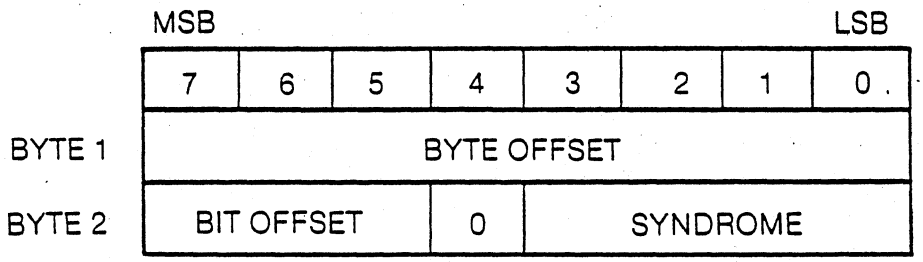


TABLE 3

03	Request Sense. This command must be issued immediately after an error. It returns 4 bytes of drive and controller sense for the specified LUN. (See copy block for exception)
04	Format Drive. Formats all blocks with ID field set according to interleave code. The data field contains E5 Hex.
05	Spare.
06	Format Track. *Formats the specified track with bad block flag cleared in all blocks of that track. Writes E5 Hex in the data fields.
07	Format Bad Track *(bad block flag). Formats the specified track with bad block flag set in the ID fields (bit 7 of the Head Address byte set). Writes E5 Hex in the data fields.
08	Read. Reads the specified number of blocks starting from initial block address given in the CDB. (See Warning above!)
09	Reserved.
0A	Write. Writes the specified number of blocks starting from initial block address given in the CDB. (See Warning above!)
0B	Seek. Initiates seek to specified block and immediately returns completion status before the seek is complete for those drives capable of overlap seek.

The track is addressed via the logical sector address, which may be any address within the desired track.

5.1.2 COMMAND DESCRIPTION (CLASS 1)

Opcode (Hex)	Description
00	Copy Blocks. Copies the specified number of blocks from Source LUN starting at the specified Logical address to Destination LUN starting at the specified Logical address. The number of sectors transferred may be from 1 to 256. The completion status byte will indicate the source LUN. If an error occurs, a Request Sense command is issued to the source LUN. The sense will indicate the type of error for the appropriate LUN. Note the data in the blocks will be truncated or appended with undefined data if the Source and Destination block sizes are not the same (e.g. Source block size - 128 bytes/sector, and Destination block size - 256 bytes/sector).

5.1.3 COMMAND DESCRIPTION (CLASS 6)

Opcode (Hex)	Description
00	Define Floppy Disk Track Format. The Track format code in byte 6 of the CDB defines the track format for the LUN. The Track Format Codes are as follows:

Track Format Code (Hex)	Description
00	Single Density, Single Sided. All tracks - FM recording, 128 bytes/sector, 26 sectors/track.
01	Single Density, Double Sided. All tracks - FM recording, 128 bytes/sector, 26 sectors/track.
02	Double Density, Single Sided. Side 0, Cylinder 0 - FM Recording, 128 bytes/sector, 26 sectors/track. All other tracks - MFM recording, 256 bytes/sector, 26 sectors/track.
03	Double Density, Double Sided. Side 0, Cylinder 0 - FM recording, 128 bytes/sector, 26 sectors/track. All other track - MFM recording, 256 bytes/sector, 26 sectors/track.

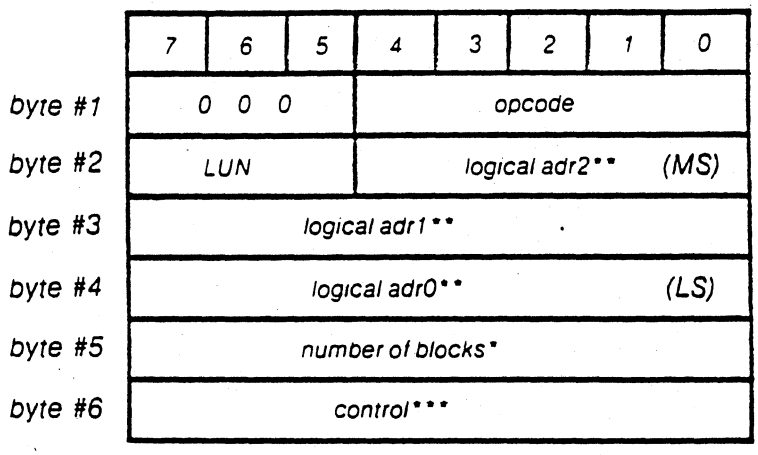
NOTE: If track format information for floppy is not specified after each reset or power-on, the default mode will be taken from the drive type selection dipswitch as follows:

Switch Setting	Mode
OFF-ON	Single density, single sided (same as track format code 00)
OFF-OFF	Single density, double sided (same as track format code 01)

Refer to Section 9.2 for switch setup instructions.

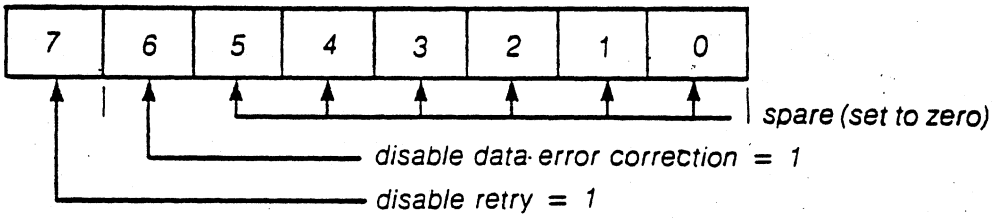
5.2 COMMAND FORMAT

5.2.1 CLASS 0 COMMANDS



	7	6	5	4	3	2	1	0
byte #1	0	0	0	opcode				
byte #2	LUN			logical adr2** (MS)				
byte #3	logical adr1**							
byte #4	logical adr0** (LS)							
byte #5	number of blocks*							
byte #6	control***							

*Interleave factor for Format, Check Track Format commands.
**Refer to Section 5.5 Logical Address.
***The control field is defined as follows:



CONTROL FIELD

5.2.2 CLASS 1 COMMANDS

	7	6	5	4	3	2	1	0
byte #1	0	0	1	opcode				
byte #2	0	LUN/s		logical adr2/s* (MS)				
byte #3	logical adr1/s*							
byte #4	logical adr0/s* (LS)							
byte #5	number of blocks							
byte #6	0	LUN/d		logical adr2/d* (MS)				
byte #7	logical adr1/d*							
byte #8	logical adr0/d* (LS)							
byte #9	spare							
byte #10	control (section 5.2.1)							

where 's' indicates the source device and 'd' indicates the destination device.
*Refer to Section 5.5 Logical Address

5.2.3 CLASS 6 COMMANDS

	7	6	5	4	3	2	1	0
byte #1	1	1	0	opcode				
byte #2	LUN			N/A				
byte #3	N/A							
byte #4	N/A							
byte #5	N/A							
byte #6	Track Format Code							

NOTE: See Class 6 Command Description for more information and default modes for floppy drives.

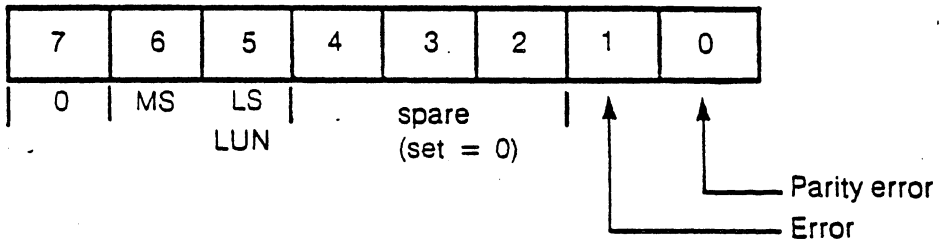
5.3 STATUS FORMAT

5.3.1 Completion Status Byte Format

At the normal termination of a command or following a fatal error, the controller will cause a status byte to be transferred from the controller to the Host. Bit 0, the least significant bit of the status byte, will be set equal to 1 if the controller detects a parity error during a command or data transfer to the controller. Bit 1 will be set = 1 if the controller detects an error condition. Bits 5 and 6 represent the LUN of the device where the error occured. If no error occurs, bit 0 - 4 will be set equal to 0.

Following the transfer of the status byte, the MSG line will be asserted to indicate a completion message. At this time the message consists of a single byte transfer with all bits set = 0.

Prior to an error condition the controller, unless diabled (see section 5.2.1 Control Field), will retry 3 times before posting the error.

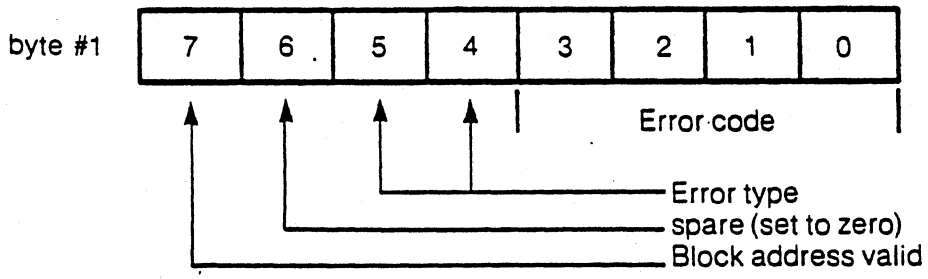


Bit 0 Parity error during transfer from host to controller.
Bit 1 Error occured during command execution.
Bit 2-4 Spare (set to zero).
Bit 5-7 Logical unit number of the drive.

5.3.2 DRIVE AND CONTROLLER SENSE BLOCK

Following an error indication from the status byte, the Host may perform a REQUEST SENSE command to obtain more detailed information about the error.

The REQUEST SENSE command will transfer a block of 4 bytes to the Host system.



	7	6	5	4	3	2	1	0
byte #2	LUN			logical adr2*				
byte #3	logical adr1*							
byte #4	logical adr0*							

*Refer to Section 5.5 Logical Address

5.4 ERROR CODES

5.4.1 TYPE 0 (DRIVE) ERROR CODES

0	No error
1	No Index signal
2	No Seek Complete
3	Write Fault (SA1000 only)
4	Drive not ready
5	Drive not selected (SA1000 only)
6	No Track 00

5.4.2 TYPE 1 (CONTROLLER) ERROR CODES

0	ID read error. ECC or CRC (floppy) error in the ID field (uncorrectable).
1	Uncorrectable data error during a read.
2	ID Address Mark not found (possibly unformated disk).
3	Data Address Mark not found.
4	Record not found. Found correct cylinder and head but not sector.
5	Seek error. R/W head positioned on a wrong cylinder and/or selected a wrong head.
6	DMA Data time out error. No Host acknowledge within 256μs.
7	Write protected. (SA800/850 only)
8	Correctable data field error. ECC error (automatic correction if not disabled).
9	Bad track found
A	Format Error. The controller detected that during the Check Track command, the format on the drive was not as expected.

5.4.3 TYPE 2 (COMMAND) ERROR CODES

0 Invalid Command received from the host.
1 Illegal logical sector address. Address is beyond the maximum address for the type of drive.
2 Illegal function for the specified drive.

5.5.4 TYPE 3 (MISC) ERROR CODES

0 RAM error. Data error detected during Sector buffer RAM diagnostic.

5.5 LOGICAL ADDRESS

The logical address is computed as follows:

Logical adr = (CYADR * HDCYL + HDADR) * SETRK + (SEADR)

Where: CYADR = cylinder address
HDADR = head address
SEADR = sector address
HDCYL = number of heads per cylinder
SETRK = number of sectors per track

Bit 0 of Logical adr 0 = the least significant bit.
Bit 4 of Logical adr 2 = the most significant bit.

Note: All addresses begin with 00.

6.0 SECTOR INTERLEAVE CODES

In order to tailor host system data transfer speed to the disk rotational speed, sector interleaving is offered. Sixteen interleave codes are offered numbered 1 to 16. Not all interleave codes will result in optimum sector interleave, therefore the interleave should be chosen carefully. In order to maintain IBM floppy disk compatibility in interleave code of 1 should be used. This will result in a non-interleave condition.

6.1 SELECTING THE RIGID DISK INTERLEAVE CODE

The interleave code given during the format command is used to calculate the logical sector number for the rigid disk as follows: Logical Sector = (Physical Sector × Interleave code) (mod 32). Note: when the logical sector number exceeds 31 the next logical sector is the lowest available physical sector. This does not always create a true modulo function.

Two examples of interleave codes are shown:

Interleave code of 2:

Physical:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Logical:	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
Physical:	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Logical:	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31

Interleave code of 11:

Physical:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Logical:	0	11	22	1	12	23	2	13	24	3	14	25	4	15	26	5
Physcial:	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Logical:	16	27	6	17	28	7	18	29	8	19	30	9	20	31	10	21

Code	Number of Disk Revolutions Required to Read One Track	Time available to Transfer one Byte of Data (including controller time)	Minimum Number of Idle Sectors Between Reads
11	3	4.7μs	2
8	4	7.0μs	3
6	6	9.4μs	4
5	7	11.7μs	5
4	8	16.4μs	7
3	11	23.4μs	10
2	16	35.1μs	15
1	32	72.5μs	31

• (for SA1400 series controllers operating with SA1000 series drives - double density, 32 sectors, 256 bytes/sector.)
Note: Other codes will work, but require more revolutions of the disk to read all sectors of one track.

TABLE 3. INTERLEAVE CODE SELECTION CHART*

7.0 DIAGNOSTIC PHILSOPHY

7.1 BOARD RESIDENT MICRODIAGNOSTIC

Fault Isolation Microdiagnostic (Optional)
The controller can be further checked out off-line by initiating explicit microdiagnostic routines via optional firmware diagnostic sets. The routines are initiated by a set of control switches. Errors will be dislayed in a set of LED's. Each microdiagnostic checks the funtionality of a particular section of the controller and is able to isolate failures in the following major categories:

ALU
Registers
Sector Buffer
ECC Logics

Fault-isolation techniques can be concentrated on the failing section.

8.0 STATUS LED ERROR INTERPRETATION

Drive/controller error conditions are displayed on the 8 LED display lights provided near the J10 DC power connector (see Figures 11). The following list of hexadecimal numbered error codes describe error meanings. Note that these error codes do not necessarily match the request sense block error codes. LED number 7 is the MSB.

01	No Index Detected
02	No Track Zero Detected
03	Illegal Logical Sector Address - beyond maximum sectors available for type of drive
04	Drive Not Selected (SA1000 only)
05	No Seek Complete Detected
06	ID Address Mark Not found (unformatted)
07	Data Address Mark Not found
08	Seek Error - R/W head not positioned on correct track
09	Record Not found - found correct cylinder and head but not sector
0A	ID ECC or CRC error (uncorrectable)
0B	DMA Timeout Error - no Host acknowledge within 256μsec after request.
0C	Invalid Command Received from Host
0D	Incorrect Data Address Mark
0E	Incorrect ID Address Mark
0F	Incorrect Cylinder Address
10	Incorrect Sector Address
11	Incorrect Head Address
12	Uncorrectable Data Field ECC or CRC error
13	Correctable Data Field ECC error
14	Drive Not Ready
15	Write Fault (SA1000 and SA4000/4100 only)
16	Spare
17	Write Protected (SA800/850 only)
18	RAM Diagnostic Error
19-1F	Spare
20	Parity Error
21	Bad Sector found - a sector within a track that has been flagged bad has been found.
22	Invalid function for this drive type.

9.0 CONTROLLER OPTION SELECTION

9.1 PARITY SELECT JUMPERS

Odd parity may be used by the Host system for data integrity verification. The controller will always output odd parity to the Host system.

Odd parity checking by the controller may be allowed or inhibited by moving a 3 position jumper plug at W2 located near the J6 Host connector (see Figure 11). With jumper at position A + B the controller will test for odd parity on all data input to the controller. With jumper at positon B + C the controller will not check for parity (normally shipped in A + B).

9.2 DRIVE TYPE SELECTION DIPSWITCH

The dipswitch settings for various types of drives for the SA1403D are shown below:

Prom Set AS30 — I, II, III, IV CUSTOMER FIRMWARE: (DIP SWITCH set-up procedure)

Location: 2H

Switch Bits	8 7	6 5	4 3	2 1	
Field Definition	LUN 0 Drive Type	LUN 1 Drive Type	LUN 2 Drive Type	LUN 3 Drive Type	OFF ON

Drive Type	Switch Setting		Description	
	Even	Odd		
0	on	on	SA1002	2 heads, 256 cylinders
1	on	off	SA1004	4 heads, 256 cylinders
2	off	on	SA800	1 head, 77 cylinders
3	off	off	SA850	2 heads, 77 cylinders

EXAMPLE:

LOCATION: 23

8 7	6 5	4 3	2 1	
LUN 0 Drive Type on on	LUN 1 Drive Type off on	LUN 2 Drive Type on off	LUN 3 Drive Type off off	OFF ON

Drive 0 is set up for SA1002
Drive 1 is set up for SA800
Drive 2 is set up for SA1004
Drive 3 is set up for SA850

10.0 TRACK FORMAT DESCRIPTION

10.1 26 SECTOR FORMAT

The 26 sector format is an IBM compatible format which employes FM single density encoding on all tracks of the single density format (IBM 3740 compatible) and on track 0, side 0 of the double density format. This format yields 26 sectors of 128 bytes per sector.

The remainder of the tracks on the double density formats are encoded with MFM double density which yields 26 sectors of 256 bytes per sector (IBM system 34 compatible). Figure 9 shows the two type of encoding utilized.

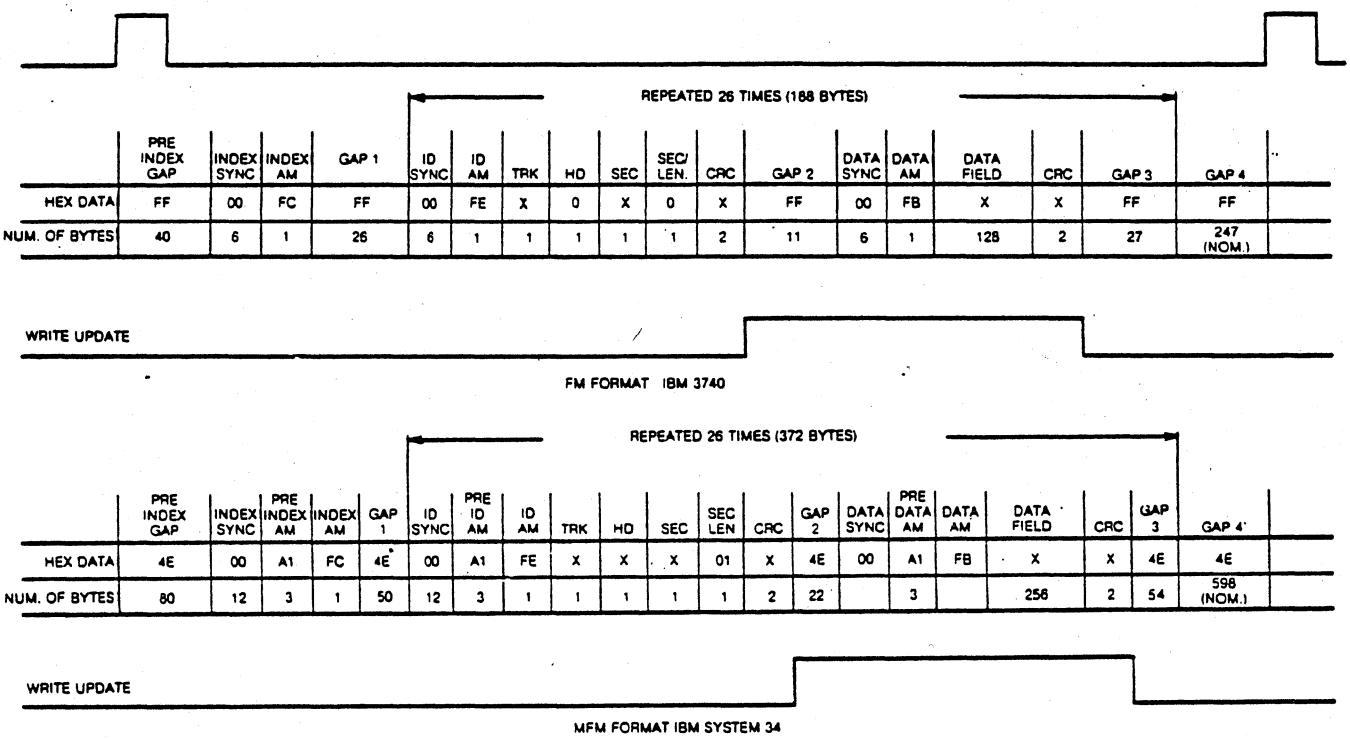


FIGURE 9. 26 SECTOR FORMAT - SA800/850

10.2 32 SECTOR FORMAT

The 32 sector format employs MFM encoding on all tracks of the SA1000. This format yields 32 sectors of 256 bytes per sector. Figure 10 shows the 32 sector format.

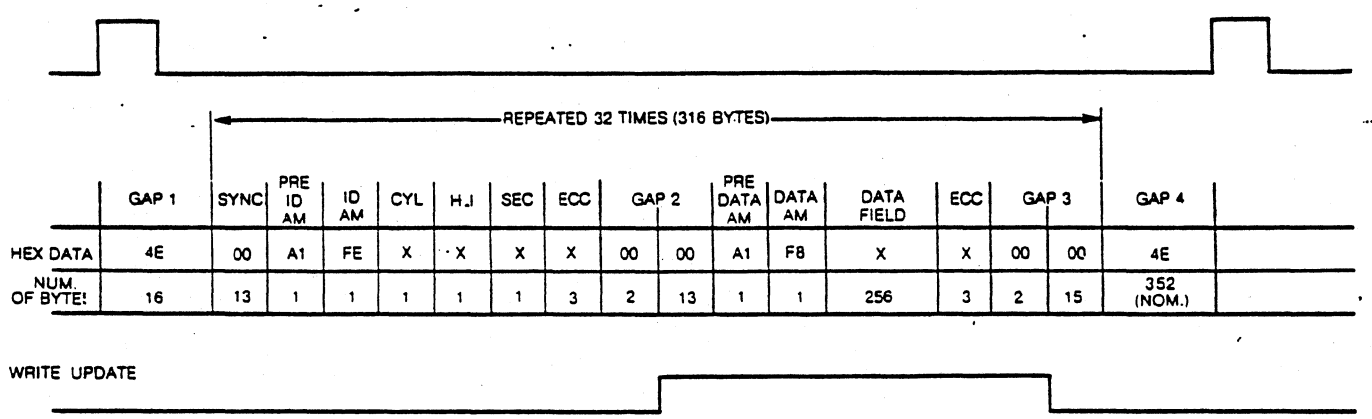


FIGURE 10. 32 SECTOR FORMAT - SA1000

11.0 DRIVE JUMPER SETTINGS

11.1 JUMPER SETTINGS FOR SA800/801 FLOPPY

The following information is contained in the SA800/801 Diskette Storage Drive OEM Manual, Shugart Associates, 1977.

Jumper Name	Function (Enabled if Jumper Installed)
A	Install enable DRSEL to drive selection
B	Install, Head Load on Drive Select
C	Remove, Drive Select loads heads
D	Remove, In Use to LED is disabled
DC	Remove, Disable Disk Change to return to controller
DS	Install enable stepper on Drive Select
DS1-4	Install one only, DS1 = LUN 0 (Drive Select)
HL	Remove, Head load on Drive Select
L	Jumper for -5V (remove for -15V), controller requires -5V only
T1	Remove, Head Load terminator
T2	Install, Pullup for Drive Select lines
T3	Install, Direction terminator
T4	Install, Step terminator
T5	Install, Write Data terminator
T6	Install, Write Gate terminator
X	Install, Head Load Enable
Y	Remove, Disable Hdld from driving LED
Z	Install drive select drives in use LED
800	Install, enables 800 index only operation
801	Remove, disables 801 mode operation

11.2 JUMPER SETTINGS FOR SA850/851 FLOPPY

Jumper Name **Function (Enabled if Jumper Installed)**

Controller is compatible with the factory jumper configuration. See SA850/851 OEM Manual.

Note: Jumpers must be set for SA850, not SA851

11.3 JUMPER SETTINGS FOR SA1000 WINCHESTER

Jumper Name **Function (Enabled if Jumper Installed)**

Controller is compatible with the factory jumper configuration. See SA1000 OEM Manual.

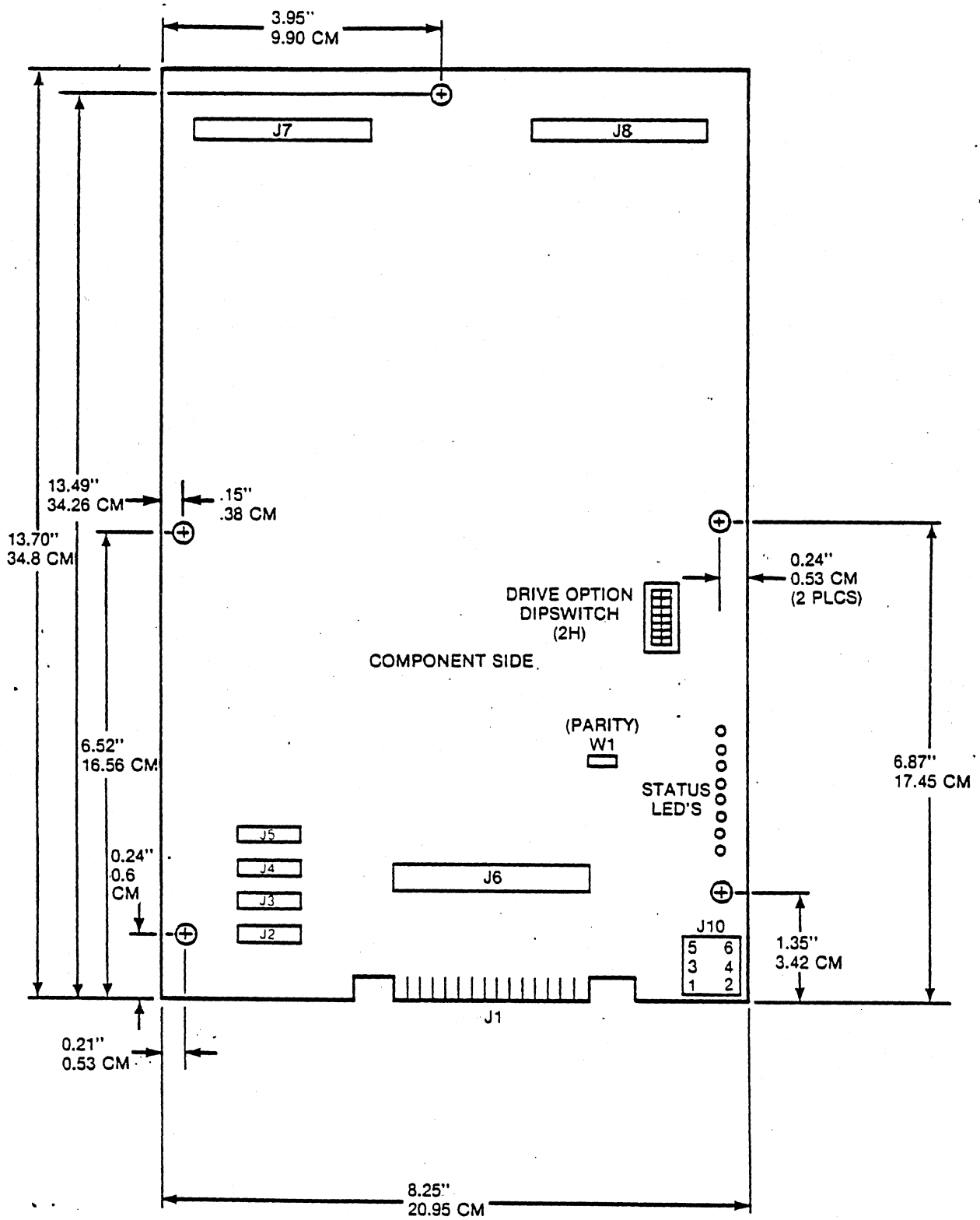


FIGURE 11. SA1403D DIMENSIONAL DRAWING

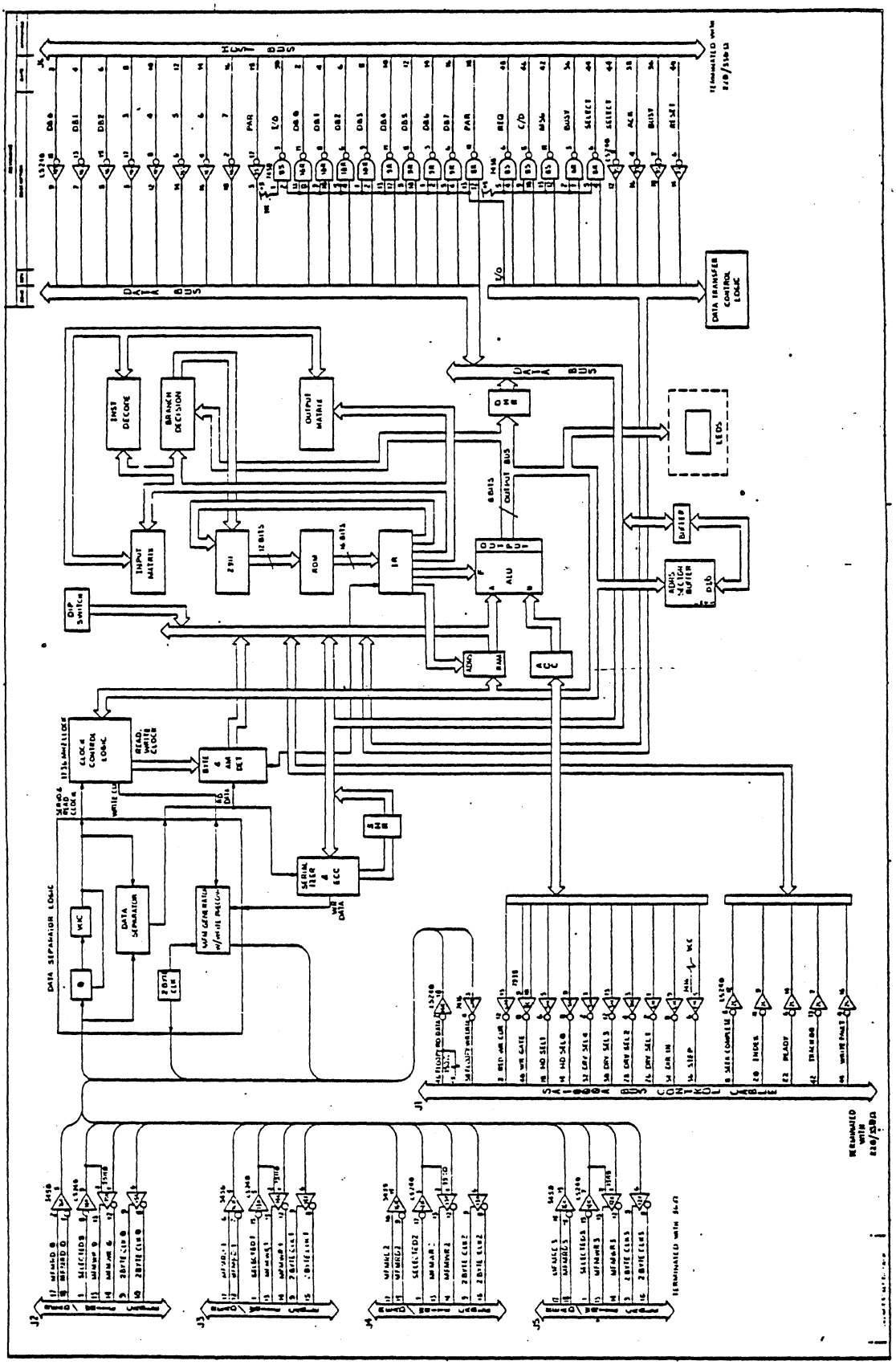


FIGURE 12. SA1403D FUNCTIONAL BLOCK DIAGRAM

Notes

APPENDIX A

CP/M-80 2.2 BIOS Programming Considerations

BIOS provides the operations necessary to access the disk drives and to interface with peripherals. The user interface with the BIOS is through a series of Entry Points. These entry points are "Jump Vectors". Each jump address corresponds to a particular subroutine which performs a specific function. The Base (+ B for the jump vectors) depends on the size of RAM memory.

BIOS Entry Vector Table

BIOS Cold Boot

Entry Point:	(Bbase + 00) - Bios
Function(s):	This entry is called only by the Boot Loader to initialize CP/M.
Argument(s):	None
Value(s) Returned:	None
Registers Saved:	None
Errors Returned:	None

BIOS Warm Boot

Entry Point:	(Bbase + 03) - Bwboot
Function(s):	Perform a Warm Start by reloading the CCP and BDOS from the disk in the A: drive, returning control to the CCP.
Argument(s):	None
Value(s) Returned:	None
Registers Saved:	None
Errors Returned:	None in registers; however, message 'Boot Err' is displayed.

BIOS Console Status
Entry Point: (Bbase + 06) - Bconst

BIOS Console Input
Entry Point: (Bbase + 09) - Bconin

BIOS Console Output
Entry point: (Bbase + 0C) - Bconot

BIOS List Output
Entry Point: (Bbase + 0F) - Bprint

BIOS Punch Output
Entry Point: (Bbase + 12) - Bpunch

BIOS Reader Input
Entry Point: (Bbase + 15) - Breadr

BIOS Home Disk
Entry Point: (Bbase + 18) - Bhome
Function(s): Sets track number to zero in preparation for disk access.
Arguments: None
Value(s) Returned: None
Registers Saved: None
Errors Returned: None

BIOS Select Disk

Entry Point:	(Bbase + 1B) - Bseld
Function(s):	Select the requested logical disk. The drive that will be logged on in further operations is the default drive (or drive A if the default drive cannot be selected).
Arguments:	(C) = drive to select (00 - 0F) (E) = even if media identification required (E) = odd if media identification previously issued and no disks removed/replaced
Value(s) Returned:	(HL) = address of CP/M-compatible Disk Parameter Header if select successful (HL) = 0 otherwise
Registers Saved:	None
Errors Returned:	None

BIOS Set Track

Entry Point:	(Bbase + 1E) - Bsett
Function(s):	Stores desired track number in preparation for a disk read or write record call.
Argument(s):	(BC) = track number
Value(s) Returned:	None
Registers Saved:	None
Errors Returned:	None

BIOS Set Sector

Entry Point	(Bbase + 21) - Bsets
Function(s):	Stores desired sector number in preparation for a read or write record call.
Argument(s):	(BC) = sector number
Value(s) Returned:	None
Registers Saved:	None
Errors Returned:	None

BIOS Set DMA Address

Entry Point:	(Bbase + 24) - Bsetd
Function(s):	Stores desired transfer address in preparation for a read or write a record call.
Argument(s):	(BC) = transfer address
Value(s) Returned:	None
Registers Saved:	None
Errors Returned:	None

BIOS Read Sector

Entry Point:	(Bbase + 27) - Bread
Function(s)	Transfer one 128 (decimal) byte record from the selected disk to the current DMA transfer address.
Argument(s):	Bseld, Bsett, Bsctrn, Bsets, Bsetd previously called.
Value(s) Returned:	None
Registers Saved:	none
Errors Returned:	(A) = 00 if no error (A) = FF if error

BIOS Write Sector

Entry Point:	(Bbase + 2A) - Bwritt
Function(s):	Transfer one 128 (decimal) byte record from the current DMA transfer address to the selected disk.
Argument(s):	Bseld, Bsett, Bsctrn, Bsets, Bsetd previously called.
Value(s) Returned:	None
Registers Saved:	None
Errors Returned:	(A) = 00 if no error (A) = FF if error

BIOS List Status

Entry Point;	(Bbase + 2D) - Bprnts

BIOS Sector Translate

Entry Point:	(Bbase + 30) - Bsctrn
Function(s):	Translate a logical sector number into a physical sector number in preparation for a call to Bsets, the BIOS set sector call.
Argument(s):	(BC) = Sector number (0 < = (BC) < sectors per track
	(DE) = Skew table address obtained from the CP/M Disk Parameter Header
Value(s) Returned:	(HL) = (BC) if (DE) = 0
	(L) = [(DE) + (BC)] if (DE) = 0
	(H) = (B) should be 0
Registers Saved:	None
Errors Returned:	None

Notes

APPENDIX B

Monitor Entry Vector Table

F000H	Cold start monitor
F003H	Warm start monitor
F006H	Keyboard status
F009H	Keyboard input
F00CH	CRT output
F00FH	Fast CRT output from C
F012H	SIO channel B input status
F015H	SIO channel B input
F018H	SIO channel B output
F01BH	Drive select
F01EH	Home r/w head
F021H	Seek to track
F024H	Read sector
F027H	Write sector
F02AH	Execute physical driver request
F02DH	Set direct CRT cursor
F030H	Direct CRT display
F033H	CRT memory block move
F036H	Return address of disk mapping table
F039H	Return address of day variable
F03CH	Return configuration status
F03FH	SIO channel B output ready status
F042H	Set configuration
F045H	Start screen print
F048H	Accessible 1-second interrupt
F04BH	Console status through iobyte
F04EH	Console input through iobyte
F051H	Console output through iobyte
F054H	Printer output through iobyte
F057H	Printer status through iobyte
F05AH	Communications input ready status
F05DH	Communications input data
F060H	Communications output data
F063H	Communications output ready status
F066H	Idle while i/o is pending
F069H	Record soft error

Notes

APPENDIX C

Documented System Storage and Structures

Z80-A Mode 2 Interrupt Vectors

```
FF00    SIOVO:   DEFS2    ;Z80-A SIO port B xmit buffer empty
FF02    SIOV1:   DEFS2    ;Z80-A SIO port B external/status change
FF04    SIOV2:   DEFS2    ;Z80-A SIO port B receive data available
FF06    SIOV3:   DEFS2    ;Z80-A SIO port B special receive
                          condition
FF08    SIOV4:   DEFS2    ;Z80-A SIO port  A xmit buffer empty
FF0A    SIOV5:   DEFS2    ;Z80-A SIO port A external/status change
FF0C    SIOV6:   DEFS2    ;Z80-A SIO port A receive data available
FF0E    SIOV7:   DEFS2    ;Z80-A SIO port A special receive
                          condition

FF10    CTCVO:   DEFS2    ;Z80-A CTC channel 0 interrupt
FF12*   CTCV1:   DEFS2    ;Z80-A CTC channel 1 interrupt
FF14    CTCV2:   DEFS2    ;Z80-A CTC channel 2 interrupt
FF16*   CTCV3:   DEFS2    ;Z80-A CTC channel 3 interrupt

FF18    SYSVA:   DEFS2    ;System Z80-A PIO port A interrupt
FF1A*   SYSVB:   DEFS2    ;System Z80-A PIO port B interrupt

FF1C    GENVA:   DEFS2    ;General purpose Z80-A PIO port A
                          interrupt
FF1E    GENVB:   DEFS2    ;General purpose Z80-A PIO port B
                          interrupt
```

*Vectors used by the Monitor ROM

Keyboard Data Input FIFO Variables

```
FF20    fifo:    defs 16  ;Console input fifo
FF30    fifcnt:  defs 1   ;FIFO data counter
FF31    fifin:   defs 1   ;FIFO input pointer
FF32    fifout:  defs 1   ;FIFO output pointer

FF33             defs 1   ;Round address
```

More Interrupt Vectors

```
FF34    expvec:   defs 8        ;Space for 4 vectors for expansion slot
```

Available Memory Pointers

```
FF3C    availb:   defs 2        ;Bottom of available memory
FF3E    availt:   defs 2        ;Top of available memory
```

End of documented storage locations

Logical to Physical Drive Mapping Tables

Seltab contains two bytes per logical CP/M drive A-P. The first byte is an index into the physical driver address table (see next table). The second byte is a unit number that is passed to the driver by the XQDVR dispatcher.

```
Seltab:

    A:    defb    1,0     ;Floppy unit 0
    B:    defb    1,1     ;Floppy unit 1
    C:    defb    1,2     ;Floppy unit 2
    D:    defb    1,3     ;Floppy unit 3

    E:    defb    1,4     ;Rigid partition 0
    F:    defb    1,5     ;Rigid partition 1
    G:    defb    1,6     ;Rigid partition 2
    H:    defb    1,7     ;Rigid partition 3
    I:    defb    0,0     ;Error driver
    J:    defb    0,0     ;Error driver
    K:    defb    0,0     ;Error driver
    L:    defb    0,0     ;Error driver
    M:    defb    0,0     ;Error driver
    N:    defb    0,0     ;Error driver
    O:    defb    0,0     ;Error driver
    P:    defb    0,0     ;Error driver
```

Physical Driver Address Table

Drvtab contains the addresses of several independent physical disk drivers. By convention, driver number 0 always returns a select error. Unused entries in Seltab should point to this trivial driver.

```
Drvtab:   defw    Selerr    ;Select error physical driver
          defw    Dskdvr    ;Disk driver (WD or SA)
          defw    0         ;Empty physical driver
                            ;Expansion slots

          defw    0
          defw    0
          defw    0
          defw    0
          defw    0-1       ;Mark last entry
```

Physical Driver Request Block

```
db    command     ;FF = Select
                  ;00 = Write
                  ;01 = Read
ds    1           ;For system use
db    Ldrive      ;Logical drive for request (00 - 0F)
dw    Track       ;Track number for request
dw    Sector      ;Sector number for request
dw    Address     ;Address of sector buffer for request
```

Time-of-Day and Timer Variables

```
        Milsec:     ds   2      ;Location incremented by CTC1
                                ;Interrupt
                    ds   2      ;(unused)
        Ticker:     ds   2      ;Increments once per second
        Steprt:     ds   1      ;WD1797 step rate
        Motor:      ds   1      ;Disk motor/select timeout (1 Hz)
HL→     Day:        ds   1      ;01-31
        Month:      ds   1      ;01-12
        Year:       ds   1      ;80-99
        Hour:       ds   1      ;00-23
        Minute:     ds   1      ;00-59
        Second:     ds   1      ;00-59
        Linbuf:     ds   80     ;Line buffer
```

How To Make Monitor Calls from Basic

Several of the monitor function calls return the value in the HL register if the H register equals 0, or return the value at the address pointed to by the HL register if the H register is not zero. This convention allows Microsoft Basic Users to access these functions directly. The examples listed in this section demonstrate this feature of the ROSR ROM.

```
100  '
110  ' Make 820-II Monitor call to get address of day variable, then
120  ' Print Day, Month etc.
130  '
140  DATA Day,Month,Year,Hour,Minute,Second
150  '
160  DEFINT I
170  GETTOD = &HF039:CALL GETTOD(I)          'Return Add. of Day
180  FOR X = 0 TO 5
190      READ X$
200      PRINT USING "\     \ ##;X$,PEEK(I + X)
210  NEXT X
220  END
```

```
100  '
110  ' Do configuration status call & print value returned
120  '
130  DEFINT I
140  GETCON = &HF03C:CALL GETCON(I)          'Get config status
150  PRINT CHR$(26);                 'Clear Screen
160  PRINT "The configuration status word is - ";
170  PRINT HEX$(I);
180  PRINT " (Hex)"
190  END
```

```
100  ' Example Using Line Delete To scroll screen up.
110  ' Make 820-II Monitor Call to get address of day variable
120  ' then calculate address of line input buffer variable.
130  '
140  ' Clear screen, fill screen with characters, position
150  ' Cursor back on top line, send line delete code to CRT,
160  ' This moves the line deleted from the top of the screen
170  ' To the input buffer.
180  '
190  ' Recall deleted line from line input buffer & display
200  ' on line 23 of the screen.
210  '
220  '
230  WIDTH 255
240  PRINT CHR$(5);" ";              'Remove cursor
250  DEFINT I
260  GETTOD = &HF039:CALL GETTOD(I) 'Get address of Day Variable
270  I = I + 6              'Line input buffer is at Day + 6
280  PRINT CHR$(26);          'Clear screen
290  FOR X = 1 TO 23
300      PRINT STRING$(80,CHR$(X + 64)); 'Fill Screen
310  NEXT X
320  '
330  FOR M = 1 TO 100                'Do 100 lines
340      PRINT CHR$(30);             'Put Cursor back on top line
350      PRINT CHR$(27);"R";         'Do line delete, move deleted
360                                  'Line to buffer.
370      PRINT CHR$(27);" = ";CHR$(32 + 22);CHR$(32)
380      FOR X = 0 TO 79             'Now print characters back from
390          PRINT CHR$(PEEK(I + X); 'Input buffer
400      NEXT X
410  NEXT M
420  PRINT CHR$(26);CHR$(5);CHR$(2);   'Clear screen and
430  '                                  Restore Cursor.
440  END
```

```
100  ' Example Using Line Insert To scroll screen down.
110  ' Make 820-II Monitor Call to get address of day variable
120  ' then calculate address of line input buffer variable.
130  '
140  ' Clear screen, fill screen with characters, position
150  ' Cursor back on top line, send line insert code to CRT,
160  ' This moves the line deleted from the bottom of the screen
170  ' To the input buffer.
180  '
190  ' Recall deleted line from line input buffer & display
200  ' on the first line of the screen.
210  '
220  '
230  WIDTH 255
240  PRINT CHR$(5);" ";                    'Remove cursor
250  DEFINT I
260  GETTOD = &HF039:CALL GETTOD(I) 'Get address of Day Variable
270  I = I + 6          'Line input buffer is at Day + 6
280  PRINT CHR$(26);         'Clear screen
290  FOR X = 1 TO 23
300      PRINT STRING$(80,CHR$(X + 64)); 'Fill Screen
310  NEXT X
320  '
330  FOR M = 1 TO 100:       'Do 100 lines
340      PRINT CHR$(30);    'Put Cursor back on top line
350      PRINT CHR$(27);"E"; ' Do line insert, move deleted
360                          'Line to buffer.
370      PRINT CHR$(27);" = ";CHR$(32 + 22);CHR$(32)
380      FOR X = 0 TO 79     'Now print characters back from
390              PRINT CHR$(PEEK(I + X); 'Input buffer
400      NEXT X
410  NEXT M
420  PRINT CHR$(26);CHR$(5);CHR$(2);   'Clear screen and
430  '                                  Restore Cursor.
440  END
```

Bank Switching

- The Bank control switch is bit 7 of port 1C.
  Bit 7 = 0 = Bank 1 (RAM)
  Bit 7 = 1 = Bank 0 (ROM)

- Change bit 7 only: Bits 0 through 6 should be maintained.

- Bank 0 and 1 are mutually exclusive; data movement to or from one bank will not affect the other.

- When bank switching, the driver code must be executed at C000h or above; the upper 16K (C000h-FFFFh) is common memory to both banks.

For example,

```
DI
IN A,(1Ch)      ;read port
SET 7,a         ;set bit
EI
OUT (1Ch),a     ;output

DI
IN A,(1Ch)
RES 7,a         ;reset bit
EI
OUT (1Ch)
```

```
                                        title   Balcones Operating System for the XEROX 820-II
                                        .z80


                                ;;;     Balcones Operating System for XEROX 820-II.
                                ;
                                ;       Copyright 1982 (C) Balcones Computer Corporation
                                ;
                                ;                       All rights reserved
                                ;
                                ;;                      Robert Burns, Bcc.
                                ;


0191                            rev     defl    401


                                        subttl  Symbol Definitions
                                        page
```

```
                                
FFFF                    true    equ     -1
0000                    false   equ     not true
                        
0000                    debug   equ     false           ;assemble ram loader
                        
                        ;;      Absolute Memory Addresses.
                        ;
0000                    rom     equ     01000h and debug;non resident code base
1800                    romsiz  equ     01000h+((not debug) and 0800h)
1800                    Rx1984  equ     01800h          ;prescription for the future
0800                    Lx1984  equ     00800h          ;length of future
0003                    iobyte  equ     00003h          ;i/o byte
0080                    bootld  equ     00080h          ;boot loader address
ED80                    bootbf  equ     0ed80h          ;boot loader buffer
FF00                    ram     equ     0ff00h          ;system ram page address
F000                    monitr  equ     0f000h          ;resident monitor address
3000                    crtmem  equ     03000h          ;crt memory address
3C00                    crtmax  equ     crtmem+24*128   ;crt maximum address
0030                    crtbas  equ     high crtmem     ;starting page of display ram
003C                    crttop  equ     high crtmax     ;ending page of display ram
                        
                        ;;      I/O Port Addresses.
                        ;
0000                    bauda   equ     00h             ;channel a baud rate generator
0004                    siodpa  equ     04h             ;sio data port A (communications)
0005                    siodpb  equ     05h             ;sio data port B (printer)
0006                    siocpa  equ     06h             ;sio control/status port A
0007                    siocpb  equ     07h             ;sio control/status port B
0008                    gpioda  equ     08h             ;general purpose parallel i/o A data
0009                    gpioca  equ     09h             ;general purpose parallel i/o A control
000A                    gpiodb  equ     0ah             ;general purpose parallel i/o B data
000B                    gpiocb  equ     0bh             ;general purpose parallel i/o B control
000C                    baudb   equ     0ch             ;channel b baud rate generator
0010                    wd1797  equ     10h             ;western digital disk controller base
0014                    scroll  equ     14h             ;crt bottom line scroll register
0018                    ctc     equ     18h             ;quad counter/timer circuit
0018                    ctc0    equ     18h             ;ctc channel 0 (user)
0019                    ctc1    equ     19h             ;ctc channel 1 (msec, screen print)
001A                    ctc2    equ     1ah             ;ctc channel 2 (one second prescaler)
001B                    ctc3    equ     1bh             ;ctc channel 3 (one second)
001C                    syspio  equ     1ch             ;system pio data
001D                    sysctl  equ     1dh             ;system pio control
001E                    kbddat  equ     1eh             ;keyboard data
001F                    kbdctl  equ     1fh             ;keyboard control
0028                    bellof  equ     28h             ;turn bell off
0029                    bellon  equ     29h             ;turn bell on
0030                    slsden  equ     30h             ;select single density
0031                    sldden  equ     31h             ;select double density
0034                    chrom1  equ     34h             ;select ROM 1 character generator
0035                    chrom2  equ     35h             ;select ROM 2 character generator
0036                    lowlite equ     36h             ;select low intensity attribute
0068                    async   equ     68h             ;set internal clocks for asynchronous sio A
0069                    sync    equ     69h             ;set external clocks for  synchronous sio A
                        
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Symbol Definitions

                                ;;      Configuration Status Byte Bit Definitions.
                                ;
0007                            c.keym  equ     7               ;Keyboard upper bit is passed
0006                            c.sasi  equ     6               ;Shugart SA-1403D Disk Controller
0004                            c.five  equ     4               ;Five inch micro floppies

                                ;;      Ascii.
                                ;
0004                            eot     equ     04h             ;ascii end of text
000A                            lf      equ     0ah             ;ascii line feed
000D                            cr      equ     0dh             ;ascii carriage return
0011                            xon     equ     11h             ;ascii Xon
0013                            xoff    equ     13h             ;ascii Xoff
001B                            esc     equ     1bh             ;ascii escape
001A                            clrs    equ     1ah             ;clear screen

                                ;;      Special Key Constants.
                                ;
001E                            Helpkey equ     01eh
009E                            Scrprt  equ     09eh            ;Screen Print key CTRL <HELP>
009B                            Abort   equ     09bh            ;Automatic Abort  CTRL <ESC>

                                ;;      Bell Constants.
                                ;
0035                            bltim   equ     35h             ;bell loop time
0061                            blonc   equ     61h             ;bell on   time
0061                            blofc   equ     61h             ;bell off  time

                                ;;      Assembly Options.
                                ;
8000                            o.resv  equ     1000000000000000b       ;reserved
4000                            o.auto  equ     0100000000000000b       ;auto boot A:
2000                            o.help  equ     0010000000000000b       ;help command
1000                            o.prot  equ     0001000000000000b       ;printer protocol
0800                            o.ddvr  equ     0000100000000000b       ;disk drivers
0400                            o.baud  equ     0000010000000000b       ;baud rate set command
0200                            o.inpc  equ     0000001000000000b       ;in command
0100                            o.outc  equ     0000000100000000b       ;out command
0080                            o.verf  equ     0000000010000000b       ;verify memory block
0040                            o.ramt  equ     0000000001000000b       ;simple ram test
0020                            o.disk  equ     0000000000100000b       ;console disk read/write commands
0010                            o.esct  equ     0000000000010000b       ;escape command table
0008                            o.type  equ     0000000000001000b       ;typewriter mode
0004                            o.fill  equ     0000000000000100b       ;fill memory
0002                            o.move  equ     0000000000000010b       ;move memory
0001                            o.term  equ     0000000000000001b       ;terminal scroll driver

0000                            options defl    debug    and not o.ddvr and not o.esct
0000                            options defl    options  and not o.disk and not o.resv
0000                            options defl    options  and not o.verf and not o.fill
0000                            options defl    options  and not o.ramt

BFFF                            options defl    (not debug or o.esct) and not o.auto

                                ;;      configuration sector offsets.
                                ;
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Symbol Definitions

  132     EE5F                          z.scra  equ     bootbf+255-32   ;initial screen attribute
  133     EE60                          z.stpr  equ     z.scra+1        ;floppy step rate
  134     EE62                          z.keym  equ     z.stpr+2        ;keyboard mask
  135     EE63                          z.sioA  equ     z.keym+1        ;sio A init
  136     EE6D                          z.sioB  equ     z.sioA+10       ;sio B init
  137     EE77                          z.siom  equ     z.sioB+10       ;clear to send low/high/ignore
  138     EE79                          z.siov  equ     z.siom+2        ;data carrier detect low/high/ignore
  139     EE7B                          z.xonp  equ     z.siov+2        ;Xon/Xoff protocal
  140     EE7D                          z.baua  equ     z.xonp+2        ;comm channel baud rate
  141     EE7E                          z.baub  equ     z.baua+1        ;printer baud rate
  142     EE7F                          z.iobt  equ     z.baub+1        ;initial i/o byte
  143
  144                                   ;;      parallel printer status bits.
  145                                   ;
  146     0007                          p.ackn  equ     7               ;acknowledge
  147     0006                          p.onln  equ     6               ;on line
  148     0005                          p.rdyi  equ     5               ;ready to input
  149     0004                          p.rdyo  equ     4               ;ready to output
  150     0002                          p.strb  equ     2               ;data stobe
  151     0000                          p.auto  equ     0               ;auto LF enable
  152
  153                                           subttl  Code Generation Control Macros Definitions
  154                                           page
```

```
155
156                             ;;      Rom code placement macros.
157                             ;
158                             ;;      The Common Segment holds the non-resident (banked) portion
159                             ;       of the monitor.  This segment is not copied to ram.
160                             ;
161                             ;;      The Data Segment holds the resident portion of the monitor.
162                             ;       It is moved to ram at location MONITR during initialization.
163                             ;
164                             ;;      The Code Segment holds the various Transient Commands.  Each
165                             ;       command is loaded from the ROM to the TPA when it is executed.
166                             ;
167                             ;;      The following macros keep it all straight.
168                             ;
169                             ;;      below - Generate code for rom below.
170                             ;
171                             below   macro
172                             \       segment b               ;;enable common segment
173                                     endm
174
175                             ;;      above - Generate code for ram above.
176                             ;
177                             above   macro
178                                     segment d               ;;enable data segment
179                                     endm
180
181                             ;;      Overlay - Generate code for transients.
182                             ;
183                             overlay macro   addr
184                             tloc    defl    tloc+$-cloc
185                             addr    equ     tloc+bloc+cloc-Monitr
186                                     segment c               ;;enable code segment
187                                     endm
188
189                             ;;      bseg - activate common segment.
190                             ;
191                             bseg    macro
192                                     common  /COMROM/
193                                     defs    comres
194                             sega    defl    $
195                                     endm
196
197                             ;;      segment - Activate Segment.
198                             ;
199                             segment macro   s
200                                     update                  ;;update active phase counter
201                             s&space defl    -1              ;;set enabled segment active
202                                     s&seg                   ;;activate segment code placement
203                                     .phase  s&loc           ;;set absolute segment location counter
204                                     endm
205
206                             ;;      update - Update Phase Counters.
207                             ;
208                             update  macro
209                                     if      bspace
```

```
                                comres  defl    $-rom+100h-3
                                        endif
                                        irpc    x,<bcd>
                                        if      x&space         ;;if segment active
                                        if      cspace
                                        if      tpal lt ($-cloc)
                                tpal    defl    $-cloc
                                        endif
                                        else
                                x&loc   defl    $               ;;save segment address
                                        endif
                                x&space defl    0               ;;clear segment active
                                        .dephase                ;;revert to relocatable
                                        endif
                                        endm
                                        endm

0000                            bloc    defl    rom             ;establish non-resident code base
F000                            dloc    defl    monitr          ;establish resident code base
0000                            tloc    defl    0               ;establise Transient code base
0000                            tpal    defl    0               ;establish maximum transient length
0000                            bspace  defl    0               ;preset common segment inactive
0000                            cspace  defl    0               ;preset code segment inactive
0000                            dspace  defl    0               ;preset data segment inactive
0000                            comres  defl    0               ;preset common base address

                                        subttl  Ram Loader for Testing Only
                                        page
```

```
                                        bseg
0000!                   +               defs    comres
0000!                           entry:
0000!  21 00FD!                 xcks:   ld      hl,bbase+movln
0003!  01 17FF                          ld      bc,romsiz-1
0006!  1E 00                            ld      e,0             ;preset checksum
0008!  7E                       xcks1:  ld      a,(hl)
0009!  23                              inc     hl
000A!  83                              add     a,e
000B!  5F                              ld      e,a
000C!  0B                              dec     bc
000D!  78                              ld      a,b
000E!  B1                              or      c
000F!  20 F7                           jr      nz,xcks1
0011!  7B                              ld      a,e             ;store twos complement of checksum
0012!  ED 44                           neg
0014!  77                              ld      (hl),a          ;store checksum
0015!  C3 0000                        jp      0

                                        subttl  System Initialization
                                        page
```

```
                                bbase:                          ;symbol for accessing non-resident base address
0018!                           defs    100h-($-(entry-3)),-1

00E5                    movln   equ     $-bbase
00FD                    comres  defl    100h-3

                                below                           ;generate non-resident code
0000!                   +       defs    comres

                        ;;      prs - preset storage.
                        ;
                        ;       Entry:  Power up or Reset button.
                        ;
0000    F3              prs:    di                              ;lock up system
0001    AF                      xor     a
0002    3D              prs1:   dec     a                       ;the pause that refershes
0003    20 FD                   jr      nz,prs1
0005    ED 73 FFE0              ld      (rstsp),sp              ;save partial reset state
0009    22 FFE2                 ld      (rsthl),hl              ;in case the luser go boom
000C    E1                      pop     hl                      ;pick possible return off stack
000D    22 FFE4                 ld      (rstpc),hl
0010    D9                      exx                             ;give primary registers half a break
0011    1C                      inc     e
0012    31 3839                 ld      sp,3839h                ;load strange values in SP
0015    31 4142                 ld      sp,4142h
0018    4C                      ld      c,h                     ;insure
0019    43                      ld      b,e                     ;registers
001A    4F                      ld      c,a                     ;can
001B    4E                      ld      c,(hl)                  ;forget
001C    45                      ld      b,l                     ;insure
001D    53                      ld      d,e                     ;registers
001E    43                      ld      B,e                     ;can
001F    4F                      ld      C,a                     ;copy
0020    4D                      ld      C,l
0021    50                      ld      d,b
0022    55                      ld      d,l
0023    54                      ld      d,h
0024    45                      ld      b,l
0025    52                      ld      d,d
0026    08                      ex      af,af'
0027    3E 17                   ld      a,24-1                  ;line up bottom of screen
0029    D3 14                   out     (scroll),a              ;init scroll port
002B    21 3000                 ld      hl,crtmem               ;clear display memory
002E    36 20                   ld      (hl),' '
0030    11 3001                 ld      de,crtmem+1
0033    01 0BFF                 ld      bc,crtmax-crtmem-1
0036    ED B0                   ldir                            ;pray the video hardware works
0038    31 F000                 ld      sp,monitr               ;insure monitor ram ok
003B    21 AA55         prs2:   ld      hl,0aa55h               ;walk checker board through ram
003E    C1                      pop     bc                      ;read ram
003F    E5                      push    hl                      ;write ram fast
0040    D1                      pop     de                      ;read ram fast
0041    C5                      push    bc                      ;put ram back
```

```
  316  0042  F1                        pop     af              ;and verify it
  317  0043  90                        sub     b
  318  0044  20 76                     jr      nz,err1         ;if ram failure
  319  0046  ED 52                     sbc     hl,de
  320  0048  20 72                     jr      nz,err1         ;if ram or register failure
  321  004A  3B                        dec     sp              ;advance test address
  322  004B  3F                        ccf
  323  004C  ED 7A                     adc     hl,sp
  324  004E  20 EB                     jr      nz,prs2         ;if top of memory not reached
  325  0050  31 0000                   ld      sp,stack        ;set monitor stack
  326  0053  21 0000                   ld      hl,prs          ;set rom address
  327  0056  01 1800                   ld      bc,romsiz
  328  0059  CD 00AF                   call    ccs             ;compute check sum
  329  005C  20 63                     jr      nz,err2         ;if bad rom
  330  005E  21 00E6                   ld      hl,intab        ;point to default variable table
  331  0061  06 00             prs3:   ld      b,0
  332  0063  4E                        ld      c,(hl)          ;set data block length
  333  0064  23                        inc     hl
  334  0065  5E                        ld      e,(hl)          ;set variable address in ram
  335  0066  23                        inc     hl
  336  0067  56                        ld      d,(hl).
  337  0068  23                        inc     hl              ;point to initial values
  338  0069  ED B0                     ldir                    ;copy data from rom to variables in ram
  339  006B  CB 7E                     bit     7,(hl)
  340  006D  28 F2                     jr      z,prs3          ;if more data to preset
  341  006F  23                        inc     hl              ;point to i/o init data table
  342  0070  46                 prs4:  ld      b,(hl)          ;set number of bytes to preset
  343  0071. 23                        inc     hl
  344  0072  4E                        ld      c,(hl)          ;set i/o port address
  345  0073  23                        inc     hl
  346  0074  ED B3                     otir                    ;shoot preset data to i/o device
  347  0076  CB 7E                     bit     7,(hl)
  348  0078  28 F6                     jr      z,prs4          ;if more devices require initialization
  349  007A  DB 1E                     in      a,(kbddat)      ;assert PARDY
  350  007C  ED 5E                     im      2               ;select interrupt mode 2
  351  007E  3E FF                     ld      a,high vectab   ;set interrupt vector page
  352  0080  ED 47                     ld      i,a
  353  0082  21 041B                   ld      hl,rbase        ;set resident base address
  354  0085  11 F000                   ld      de,monitr       ;set monitor address
  355  0088  01 0F00                   ld      bc,ram-monitr   ;set max resident length
  356  008B  ED B0                     ldir                    ;plant monitor upstairs
  357  008D  21 1800                   ld      hl,Rx1984       ;prognosticate
  358  0090  01 0800                   ld      bc,Lx1984
  359  0093  CD 00AF                   call    ccs
  360  0096  20 14                     jr      nz,prs5
  361  0098  2A 1FFD                   ld      hl,(Rx1984+Lx1984-3)
  362  009B  11 55AA                   ld      de,55aah
  363  009E  ED 52                     sbc     hl,de
  364  00A0  21 FAD8                   ld      hl,cmdtab
  365  00A3  11 F360                   ld      de,seltab
  366  00A6  01 FC55                   ld      bc,cloc
  367  00A9  CC 1800                   call    z,Rx1984        ;FutureShock
  368  00AC  C3 FC55            prs5:  jp      signon          ;Signon Resident Monitor
  369
  370  00AF  1E 00              ccs:   ld      e,0             ;preset ckecksum
  371  00B1  7E                 ccs1:  ld      a,(hl)
```

```
00B2  23                        inc     hl
00B3  83                        add     a,e
00B4  5F                        ld      e,a
00B5  0B                        dec     bc
00B6  78                        ld      a,b
00B7  B1                        or      c
00B8  20 F7                     jr      nz,ccs1
00BA  B3                        or      e
00BB  C9                        ret

00BC  21 00D4           err1:   ld      hl,errm1        ;set ram error message
00BF  18 03                     jr      err
00C1  21 00DD           err2:   ld      hl,errm2
00C4  11 3024           err:    ld      de,crtmem+40-(errm1/2)
00C7  01 0009                   ld      bc,errm1
00CA  ED B0                     ldir
00CC  0B                err3:   dec     bc              ;pause a while
00CD  78                        ld      a,b
00CE  B1                        or      c
00CF  20 FB                     jr      nz,err3
00D1  C3 0000                   jp      prs             ;try restart again

00D4  52 61 6D 20       errm1:  db      'Ram Error'
00D8  45 72 72 6F
00DC  72
00DD  52 6F 6D 20       errm2:  db      'Rom Error'
00E1  45 72 72 6F
00E5  72
0009                    errm1   equ     ($-errm1)/2

                        ;;      initialize the interrupt vector table
                        ;
00E6  02                intab:  defb    2
00E7  FF1A                      defw    sysvec+2
00E9  F140                      defw    -keysrv         ;parallel keyboard interrupt vector

00EB  06                        defb    6
00EC  FF12                      defw    ctcvec+2
00EE  F1FD                      defw    milli           ;one millisecond interrupt timer
00F0  0000                      defw    0
00F2  F192                      defw    timer           ;one second timer interrupt vector
                        ;
                        ;       init keyboard fifo
                        ;
00F4  03                        defb    3
00F5  FF30                      defw    fifcnt
00F7  00                        defb    0               ;fifo count
00F8  00                        defb    0               ;fifo in
00F9  00                        defb    0               ;fifo out
                        ;
                        ;       initialize the crt display
                        ;
00FA  08                        defb    8
00FB  FFAC                      defw    cursor
00FD  3000                      defw    crtmem          ;base address is 3000h
00FF  02                        defb    02              ;use non-blinking box cursor
```

```
 428  0100  3000                        defw    crtmem          ;direct crt memory output address
 429  0102  17                          defb    23              ;initial scroll base
 430  0103  00                          defb    0               ;initial leadin
 431  0104  00                          defb    0               ;initial attribute
 432                              ;
 433                              ;       Initialize configurable parameter addresses
 434                              ;
 435  0105  0C                          defb    2*numcon
 436  0106  FFBF                        defw    contbl          ;configure table address
 437  0108  F10C                cfinit: defw    siomsk          ;printer output ready mask
 438  010A  F10E                        defw    sioval          ;printer output ready value
 439  010C  F115                        defw    xonenb          ;Xon / Xoff enable/disable (NOP or RET)
 440  010E  FF54                        defw    steprt          ;step rate for wd1797
 441  0110  FFCB                        defw    spare1
 442  0112  FFCC                        defw    spare2
 443  0006                      numcon  equ     ($-cfinit)/2
 444
 445  0114  04                          defb    2*2
 446  0115  FF3C                        defw    availb
 447  0117  F7FD                        defw    iobloc+iobdvs
 448  0119  FC80                        defw    ram-280h
 449
 450  011B  FF                          defb    -1              ;end of variable init table
 451
 452                              ;;      I/O port initialization.
 453                              ;
 454                              ;       initialize system pio for use as bank-switch,
 455                              ;       configuration select and parallel keyboard input
 456                              ;
 457
 458  011C  01 1D                       defb    1,sysctl
 459  011E  4F                          defb    01001111b       ;select input mode
 460
 461  011F  01 1C                       defb    1,syspio
 462  0121  80                          defb    10000000b       ;enable ROM
 463
 464  0122  03 1D                       defb    3,sysctl
 465  0124  CF                          defb    11001111b       ;put system pio in bit mode
 466  0125  3F                          defb    00111111b       ;make bits 5, 4, 3, 2, 1,  and 0 be inputs
 467  0126  07                          defb    00000111b       ;disable interrupts
 468
 469  0127  03 1F                       defb    3,kbdctl
 470  0129  4F                          defb    01001111b       ;put keyboard port in input mode
 471  012A  1A                          defb    sysvec+2        ;load keyboard interrupt vector
 472  012B  83                          defb    10000011b       ;enable interrupts
 473                              ;
 474                              ;       Initialize Counter Timer Circuit.
 475                              ;
 476  012C  02 18                       defb    2,ctc0
 477  012E  03                          defb    00000011b       ;reset timer
 478  012F  10                          defb    low ctcvec      ;base interrupt vector for ctc
 479
 480  0130  02 19                       defb    2,ctc1
 481  0132  07                          defb    00000111b       ;start timer, but no interrupts
 482  0133  FA                          defb    250             ;ctc1 period = 1 msec
 483
```

```
                                defb    2,ctc2
                                defb    00100111b       ;put ctc2 in timer / 256 mode (64 usec/count)
                                defb    125             ;ctc2 period = 8 msec

                                defb    2,ctc3
                                defb    11000111b       ;put ctc3 in counter mode with interrupt
                                defb    125             ;ctc3 period = 125*8 msec = 1 second
                        ;
                        ;       initialize sio channel b for asynchronous serial
                        ;       interface to printer or terminal
                        ;
                                defb    10,siocpb
                                defb    1               ;select register #1
                                defb    00000000b       ;disable interrupts
                                defb    2               ;select register #2
                                defb    low siovec      ;base sio interrupt vector
                                defb    3               ;select register #3
                                defb    01000001b       ;7 bits/rx characters
                                defb    4               ;select register #4
                                defb    01000111b       ;16x clock, 1 stop bit, even parity enabled
                                defb    5               ;select register #5
                                defb    10101010b       ;DTR, 7 bits/tx character, Tx enb, RTS

                                defb    1,baudb
                                defb    0111b           ;default clock is 1200 bps

                        ;
                        ;       initialize communications port for async modem interface
                        ;
                                defb    8,siocpa
                                defb    1               ;select register #1
                                defb    00000000b       ;disable interrupts
                                defb    3               ;select register #3
                                defb    01000001b       ;7 bits/rx characters
                                defb    4               ;select register #4
                                defb    01000111b       ;16x clock, 1 stop bit, even parity enabled
                                defb    5               ;select register #5
                                defb    10101010b       ;DTR, 7 bits/tx character, Tx enb, RTS

                                defb    1,bauda
                                defb    0101b           ;default clock is 300 bps

                                defb    1,async         ;set internal Rx+Tx clocks
                                defb    0
                        ;
                        ;       initialize PIO for Centronics style printer
                        ;
                                defb    3,gpioca
                                defb    11001111b       ;mode 3
                                defb    00000000b       ;all output
                                defb    00000111b       ;no interrupts

                                defb    3,gpiocb
                                defb    11001111b       ;mode 3
                                defb    11110000b       ;upper nibble in, lower out
                                defb    00000111b       ;no interrupts
```

```
                                        defb    1,gpiodb
0165    01 0A                           defb    (1 shl p.strb) or (1 shl p.auto)
0167    05
                                        defb    -1              ;end of i/o init table
0168    FF
                                        subttl  Resident Monitor Entry Points
                                        page
```

```
                                ;;      Resident monitor entry points.
                                ;
                                ;       This Entry Point Vector provides the only reliable access
                                ;       to services provided by the Resident Monitor.  Any access
                                ;       to code in the Monitor or its Ram page past the keyboard
                                ;       variables is not allowed.  Future releases of the Resident
                                ;       Monitor will always provide compatability with these entry
                                ;       vectors.
                                ;
                                ;       This restriction also applies to the Resident Monitor Ram
                                ;       Page at the top of memory.  Access to Ram Variables must
                                ;       be obtained through the appropriate entry vector.
                                ;
                                        above
0266!                                   cseg
F000    C3 F07C         cold:   jp      restart         ;monitor restart
F003    C3 FA62         warm:   jp      prompt          ;monitor entry point
F006    C3 F0CD         const:  jp      kbdst           ;console status to A
F009    C3 F0D8         conin:  jp      kbdin           ;console input to A
F00C    C3 F2F1         conout: jp      crtout          ;console output from A
F00F    C3 F2FE                 jp      fastcrt         ;fast crt output from C
F012    C3 F0E5                 jp      siost           ;sio channel b status to A
F015    C3 F0F0                 jp      sioin           ;sio channel b input to A
F018    C3 F0F8                 jp      sioout          ;sio channel b output from A
F01B    C3 FA17                 jp      select          ;select drive in C
F01E    C3 FA3C                 jp      home            ;home r/w head
F021    C3 FA3E                 jp      seek            ;seek to track in C
F024    C3 FA48                 jp      read            ;read sector  C -> buffer @ HL
F027    C3 FA44                 jp      write           ;write sector C <- buffer @ HL
F02A    C3 F344                 jp      xqdvr           ;execute physical driver request @ HL
F02D    C3 F284                 jp      setcur          ;set direct crt cursor from HL
F030    C3 F288                 jp      outcur          ;direct crt display
F033    C3 F2A3                 jp      crtldir         ;crt memory block move ala' LDIR
F036    C3 F097                 jp      getsel          ;return address of disk mapping table to HL
F039    C3 F086         dayti:  jp      daytim          ;return address of Time-of-Day
F03C    C3 F08B                 jp      config          ;return configuration status
F03F    C3 F105                 jp      siordy          ;sio channel b output ready status
F042    C3 F0A4                 jp      setcon          ;set configuration
F045    C3 F0BF                 jp      ssp             ;start screen print
F048    C3 F13F         usrsec: jp      nulint          ;user accessible 1 second interrupt
F04B    C3 F7A3                 jp      iocons          ;console status through iobyte
F04E    C3 F7AF                 jp      ioconi          ;console input  through iobyte
F051    C3 F796                 jp      iocono          ;console output through iobyte
F054    C3 F7BB                 jp      iolist          ;printer output through iobyte
F057    C3 F7CC                 jp      iolsts          ;printer status through iobyte
F05A    C3 F770                 jp      comins          ;communications input ready status
F05D    C3 F775                 jp      cominp          ;communications input data to A
F060    C3 F77F                 jp      comout          ;communications output data from C
F063    C3 F788                 jp      comots          ;communications output ready status
F066    C3 F13F         idle:   jp      nulint          ;idle while i/o is pending
F069    C3 F0D2         softv:  jp      soft            ;record soft error
F06C                            defs    16,-1           ;space for option rom linkage

                                subttl  Monitor Function Processors
                                page
```

```
                                ;;      Monitor Restart.
                                ;
F07C    F3              restart:di                      ;lock system
F07D    DB 1C                   in      a,(syspio)
F07F    F6 80                   or      1 shl 7         ;enable banked rom
F081    D3 1C                   out     (syspio),a
F083    C3 0000                 jp      prs             ;reload monitor from rom or ram

                                ;;      Daytim - Return Address of Time-of-Day.
                                ;
F086    11 FF56         daytim: ld      de,day          ;point to day of month
F089    18 0F                   jr      retval

                                ;;      Config - Return Configuration Status Byte.
                                ;
F08B    3A F0E3         config: ld      a,(mask)        ;turn keyboard mask into c.keym
F08E    E6 80                   and     080h
F090    F6 00                   or      0
F091                    confg   equ     $-1     ;*****=>;This word stored by Preset
F092    5F                      ld      e,a
F093    16 01                   ld      d,rev-400       ;return revision level
F095    18 03                   jr      retval

                                ;;      getsel - Get address of Select table.
                                ;
F097    11 F360         getsel: ld      de,Seltab       ;set select table address

                                ;;      Retval - Return Value to Caller.
                                ;
F09A    24              retval: inc     h               ;see if high level language call
F09B    25                      dec     h
F09C    28 03                   jr      z,retvl         ;if assembly level call
F09E    73                      ld      (hl),e          ;store answer in variable
F09F    23                      inc     hl
F0A0    72                      ld      (hl),d
F0A1    EB              retvl:  ex      de,hl           ;leave result in HL as well
F0A2    FB              eiret:  ei
F0A3    C9                      ret

                                ;;      setcon - set configuration.
                                ;
F0A4    7E              setcon: ld      a,(hl)          ;get configuration table index
F0A5    CB BF                   res     7,a
F0A7    FE 06                   cp      numcon
F0A9    D0                      ret     nc              ;if index out of range
F0AA    5F                      ld      e,a
F0AB    7E                      ld      a,(hl)          ;get read/write flag
F0AC    23                      inc     hl
F0AD    46                      ld      b,(hl)          ;get configuration data
F0AE    16 00                   ld      d,0
F0B0    21 FFBF                 ld      hl,contbl       ;set address of configuration table addresses
F0B3    19                      add     hl,de
F0B4    19                      add     hl,de
F0B5    5E                      ld      e,(hl)          ;get configurable byte address
```

```
  659     F0B6    23                              inc     hl
  660     F0B7    56                              ld      d,(hl)
  661     F0B8    EB                              ex      de,hl
  662     F0B9    CB 7F                           bit     7,a             ;set direction
  663     F0BB    7E                              ld      a,(hl)          ;get previous value
  664     F0BC    C8                              ret     z               ;if asking current configuration
  665     F0BD    70                              ld      (hl),b          ;store new configuration
  666     F0BE    C9                              ret
  667
  668                                     ;;      ssp - start screen print.
  669                                     ;
  670     F0BF    3E 67                   ssp:    ld      a,3+((24+1) shl 2)      ;start with cr/lf
  671     F0C1    32 F20E                         ld      (spact),a
  672     F0C4    AF                              xor     a
  673     F0C5    32 F224                         ld      (spcnt),a
  674     F0C8    3E 81                           ld      a,81h           ;start millisecond timer
  675     F0CA    D3 19                           out     (ctc1),a
  676     F0CC    C9                              ret
  677
  678                                             subttl  Console / Printer Drivers
  679                                             page
```

```
                                        above                   ;run this code upstairs
00CD"                           +       d&seg

                                ;;      kbdst - keyboard status.
                                ;
                                ;       Returns A =  0 if no char
                                ;               A = -1 if char available
                                ;
F0CD    3A FF30                 kbdst:  ld      a,(fifcnt)      ;get input fifo bytecount
F0D0    B7                              or      a
F0D1    C8                              ret     z               ;if keyboard queue is empty

                                ;;      soft - record soft error.
                                ;
F0D2    F6 FF                   soft:   or      -1              ;set ready / error status
F0D4    C9                              ret

                                ;;      kbdin - Keyboard Input.
                                ;
                                ;       Returns A = character
                                ;
F0D5    CD F066                 kbdin1: call    idle            ;idle cpu
F0D8    CD F0CD                 kbdin:  call    kbdst
F0DB    28 F8                           jr      z,kbdin1        ;loop until keyboard input ready
F0DD    E5                              push    hl
F0DE    CD F130                         call    remove          ;get keyboard entry
F0E1    E1                              pop     hl
F0E2    E6 7F                   kbmask: and     07fh
F0E3                            mask    equ     $-1     ;*****=>;this byte modified by ESC 0/1
F0E4    C9                              ret

                                ;;      siost - sio channel b input ready status.
                                ;
F0E5    DB 07                   siost:  in      a,(siocpb)      ;get sio status register
F0E7    E6 01                           and     00000001b
F0E9    C8                              ret     z               ;if no data available
F0EA    3E FF                           ld      a,-1
F0EC    C9                              ret

                                ;;      sioin - Sio channel b input character.
                                ;
F0ED    CD F066                 sioin1: call    idle            ;idle cpu
F0F0    CD F0E5                 sioin:  call    siost           ;test console status
F0F3    28 F8                           jr      z,sioin1        ;loop until data is
F0F5    DB 05                           in      a,(siodpb)      ;ready at sio data port
F0F7    C9                              ret

                                ;;      sioout - Sio channel B output character.
                                ;
F0F8    F5                      sioout: push    af
F0F9    CD F105                 siox1:  call    siordy
F0FC    CC F066                         call    z,idle          ;idle cpu if transmitter not ready
F0FF    28 F8                           jr      z,siox1
F101    F1                              pop     af
```

```
735     F102    D3 05                   out     (siodpb),a      ;output data to sio
736     F104    C9                      ret
737
738                             ;;      siordy - Sio channel B output ready ststus.
739                             ;
740     F105    3E 10           siordy: ld      a,10h           ;reset status latch
741     F107    D3 07                   out     (siocpb),a
742     F109    DB 07                   in      a,(siocpb)
743     F10B    E6 04                   and     00000100b       ;test tbe status bit
744     F10C                    siomsk  equ     $-1     ;*****=>;modified at run time
745     F10D    EE 04                   xor     00000100b
746     F10E                    sioval  equ     $-1     ;*****=>;modified at run time
747     F10F    28 02                   jr      z,siord1        ;if hardware is ready
748     F111    AF                      xor     a
749     F112    C9                      ret
750     F113    F6 FF           siord1: or      -1              ;set ready status
751     F115    00              xonenb: nop             ;*****=>;put RET here to disable Xon/Xoff
752     F116    CD F0E5                 call    siost
753     F119    28 11                   jr      z,siord3        ;if input not available
754     F11B    CD F0F0                 call    sioin
755     F11E    E6 7F                   and     7fh
756     F120    D6 13                   sub     Xoff
757     F122    28 05                   jr      z,siord2        ;if printer said Stop
758     F124    D6 FE                   sub     Xon-Xoff
759     F126    20 04                   jr      nz,siord3       ;if not Resume
760     F128    2F                      cpl                     ;set printer ready
761     F129    32 F12D         siord2: ld      (xofflg),a
762     F12C    3E FF           siord3: ld      a,-1
763     F12D                    xofflg  equ     $-1     ;*****=>;set ^S pending flag
764     F12E    B7                      or      a
765     F12F    C9                      ret
766
767                             ;;      Remove - remove key from fifo.
768                             ;
769     F130    21 FF30         remove: ld      hl,fifcnt       ;decrement fifo count
770     F133    35                      dec     (hl)
771     F134    21 FF32                 ld      hl,fifout       ;point hl to fifo output offset
772     F137    34              index:  inc     (hl)            ;advance fifo pointer
773     F138    CB A6                   res     4,(hl)          ;modulo 16
774     F13A    3E 20                   ld      a,low fifo
775     F13C    86                      add     a,(hl)          ;index into fifo by offset
776     F13D    6F                      ld      l,a
777     F13E    7E                      ld      a,(hl)          ;fetch character in fifo
778     F13F    C9              nulint: ret
779
780                                     subttl  Interrupt Service Routines
781                                     page
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Interrupt Service Routines

  782
  783                                   ;;      isr - interrupt service routines.
  784                                   ;
  785                                   service macro
  786                                           ld      (savstk),sp     ;;;save user stack pointer and
  787                                           ld      sp,intstk       ;;;switch to local stack
  788                                           push    hl              ;;;save machine state
  789                                           push    af
  790                                           endm
  791
  792                                   ;;      keysrv - parallel keyboard interrupt service.
  793                                   ;
  794     F140                          keysrv: service                 ;save state
  795     F140    ED 73 F1EC      +             ld      (savstk),sp
  796     F144    31 FF50         +             ld      sp,intstk
  797     F147    E5              +             push    hl
  798     F148    F5              +             push    af
  799     F149    C5                            push    bc
  800     F14A    DB 1E                         in      a,(kbddat)      ;read keyboard input port
  801     F14C    2F                            cpl
  802     F14D    FE 9E                         cp      Scrprt
  803     F14F    20 16                         jr      nz,key2         ;if not screen print key
  804     F151    3A F20E                       ld      a,(spact)
  805     F154    B7                            or      a
  806     F155    28 0B                         jr      z,key1          ;if screen not printing now
  807     F157    3E 07                         ld      a,3+(1 shl 2)   ;set state to cr/lf/stop
  808     F159    32 F20E                       ld      (spact),a       ;set screen print state
  809     F15C    AF                            xor     a
  810     F15D    32 F224                       ld      (spcnt),a       ;restart character counter
  811     F160    18 2D                         jr      key5
  812     F162    CD F0BF                 key1: call    ssp             ;start screen print
  813     F165    18 28                         jr      key5
  814     F167    21 FF30                 key2: ld      hl,fifcnt       ;bump input fifo character count
  815     F16A    FE 9B                         cp      Abort           ;check user Abort Key
  816     F16C    28 11                         jr      z,key3          ;warm start system
  817     F16E    4F                            ld      c,a
  818     F16F    7E                            ld      a,(hl)
  819     F170    3C                            inc     a
  820     F171    FE 10                         cp      16
  821     F173    30 1A                         jr      nc,key5         ;exit now if fifo is full
  822     F175    77                            ld      (hl),a
  823     F176    21 FF31                       ld      hl,fifin        ;point hl to fifo input offset
  824     F179    CD F137                       call    index
  825     F17C    71                            ld      (hl),c          ;store character in fifo @ hl
  826     F17D    18 10                         jr      key5
  827     F17F    CD F1EF                 key3: call    retins          ;release Pio interrupt controller
  828     F182    06 03                         ld      b,3
  829     F184    36 00                   key4: ld      (hl),0          ;clear fifo count
  830     F186    2C                            inc     l               ;and fifo in/out pointers
  831     F187    10 FB                         djnz    key4
  832     F189    CD F293                       call    crtoff          ;turn crt memory off
  833     F18C    CD F003                       call    warm            ;and warm start system
  834     F18F    C1                      key5: pop     bc
  835     F190    18 57                         jr      rfi             ;return from interrupt
  836
```

```
                                ;;      timer - Once per second interrupt.
                                ;
F192                            timer:  service
F192    ED 73 F1EC      +               ld      (savstk),sp
F196    31 FF50         +               ld      sp,intstk
F199    E5              +               push    hl
F19A    F5              +               push    af
F19B    2A FF52                         ld      hl,(tikcnt)     ;advance binary seconds counter
F19E    23                              inc     hl
F19F    22 FF52                         ld      (tikcnt),hl
F1A2    CD F048                         call    usrsec          ;invoke user's interrupt routine
F1A5    21 FF55                         ld      hl,timout       ;decrement disk turn-off timer
F1A8    35                              dec     (hl)
F1A9    20 06                           jr      nz,timer1       ;exit if not timed out yet
F1AB    DB 1C                           in      a,(syspio)
F1AD    E6 F8                           and     11111000b       ;disable all drive selects which
F1AF    D3 1C                           out     (syspio),a      ;turns off spindle motors
F1B1    C5                      timer1: push    bc
F1B2    06 02                           ld      b,2
F1B4    3E 3B                           ld      a,59
F1B6    21 FF5B                         ld      hl,secs         ;point at wall clock
F1B9    34                      timer2: inc     (hl)            ;increment seconds
F1BA    BE                              cp      (hl)
F1BB    30 2B                           jr      nc,timer3       ;if not one minute or hour
F1BD    36 00                           ld      (hl),0
F1BF    2B                              dec     hl
F1C0    10 F7                           djnz    timer2
F1C2    3E 17                           ld      a,23
F1C4    34                              inc     (hl)            ;increment hours
F1C5    BE                              cp      (hl)
F1C6    30 20                           jr      nc,timer3       ;if not one day
F1C8    36 00                           ld      (hl),0
F1CA    D5                              push    de
F1CB    2B                              dec     hl
F1CC    2B                              dec     hl
F1CD    4E                              ld      c,(hl)          ;get month
F1CE    2B                              dec     hl              ;point to day
F1CF    11 F1F0                         ld      de,dpm-1        ;point to day/month table
F1D2    EB                              ex      de,hl
F1D3    09                              add     hl,bc
F1D4    7E                              ld      a,(hl)          ;get number of days
F1D5    EB                              ex      de,hl
F1D6    D1                              pop     de
F1D7    34                              inc     (hl)            ;increment day
F1D8    BE                              cp      (hl)
F1D9    30 0D                           jr      nc,timer3       ;if not end of month
F1DB    36 01                           ld      (hl),1          ;reset day in month
F1DD    23                              inc     hl
F1DE    34                              inc     (hl)            ;increment month
F1DF    3E 0B                           ld      a,11
F1E1    BE                              cp      (hl)
F1E2    30 04                           jr      nc,timer3       ;if not new years eve
F1E4    36 01                           ld      (hl),1          ;wrap december to january
F1E6    23                              inc     hl
F1E7    34                              inc     (hl)            ;signal Guy Lombardo
F1E8    C1                      timer3: pop     bc
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Interrupt Service Routines

  893
  894                                   ;;      rfi - return From Interrupt.
  895                                   ;
  896   F1E9   F1                       rfi:    pop     af
  897   F1EA   E1                               pop     hl
  898   F1EB   31 0000                          ld      sp,0            ;restore stack
  899   F1EC                            savstk  equ     $-2     ;*****=>;this word modified at runtime
  900   F1EE   FB                               ei                      ;re-enable interrupts and return
  901   F1EF   ED 4D                    retins: reti
  902
  903                                   ;;      Table of days per month.
  904                                   ;
  905   F1F1   1F                       dpm:    db      31              ;january
  906   F1F2   1C                               db      28              ;febuary
  907   F1F3   1F                               db      31              ;march
  908   F1F4   1E                               db      30              ;april
  909   F1F5   1F                               db      31              ;may
  910   F1F6   1E                               db      30              ;june
  911   F1F7   1F                               db      31              ;july
  912   F1F8   1F                               db      31              ;august
  913   F1F9   1E                               db      30              ;september
  914   F1FA   1F                               db      31              ;october
  915   F1FB   1E                               db      30              ;november
  916   F1FC   1F                               db      31              ;december
  917
  918                                   ;;      milli - Millisecond timer interrupt service.
  919                                   ;
  920   F1FD                            milli:  service
  921   F1FD   ED 73 F1EC       +               ld      (savstk),sp
  922   F201   31 FF50          +               ld      sp,intstk
  923   F204   E5               +               push    hl
  924   F205   F5               +               push    af
  925   F206   2A FF50                          ld      hl,(Milsec)
  926   F209   23                               inc     hl              ;increment millisecond counter
  927   F20A   22 FF50                          ld      (Milsec),hl
  928   F20D   3E 00                            ld      a,0             ;set screen print flag
  929   F20E                            spact   equ     $-1     ;*****=>;this byte modified at runtime
  930   F20F   67                               ld      h,a
  931   F210   E6 03                            and     3
  932   F212   28 6D                            jr      z,mil6          ;if not printing screen
  933   F214   6F                               ld      l,a
  934   F215   CD F105                          call    siordy          ;get printer status
  935   F218   28 67                            jr      z,mil6          ;if printer not ready
  936   F21A   2D                               dec     l
  937   F21B   20 48                            jr      nz,mil2         ;if not character print state
  938   F21D   DB 1C                            in      a,(syspio)      ;get pio state
  939   F21F   F5                               push    af
  940   F220   CD F29C                          call    crton
  941   F223   3E 00                            ld      a,0             ;get character count
  942   F224                            spcnt   equ     $-1     ;*****=>;byte modified at runtime
  943   F225   3D                               dec     a
  944   F226   FA F236                          jp      m,mil0          ;if end of line
  945   F229   32 F224                          ld      (spcnt),a
  946   F22C   21 0000                          ld      hl,0            ;set next character address
  947   F22D                            spaddr  equ     $-2     ;*****=>;word modified at runtime
  948   F22F   7E                               ld      a,(hl)
```

```
 949     F230    D3 05                   out     (siodpb),a      ;fire hammer
 950     F232    2C                      inc     l               ;advance screen cursor
 951     F233    AF                      xor     a               ;do not advance state
 952     F234    18 22                   jr      mill1           ;if not end of line
 953     F236    3E 61           mil0:   ld      a,l+(24 shl 2)  ;set address of next print line
 954     F238    94                      sub     h
 955     F239    1F                      rra
 956     F23A    CB 2F                   sra     a
 957     F23C    CD F31E                 call    cca             ;compute cursor address
 958     F23F    E5                      push    hl              ;save next line address
 959     F240    C5                      push    bc
 960     F241    06 50                   ld      b,80            ;delete  trailing blanks
 961     F243    7D                      ld      a,l
 962     F244    80                      add     a,b
 963     F245    6F                      ld      l,a
 964     F246    2D              mil01:  dec     l
 965     F247    7E                      ld      a,(hl)          ;get next character
 966     F248    E6 7F                   and     7fh
 967     F24A    FE 20                   cp      ' '
 968     F24C    20 02                   jr      nz,mil02        ;if not trailing blank
 969     F24E    10 F6                   djnz    mil01
 970     F250    78              mil02:  ld      a,b
 971     F251    32 F224                 ld      (spcnt),a       ;set number of characters to print
 972     F254    C1                      pop     bc
 973     F255    E1                      pop     hl
 974     F256    3E 03                   ld      a,3             ;set CR next state
 975     F258    22 F22D         mill1:  ld      (spaddr),hl     ;set next display address
 976     F25B    21 F20E                 ld      hl,spact        ;set state variable
 977     F25E    B6                      or      (hl)            ;advance state
 978     F25F    77                      ld      (hl),a
 979     F260    F1                      pop     af              ;get pio back
 980     F261    D3 1C                   out     (syspio),a
 981     F263    18 1C                   jr      mill6
 982     F265    2D              mill2:  dec     l               ;check next state
 983     F266    21 F20E                 ld      hl,spact        ;set state address
 984     F269    20 11                   jr      nz,mill4        ;if not lf state
 985     F26B    7E                      ld      a,(hl)
 986     F26C    D6 04                   sub     1 shl 2         ;advance line counter
 987     F26E    77                      ld      (hl),a
 988     F26F    FE FE                   cp      2-(1 shl 2)
 989     F271    20 05                   jr      nz,mill3
 990     F273    3E 01                   ld      a,1             ;disable ctc interrupt
 991     F275    D3 19                   out     (ctc1),a
 992     F277    77                      ld      (hl),a
 993     F278    3E 0A           mill3:  ld      a,lf            ;set line feed
 994     F27A    18 02                   jr      mill5
 995     F27C    3E 0D           mill4:  ld      a,cr            ;set carriage return
 996     F27E    D3 05           mill5:  out     (siodpb),a      ;move paper or carriage
 997     F280    35                      dec     (hl)
 998     F281    C3 F1E9         mill6:  jp      rfi             ;return from interrupt
 999
1000                                    subttl  Crt Driver
1001                                    page
```

```
Balcones Operating System for the XEROX 820-II, MACRO-80 3.44   09-Dec-81
Crt Driver

 1002
 1003                                   ;;      setcur - set direct display cursor position.
 1004                                   ;
 1005    F284    22 FFAF                setcur: ld      (dircur),hl     ;set up cursor address
 1006    F287    C9                             ret
 1007
 1008                                   ;;      outcur - store character directly to crt memory.
 1009                                   ;
 1010    F288    CD F29C                outcur: call    crton           ;turn on crt bank
 1011    F28B    2A FFAF                        ld      hl,(dircur)     ;fetch direct cursor
 1012    F28E    71                             ld      (hl),c          ;store character
 1013    F28F    23                             inc     hl
 1014    F290    22 FFAF                        ld      (dircur),hl
 1015
 1016                                   ;;      crtoff - turn crt ram off.
 1017                                   ;
 1018    F293    F3                     crtoff: di                      ;lock pio access
 1019    F294    DB 1C                          in      a,(syspio)
 1020    F296    CB BF                  crtof1: res     7,a             ;reset crt bank enable
 1021    F298    FB                             ei                      ;unlock pio access
 1022    F299    D3 1C                  crton1: out     (syspio),a
 1023    F29B    C9                             ret
 1024
 1025                                   ;;      crton - turn crt ram on.
 1026                                   ;
 1027    F29C    F3                     crton:  di                      ;lock time-out interrupt
 1028    F29D    DB 1C                          in      a,(syspio)      ;get pio status
 1029    F29F    CB FF                          set     7,a             ;enable bank
 1030    F2A1    18 F6                          jr      crton1
 1031
 1032                                   ;;      block move from/to crt memory.
 1033                                   ;
 1034                                   ;       Entry:  HL = Source address
 1035                                   ;               DE = Destination address
 1036                                   ;               BC = Number of bytes to move
 1037                                   ;               A  = 0 - Move crt ram to crt ram
 1038                                   ;               A  < 0 - Move sys ram to crt ram
 1039                                   ;               A  > 0 - Move crt ram to sys ram
 1040                                   ;
 1041    F2A3    ED 73 F31B             crtldir:ld      (usrstk),sp     ;do not use callers stack
 1042    F2A7    31 FFE0                        ld      sp,crtstk       ;since it may disappear
 1043    F2AA    A7                             and     a               ;set entry conditions
 1044    F2AB    CD F29C                        call    crton
 1045    F2AE    28 37                          jr      z,crtmv         ;block move within crt ram
 1046    F2B0    F2 F2B5                        jp      p,ldir2         ;if move from crt ram to system ram
 1047    F2B3    EE 80                  ldir1:  xor     80h
 1048    F2B5    D3 1C                  ldir2:  out     (syspio),a      ;enable source bank
 1049    F2B7    E5                             push    hl              ;save move source address
 1050    F2B8    21 FFB0                        ld      hl,-80          ;count down one transfer buffer
 1051    F2BB    ED 4A                          adc     hl,bc
 1052    F2BD    E3                             ex      (sp),hl         ;save overflow, retrieve source address
 1053    F2BE    FA F2C4                        jp      m,ldir3         ;if less than one buffer
 1054    F2C1    01 0050                        ld      bc,80           ;transfer one buffer
 1055    F2C4    C5                     ldir3:  push    bc              ;save byte count
 1056    F2C5    D5                             push    de              ;save destination address
```

```
1057    F2C6  11 FF5C                    ld      de,linbuf       ;set upper buffer
1058    F2C9  ED B0                      ldir                    ;move data to upper ram
1059    F2CB  D1                         pop     de
1060    F2CC  C1                         pop     bc
1061    F2CD  DB 1C                      in      a,(syspio)      ;enable destination bank
1062    F2CF  EE 80                      xor     80h
1063    F2D1  D3 1C                      out     (syspio),a
1064    F2D3  E5                         push    hl              ;save source address
1065    F2D4  21 FF5C                    ld      hl,linbuf       ;set upper  buffer
1066    F2D7  ED B0                      ldir                    ;move data from buffer to destination
1067    F2D9  E1                         pop     hl
1068    F2DA  C1                         pop     bc              ;retrieve bytes left to transfer
1069    F2DB  78                         ld      a,b
1070    F2DC  A7                         and     a
1071    F2DD  FA F2E9                    jp      m,crtmvo        ;no more move, turn crt ram off and return
1072    F2E0  B1                         or      c
1073    F2E1  28 06                      jr      z,crtmvo        ;if no more
1074    F2E3  DB 1C                      in      a,(syspio)
1075    F2E5  18 CC                      jr      ldir1           ;continue transfer one buffer at a crack
1076
1077    F2E7  ED B0              crtmv:. ldir
1078    F2E9  CD F293            crtmvo: call    crtoff          ;turn crt ram off
1079    F2EC  ED 7B F31B                 ld      sp,(usrstk)
1080    F2F0  C9                         ret
1081
1082                                     subttl  Resident Crt Driver.
1083                                     page
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Resident Crt Driver.

                                ;;      crtout - Crt Output Driver.
                                ;
                                ;       Entry:  Character in register A
                                ;               16 bytes of stack space available
                                ;       Exit:   Char displayed, all registers saved
                                ;
F2F1    E5              crtout: push    hl              ;maintain users registers on his stack
F2F2    D5                      push    de
F2F3    C5                      push    bc
F2F4    F5                      push    af
F2F5    4F                      ld      c,a             ;set character to process
F2F6    CD F2FE                 call    fastcrt         ;process character quickly
F2F9    F1                      pop     af              ;restore callers registers
F2FA    C1                      pop     bc
F2FB    D1                      pop     de
F2FC    E1                      pop     hl
F2FD    C9                      ret

                                ;;      fastcrt - fast crt driver.
                                ;
                                ;       Entry:  Character in C
                                ;       Exit:   The only register preserved is SP
                                ;               Peeking in register A reveals valuable characters.
                                ;
F2FE    ED 73 F31B      fastcrt:ld      (usrstk),sp     ;do not use callers stack
F302    31 FFE0                 ld      sp,crtstk       ;since it may disappear
F305    DD E5                   push    ix
F307    DB 1C                   in      a,(syspio)      ;read system pio
F309    B7                      or      a               ;set bank enable status
F30A    F5                      push    af              ;save status for exit code
F30B    CD F29C                 call    crton           ;turn on crt memory
F30E    FB                      ei                      ;enable interrupts
F30F    CD 0169                 call    crtdvr          ;execute crt driver rom
F312    F1                      pop     af              ;get previous bank enable status
F313    F4 F293                 call    p,crtoff        ;disable bank now if it was disabled on entry
F316    DD E1                   pop     ix
F318    3E 00                   ld      a,0             ;sneak balcones golden characters to FAST users
F319                    gold    equ     $-1
F31A    31 F31B                 ld      sp,usrstk       ;restore callers stack
F31B                    usrstk  equ     $-2     ;*****=>;this operand word is modified at runtime
F31D    C9                      ret

                                ;;      cca - compute cursor address.
                                ;
                                ;       Entry:  A = Row
                                ;
F31E    67              cca:    ld      h,a
F31F    3A FFB1                 ld      a,(base)
F322    84                      add     a,h
F323    3C              cca1:   inc     a               ;entry with base absolute
F324    D6 18           cca2:   sub     24              ;ditto
F326    30 FC                   jr      nc,cca2
F328    C6 78                   add     a,24+2*crtbas
F32A    67                      ld      h,a
```

```
1139    F32B    2E 00                   ld      l,0
1140    F32D    CB 2C                   sra     h
1141    F32F    CB 1D                   rr      l
1142    F331    C9                      ret
1143
1144                            ;;      rstatt - Restore Previous Attribute.
1145                            ;
1146    F332    01 0000         rstatt: ld      bc,0            ;execute previous attribute routine
1147    F333                    lstatt  equ     $-2
1148    F335    C5                      push    bc
1149    F336    C9                      ret
1150
1151    F337    E5              setprv: push    hl
1152    F338    21 01CF                 ld      hl,setlow
1153    F339                    prvatt  equ     $-2
1154    F33B    22 F333                 ld      (lstatt),hl
1155    F33E    ED 43 F339              ld      (prvatt),bc
1156    F342    E1                      pop     hl
1157    F343    C9                      ret
1158
1159                                    subttl  Rom-resident Crt Driver
1160                                    page
```

```
                                        below                   ;execute in banked rom
0000!                           +       defs    comres
                                ;;      crtdvr - Crt Driver Proper.
                                ;
0169    2A FFAC                 crtdvr: ld      hl,(cursor)     ;set cursor address
016C    3A FFB4                         ld      a,(chrsav)      ;retrieve character under cursor
016F    77                              ld      (hl),a          ;replace character under cursor
0170    32 F319                         ld      (gold),a        ;bury balcones gold
0173    3A FFB2                         ld      a,(leadin)      ;set leadin state
0176    B7                              or      a
0177    20 1D                           jr      nz,crtd2        ;if processing escape sequence
0179    3A F0E3                         ld      a,(mask)        ;get keyboard mask
017C    A1                              and     c
017D    4F                              ld      c,a
017E    FE 20                           cp      ' '
0180    38 14                           jr      c,crtd2         ;if control code
0182    3A FFB3                 crtd1:  ld      a,(attrib)
0185    B1                              or      c
0186    77                              ld      (hl),a          ;store displayable character
0187    2C                              inc     l               ;advance pointer to next column
0188    7D                              ld      a,l
0189    E6 7F                           and     01111111b       ;extract column# from hl
018B    FE 50                           cp      80
018D    38 0A                           jr      c,crtd3         ;if end of line not reached
018F    AD                              xor     l
0190    6F                              ld      l,a             ;return cursor to left side
0191    CD 02F7                         call    lfeed           ;execute line feed
0194    18 03                           jr      crtd3
0196    CD 023D                 crtd2:  call    contrl          ;process control character
0199    22 FFAC                 crtd3:  ld      (cursor),hl     ;save cursor pointer for next time
019C    7E                              ld      a,(hl)          ;get character at new cursor location
019D    32 FFB4                         ld      (chrsav),a      ;save for next time 'CRTOUT' is called
01A0    3A FFAE                         ld      a,(csrchr)      ;get cursor character
01A3    FE 20                           cp      ' '
01A5    C8                              ret     z               ;if no cursor
01A6    4F                              ld      c,a
01A7    7E                              ld      a,(hl)
01A8    CB BF                           res     7,a
01AA    FE 20                           cp      ' '
01AC    79                              ld      a,c             ;set character used for cursor
01AD    28 03                           jr      z,crtd4         ;if character is a space
01AF    7E                              ld      a,(hl)          ;toggle attribute
01B0    EE 80                           xor     80h
01B2    77                      crtd4:  ld      (hl),a          ;store cursor character
01B3    C9                              ret

                                ;;      multi - Process multiple character escape sequence.
                                ;
01B4    EB                      multi:  ex      de,hl           ;unconditionally reset the lead-in
01B5    36 00                           ld      (hl),0          ;state to zero
01B7    EB                              ex      de,hl
01B8    3D                              dec     a
01B9    20 4E                           jr      nz,setxyl       ;if not initial state
```

```
1216    01BB    79                      ld      a,c             ;get second character of sequence
1217
1218                                    if      (options and o.esct) ne 0
1219                            ;;      Escape table search
1220                            ;
1221    01BC    E5                      push    hl
1222    01BD    21 028A                 ld      hl,esctab
1223    01C0    01 0011                 ld      bc,esctbl
1224    01C3    ED B1           search: cpir
1225    01C5    09                      add     hl,bc
1226    01C6    09                      add     hl,bc
1227    01C7    09                      add     hl,bc
1228    01C8    4E                      ld      c,(hl)
1229    01C9    23                      inc     hl
1230    01CA    46                      ld      b,(hl)
1231    01CB    E1                      pop     hl
1232    01CC    C0                      ret     nz
1233    01CD    C5                      push    bc
1234    01CE    C9                      ret
1235                                    endif                   ;options and o.esct
1236
1237                            ;;      Set attribute modes.
1238                            ;
1239    01CF    D3 36           setlow: out     (lowlite),a     ;set lo-light mode
1240    01D1    C3 F337                 jp      setprv
1241
1242    01D4    D3 35           setbli: out     (chrom2),a      ;select rom 2
1243    01D6    AF                      xor     a               ;select standard char set
1244    01D7    18 08                   jr      model
1245
1246    01D9    D3 35           setinv: out     (chrom2),a      ;select rom 2
1247    01DB    18 02                   jr      mode
1248
1249    01DD    D3 34           setgra: out     (chrom1),a      ;select rom 1
1250    01DF    3E 40           mode:   ld      a,40h           ;select alternate char set
1251    01E1    CD F337         model:  call    setprv          ;set up previous attribute
1252    01E4    47                      ld      b,a
1253    01E5    F3                      di                      ;lock system
1254    01E6    DB 1C                   in      a,(syspio)      ;read system pio
1255    01E8    CB B7                   res     6,a             ;clear rom select bit
1256    01EA    B0                      or      b
1257    01EB    FB                      ei                      ;unlock
1258    01EC    D3 1C                   out     (syspio),a      ;set or reset display mode
1259    01EE    C9                      ret
1260
1261                            ;;      Enable/disable (D7) display of selected attribute mode
1262                            ;
1263    01EF    3E 80           enatr:  ld      a,80h
1264    01F1    06                      db      6 ;ld b,        ;skip xor
1265    01F2    AF              disatr: xor     a
1266    01F3    32 FFB3         disl:   ld      (attrib),a
1267    01F6    C9                      ret
1268
1269                            ;;      setmsk - Select 7 or 8 bit data  from keyboard
1270                            ;
1271    01F7    0F              setmsk: rrca                    ;get low order bit as upper bit mask
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Rom-resident Crt Driver

 1272    01F8    F6 7F                          or      7fh
 1273    01FA    4F                             ld      c,a
 1274    01FB    11 F0E3                        ld      de,mask
 1275    01FE    1A                             ld      a,(de)
 1276    01FF    32 F319                        ld      (gold),a        ;stash balcones gold
 1277    0202    79                             ld      a,c
 1278    0203    12                             ld      (de),a          ;store keyboard mask
 1279    0204    C9                             ret
 1280
 1281                            ;;      Process cursor position sequence.
 1282                            ;
 1283    0205    3E 02           setxy:  ld      a,2
 1284    0207    12                              ld      (de),a          ;make leadin=2 next time
 1285    0208    C9                              ret
 1286
 1287    0209    3D              setxy1: dec     a
 1288    020A    20 10                           jr      nz,m3tst        ;if not in state 2
 1289    020C    6F                              ld      l,a             ;clear low cursor pos
 1290    020D    3E 03                           ld      a,3
 1291    020F    12                              ld      (de),a          ;set state 3 for next time
 1292    0210    79              setrow: ld      a,c
 1293    0211    E6 7F                           and     07fh            ;strip parity bit
 1294    0213    D6 20                           sub     ' '
 1295    0215    D8                              ret     c               ;if illegal character
 1296    0216    FE 18                           cp      24
 1297    0218    D0                              ret     nc
 1298    0219    C3 F31E                         jp      cca             ;compute cursor address
 1299
 1300    021C    3D              m3tst:  dec     a
 1301    021D    20 0C                           jr      nz,m4tst        ;if not ready for column
 1302    021F    79              setcol: ld      a,c
 1303    0220    E6 7F                           and     07fh            ;strip parity bit
 1304    0222    D6 20                           sub     ' '             ; of esc,'=',row,col sequence
 1305    0224    D8                              ret     c
 1306    0225    FE 50                           cp      80
 1307    0227    D0                              ret     nc
 1308    0228    B5                              or      l               ;merge in col# with l
 1309    0229    6F                              ld      l,a
 1310    022A    C9                              ret
 1311
 1312    022B    3D              m4tst:  dec     a
 1313    022C    20 04                           jr      nz,m5tst        ;if not escape state 4
 1314    022E    D1                              pop     de              ;pitch address of crtd3
 1315    022F    C3 0182                         jp      crtd1           ;display character in C
 1316
 1317    0232    3A FFAE         m5tst:  ld      a,(csrchr)
 1318    0235    32 F319                         ld      (gold),a        ;stash balcones gold
 1319    0238    79                              ld      a,c
 1320    0239    32 FFAE                         ld      (csrchr),a      ;store new cursor character
 1321    023C    C9                              ret
 1322
 1323                            ;;      contrl - process control character.
 1324                            ;
 1325    023D    11 FFB2         contrl: ld      de,leadin       ;point at leadin state
 1326    0240    D2 01B4                 jp      nc,multi        ;if multi code sequence in progress
 1327    0243    FE 05                   cp      'E'-64
```

```
1328    0245  D8                        ret     c               ;control char is out of range
1329    0246  E5                        push    hl              ;save cursor
1330    0247  21 024A                   ld      hl,ctltab-('E'-64)*2
1331    024A  06 00                     ld      b,0
1332    024C  09                        add     hl,bc           ;index through control character table
1333    024D  09                        add     hl,bc
1334    024E  4E                        ld      c,(hl)
1335    024F  23                        inc     hl
1336    0250  46                        ld      b,(hl)          ;get address of control subroutine
1337    0251  E1                        pop     hl
1338    0252  C5                        push    bc
1339    0253  C9                        ret                     ;execute control code driver
1340
1341                                    if      (options and o.esct) ne 0
1342    0254  02C5              ctltab: defw    defcur          ;Ctrl-e is define new cursor character
1343    0256  F332                      defw    rstatt          ;Ctrl-f is restore previous attribute mode
1344    0258  032F                      defw    bell            ;Ctrl-g is the bell
1345    025A  02CE                      defw    bakspc          ;Ctrl-h is cursor left
1346    025C  031F                      defw    tab             ;Ctrl-i is tab
1347    025E  02F7                      defw    lfeed           ;Ctrl-j is cursor down
1348    0260  02DC                      defw    upcsr           ;Ctrl-k is cursor up
1349    0262  02D4                      defw    forspc          ;Ctrl-l is cursor right
1350    0264  02F2                      defw    return          ;Ctrl-m is carriage return
1351    0266  02C0                      defw    nono            ;Ctrl-n is not acceptable
1352    0268  02C0                      defw    nono            ;Ctrl-o is not acceptable
1353    026A  02C0                      defw    nono            ;Ctrl-p is not acceptable
1354    026C  0361                      defw    clreos          ;Ctrl-q is clear to end-of-screen
1355    026E  02C0                      defw    nono            ;Ctrl-r is not acceptable
1356    0270  02C0                      defw    nono            ;Ctrl-s is not acceptable
1357    0272  02C0                      defw    nono            ;Ctrl-t is not acceptable
1358    0274  02C0                      defw    nono            ;Ctrl-u is not acceptable
1359    0276  02C0                      defw    nono            ;Ctrl-v is not acceptable
1360    0278  02C0                      defw    nono            ;Ctrl-w is not acceptable
1361    027A  0344                      defw    clreol          ;Ctrl-x is clear to end-of-line
1362    027C  02C0                      defw    nono            ;Ctrl-y is not acceptable
1363    027E  0357                      defw    clrscn          ;Ctrl-z is clear screen
1364    0280  02BD                      defw    escape          ;Ctrl-[ is escape
1365    0282  02C0                      defw    nono            ;Ctrl-\ is not acceptable
1366    0284  02C0                      defw    nono            ;Ctrl-] is not acceptable
1367    0286  02C9                      defw    homeup          ;Ctrl-^ is home up
1368    0288  02C1                      defw    stuff           ;Ctrl-_ is display control chars
1369
1370    0036                    ctlsiz  equ     $-ctltab
1371
1372                            ;;      Escape sequence table.
1373                            ;
1374                            ;       Maintains functional compatibility with terminals supporting
1375                            ;       ADM-3a style supersets.
1376                            ;
1377    028A  28                esctab: db      '('             ;disable attribute
1378    028B  29                        db      ')'             ;enable atribute
1379    028C  2A                        db      '*'             ;clear screen
1380    028D  30                        db      '0'             ;strip keyboard upper bit
1381    028E  31                        db      '1'             ;pass keyboard upper bit
1382    028F  34                        db      '4'             ;char font and blinking
1383    0290  35                        db      '5'             ;char font and graphics
```

```
1384  0291  36                        db      '6'             ;char font and blinking
1385  0292  37                        db      '7'             ;char font and inverse video
1386  0293  38                        db      '8'             ;char font and lo-light
1387  0294  3D                        db      '='             ;position cursor
1388  0295  45                        db      'E'             ;line insert
1389  0296  51                        db      'Q'             ;character insert
1390  0297  52                        db      'R'             ;line delete
1391  0298  57                        db      'W'             ;character delete
1392  0299  54                        db      'T'             ;clear to end of line
1393  029A  59                        db      'Y'             ;clear to end of screen
1394  0011                    esctbl  equ     $-esctab
1395
1396  029B  0361              escadr: defw    clreos
1397  029D  0344                      defw    clreol
1398  029F  03F5                      defw    chrdel
1399  02A1  037C                      defw    lindel
1400  02A3  03DC                      defw    chrins
1401  02A5  03A4                      defw    linins
1402  02A7  0205                      defw    setxy
1403  02A9  01CF                      defw    setlow
1404  02AB  01D9                      defw    setinv
1405  02AD  01D4                      defw    setbli
1406  02AF  01DD                      defw    setgra
1407  02B1  01D4                      defw    setbli
1408  02B3  01F7                      defw    setmsk
1409  02B5  01F7                      defw    setmsk
1410  02B7  0357                      defw    clrscn
1411  02B9  01EF                      defw    enatr
1412  02BB  01F2                      defw    disatr
1413                                  ifl
1414                                  if      ($-escadr)/2 ne esctbl
1415                                  .printx Escape table mismatch
1416                                  endif
1417                                  endif
1418                                  endif                   ;o.esct and options
1419
1420                    ;;            escape - Initialize escape sequence.
1421                    ;
1422  02BD  3E 01       escape: ld      a,1
1423  02BF  12                  ld      (de),a          ;set sequence state
1424  02C0  C9                nono:   ret                     ;for escape processing
1425
1426                    ;;            stuff - Enable next char to be stored directly.
1427                    ;
1428  02C1  3E 04       stuff:  ld      a,4
1429  02C3  12                  ld      (de),a          ;set sequence state
1430  02C4  C9                  ret                     ;for control char output mode
1431
1432                    ;;            defcur - Enable next chara to be new cursor.
1433                    ;
1434  02C5  3E 05       defcur: ld      a,5
1435  02C7  12                  ld      (de),a
1436  02C8  C9                  ret
1437
1438                    ;;            homeup - Move cursor to upper left.
1439                    ;
```

```
1440    02C9    0E 20           homeup: ld      c,' '           ;use cursor addressing routine
1441    02CB    C3 0210                 jp      setrow          ;to do homeup almost for free
1442
1443                            ;;      bakspc - Move cursor left.
1444                            ;
1445    02CE    7D              bakspc: ld      a,l             ;check for left margin
1446    02CF    E6 7F                   and     01111111b
1447    02D1    C8                      ret     z               ;abort if in leftmost column
1448    02D2    2D                      dec     l               ;back up cursor pointer
1449    02D3    C9                      ret
1450
1451                            ;;      forspc - Move cursor right.
1452                            ;
1453    02D4    7D              forspc: ld      a,l             ;check for rightmost colunm
1454    02D5    E6 7F                   and     01111111b
1455    02D7    FE 4F                   cp      79
1456    02D9    D0                      ret     nc              ;do nothing if already there
1457    02DA    2C                      inc     l
1458    02DB    C9                      ret                     ;else advance the cursor pointer
1459
1460                            ;;      upscr - Move cursor up.
1461                            ;
1462    02DC    11 FF80         upcsr:  ld      de,-128         ;subtract 1 from row# component
1463    02DF    19                      add     hl,de           ; of cursor pointer in hl
1464    02E0    7C                      ld      a,h
1465    02E1    FE 30                   cp      crtbas          ;check for underflow of pointer
1466    02E3    D0                      ret     nc
1467    02E4    26 3B                   ld      h,crttop-1      ;wrap cursor around modulo 3k
1468    02E6    C9                      ret
1469
1470                            ;;      dncsr - Move cursor down.
1471                            ;
1472    02E7    11 0080         dncsr:  ld      de,128          ;add 1 to row# component
1473    02EA    19                      add     hl,de           ; of cursor pointer in hl
1474    02EB    7C                      ld      a,h
1475    02EC    FE 3C                   cp      crttop          ;check for overflow of pointer
1476    02EE    D8                      ret     c
1477    02EF    26 30                   ld      h,crtbas        ;reset pointer modulo 128*24
1478    02F1    C9                      ret
1479
1480                            ;;      return - Move cursor to left side.
1481                            ;
1482    02F2    7D              return: ld      a,l             ;clear column
1483    02F3    E6 80                   and     10000000b
1484    02F5    6F                      ld      l,a             ;move cursor pointer back
1485    02F6    C9                      ret                     ; to start of line
1486
1487                            ;;      lfeed - Move cursor down with scroll.
1488                            ;
1489    02F7    7D              lfeed:  ld      a,l
1490    02F8    17                      rla
1491    02F9    7C                      ld      a,h
1492    02FA    17                      rla                     ;extract row# component of hl
1493    02FB    E6 1F                   and     00011111b
1494    02FD    4F                      ld      c,a             ;copy row# into c for scroll test
1495    02FE    CD 02E7                 call    dncsr           ;move cursor to next row down
```

```
1496  0301  3A FFB1                  ld      a,(base)        ;test if cursor was on bottom row
1497  0304  91                       sub     c               ;of screen before moving down
1498  0305  32 F319                  ld      (gold),a        ;send scroll flag back to fast users
1499  0308  C0                       ret     nz              ;exit if not at bottom
1500  0309  E5                       push    hl              ;prepare scroll screen up
1501  030A  3A F31C                  ld      a,(usrstk+1)
1502  030D  3C                       inc     a
1503  030E  C4 040F                  call    nz,bbg          ;bury balcones gold, unless monitor running
1504  0311  CD 0341                  call    clrlin          ;fill top line with spaces
1505  0314  29                       add     hl,hl
1506  0315  7C                       ld      a,h             ;get row# component of hl into a
1507  0316  E6 1F                    and     00011111b
1508  0318  32 FFB1                  ld      (base),a        ;store new base line#
1509  031B  D3 14                    out     (scroll),a      ;scroll top line to bottom
1510  031D  E1                       pop     hl              ;restore cursor
1511  031E  C9                       ret
1512
1513                          ;;      tab - Move cursor to next tab stop.
1514                          ;
1515  031F  11 0008           tab:    ld      de,8            ;tabs are every 8 columns
1516  0322  7D                       ld      a,l             ;get column component of
1517  0323  E6 78                    and     01111000b       ;previous tab position
1518  0325  83                       add     a,e
1519  0326  FE 50                    cp      80
1520  0328  D0                       ret     nc              ;if next tab column past end of line
1521  0329  7D                       ld      a,l
1522  032A  E6 F8                    and     11111000b       ;advance cursor to next tab stop
1523  032C  6F                       ld      l,a
1524  032D  19                       add     hl,de
1525  032E  C9                       ret
1526
1527                          ;;      bell - Move speaker back and forth.
1528                          ;
1529  032F  3E 35             bell:   ld      a,bltim         ;Bell time constant
1530  0331  D3 29             bell1:  out     (bellon),a      ;push speaker out
1531  0333  06 61                    ld      b,blonc         ;set Bell on time constant
1532  0335  10 FE                    djnz    $               ;pause B*2 micro seconds
1533  0337  D3 28                    out     (bellof),a      ;yank speaker in
1534  0339  06 61                    ld      b,blofc         ;set Bell off time constant
1535  033B  10 FE                    djnz    $
1536  033D  3D                       dec     a
1537  033E  20 F1                    jr      nz,bell1        ;if more noise to make
1538  0340  C9                       ret
1539
1540                          ;;      clrlin - Clear line.
1541                          ;
1542  0341  CD 02F2           clrlin: call    return          ;return cursor and fall through clreol
1543
1544                          ;;      clreol - Clear to end of line.
1545                          ;
1546  0344  7D                clreol: ld      a,l
1547  0345  E6 7F                    and     01111111b       ;get column component of cursor posistion
1548  0347  FE 50                    cp      80
1549  0349  D0                       ret     nc              ;if someone busted curpos
1550  034A  ED 44                    neg
1551  034C  C6 50                    add     a,80            ;calculate number of characters to clear
```

```
1552  034E  47                        ld      b,a
1553  034F  7D                        ld      a,l             ;save cursor column
1554  0350  36 20               clr1:  ld      (hl),' '        ;clear next location
1555  0352  2C                        inc     l
1556  0353  10 FB                     djnz    clr1            ;if end of line not clear
1557  0355  6F                        ld      l,a             ;restore cursor column
1558  0356  C9                        ret
1559
1560                            ;;      clrscn - clear visible screen memory.
1561                            ;
1562  0357  21 3000             clrscn: ld      hl,crtmem       ;home cursor
1563  035A  3E 17                      ld      a,23
1564  035C  32 FFB1                    ld      (base),a        ;put line 23 at bottom of screen
1565  035F  D3 14                      out     (scroll),a      ;note scroll register gets A8-A12, not d0-d7
1566
1567                            ;;      clreos - clear to end of screen.
1568                            ;
1569  0361  CD 0344             clreos: call    clreol          ;clear remainder of current row
1570  0364  E5                        push    hl              ;save cursor location
1571  0365  ED 4B FFB1          clrs1:  ld      bc,(base)       ;set bottom screen row to c
1572  0369  7D                        ld      a,l
1573  036A  17                        rla
1574  036B  7C                        ld      a,h
1575  036C  17                        rla                     ;get row# component of hl into a
1576  036D  E6 1F                     and     00011111b
1577  036F  B9                        cp      c
1578  0370  28 08                     jr      z,clrs2         ;if hl is on bottom row of screen
1579  0372  CD 02E7                   call    dncsr           ;point hl to next row
1580  0375  CD 0341                   call    clrlin          ;and fill that line with spaces
1581  0378  18 EB                     jr      clrs1
1582  037A  E1                  clrs2:  pop     hl              ;restore original cursor pointer
1583  037B  C9                        ret
1584
1585                            ;;      lindel - Line delete.
1586                            ;
1587  037C  E5                  lindel: push    hl              ;save cursor address
1588  037D  CD 040F                   call    bbg             ;bury balcones gold
1589  0380  29                        add     hl,hl
1590  0381  7C                        ld      a,h
1591  0382  E6 1F                     and     00011111b       ;extract row
1592  0384  ED 4B FFB0                ld      bc,(base-1)     ;get base screen row in b
1593  0388  CD 03D1             lind1:  call    smp             ;set move parameters
1594  038B  B8                        cp      b
1595  038C  28 10                     jr      z,lind2         ;if last line
1596  038E  C5                        push    bc              ;b=last line, c=row
1597  038F  01 0050                   ld      bc,80
1598  0392  ED B0                     ldir
1599  0394  C1                        pop     bc
1600  0395  79                        ld      a,c
1601  0396  3C                        inc     a
1602  0397  FE 18                     cp      24
1603  0399  38 ED                     jr      c,lind1
1604  039B  AF                        xor     a               ;wrap
1605  039C  18 EA                     jr      lind1           ;move next line
1606  039E  EB                  lind2:  ex      de,hl
1607  039F  CD 0341             lind3:  call    clrlin
```

```
1608    03A2    E1                      pop     hl              ;restore original cursor
1609    03A3    C9                      ret
1610
1611                            ;;      linins - Line insert.
1612                            ;
1613    03A4    E5              linins: push    hl              ;save cursor position
1614    03A5    3E 17                   ld      a,23            ;get bottom line
1615    03A7    CD F31E                 call    cca             ;compute cursor address
1616    03AA    CD 040F                 call    bbg             ;bury balcones gold
1617    03AD    E1                      pop     hl
1618    03AE    E5                      push    hl
1619    03AF    29                      add     hl,hl
1620    03B0    7C                      ld      a,h
1621    03B1    E6 1F                   and     00011111b       ;extract cursor row
1622    03B3    47                      ld      b,a
1623    03B4    3A FFB1                 ld      a,(base)        ;set last line
1624    03B7    B8              lini1:  cp      b
1625    03B8    28 13                   jr      z,lini3         ;if move complete
1626    03BA    3D                      dec     a
1627    03BB    F2 03C0                 jp      p,lini2
1628    03BE    3E 17                   ld      a,23
1629    03C0    CD 03D1         lini2:  call    smp             ;set move parameters
1630    03C3    EB                      ex      de,hl
1631    03C4    C5                      push    bc
1632    03C5    01 0050                 ld      bc,80
1633    03C8    ED B0                   ldir
1634    03CA    C1                      pop     bc
1635    03CB    18 EA                   jr      lini1           ;move next line
1636    03CD    E1              lini3:  pop     hl
1637    03CE    E5                      push    hl              ;restore cursor
1638    03CF    18 CE                   jr      lind3           ;clear cursor line
1639
1640                            ;;      smp - Set move parameters.
1641                            ;
1642    03D1    4F              smp:    ld      c,a             ;save row
1643    03D2    CD F324                 call    cca2
1644    03D5    EB                      ex      de,hl
1645    03D6    79                      ld      a,c
1646    03D7    CD F323                 call    cca1
1647    03DA    79                      ld      a,c
1648    03DB    C9                      ret
1649
1650                            ;;      chrins - Character insert.
1651                            ;
1652    03DC    E5              chrins: push    hl
1653    03DD    7D                      ld      a,l             ;set cursor column
1654    03DE    E6 7F                   and     01111111b       ;set move length = 79-column
1655    03E0    ED 44                   neg
1656    03E2    C6 4F                   add     a,79
1657    03E4    47                      ld      b,a             ;number of chars to move
1658    03E5    7E                      ld      a,(hl)          ;get char under cursor
1659    03E6    36 20                   ld      (hl),' '        ;clear char under cursor
1660    03E8    28 06                   jr      z,chrin2        ;if cursor in last column
1661    03EA    2C              chrin1: inc     l
1662    03EB    4E                      ld      c,(hl)
1663    03EC    77                      ld      (hl),a
```

```
1664    03ED    79                      ld      a,c
1665    03EE    10 FA                   djnz    chrin1
1666    03F0    32 F319         chrin2: ld      (gold),a        ;shift line into the gold mine
1667    03F3    E1                      pop     hl
1668    03F4    C9                      ret
1669
1670                            ;;      chrdel - Character delete.
1671                            ;
1672    03F5    E5              chrdel: push    hl
1673    03F6    7D                      ld      a,l
1674    03F7    E6 7F                   and     01111111b       ;isolate cursor column
1675    03F9    ED 44                   neg
1676    03FB    C6 4F                   add     a,79
1677    03FD    4F                      ld      c,a             ;number of chars to move = 79-column
1678    03FE    06 00                   ld      b,0
1679    0400    54                      ld      d,h
1680    0401    5D                      ld      e,l
1681    0402    1A                      ld      a,(de)
1682    0403    32 F319                 ld      (gold),a        ;mine balcones gold
1683    0406    23                      inc     hl
1684    0407    C4 0418                 call    nz,ldirx
1685    040A    EB                      ex      de,hl
1686    040B    36 20                   ld      (hl),' '        ;blank last char on line
1687    040D    E1                      pop     hl              ;restore cursor
1688    040E    C9                      ret
1689
1690                            ;;      bbg - bury balcones gold.
1691                            ;
1692    040F    CD 02F2         bbg:    call    return
1693    0412    11 FF5C                 ld      de,linbuf
1694    0415    01 0050                 ld      bc,80
1695    0418    ED B0           ldirx:  ldir
1696    041A    C9                      ret
1697
1698                                    subttl  Logical to Physical Driver Executioner
1699                                    page
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Logical to Physical Driver Executioner -

                                        above                   ;code goes in ram
0518!                           +       d&seg

                                ;;      Xqdvr - Execute Physical Driver.
                                ;
                                ;       Entry:  HL = Pointer to Physical Drive Request Block
                                ;
                                ;
F344    23                      xqdvr:  inc     hl              ;point at physical unit
F345    E5                              push    hl
F346    23                              inc     hl
F347    5E                              ld      e,(hl)          ;set logical drive
F348    21 F360                         ld      hl,Seltab       ;Set driver mapping table address
F34B    16 00                           ld      d,0
F34D    19                              add     hl,de           ;index into driver select table
F34E    19                              add     hl,de
F34F    5E                              ld      e,(hl)          ;set physical driver index
F350    23                              inc     hl
F351    7E                              ld      a,(hl)          ;set physical unit
F352    21 F380                         ld      hl,Drvtab       ;set Driver table address
F355    19                              add     hl,de
F356    19                              add     hl,de
F357    5E                              ld      e,(hl)          ;set physical driver address
F358    23                              inc     hl
F359    56                              ld      d,(hl)
F35A    E1                              pop     hl              ;recover request block address
F35B    77                              ld      (hl),a          ;store physical unit
F35C    2B                              dec     hl
F35D    D5                              push    de              ;execute physical driver
F35E    C9                              ret

                                        subttl  Physical Disk Driver Area
                                        page
```

```
                                ;;      Physical Disk Driver Area.
                                ;
                                        above
035F                    +               d&seg
                                ;;      Waste space to get drivers on page boundry.
                                ;
F35F                            Wasted:
000F                            bndry   equ     0fh
                                        if      ($ and bndry) ne 0
F35F                                    defs    bndry+1-($ and bndry),-1
                                        endif

                                ;;      Logical to Physical Drive Mapping Tables.
                                ;
                                ;       Seltab contains two bytes per logical CP/M drive A-P.
                                ;       The first byte is an index into the physical driver
                                ;       address table below.  The second byte is a unit number
                                ;       that is  passed to the driver by the standard deblocker.
                                ;
F360                            Seltab:

F360    01 00                   .A:     defb    1,0             ;Floppy Unit 0
F362    01 01                   .B:     defb    1,1             ;Floppy Unit 1
F364    01 02                   .C:     defb    1,2             ;Floppy Unit 2
F366    01 03                   .D:     defb    1,3             ;Floppy Unit 3

F368    01 04                   .E:     defb    1,4             ;Rigid Partition 0
F36A    01 05                   .F:     defb    1,5             ;Rigid Partition 1
F36C    01 06                   .G:     defb    1,6             ;Rigid Partition 2
F36E    01 07                   .H:     defb    1,7             ;Rigid Partition 3

F370    00 00                   .I:     defb    0,0             ;Error Driver
F372    00 00                   .J:     defb    0,0             ;Error Driver
F374    00 00                   .K:     defb    0,0             ;Error Driver
F376    00 00                   .L:     defb    0,0             ;Error Driver

F378    00 00                   .M:     defb    0,0             ;Error Driver
F37A    00 00                   .N:     defb    0,0             ;Error Driver
F37C    00 00                   .O:     defb    0,0             ;Error Driver
F37E    00 00                   .P:     defb    0,0             ;Error Driver

                                ;;      Physical Driver Address Table.
                                ;
                                ;       Drvtab contains the addresses of several independent
                                ;       physical disk drivers.  By convention, driver # 0 always
                                ;       returns a select error.
                                ;
F380    F42A                    Drvtab: defw    Selerr          ;select error physical driver
F382    F4B0                            defw    Dskdvr          ;Disk driver (WD or SA)
F384    0000                            defw    0               ;Empty physical driver expansion slots
F386    0000                            defw    0
F388    0000                            defw    0
F38A    0000                            defw    0
F38C    0000                            defw    0
```

```
1789    F38E    FFFF                    defw    0-1             ;mark last empty expansion slot
1790
1791                            ;;      Overlaid Monitor Ram Address Definitions.
1792                            ;
1793    FC80                    dirbuf  equ     0fc80h          ;director buffer
1794    FD00                    chk00   equ     0fd00h          ;Directory Check Vector for Floppy Drive 0
1795    FD20                    chk01   equ     0fd20h          ;Directory Check Vector for Floppy Drive 1
1796    FD40                    chk02   equ     0fd40h          ;Directory Check Vector for Floppy Drive 2
1797    FD60                    chk03   equ     0fd60h          ;Directory Check Vector for Floppy Drive 3
1798    0000                    chk04   equ     0               ;No Check Vector for Rigid Partition 0
1799    0000                    chk05   equ     0               ;No Check Vector for Rigid Partition 1
1800    0000                    chk06   equ     0               ;No Check Vector for Rigid Partition 2
1801    0000                    chk07   equ     0               ;No Check Vector for Rigid Partition 3
1802
1803    FD80                    all00   equ     0fd80h          ;Floppy Drive 0 Allocation Vector
1804    FDA0                    all01   equ     0fda0h          ;Floppy Drive 1 Allocation Vector
1805    FDC0                    all02   equ     0fdc0h          ;Floppy Drive 2 Allocation Vector
1806    FDE0                    all03   equ     0fde0h          ;Floppy Drive 3 Allocation Vector
1807    FE00                    all04   equ     0fe00h          ;Rigid Partition Allocation vectors
1808    FE80                    all05   equ     0fe80h
1809    FEC0                    all06   equ     0fec0h
1810    FEE0                    all07   equ     0fee0h
1811
1812                            subttl  Disk Parameter Headers
1813                            page
```

```
                                ;;      Disk Parameter Headers.
                                ;
F390    0000 0000       Dpbase: dw      0,0,0,0         ;Floppy Drive 0
F394    0000 0000
F398    FC80 0000               dw      dirbuf,0
F39C    FD00 FD80               dw      chk00,all00

F3A0    0000 0000               dw      0,0,0,0         ;Floppy Drive 1
F3A4    0000 0000
F3A8    FC80 0000               dw      dirbuf,0
F3AC    FD20 FDA0               dw      chk01,all01

F3B0    0000 0000               dw      0,0,0,0         ;Floppy Drive 2
F3B4    0000 0000
F3B8    FC80 0000               dw      dirbuf,0
F3BC    FD40 FDC0               dw      chk02,all02

F3C0    0000 0000               dw      0,0,0,0         ;Floppy Drive 3
F3C4    0000 0000
F3C8    FC80 0000               dw      dirbuf,0
F3CC    FD60 FDE0               dw      chk03,all03

F3D0    0000 0000               dw      0,0,0,0         ;Rigid Partition 0
F3D4    0000 0000
F3D8    FC80 F470               dw      dirbuf,Dpbrg4
F3DC    0000 FE00               dw      chk04,all04

F3E0    0000 0000               dw      0,0,0,0         ;Rigid Partition 1
F3E4    0000 0000
F3E8    FC80 F480               dw      dirbuf,Dpbrg5
F3EC    0000 FE80               dw      chk05,all05

F3F0    0000 0000               dw      0,0,0,0         ;Rigid Partition 2
F3F4    0000 0000
F3F8    FC80 F490               dw      dirbuf,Dpbrg6
F3FC    0000 FEC0               dw      chk06,all06

F400    0000 0000               dw      0,0,0,0         ;Rigid Partition 3
F404    0000 0000
F408    FC80 F4A0               dw      dirbuf,Dpbrg7
F40C    0000 FEE0               dw      chk07,all07

                                subttl  Sector Translate Tables
                                page
```

```
1859
1860                                    ;;      Sector Translation Tables.
1861                                    ;
1862                                    ;       For 8 inch single density drives.
1863                                    ;       Skew by 6
1864                                    ;
1865    F410    01 07 0D 13             trn6:   db      01,07,13,19
1866    F414    19 05 0B 11                     db      25,05,11,17
1867    F418    17 03 09 0F                     db      23,03,09,15
1868    F41C    15 02 08 0E                     db      21,02,08,14
1869    F420    14 1A 06 0C                     db      20,26,06,12
1870    F424    12 18 04 0A                     db      18,24,04,10
1871    F428    10 16                           db      16,22
1872
1873    F42A    21 0000                 selerr: ld      hl,0
1874    F42D    F6 FF                           or      -1
1875    F42F    C9                              ret
1876
1877                                            subttl  Floppy Disk Parameter Blocks
1878                                            page
```

```
1879
1880                                    ;;      Floppy Disk Parameter Blocks, one per media format.
1881                                    ;
1882    F430                            dpb8s:
1883                                    ;       Single Density, Single side
1884
1885    F430    001A                            dw      26              ;spt
1886    F432    03 07 00                        db      3,7,0           ;blkshf, blkmsk, nullmsk
1887    F435    00F2 003F                       dw      242,63,192,16,2 ;dsw,dirm,alloc01,chksiz,trk off
1888    F439    00C0 0010
1889    F43D    0002
1890    F43F    00                              db      0               ;128 byte sectors
1891
1892                                    ;       Single Density, Double Side
1893
1894    F440    001A                            dw      26              ;spt
1895    F442    04 0F 01                        db      4,15,1          ;blkshf, blkmsk, nullmsk
1896    F445    00F6 007F                       dw      246,127,192,16,2;dsw,dirm,alloc01,chksiz,trk off
1897    F449    00C0 0010
1898    F44D    0002
1899    F44F    00                              db      0               ;128 byte sectors
1900
1901    F450                            dpb8d:
1902                                    ;       Double Density, Single Side
1903
1904    F450    0034                            dw      2*26            ;spt
1905    F452    04 0F 01                        db      4,15,1          ;blkshf, blkmsk, nullmsk
1906    F455    00F2 007F                       dw      242,127,192,32,2;dsw,dirm,alloc01,chksiz,trk off
1907    F459    00C0 0020
1908    F45D    0002
1909    F45F    81                              db      81h             ;256 byte sectors, track zero single density
1910
1911                                    ;       Double Density, Double Side
1912
1913    F460    0034                            dw      2*26            ;spt
1914    F462    05 1F 03                        db      5,31,3          ;blkshf, blkmsk, nullmsk
1915    F465    00F6 007F                       dw      246,127,192,32,2;dsw,dirm,alloc01,chksiz,trk off
1916    F469    00C0 0020
1917    F46D    0002
1918    F46F    81                              db      81h             ;256 byte sectors, track zero single density
1919
1920                                    subttl  Micro Floppy Disk Parameter Blocks
1921                                    page
```

```
                                ;       Single Density, Single Side
F470    0012                            dw      18              ;spt
F472    03 07 00                        db      3,7,0           ;blkshf, blkmsk, nulmsk
F475    0052 001F                       dw      82,31,128,8,3   ;dsw,dirm,alloc01,chksiz,trk off
F479    0080 0008
F47D    0003
F47F    00                              db      0               ;128 byte sectors

                                ;       Single Density, Double Side
F480    0012                            dw      18              ;spt
F482    03 07 00                        db      3,7,0           ;blkshf, blkmsk, nulmsk
F485    00AC 001F                       dw      172,31,128,8,3  ;dsw,dirm,alloc01,chksiz,trk off
F489    0080 0008
F48D    0003
F48F    00                              db      0               ;128 byte sectors

F490                            dpb5d:

                                ;       Double Density, Single Side
F490    0022                            dw      17*2            ;spt
F492    03 07 00                        db      3,7,0           ;blkshf, blkmsk, nulmsk
F495    009C 003F                       dw      156,63,192,16,3 ;dsw,dirm,alloc01,chksiz,trk off
F499    00C0 0010
F49D    0003
F49F    81                              db      81h             ;256 byte sectors, track zero single density

                                ;       Double Density, Double Side
F4A0    0022                            dw      17*2            ;spt
F4A2    04 0F 01                        db      4,15,1          ;blkshf, blkmsk, nulmsk
F4A5    00A2 003F                       dw      162,63,192,16,3 ;dsw,dirm,alloc01,chksiz,trk off
F4A9    00C0 0010
F4AD    0003
F4AF    81                              db      81h             ;256 byte sectors, track zero single density

                                subttl  Western Digital WD-1797-02 Floppy Disk Driver
                                page
```

```
1965
1966                            ;;      Standard Disk Driver Interface Definitions.
1967                            ;
1968                            ;       The main entry point (FLOPPY) is called with HL pointing
1969                            ;       to a disk driver request block.  All information is passed
1970                            ;       in this request as follows:
1971                            ;
1972                            ;       HL->    db      command ;1 = read, 0 = write, -1 = select dph
1973                            ;               db      phunit  ;physical unit for request (0-3)
1974                            ;               db      cpunit  ;CP/M logical drive for request (0-15)
1975                            ;               dw      track   ;CP/M track number (offset already applied)
1976                            ;               dw      sector  ;Phys sector number (after deblocking)
1977                            ;               dw      address ;CP/M dma transfer address
1978                            ;
1979                                    subttl  Assembly Constants
1980                                    page
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Assembly Constants

 1981
 1982
 1983   0066                    NMI     equ     00066h          ;address of non maskable interrupt
 1984
 1985                           ;;      WD 1797 I/O port addresses.
 1986                           ;
 1987   0010                    wdsr    equ     10h             ;status
 1988   0010                    wdcr    equ     10h             ;command
 1989   0011                    wdtr    equ     11h             ;track
 1990   0012                    wdsn    equ     12h             ;sector
 1991   0013                    wddt    equ     13h             ;data
 1992   001C                    wdsl    equ     1ch             ;drive select port
 1993   0030                    wdsd    equ     30h             ;select single density
 1994   0031                    wddd    equ     31h             ;select double density
 1995
 1996                           ;;      External Disk Parameter Tables.
 1997                           ;
 1998
 1999   0007                    fm.un   equ     7
 2000   0004                    fm.ds   equ     4
 2001   0005                    fm.dd   equ     5
 2002   0006                    fm.fv   equ     6
 2003   00A0                    fm.ddss equ     (1 shl fm.dd) or (1 shl fm.un)
 2004
 2005   004D                    ntrk8   equ     77
 2006   0028                    ntrk5   equ     40
 2007
 2008   0004                    c.8in   equ     4
 2009   0005                    c.two   equ     5
 2010   0006                    timou   equ     6               ;motor / select time out
 2011   000A                    dpbofs  equ     10              ;offset in dph for dpb address
 2012
 2013                           subttl  Floppy Disk Driver Proper
 2014                           page
```

```
2015
2016                                    ;;      Western Digital Floppy Disk Driver.
2017                                    ;
2018    F4B0    CD F647                 Dskdvr: call    rdc             ;un hang busy controller
2019    F4B3    7E                              ld      a,(hl)          ;set command
2020    F4B4    23                              inc     hl              ;point to unit
2021    F4B5    32 F4E7                         ld      (rdop),a
2022    F4B8    3C                              inc     a
2023    F4B9    28 55                           jr      z,selec         ;if select command
2024    F4BB    06 0A                           ld      b,10            ;set retry count
2025    F4BD    C5                      flop1:  push    bc              ;save count
2026    F4BE    E5                              push    hl              ;save command
2027    F4BF    7E                              ld      a,(hl)          ;set unit select
2028    F4C0    CD'F544                         call    selunt
2029    F4C3    FA'F506                         jp      m,flop5         ;if unit not ready
2030    F4C6    23                              inc     hl
2031    F4C7    23                              inc     hl
2032    F4C8    4E                              ld      c,(hl)          ;set track low
2033    F4C9    CD F5A3                         call    seekx           ;position disk
2034    F4CC    4E                              ld      c,(hl)          ;retrieve track low
2035    F4CD    20 37                           jr      nz,flop5        ;if unrecoverable error
2036    F4CF    23                              inc     hl              ;track high
2037    F4D0    23                              inc     hl
2038    F4D1    13                              inc     de              ;point to second byte of track table entry
2039    F4D2    1A                              ld      a,(de)          ;get diskette type
2040    F4D3    E6 18                           and     18h
2041    F4D5    7E                              ld      a,(hl)          ;sector low
2042    F4D6    20 06                           jr      nz,flop2        ;if single density, cp/m skews
2043    F4D8    79                              ld      a,c             ;get current logical track
2044    F4D9    B7                              or      a
2045    F4DA    7E                              ld      a,(hl)          ;set sector
2046    F4DB    28 01                           jr      z,flop2         ;if single density track zero
2047    F4DD    3C                              inc     a               ;translate for double density
2048    F4DE    D3 12                   flop2:  out     (wdsn),a        ;set sector to read in 1791
2049    F4E0    23                              inc     hl              ;skip sector high
2050    F4E1    23                              inc     hl              ;dmal
2051    F4E2    5E                              ld      e,(hl)          ;set transfer address to HL
2052    F4E3    23                              inc     hl              ;dmah
2053    F4E4    56                              ld      d,(hl)
2054    F4E5    EB                              ex      de,hl
2055    F4E6    3E 00                           ld      a,0             ;set read/write switch
2056    F4E7                            rdop    equ     $-1
2057    F4E8    B7                              or      a
2058    F4E9    0E A8                           ld      c,0a8h          ;preset write command
2059    F4EB    3E A3                           ld      a,0a3h          ;set second part of OUTI
2060    F4ED    28 03                           jr      z,flop3         ;if write
2061    F4EF    0E 88                           ld      c,088h          ;turn write command into read command
2062    F4F1    3D                              dec     a               ;turn OUTI into INI
2063    F4F2    32 F4FE                 flop3:  ld      (rdwra),a       ;set up i/o direction
2064    F4F5    3E 00                           ld      a,0
2065    F4F6                            rdwrs   equ     $-1             ;set side compare flag
2066    F4F7    81                              add     a,c
2067    F4F8    4F                              ld      c,a
2068    F4F9    CD F61D                         call    stc             ;start transfer
2069    F4FC    76                      flop4:  halt                    ;wait for DRQ or INT
```

```
2070    F4FD    ED A2                   ini                     ;transfer next byte
2071    F4FE                    rdwra   equ     $-1
2072    F4FF    20 FB                   jr      nz,flop4        ;if transfer not complete
2073    F501    CD F639                 call    ttc             ;terminate transfer command
2074    F504    E6 DF                   and     0dfh            ;set error bits
2075    F506    E1                      pop     hl              ;recover command pointer
2076    F507    C1                      pop     bc
2077    F508    C8                      ret     z               ;if no errors
2078    F509    D8                      ret     c               ;if 5.25" not ready
2079    F50A    CD F069                 call    softv
2080    F50D    10 AE                   djnz    flop1           ;if retry not exceeded
2081    F50F    C9                      ret
2082
2083                            ;;      select - select dph for unit.
2084                            ;
2085    F510    7E              selec:  ld      a,(hl)          ;set unit
2086    F511    FE 04                   cp      4
2087    F513    D2 F42A                 jp      nc,selerr       ;if bad unit select
2088    F516    26 00                   ld      h,0
2089    F518    CD F039                 call    dayti           ;set address of timers
2090    F51B    2B                      dec     hl              ;point to motor timer
2091    F51C    22 F559                 ld      (mtradr),hl
2092    F51F    2B                      dec     hl              ;point to configurable step rate
2093    F520    22 F66E                 ld      (stpadr),hl     ;store address of step rate for media selector
2094    F523    26 00                   ld      h,0
2095    F525    6F                      ld      l,a
2096    F526    29                      add     hl,hl           ;multiply by 2**4
2097    F527    29                      add     hl,hl
2098    F528    29                      add     hl,hl
2099    F529    29                      add     hl,hl
2100    F52A    11 F390                 ld      de,Dpbase       ;set address of disk parameter headers
2101    F52D    19                      add     hl,de           ;set dph address
2102    F52E    E5                      push    hl
2103    F52F    CD F65A                 call    smf             ;set media format
2104    F532    E1                      pop     hl
2105    F533    CA F42A                 jp      z,selerr        ;if no media
2106    F536    71                      ld      (hl),c          ;store translate table
2107    F537    23                      inc     hl
2108    F538    70                      ld      (hl),b
2109    F539    01 000A                 ld      bc,dpbofs
2110    F53C    09                      add     hl,bc           ;point to dpb addr and clear carry
2111    F53D    72                      ld      (hl),d          ;fill in dpb address
2112    F53E    2B                      dec     hl
2113    F53F    73                      ld      (hl),e
2114    F540    ED 42                   sbc     hl,bc           ;point back to dph
2115    F542    AF                      xor     a
2116    F543    C9                      ret
2117
2118                            ;;      select physical unit.
2119                            ;
2120    F544    4F              selunt: ld      c,a             ;save select
2121    F545    EB                      ex      de,hl
2122    F546    21 F700                 ld      hl,trktbl+1     ;set track / density table address
2123    F549    06 00                   ld      b,0
2124    F54B    09                      add     hl,bc
2125    F54C    09                      add     hl,bc
```

```
2126    F54D  22 F6D1                 ld      (smfa),hl       ;save address for media selector
2127    F550  EB                      ex      de,hl
2128    F551  1A                      ld      a,(de)          ;get previous density switch
2129    F552  1B                      dec     de              ;point to track word
2130    F553  CD F598                 call    selden          ;select density
2131    F556  3E 06                   ld      a,timou         ;start motor / select timer
2132    F558  32 F559                 ld      (mtradr),a
2133    F559                  mtradr  equ     $-2             ;address filled in by once only routine
2134    F55B  FB                      ei                      ;insure clock enabled
2135    F55C  CB 89                   res     1,c             ;map C->A, D->B
2136    F55E  DB 1C                   in      a,(wdsl)        ;read current select
2137    F560  47                      ld      b,a
2138    F561  E6 F8                   and     not 7
2139    F563  B1                      or      c               ;insert new select
2140    F564  3C                      inc     a               ;0-1, 1-2
2141    F565  D3 1C                   out     (wdsl),a        ;select drive
2142    F567  A8                      xor     b
2143    F568  E6 03                   and     3
2144    F56A  28 25                   jr      z,sel3          ;if drive select identical
2145    F56C  3E FF                   ld      a,-1            ;force track position recovery
2146    F56E  12                      ld      (de),a
2147    F56F  CB 60                   bit     c.8in,b         ;test 8/5 status
2148    F571  20 1E                   jr      nz,sel3         ;if 8"
2149    F573  CD F647                 call    rdc             ;set type I status
2150    F576  06 08                   ld      b,2*4           ;watch for four holes (8 transitions)
2151    F578  E5              sellw:  push    hl
2152    F579  2A F559                 ld      hl,(mtradr)     ;get address of motor select timer
2153    F57C  7E                      ld      a,(hl)
2154    F57D  E1                      pop     hl
2155    F57E  D6 04                   sub     timou-2         ;look for 1-2 seconds
2156    F580  D8                      ret     c               ;if drive not spinning
2157    F581  DB 10                   in      a,(wdsr)
2158    F583  E6 02                   and     2
2159    F585  28 F1           sel2:   jr      z,sellw         ;if index not under light
2160    F587  3A F585                 ld      a,(sel2)        ;switch index polarity
2161    F58A  EE 08                   xor     8               ;(jr z) xor (jr nz)
2162    F58C  32 F585                 ld      (sel2),a
2163    F58F  10 E7                   djnz    sellw           ;wait for at least three revolutions
2164    F591  DB 10           sel3:   in      a,(wdsr)        ;set ready status
2165    F593  E6 80                   and     80h
2166    F595  C9                      ret
2167
2168    F596  3E 18           seldns: ld      a,18h           ;set track zero single density
2169    F598  32 F632         selden: ld      (dsw),a         ;store switch for read/write routines
2170    F59B  E6 18                   and     18h
2171    F59D  D3 31                   out     (wddd),a        ;pre-select dual density
2172    F59F  C8                      ret     z               ;if dual density
2173    F5A0  D3 30                   out     (wdsd),a        ;select single density
2174    F5A2  C9                      ret
2175
2176                          ;;      seek - position disk.
2177                          ;
2178    F5A3  79              seekx:  ld      a,c             ;set new track
2179    F5A4  B7                      or      a
2180    F5A5  CC F596                 call    z,seldns        ;force single density track 0
2181    F5A8  13                      inc     de
```

```
2182    F5A9    1A                      ld      a,(de)
2183    F5AA    1B                      dec     de
2184    F5AB    E6 01                   and     1
2185    F5AD    28 15                   jr      z,sek1          ;if one sided diskette
2186    F5AF    DB 1C                   in      a,(wds1)
2187    F5B1    CB 67                   bit     c.8in,a
2188    F5B3    06 4D                   ld      b,ntrk8         ;set number of eight inch tracks
2189    F5B5    20 02                   jr      nz,sek0         ;if 8" drives
2190    F5B7    06 28                   ld      b,ntrk5
2191    F5B9    79              sek0:   ld      a,c             ;set seek track
2192    F5BA    B8                      cp      b
2193    F5BB    3E 00                   ld      a,0             ;preset side 0
2194    F5BD    38 05                   jr      c,sek1          ;if side 0
2195    F5BF    79                      ld      a,c
2196    F5C0    90                      sub     b               ;wrap to side 1
2197    F5C1    4F                      ld      c,a
2198    F5C2    3E 02                   ld      a,2             ;set side 1
2199    F5C4    32 F4F6         sek1:   ld      (rdwrs),a       ;store F1 (update SSO)
2200    F5C7    87                      add     a,a             ;move into select port position
2201    F5C8    47                      ld      b,a
2202    F5C9    F3                      di
2203    F5CA    DB 1C                   in      a,(wds1)
2204    F5CC    CB 97                   res     2,a
2205    F5CE    B0                      or      b
2206    F5CF    FB                      ei
2207    F5D0    D3 1C                   out     (wds1),a        ;send out REAL SSO
2208    F5D2    1A                      ld      a,(de)          ;check current position
2209    F5D3    D3 11                   out     (wdtr),a        ;inform 1797 of current track
2210    F5D5    B9                      cp      c
2211    F5D6    28 17                   jr      z,seek3         ;if position ok, load head
2212    F5D8    3C                      inc     a               ;check for forced recovery
2213    F5D9    CC F5F8                 call    z,rse           ;recover seek errors
2214    F5DC    28 0D                   jr      z,seek1         ;if error not recoverable
2215    F5DE    79              seek0:  ld      a,c             ;set new track
2216    F5DF    D3 13                   out     (wddt),a        ;in data register
2217    F5E1    3E 1C                   ld      a,1ch           ;set seek with verify command
2218    F5E3    CD F643                 call    isc             ;issue step command
2219    F5E6    E6 98                   and     98h
2220    F5E8    79                      ld      a,c             ;update current track
2221    F5E9    28 02                   jr      z,seek2         ;if no errors
2222    F5EB    F6 FF           seek1:  or      -1              ;force recovery next time
2223    F5ED    12              seek2:  ld      (de),a
2224    F5EE    C9                      ret
2225    F5EF    CD F647         seek3:  call    rdc             ;set type I status
2226    F5F2    E6 20                   and     20h             ;test head load
2227    F5F4    28 E8                   jr      z,seek0         ;if head is not loaded
2228    F5F6    AF              retzr:  xor     a               ;say seek complete
2229    F5F7    C9                      ret
2230
2231                            ;;      rse - recover seek error.
2232                            ;
2233    F5F8    C5              rse:    push    bc
2234    F5F9    CD F605                 call    rdid            ;read id mark
2235    F5FC    20 05                   jr      nz,rse1         ;if track position identified
2236    F5FE    CD F641                 call    recal           ;recalibrate
2237    F601    E6 04                   and     4               ;verify track zero flag set
```

```
2238    F603    C1                  rsel:   pop     bc
2239    F604    C9                          ret
2240
2241                                ;;      rdid - read id mark.
2242                                ;
2243    F605    0E C4               rdid:   ld      c,0c4h          ;set Read Address Command
2244    F607    CD F61D                     call    stc             ;start transfer command
2245    F60A    76                          halt                    ;wait for interrupt
2246    F60B    ED 40                       in      b,(c)           ;first byte is track
2247    F60D    76                          halt
2248    F60E    ED 48                       in      c,(c)           ;second byte is side, pitch next 4
2249    F610    CD F639                     call    ttc             ;terminate transfer command
2250    F613    E6 98                       and     98h             ;ignore lost data
2251    F615    20 04                       jr      nz,rdid1        ;if track not identified
2252    F617    78                          ld      a,b
2253    F618    D3 11                       out     (wdtr),a        ;tell 1797 track head is on now
2254    F61A    F6                          defb    0f6h            ;or xra to set NZ
2255    F61B    AF                  rdid1:  xor     a               ;set track not found
2256    F61C    C9                          ret
2257
2258                                ;;      stc - start transfer command.
2259                                ;
2260    F61D    F3                  stc:    di                      ;lock normal interrupts
2261    F61E    3A 0066                     ld      a,(NMI)         ;save byte at NMI address
2262    F621    32 F63A                     ld      (ttca),a
2263    F624    3E C9                       ld      a,0c9h          ;store RET there
2264    F626    32 0066                     ld      (NMI),a
2265    F629    79                          ld      a,c             ;retrieve command
2266    F62A    01 1413                     ld      bc,wddt+20*256  ;1797 access timer / data port
2267    F62D    D3 10                       out     (wdcr),a        ;issue command
2268    F62F    10 FE                       djnz    $               ;pause 60 usec
2269    F631    3E 00                       ld      a,0
2270    F632                        dsw     equ     $-1             ;density switch
2271    F633    E6 18                       and     18h             ;say ready and density
2272    F635    C8                          ret     z               ;if double density
2273    F636    06 80                       ld      b,128           ;set 128 byte single density sectors
2274    F638    C9                          ret
2275
2276                                ;;      ttc - terminate transfer command.
2277                                ;
2278    F639    3E 00               ttc:    ld      a,0             ;restore location 66
2279    F63A                        ttca    equ     $-1
2280    F63B    32 0066                     ld      (NMI),a
2281    F63E    FB                          ei                      ;take interrupts now
2282    F63F    18 0A                       jr      woc             ;wait for 1797 to complete
2283
2284                                ;;      recalibrate drive.
2285                                ;
2286    F641    AF                  recal:  xor     a               ;set restore command / track 0
2287    F642    12                          ld      (de),a          ;set track zero
2288
2289                                ;;      isc - issue step command.
2290                                ;
2291    F643    F6 01               isc:    or      1               ;insert step rate
2292    F644                        stepr   equ     $-1     ;*****=>;modify here for step rate change
2293    F645    18 02                       jr      icc
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Floppy Disk Driver Proper

 2294     F647    3E D0                rdc:    ld      a,0d0h          ;terminate and set type I status
 2295
 2296                                  ;;      icc  - issue controller command.
 2297                                  ;
 2298     F649    D3 10                icc:    out     (wdcr),a        ;issue 1797 command
 2299
 2300                                  ;;      woc  - wait operation complete.
 2301                                  ;
 2302     F64B    3E 14                woc:    ld      a,20            ;set 60 usec delay
 2303     F64D    3D                   woc1:   dec     a
 2304     F64E    20 FD                        jr      nz,woc1
 2305     F650    CD F066              woc2:   call    idle            ;idle cpu
 2306     F653    DB 10                        in      a,(wdsr)        ;set 1797 status
 2307     F655    CB 47                        bit     0,a
 2308     F657    20 F7                        jr      nz,woc2         ;if busy, wait
 2309     F659    C9                           ret
 2310
 2311                                          subttl  Media Format Selector
 2312                                          page
```

```
                                ;;      smf - set media format.
                                ;
                                ;       entry:  unit in A
                                ;
                                ;       exit:   DE = dpb address
                                ;               BC = translate table
                                ;
F65A  CD F544           smf:    call    selunt          ;select unit
F65D  FA F5F6                   jp      m,retzr         ;if disk not ready
F660  21 F6D5                   ld      hl,dtype
F663  36 A0                     ld      (hl),fm.ddss    ;start out double density, single side, retry
F665  DB 1C                     in      a,(wdsl)        ;read select status
F667  CB 67                     bit     c.8in,a         ;test 8" / 5" status
F669  20 02                     jr      nz,smf0         ;if 8 inch drives
F66B  CB F6                     set     fm.fv,(hl)      ;move up to 5.25" dpbs
F66D  3A F66E           smf0:   ld      a,(stpadr)      ;set configurable step rate for 8" drives
F66E                    stpadr  equ     $-2
F670  E6 03                     and     3               ;just so seeks aren't formats
F672  32 F644                   ld      (stepr),a       ;save step rate in seek command
F675  CD F641           smf0a:  call    recal           ;establish position
F678  E6 84                     and     84h
F67A  C8                        ret     z               ;if not on track zero
F67B  FA F5F6                   jp      m,retzr         ;if unit not ready
F67E  D3 31                     out     (wddd),a        ;set double density
F680  3E FF                     ld      a,-1
F682  12                        ld      (de),a          ;clear drive on track
F683  3E 02                     ld      a,2             ;use track 2 for density select
F685  D3 13                     out     (wddt),a
F687  3E 18                     ld      a,18h           ;seek / no verify
F689  CD F643                   call    isc             ;issue seek command
F68C  3E 1C                     ld      a,1ch           ;find id mark
F68E  D3 10                     out     (wdcr),a        ;start verify
F690  01 0000                   ld      bc,0            ;set timers
F693  10 FE             smf1:   djnz    $               ;pause
F695  DB 10                     in      a,(wdsr)
F697  CB 47                     bit     0,a
F699  28 08                     jr      z,smf1a         ;if command completed
F69B  0D                        dec     c
F69C  20 F5                     jr      nz,smf1         ;if more time
F69E  CD F647                   call    rdc             ;terminate seek
F6A1  3E 18                     ld      a,18h           ;set pseudo record not found
F6A3  E6 18             smf1a:  and     18h             ;check record not found / crc error
F6A5  13                        inc     de              ;point to density word in track table
F6A6  12                        ld      (de),a
F6A7  1B                        dec     de
F6A8  28 14                     jr      z,smf2          ;if density select successful
F6AA  D3 30                     out     (wdsd),a        ;use single density
F6AC  3E 1C                     ld      a,1ch           ;verify single density
F6AE  CD F643                   call    isc             ;issue seek
F6B1  E6 18                     and     18h
F6B3  28 07                     jr      z,smf1b         ;if single density successful
F6B5  CB 7E                     bit     fm.un,(hl)      ;test retry
F6B7  CB BE                     res     fm.un,(hl)      ;clear retry
```

```
2368    F6B9    20 BA                       jr      nz,smf0a        ;if retry
2369    F6BB    C9                          ret                     ;return select error
2370    F6BC    CB AE               smf1b:  res     fm.dd,(hl)      ;back up to single density dpbs
2371    F6BE    CB BE               smf2:   res     fm.un,(hl)      ;clear retry
2372    F6C0    DB 1C                       in      a,(wdsl)
2373    F6C2    CB D7                       set     2,a             ;select side 2
2374    F6C4    D3 1C                       out     (wdsl),a
2375    F6C6    CD F605                     call    rdid            ;read id mark
2376    F6C9    28 09                       jr      z,smf4          ;if no id found, must be one side
2377    F6CB    0D                          dec     c
2378    F6CC    20 06                       jr      nz,smf4         ;if side 1 ID not read
2379    F6CE    CB E6                       set     fm.ds,(hl)      ;bump up to two sided dpbs
2380    F6D0    21 F6D1                     ld      hl,smfa         ;set double sided status in track table
2381    F6D1                        smfa    equ     $-2
2382    F6D3    34                          inc     (hl)
2383    F6D4    21 0000             smf4:   ld      hl,0            ;set diskette type
2384    F6D5                        dtype   equ     $-2
2385    F6D7    7D                          ld      a,l             ;save type
2386    F6D8    4C                          ld      c,h             ;preset no translate
2387    F6D9    44                          ld      b,h
2388    F6DA    11 F430                     ld      de,dpb8s        ;set base of disk parameter blocks
2389    F6DD    19                          add     hl,de
2390    F6DE    EB                          ex      de,hl           ;return DPB address in DE
2391    F6DF    CB 6F                       bit     fm.dd,a
2392    F6E1    C0                          ret     nz              ;if diskette is double density
2393    F6E2    01 F6ED                     ld      bc,trn5         ;preset 5.25" skew table
2394    F6E5    CB 77                       bit     fm.fv,a
2395    F6E7    C0                          ret     nz              ;if diskette is small
2396    F6E8    01 F410                     ld      bc,trn6         ;set 8" translate
2397    F6EB    3C                          inc     a               ;force NZ
2398    F6EC    C9                          ret
2399
2400                                ;;      Skew by 5 translate table.
2401                                ;
2402    F6ED    01 06 0B 10         trn5:   db      01,06,11,16
2403    F6F1    03 08 0D 12                 db      03,08,13,18
2404    F6F5    05 0A 0F 02                 db      05,10,15,02
2405    F6F9    07 0C 11 04                 db      07,12,17,04
2406    F6FD    09 0E                       db      09,14
2407
2408    F6FF    7F 00 C0 00         trktbl: db      7fh,0,0c0h,0,20h,0,2,0,81h
2409    F703    20 00 02 00
2410    F707    81
2411
2412    F708                        rigdpb  equ     0f708h
2413    F770                        iobloc  equ     0f770h
2414
2415                                        above
2416    0708"                   +           d&seg
2417
2418                                        .dephase
2419                                        .phase  0f470h
2420    F470                        sasstr  equ     $
2421
2422                                        Subttl  Rigid Partition Disk Parameter Blocks.
2423                                        page
```

```
2424
2425                                    ;       There are 16 4k Blocks per pseudo track.
2426                                    ;       Track Zero (2 Cylinders) are reserved for CP/M boot.
2427                                    ;
2428    0040                            Nt4     equ     64              ;Number of Tracks on Partition 0
2429    0020                            Nt5     equ     32              ;Number of Tracks on Partition 1
2430    0010                            Nt6     equ     16              ;Number of Tracks on Partition 2
2431    0010                            Nt7     equ     16              ;Number of Tracks on Partition 3
2432
2433    0000                            ..      aset    0               ;First usable track
2434                                            irpc    n,<4567>
2435                                    ..      aset    ..+1            ;reserve system track
2436                                    Dsm&n   equ     Nt&n*16-17
2437                                    Rtk&n   equ     ..
2438                                    ..      aset    ..+Nt&n-1
2439                                    Dpbrg&n:dw      512             ;spt
2440                                            db      5,31            ;blkshf, blkmsk
2441                                            db      3+2*(Dsm&n ge 256);exm
2442                                            dw      Dsm&n           ;dsm
2443                                            dw      511             ;dirmax
2444                                            db      -1              ;alloc0 (reserve additional dir space)
2445                                            db      0               ;alloc1
2446                                            dw      0               ;check size
2447                                            dw      Rtk&n           ;track offset
2448                                            db      1               ;256 byte sectors
2449                                            endm
2450    F470    0200            +       Dpbrg&4:dw      512             ;spt
2451    F472    05 1F           +               db      5,31            ;blkshf, blkmsk
2452    F474    01              +               db      3+2*(Dsm&4 ge 256);exm
2453    F475    03EF            +               dw      Dsm&4           ;dsm
2454    F477    01FF            +               dw      511             ;dirmax
2455    F479    FF              +               db      -1              ;alloc0 (reserve additional dir space)
2456    F47A    00              +               db      0               ;alloc1
2457    F47B    0000            +               dw      0               ;check size
2458    F47D    0001            +               dw      Rtk&4           ;track offset
2459    F47F    01              +               db      1               ;256 byte sectors
2460    F480    0200            +       Dpbrg&5:dw      512             ;spt
2461    F482    05 1F           +               db      5,31            ;blkshf, blkmsk
2462    F484    01              +               db      3+2*(Dsm&5 ge 256);exm
2463    F485    01EF            +               dw      Dsm&5           ;dsm
2464    F487    01FF            +               dw      511             ;dirmax
2465    F489    FF              +               db      -1              ;alloc0 (reserve additional dir space)
2466    F48A    00              +               db      0               ;alloc1
2467    F48B    0000            +               dw      0               ;check size
2468    F48D    0041            +               dw      Rtk&5           ;track offset
2469    F48F    01              +               db      1               ;256 byte sectors
2470    F490    0200            +       Dpbrg&6:dw      512             ;spt
2471    F492    05 1F           +               db      5,31            ;blkshf, blkmsk
2472    F494    03              +               db      3+2*(Dsm&6 ge 256);exm
2473    F495    00EF            +               dw      Dsm&6           ;dsm
2474    F497    01FF            +               dw      511             ;dirmax
2475    F499    FF              +               db      -1              ;alloc0 (reserve additional dir space)
2476    F49A    00              +               db      0               ;alloc1
2477    F49B    0000            +               dw      0               ;check size
2478    F49D    0061            +               dw      Rtk&6           ;track offset
```

```
2479    F49F    01          +           db      1               ;256 byte sectors
2480    F4A0    0200        +   Dpbrg&7:dw      512             ;spt
2481    F4A2    05 1F       +           db      5,31            ;blkshf, blkmsk
2482    F4A4    03          +           db      3+2*(Dsm&7 ge 256);exm
2483    F4A5    00EF        +           dw      Dsm&7           ;dsm
2484    F4A7    01FF        +           dw      511             ;dirmax
2485    F4A9    FF          +           db      -1              ;alloc0 (reserve additional dir space)
2486    F4AA    00          +           db      0               ;alloc1
2487    F4AB    0000        +           dw      0               ;check size
2488    F4AD    0071        +           dw      Rtk&7           ;track offset
2489    F4AF    01          +           db      1               ;256 byte sectors
2490
2491                                    subttl  SA1403 - Shugart / DTC SASI Driver
2492                                    page
```

```
                                ;;      SA-1403D SASI driver.
                                ;
EE00                            rgdbuf  equ     0ee00h          ;rigid parameter load buffer

                                ;;      Sasi Pio Port Addresses.
                                ;
0011                            pioAs   equ     11h             ;Pio A Status
0010                            pioAd   equ     pioAs xor 01b
0013                            pioBs   equ     pioAs xor 10b
0012                            pioBd   equ     pioAs xor 11b

0010                            Sasid   equ     pioAd           ;bus data
0012                            Sasic   equ     pioBd           ;bus control
0012                            Sasis   equ     pioBd           ;bus status

001C                            syspio  equ     1ch             ;system configuration port

                                ;;      Sasi controller status bit definitions.
                                ;
0000                            b.bsy   equ     00              ;(in)  controller busy status
0001                            b.msg   equ     01              ;(in)  status byte completion status
0002                            b.cd    equ     02              ;(in)  control byte or data byte transfer
0003                            b.req   equ     03              ;(in)  controller request for data/command
0004                            b.io    equ     04              ;(in)  data transfer direction
0005                            b.sel   equ     05              ;(out) controller select
0006                            b.par   equ     06              ;(in)  buss parity error
0007                            b.rst   equ     07              ;(out) controller reset

                                ;;      Logical Unit Assignments.
                                ;
0000                            falun   equ     0               ;A: Lun
0001                            fblun   equ     1               ;B: Lun
0000                            fclun   equ     0               ;C: Lun
0002                            fdlun   equ     2               ;D: Lun
0003                            rglun   equ     3               ;E: Lun

                                        subttl  Sasi Class Code Definitions
                                        page
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Sasi Class Code Definitions

 2533
 2534                                ;;      Class Command Codes for Prom Set AS31*
 2535                                ;
 2536                                ;       DTC Reference Manual Dated February 4, 1981.
 2537                                ;
 2538                                ;;      class  0 commands.
 2539                                ;
 2540    0000                        c.trdy  equ     00h             ;test ready status
 2541    0001                        c.recal equ     01h             ;recalibrate drive
 2542    0002                        c.rsyn  equ     02h             ;request syndrome
 2543    0003                        c.rqsn  equ     03h             ;request sense after error
 2544    0004                        c.fmat  equ     04h             ;format drive
 2545    0005                        c.vtrk  equ     05h             ;verify track format
 2546    0006                        c.ftrk  equ     06h             ;format single track
 2547    0007                        c.flaw  equ     07h             ;format track with flaw
 2548    0008                        c.read  equ     08h             ;read data
 2549    0009                        c.wrpr  equ     09h             ;write protect sector
 2550    000A                        c.writ  equ     0ah             ;write data
 2551    000B                        c.seek  equ     0bh             ;initiate seek
 2552    000C                        c.init  equ     0ch             ;inititialize drive
 2553
 2554                                ;;      Class 6 commands.
 2555                                ;
 2556    00C0                        c.flpy  equ     0c0h            ;define floppy disk format
 2557
 2558                                ;;      Floppy Format Codes.
 2559                                ;
 2560    0000                        fmds    equ     0               ;double side bit
 2561    0001                        fmdd    equ     1               ;double density bit
 2562    0002                        fm.sz   equ     2               ;sector size bit
 2563    0003                        fm.wr   equ     3               ;log2(fm.ddds+1)
 2564
 2565    0000                        fm.sdss equ     00h             ;Single Density, Single Sided
 2566    0001                        fm.sdds equ     01h             ;Single Density, Double Sided
 2567    0006                        fmddss  equ     06h             ;Double Density, Single Sided
 2568    0007                        fm.ddds equ     07h             ;Double Density, Double Sided
 2569    0080                        fm.hard equ     80h             ;Rigid
 2570
 2571                                ;;      Class 7 commands.
 2572                                ;
 2573    00E0                        c.tram  equ     0e0h            ;test ram buffer
 2574
 2575                                ;;      Message Macros.
 2576                                ;
 2577                                pmsg    macro   n,msg
 2578                                        if1
 2579                                        .printx +MSG N+
 2580                                        endif
 2581                                        endm
 2582
 2583                                phex    macro   n,m
 2584                                        .radix  16
 2585                                        pmsg    %(n),<m>
 2586                                        .radix  10
 2587                                        endm
```

```
2588
2589                                    subttl  Sasi Physical Driver.
2590                                    page
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Sasi Physical Driver.

2591
2592                                    ;;      Sa1403 - Sasi Physical Driver.
2593                                    ;
2594
2595    F4B0    06 06                   Sa1403: ld      b,6             ;set retry count
2596    F4B2    7E                      sas0a:  ld      a,(hl)          ;set driver operation
2597    F4B3    23                              inc     hl              ;point to unit
2598    F4B4    3C                              inc     a
2599    F4B5    28 49                           jr      z,sselec        ;if select DPH
2600    F4B7    E5                              push    hl
2601    F4B8    C5                              push    bc
2602    F4B9    3D                              dec     a
2603    F4BA    3E 0A                           ld      a,c.writ        ;preset write opcode
2604    F4BC    28 02                           jr      z,sas0
2605    F4BE    3E 08                           ld      a,c.read        ;assume read
2606    F4C0    32 F6F0                 sas0:   ld      (opcode),a      ;set Sasi opcode
2607    F4C3    7E                              ld      a,(hl)          ;get driver unit
2608    F4C4    E5                              push    hl
2609    F4C5    CD F5AF                         call    mlu             ;map to logical unit
2610    F4C8    E1                              pop     hl
2611    F4C9    23                              inc     hl              ;ignore cpm dsk
2612    F4CA    23                              inc     hl              ;track low
2613    F4CB    56                              ld      d,(hl)          ;set track
2614    F4CC    23                              inc     hl
2615    F4CD    23                              inc     hl
2616    F4CE    5E                              ld      e,(hl)          ;set sector
2617    F4CF    CD F6D7                         call    cwp             ;check write protect
2618    F4D2    20 22                           jr      nz,sas2         ;if write protected and track > 0
2619    F4D4    E5                              push    hl              ;save request block address
2620    F4D5    CD F5C2                         call    mpa             ;map physical address to logical address
2621    F4D8    21 F6F0                         ld      hl,opcode
2622    F4DB    CD F643                         call    iccs            ;issue controller command
2623    F4DE    E1                              pop     hl              ;get pointer to low sector back
2624    F4DF    23                              inc     hl              ;ignore sector high
2625    F4E0    23                              inc     hl              ;dma low
2626    F4E1    5E                              ld      e,(hl)
2627    F4E2    23                              inc     hl              ;dma high
2628    F4E3    56                              ld      d,(hl)
2629    F4E4    EB                              ex      de,hl           ;set transfer address to HL
2630    F4E5    06 00                           ld      b,0             ;set sector length
2631    F4E6                            seclen  equ     $-1
2632    F4E7    3A F6F0                         ld      a,(opcode)
2633    F4EA    FE 0A                           cp      c.writ
2634    F4EC    28 05                           jr      z,sas1          ;if write command
2635    F4EE    CD F65F                         call    tdi             ;transfer data in
2636    F4F1    18 03                           jr      sas2
2637    F4F3    CD F656                 sas1:   call    tdo             ;transfer data out
2638    F4F6    C1                      sas2:   pop     bc
2639    F4F7    E1                              pop     hl
2640    F4F8    2B                              dec     hl
2641    F4F9    C8                              ret     z               ;if no errors
2642    F4FA    CD F069                         call    softv           ;report soft error
2643    F4FD    10 B3                           djnz    sas0a           ;if more retries
2644    F4FF    C9                              ret                     ;return error
2645
```

```
2646                                    subttl  Physical Driver Select
2647                                    page
```

```
                                ;;      Select - Physical Driver Select.
                                ;
F500    7E              sselec: ld      a,(hl)          ;set physical unit
F501    FE 08                   cp      8               ;verify in range
F502                    numunt  equ     $-1
F503    3F                      ccf
F504    D4 F50E                 call    nc,smfs         ;set media format
F507    D0                      ret     nc              ;if media identified

                                ;;      selerr - Select Error Driver.
                                ;
F508    21 0000         xselerr: ld     hl,0            ;Select Error Driver
F50B    F6 FF           selerl: or      -1
F50D    C9                      ret

                                ;;      smfs - Set Media Format.
                                ;
                                ;       entry:  A = Driver unit index
                                ;       Exit:   HL = DPH address, if no carry
                                ;
F50E    CD F6F6         smfs:   call    first           ;execute first time only routine
F511    CD F5AF                 call    mlu             ;map to logical unit
F514    EB                      ex      de,hl           ;get dph index to hl
F515    7D                      ld      a,l             ;and A
F516    29                      add     hl,hl           ;index *16
F517    29                      add     hl,hl
F518    29                      add     hl,hl
F519    29                      add     hl,hl
F51A    11 F390                 ld      de,Dpbase       ;set base of Disk Parameter Headers
F51D    19                      add     hl,de
F51E    FE 04                   cp      4
F520    D0                      ret     nc              ;if rigid unit
F521    E5                      push    hl              ;save dph address
F522    3E 80                   ld      a,80h           ;disable error recovery
F524    32 F6F5                 ld      (dctrl),a
F527    32 F5F1                 ld      (lastfm+1),a
F52A    0A                      ld      a,(bc)          ;always try double side first
F52B    F6 01                   or      1 shl fmds
F52D    02                      ld      (bc),a
F52E    3E 07                   ld      a,8-1           ;try each type two times
F530    32 F53D         smfs1:  ld      (smfsa),a       ;set retry count
F533    C5                      push    bc              ;save define format table address
F534    CD F57A                 call    cdd             ;check drive density
F537    C1                      pop     bc
F538    60                      ld      h,b             ;set format table address
F539    69                      ld      l,c
F53A    28 14                   jr      z,smfs2         ;if diskette type identified
F53C    3E 00                   ld      a,0
F53D                    smfsa   equ     $-1             ;diskette type retry counter
F53E    D6 01                   sub     1
F540    38 31                   jr      c,smfs4         ;if media not identified
F542    35                      dec     (hl)            ;advance disk type code
F543    F2 F548                 jp      p,smfs1a        ;if no wrap
F546    36 07                   ld      (hl),fm.ddds
```

```
2703    F548    CB 4E               smfs1a: bit     fmdd,(hl)       ;test for double density
2704    F54A    20 E4                       jr      nz,smfs1        ;if double density
2705    F54C    CB 96                       res     fm.sz,(hl)      ;set sector size = 128
2706    F54E    18 E0                       jr      smfs1           ;try more diskette types
2707    F550    57                  smfs2:  ld      d,a             ;preset no translate
2708    F551    5A                          ld      e,d
2709    F552    CB 4E                       bit     fmdd,(hl)
2710    F554    20 03                       jr      nz,smfs3        ;if diskette is double density
2711    F556    11 F410                     ld      de,trn6         ;set single density translate
2712    F559    E1                  smfs3:  pop     hl              ;get dph address
2713    F55A    E5                          push    hl
2714    F55B    73                          ld      (hl),e          ;store translate address
2715    F55C    23                          inc     hl
2716    F55D    72                          ld      (hl),d
2717    F55E    11 0009                     ld      de,10-1
2718    F561    19                          add     hl,de           ;point to dpb address in dph
2719    F562    0A                          ld      a,(bc)          ;get selected format
2720    F563    E6 03                       and     3
2721    F565    EB                          ex      de,hl
2722    F566    6F                          ld      l,a
2723    F567    29                          add     hl,hl           ;index by 16
2724    F568    29                          add     hl,hl
2725    F569    29                          add     hl,hl
2726    F56A    29                          add     hl,hl
2727    F56B    01 F430                     ld      bc,dpb8s        ;set dpb base
2728    F56E    09                          add     hl,bc           ;set dpb address (clears carry)
2729    F56F    EB                          ex      de,hl           ;recover dpb pointer address in dph
2730    F570    73                          ld      (hl),e
2731    F571    23                          inc     hl
2732    F572    72                          ld      (hl),d
2733    F573    E1                  smfs4:  pop     hl              ;get dph address
2734    F574    3E 00                       ld      a,0             ;enable error recovery
2735    F576    32 F6F5                     ld      (dctrl),a
2736    F579    C9                          ret
2737
2738                                ;;      cdd - check drive density.
2739                                ;
2740    F57A    0A                  cdd:    ld      a,(bc)          ;get attempted side
2741    F57B    E6 01                       and     1               ;try side 1 on ds, 0 on ss
2742    F57D    11 0201                     ld      de,2*256+1
2743    F580    28 02                       jr      z,cdd0          ;if single side
2744    F582    16 4F                       ld      d,77+2          ;use back side
2745    F584    CD F5C2             cdd0:   call    mpa             ;map physical address
2746    F587    21 F6F0                     ld      hl,opcode
2747    F58A    36 08                       ld      (hl),c.read
2748    F58C    CD F643                     call    iccs            ;issue controller command
2749    F58F    CD F6CE                     call    sim             ;set input mode
2750    F592    CD F687             cdd1:   call    wfr             ;wait for req
2751    F595    20 04                       jr      nz,cdd2         ;if timeout or status, not data requested
2752    F597    ED 78                       in      a,(c)           ;eat sector
2753    F599    18 F7                       jr      cdd1
2754    F59B    CD F669             cdd2:   call    wcc             ;wait command complete
2755    F59E    C9                          ret
2756
2757                                ;;      p21 - Physical to Logical Mapping Table.
2758                                ;
```

```
2759                                    ;       First byte is the SA1403 Logical Unit Number
2760                                    ;       Second byte is extended drive format code
2761                                    ;
2762    F59F    00 00                   p2l:    db      falun shl 5,fm.sdss     ;Floppy Drive 0
2763    F5A1    20 01                           db      fblun shl 5,fm.sdds     ;Floppy Drive 1
2764    F5A3    00 06                           db      falun shl 5,fmddss      ;Floppy Drive 0
2765    F5A5    40 07                           db      fdlun shl 5,fm.ddds     ;Floppy Drive 2
2766    F5A7    60 80                           db      rglun shl 5,fm.hard     ;Rigid Partition 0
2767    F5A9    60 80                           db      rglun shl 5,fm.hard     ;Rigid Partition 1
2768    F5AB    60 80                           db      rglun shl 5,fm.hard     ;Rigid Partition 2
2769    F5AD    60 80                           db      rglun shl 5,fm.hard     ;Rigid Partition 3
2770
2771                                    ;;      Mlu - Map Logical Unit.
2772                                    ;
2773                                    ;       Entry:  A  =  Physical Driver Unit from Deblocker
2774                                    ;       Exit:   DE =  dph index
2775                                    ;               BC = pointer to format code
2776                                    ;               Sasi LUN stored in command block
2777                                    ;
2778    F5AF    21 F59F                 mlu:    ld      hl,p2l          ;set Rigid Physical to Logical map table
2779    F5B2    16 00                           ld      d,0
2780    F5B4    5F                              ld      e,a
2781    F5B5    19                              add     hl,de           ;point to table entry
2782    F5B6    19                              add     hl,de
2783    F5B7    7E                              ld      a,(hl)          ;get Sasi LUN
2784    F5B8    32 F6F1                         ld      (lun),a         ;store in read/write command
2785    F5BB    32 F6EB                         ld      (deflun),a      ;store in define command
2786    F5BE    23                              inc     hl              ;point to format descriptor
2787    F5BF    44                              ld      b,h
2788    F5C0    4D                              ld      c,l
2789    F5C1    C9                              ret
2790
2791                                    ;;      Mpa - Map Physical Address.
2792                                    ;
2793                                    ;       Entry:  BC = p2l format code address
2794                                    ;               D  = Track
2795                                    ;               E  = Sector
2796                                    ;
2797                                    ;       Exit:   Logical Address set in command block
2798                                    ;               Seclen set for transfer
2799                                    ;               Controller notified of floppy format
2800                                    ;
2801    F5C2    21 F4E6                 mpa:    ld      hl,seclen       ;preset sector length
2802    F5C5    36 00                           ld      (hl),0
2803    F5C7    EB                              ex      de,hl           ;preset Laddr = Paddr for rigid
2804    F5C8    0A                              ld      a,(bc)          ;get drive format code
2805    F5C9    FE 80                           cp      fm.hard
2806    F5CB    28 46                           jr      z,mpa5          ;if Rigid Partition
2807    F5CD    FE 06                           cp      fmddss
2808    F5CF    38 06                           jr      c,mpa1          ;if single density
2809    F5D1    2C                              inc     l               ;advance sector from 0->25 to 1->26
2810    F5D2    7C                              ld      a,h
2811    F5D3    B7                              or      a
2812    F5D4    20 04                           jr      nz,mpa2         ;if not on track zero
2813    F5D6    2D                              dec     l
2814    F5D7    3E 80                   mpa1:   ld      a,128           ;set short sector
```

```
2815    F5D9    12                          ld      (de),a
2816    F5DA    0A              mpa2:       ld      a,(bc)
2817    F5DB    CB 47                       bit     fmds,a
2818    F5DD    28 0A                       jr      z,mpa22         ;if not double sided disk
2819    F5DF    7C                          ld      a,h             ;map first 77 tracks to side zero
2820    F5E0    FE 4D                       cp      77
2821    F5E2    38 02                       jr      c,mpa21         ;if side zero tracks
2822    F5E4    D6 4D                       sub     77              ;offset to back side
2823    F5E6    3F              mpa21:      ccf
2824    F5E7    8F                          adc     a,a
2825    F5E8    67                          ld      h,a
2826    F5E9    E5              mpa22:      push    hl              ;save track/sector
2827    F5EA    0A                          ld      a,(bc)          ;get floppy format
2828    F5EB    2A F6EB                     ld      hl,(deflun)     ;get new unit
2829    F5EE    67                          ld      h,a
2830    F5EF    11 FFFF                     ld      de,-1           ;get previously used format/lun
2831    F5F0            lastfm      equ     $-2
2832    F5F2    22 F5F0                     ld      (lastfm),hl     ;save this format/unit for next time
2833    F5F5    B7                          or      a
2834    F5F6    ED 52                       sbc     hl,de
2835    F5F8    28 0C                       jr      z,mpa3          ;if unit and format same as last time
2836    F5FA    32 F6EF                     ld      (flpfrm),a
2837    F5FD    21 F6EA                     ld      hl,deflpy       ;issue define floppy command
2838    F600    CD F643                     call    iccs
2839    F603    CD F669                     call    wcc
2840    F606    E1              mpa3:       pop     hl              ;recover track / sector
2841    F607    44                          ld      b,h             ;set track
2842    F608    11 001A                     ld      de,26           ;compute sector-26-1+(Track+1)*26
2843    F60B    62                          ld      h,d             ;clear upper track
2844    F60C    37                          scf
2845    F60D    ED 52                       sbc     hl,de           ;adjust sector
2846    F60F    04                          inc     b               ;force one pass
2847    F610    19              mpa4:       add     hl,de           ;multiply track by sectors/track
2848    F611    10 FD                       djnz    mpa4            ;if multiply incomplete
2849    F613    7C              mpa5:       ld      a,h             ;swap H & L
2850    F614    65                          ld      h,l
2851    F615    6F                          ld      l,a
2852    F616    22 F6F2                     ld      (addrh),hl      ;Store address in command block
2853    F619    C9                          ret
2854
2855                                        subttl  Sasi Bus Control Interface
2856                                        page
```

```
                                ;;      gca - get controller attention.
                                ;
F61A    CD F6AD         gca:    call    reset           ;reset controller if required
F61D    7E                      ld      a,(hl)          ;get command
F61E    FE 01                   cp      c.recal
F620    3E 0A                   ld      a,9+1           ;set 9+ second time-out
F622    28 02                   jr      z,gca0          ;if recalibrate, use long time-out
F624    3E 03                   ld      a,2+1           ;use short time-out
F626    32 F627         gca0:   ld      (gcaa),a
F627                    gcaa    equ     $-2     ;*****=>;monitor one second timer address goes here
F629    CD F6D2                 call    som             ;set output mode
F62C    3E 01                   ld      a,1             ;Set sasi controller address
F62E    D3 10                   out     (Sasid),a
F630    3E 20                   ld      a,1 shl b.sel   ;assert Select Line
F632    D3 12                   out     (Sasic),a
F634    DB 12           gca1:   in      a,(Sasis)       ;get sasi status
F636    0F                      rrca                    ;get busy bit in C
F637    38 06                   jr      c,gca2          ;if controller is ready
F639    CD F6A5                 call    cft             ;check for time-out
F63C    F2 F634                 jp      p,gca1          ;if not timed out
F63F    AF              gca2:   xor     a
F640    D3 12                   out     (Sasic),a       ;drop Select
F642    C9                      ret
                                ;;      iccs  - issue Controller Command.
                                ;
                                ;       HL => Sasi command block
                                ;
F643    7E              iccs:   ld      a,(hl)          ;peek at opcode
F644    FE 04                   cp      c.fmat
F646    C8                      ret     z               ;do not allow format entire disk
F647    CD F61A                 call    gca             ;get controller attention
F64A    01 0610                 ld      bc,Sasid+6*256  ;set port / command block length
F64D    CD F687         iccs1:  call    wfr             ;wait for REQ
F650    C8                      ret     z               ;if data requested
F651    ED A3                   outi                    ;send next byte
F653    20 F8                   jr      nz,iccs1
F655    C9                      ret
                                ;;      tdo - transmit data out.
                                ;
F656    CD F687         tdo:    call    wfr             ;wait for req
F659    20 0E                   jr      nz,wcc          ;if not data requested
F65B    ED B3                   otir                    ;pitch sector out
F65D    18 0A                   jr      wcc
                                ;;      tdi - transmit data in.
                                ;
F65F    CD F6CE         tdi:    call    sim             ;set input mode
F662    CD F687                 call    wfr             ;wait for req
F665    20 02                   jr      nz,wcc          ;if status, not data requested
F667    ED B2                   inir                    ;read sector
                                ;;      wcc  - wait command complete.
```

```
2912                                    ;
2913    F669    CD F6CE                 wcc:    call    sim             ;set input mode
2914    F66C    CD F687                         call    wfr             ;wait for REQ
2915    F66F    28 2A                           jr      z,ecr           ;if controller not providing status
2916    F671    ED 78                           in      a,(c)           ;read completion status
2917    F673    E6 03                           and     3               ;ignore unused bits
2918    F675    47                              ld      b,a
2919    F676    CD F687                         call    wfr             ;wait for REQ
2920    F679    28 20                           jr      z,ecr           ;if not status
2921    F67B    DB 12                           in      a,(Sasis)       ;recover status
2922    F67D    CB 4F                           bit     b.msg,a
2923    F67F    28 1A                           jr      z,ecr           ;if not message byte
2924    F681    ED 78                           in      a,(c)           ;read message byte
2925    F683    20 16                           jr      nz,ecr          ;if last byte not zero
2926    F685    B0                              or      b               ;set Sasi error status byte
2927    F686    C9                              ret
2928
2929                                    ;;      wfr - wait for REQ.
2930                                    ;
2931                                    ;       Exit:   A < 0   Timer Expired
2932                                    ;               A = 0   Request is for data
2933                                    ;               A > 0   Request is for control
2934                                    ;
2935    F687    CD F6A5                 wfr:    call    cft             ;check for time-out
2936    F68A    FA F69A                         jp      m,wfr1          ;if controller hung
2937    F68D    DB 12                           in      a,(Sasis)       ;read sasi status
2938    F68F    CB 5F                           bit     b.req,a
2939    F691    28 F4                           jr      z,wfr           ;if request not asserted
2940    F693    CB 77                           bit     b.par,a         ;check buss parity
2941    F695    20 03                           jr      nz,wfr1         ;if parity error
2942    F696                            wfra    equ     $-1
2943    F697    E6 04                           and     1 shl b.cd      ;test control / data bit
2944    F699    C9                              ret
2945    F69A    F1                      wfr1:   pop     af              ;pitch return address
2946
2947                                    ;;      ecr - Enable Controller Reset.
2948                                    ;
2949    F69B    AF                      ecr:    xor     a               ;enable controller reset next time
2950    F69C    32 F6AD                         ld      (reset),a       ;by placing NOP at reset entry point
2951    F69F    F6 FF                           or      -1              ;return error status
2952    F6A1    32 F5F0                         ld      (lastfm),a      ;force define floppy format
2953    F6A4    C9                              ret
2954
2955                                    ;;      Cft - Check for Time-out.
2956                                    ;
2957    F6A5    CD F066                 cft:    call    idle            ;idle cpu
2958    F6A8    3A 0000                         ld      a,(0)
2959    F6A9                            cfta    equ     $-2     ;*****=>;This word gets the address of the timer
2960                                            public  cfta
2961    F6AB    B7                              or      a
2962    F6AC    C9                              ret
2963
2964                                    ;;      Reset - Reset Controller.
2965                                    ;
2966                                    ;       This routine is called prior to every command that is
2967                                    ;       issued to the controller, but disables itself after
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Sasi Bus Control Interface

2968                                    ;       running.  When, and if the controller times-out, this
2969                                    ;       routine is re-enabled.  Thus, the controller will be
2970                                    ;       reset again before the next command is issued.
2971                                    ;
2972    F6AD    00                      reset:  nop             ;*****=>;Note RET gets put here after reset
2973                                                            ;       NOP gets put there if time-out
2974    F6AE    3E CF                           ld      a,11001111b     ;initialize pio in mode 3
2975    F6B0    D3 13                           out     (pioBs),a
2976    F6B2    3E 5F                           ld      a,01011111b     ;d7, d5 are outputs
2977    F6B4    D3 13                           out     (pioBs),a
2978    F6B6    3E 80                           ld      a,1 shl b.rst   ;assert reset to controller
2979    F6B8    D3 12                           out     (Sasic),a
2980    F6BA    AF                              xor     a
2981    F6BB    D3 12                           out     (Sasic),a       ;de-assert reset
2982    F6BD    3E C9                           ld      a,0c9h
2983    F6BF    32 F6AD                         ld      (reset),a       ;disable reset until time-out
2984    F6C2    E5                              push    hl
2985    F6C3    21 F6E8                         ld      hl,rgrecal
2986    F6C6    CD F643                         call    iccs            ;issue recursive rigid recalibrate
2987    F6C9    CD F669                         call    wcc
2988    F6CC    E1                              pop     hl
2989    F6CD    C9                              ret
2990
2991                                    ;;      Sim - Set Input Mode.
2992                                    ;
2993    F6CE    3E 4F                   sim:    ld      a,01001111b     ;set pio A input mode
2994    F6D0    18 02                           jr      soml
2995
2996                                    ;;      Som - Set Output Mode.
2997                                    ;
2998    F6D2    3E 0F                   som:    ld      a,00001111b     ;set pio A output mode
2999    F6D4    D3 11                   soml:   out     (pioAs),a
3000    F6D6    C9                              ret
3001
3002                                    ;;      cwp - check write protect.
3003                                    ;
3004    F6D7    0A                      cwp:    ld      a,(bc)          ;get drive type
3005    F6D8    E6 80                           and     fm.hard
3006    F6DA    C8                              ret     z               ;if not rigid disk access
3007    F6DB    3E 00                           ld      a,0             ;get dirty parameter flag
3008    F6DC                            rdonly  equ     $-1
3009    F6DD    B7                              or      a
3010    F6DE    C8                              ret     z               ;if not write protected
3011    F6DF    7A                              ld      a,d
3012    F6E0    B7                              or      a
3013    F6E1    C8                              ret     z               ;if track zero request
3014    F6E2    3A F6F0                         ld      a,(opcode)
3015    F6E5    D6 08                           sub     c.read          ;allow reads, but no writes to file system
3016    F6E7    C9                              ret
3017
3018                                            subttl  Sasi Command Blocks
3019                                            page
```

```
3020
3021
3022                                    ;;      Sasi Command Blocks.
3023                                    ;
3024
3025    F6E8    01                      rgrecal:db      c.recal
3026    F6E9    60                      reclun: db      3 shl 5
3027
3028    F6EA    C0                      deflpy: db      c.flpy          ;define floppy format
3029    F6EB    00                      deflun: db      0
3030    F6EC    00 00 00                        db      0,0,0
3031    F6EF    00                      flpfrm: db      0
3032
3033    F6F0    00                      opcode: db      0               ;Class code / Operation
3034    F6F1    00                      lun:    db      0               ;Logical Unit & Logical Address 20-16
3035    F6F2    00                      addrh:  db      0               ;               Logical Address 15-8
3036    F6F3    00                      addrl:  db      0               ;               Logical Address  7-0
3037    F6F4    01                      nblk:   db      1               ;Number of Blocks
3038    F6F5    00                      dctrl:  db      0               ;Error Retry Disable Control word
3039
3040                                            subttl  Overlayable Initialization Code
3041                                            page
```

```
3042
3043                                    ;;      First - First time only.
3044                                    ;
3045    F6F6    F5                      first:  push    af
3046                                            phex    $-dskdvr,<SASI Resident Length is>
3047    0010                    +               .radix  16
3048    000A                    +               .radix  10
3049    F6F7    3E C9                           ld      a,0c9h          ;nuke self first time
3050    F6F9    32 F6F6                         ld      (first),a
3051    F6FC    26 00                           ld      h,0             ;indicate return register value
3052    F6FE    CD F039                         call    dayti           ;get address of monitor timer
3053    F701    2B                              dec     hl
3054    F702    22 F6A9                         ld      (cfta),hl       ;store address of timer for check routine
3055    F705    22 F627                         ld      (gcaa),hl       ;and for command startup
3056
3057    F708    21 F767                 first1: ld      hl,cnfdpb       ;point to physical driver read command
3058    F70B    CD F4B0                         call    sai403          ;read partition parameters
3059    F70E    B7                              or      a
3060    F70F    CC F723                         call    z,cpb           ;if no errors then check parameter blocks
3061    F712    28 0D                           jr      z,first2        ;if parameters are loaded
3062    F714    21 F76C                         ld      hl,cnfdpb+5     ;try backup heads
3063    F717    7E                              ld      a,(hl)          ;get logical sector
3064    F718    C6 20                           add     a,32            ;advance to next head
3065    F71A    77                              ld      (hl),a
3066    F71B    20 EB                           jr      nz,first1       ;if 4 heads and 2 cylinders not attempted
3067    F71D    2F                              cpl                     ;set tracks > 0 read only
3068    F71E    32 F6DC                         ld      (rdonly),a
3069    F721    F1                      first2: pop     af
3070    F722    C9                              ret
3071
3072                                    ;;      cpb - check parameter blocks.
3073                                    ;
3074    F723    21 EE00                 cpb:    ld      hl,rgdbuf       ;point to dpb buffer
3075    F726    3A F76C                         ld      a,(cnfdpb+5)    ;get sector this dpb set came from
3076    F729    FE 20                           cp      32
3077    F72B    20 04                           jr      nz,cpb1         ;if not primary set
3078    F72D    7E                              ld      a,(hl)
3079    F72E    FE E5                           cp      0e5h
3080    F730    C8                              ret     z               ;use default dpbs if none configured
3081    F731    11 000F                 cpb1:   ld      de,16-1         ;set offset from high spt to deblock control
3082    F734    06 04                           ld      b,4             ;verify 4 dpbs
3083    F736    7E                      cpb2:   ld      a,(hl)          ;set low sectors / track
3084    F737    B7                              or      a
3085    F738    C0                              ret     nz              ;if bummer sectors / track
3086    F739    19                              add     hl,de           ;advance to deblock control
3087    F73A    7E                              ld      a,(hl)
3088    F73B    E6 87                           and     87h
3089    F73D    CB 2F                           sra     a
3090    F73F    CB 2F                           sra     a
3091    F741    C0                              ret     nz              ;if bad deblocking constant
3092    F742    23                              inc     hl
3093    F743    10 F1                           djnz    cpb2
3094    F745    2B                              dec     hl
3095    F746    11 F4AF                         ld      de,Dpbrg4+16*4-1
3096    F749    01 0040                         ld      bc,16*4
```

```
3097    F74C    ED B8                       lddr                    ;move dpbs into place
3098    F74E    01 0010                     ld      bc,16           ;set 16 bytes/dpx
3099    F751    11 F3DE                     ld      de,Dpbase+4*16+14 ;point at alloc vector pointer
3100    F754    21 F47D                     ld      hl,Dpbrg4+13    ;point at reserved tracks
3101    F757    3E 04                       ld      a,4             ;count off 4 partitions
3102    F759    F5                  cpb3:   push    af
3103    F75A    7E                          ld      a,(hl)          ;get reserved tracks for partition
3104    F75B    3D                          dec     a               ;just so nice numbers come out
3105    F75C    87                          add     a,a             ;16 blks/track / 8 blks/byte = 2 bytes/track
3106    F75D    12                          ld      (de),a          ;store low allocation vector address
3107    F75E    09                          add     hl,bc           ;advance to next dpb
3108    F75F    EB                          ex      de,hl
3109    F760    09                          add     hl,bc           ;advance to next dph
3110    F761    EB                          ex      de,hl
3111    F762    F1                          pop     af
3112    F763    3D                          dec     a
3113    F764    20 F3                       jr      nz,cpb3         ;if more to allocate
3114    F766    C8                          ret     z               ;return success
3115
3116    F767    01 04 00            cnfdpb: db      01,4,0          ;read partition 0
3117    F76A    0000                        dw      0               ;track zero
3118    F76C    0020                        dw      32              ;sector 32
3119    F76E    EE00                        dw      rgdbuf          ;rigid parameter table buffer
3120
3121    0300                        sasidl  equ     $-sasstr
3122                                        .dephase
3123    FA08                        dloc    defl    dloc+sasidl
3124                                        .phase  dloc
3125
3126                                        above
3127    0A08"                   +           d&seg
3128
3129    FA08                        Dvrlmt:                         ;disk driver limit
3130
3131    FA08                        rqtop   equ     $               ;set required top of resident monitor
3132
3133    FA08    21 0000             slerr:  ld      hl,0
3134    FA0B    F6 FF                       or      -1
3135    FA0D    C9                          ret
3136
3137                                        subttl  820 Style Disk Driver Emulator
3138                                        page
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
820 Style Disk Driver Emulator

 3139
 3140                                   ;;      820 Style Disk Driver Emulator.
 3141                                   ;
 3142                                           above                   ;generate code in ram
 3143     0A0E"                         +       d&seg
 3144
 3145     FA0E    FF                    phycmd: defb    -1              ;physical Driver Command
 3146     FA0F    FF                    phyunt: defb    -1              ;physical unit
 3147     FA10    00                    phydrv: defb    00              ;logical unit
 3148     FA11    0000                  phytrk: defw    00              ;track
 3149     FA13    0001                  physec: defw    01              ;sector
 3150     FA15    ED80                  phydma: defw    bootbf          ;dma address
 3151
 3152                                   ;;      Select - Select Unit for I/O.
 3153                                   ;
 3154                                   ;       Entry:  C  = Unit
 3155                                   ;       Exit:   A  = 0  if no errors
 3156                                   ;               A  = -1 if errors
 3157                                   ;
 3158     FA17    79                    select: ld      a,c             ;set drive selected
 3159     FA18    FE 08                         cp      8
 3160     FA1A    30 1D                         jr      nc,sel1
 3161     FA1C    32 FA10                       ld      (phydrv),a      ;save logical CP/M drive
 3162     FA1F    21 FA5A                       ld      hl,seltbl       ;set select table address
 3163     FA22    06 00                         ld      b,0
 3164     FA24    09                            add     hl,bc           ;index into select table
 3165     FA25    7E                            ld      a,(hl)
 3166     FA26    B7                            or      a
 3167     FA27    EB                            ex      de,hl
 3168     FA28    67                            ld      h,a             ;in case previous select worked, say no dph
 3169     FA29    6F                            ld      l,a             ;to internal routines
 3170     FA2A    C8                            ret     z               ;if drive has already been selected
 3171     FA2B    D5                            push    de              ;save table address
 3172     FA2C    06 FF                         ld      b,-1            ;set Select operation
 3173     FA2E    CD FA51                       call    xqphys          ;execute physical driver
 3174     FA31    7D                            ld      a,l             ;get returned dph address
 3175     FA32    B4                            or      h
 3176     FA33    D1                            pop     de              ;retrieve select table address
 3177     FA34    28 03                         jr      z,sel1          ;if select unsuccessful
 3178     FA36    AF                            xor     a               ;return no errors
 3179     FA37    12                            ld      (de),a          ;prevent more density re-selects
 3180     FA38    C9                            ret
 3181     FA39    F6 FF                 sel1:   or      -1              ;return error
 3182     FA3B    C9                            ret
 3183
 3184                                   ;;      Home - Position to track zero.
 3185                                   ;
 3186     FA3C    0E 00                 home:   ld      c,0             ;force track zero
 3187
 3188                                   ;;      Seek - Seek Track.
 3189                                   ;
 3190                                   ;       Entry:  C  = Track to read/write from next
 3191                                   ;
 3192     FA3E    79                    seek:   ld      a,c
 3193     FA3F    32 FA11                       ld      (phytrk),a
```

```
3194    FA42    AF                      xor     a               ;return no errors
3195    FA43    C9                      ret
3196
3197                            ;;      Write - Write Physical Sector.
3198                            ;
3199    FA44    06 00           write:  ld      b,0             ;set Write operation
3200    FA46    18 02                   jr      rdwr
3201
3202                            ;;      Read - Read Physical Sector.
3203                            ;
3204    FA48    06 01           read:   ld      b,1             ;set read operation
3205
3206                            ;;      Rdwr - Read/Write Processor.
3207                            ;
3208                            ;       Entry:  C  = Sector
3209                            ;               HL = Transfer Address
3210                            ;       Exit:   A  = 0 if no errors
3211                            ;               A  = -1 if errors
3212                            ;
3213    FA4A    79              rdwr:   ld      a,c
3214    FA4B    32 FA13                 ld      (physec),a      ;set physical sector
3215    FA4E    22 FA15                 ld      (phydma),hl     ;set transfer address
3216
3217                            ;;      xqphys - Internal Execute Physical Driver.
3218                            ;
3219                            ;       Entry:  B  = -1 for Select
3220                            ;               B  =  0 for Write
3221                            ;               B  =  1 for Read
3222                            ;
3223    FA51    21 FA0E         xqphys: ld      hl,phycmd       ;point to physical command block
3224    FA54    70                      ld      (hl),b          ;store operation
3225    FA55    CD F344                 call    xqdvr           ;execute driver
3226    FA58    B7                      or      a               ;set flags
3227    FA59    C9                      ret
3228
3229                            ;;      Emulator Disk I/O Ram.
3230                            ;
3231    FA5A    FF FF FF FF     seltbl: defb    -1,-1,-1,-1     ;drive already selected table
3232    FA5E    FF FF FF FF             defb    -1,-1,-1,-1
3233
3234                                    subttl  Command processor
3235                                    page
```

```
                                        above                   ;put code upstairs
0A62"                           +       d&seg
                                ;;      prompt user for command.
                                ;
FA62    FB                      prompt: ei
FA63    31 0000                         ld      sp,stack        ;reset system stack
FA66    CD FC3D                         call    pnext
FA69    0D 0A                           defb    cr,lf
FA6B    2A 20                           defm    '* '
FA6D    04                              defb    eot
FA6E    21 FF5C                         ld      hl,linbuf
FA71    0E 50                           ld      c,80            ;buffer of 80 chars (ver. 2.0)
FA73    CD FB37                         call    getlin          ;input a bufered console line
FA76    38 51                           jr      c,what          ;print 'what ?' if input error
FA78    3A FF5C                 autobt: ld      a,(linbuf)      ;get first character in line
FA7B    FE 0D                           cp      cr
FA7D    28 E3                           jr      z,prompt        ;jump if a null line
FA7F    D6 40                           sub     '@'
FA81    FE 1B                           cp      'Z'-'@'+1
FA83    30 44                           jr      nc,what         ;if not letter
FA85    87                              add     a,a
FA86    4F                              ld      c,a
FA87    06 00                           ld      b,0
FA89    21 FAD9                         ld      hl,cmdtab+1     ;index command table with character
FA8C    09                              add     hl,bc
FA8D    7E                              ld      a,(hl)
FA8E    2B                              dec     hl
FA8F    6E                              ld      l,(hl)          ;get address of command processor
FA90    67                              ld      h,a
FA91    E6 80                           and     80h
FA93    20 17                           jr      nz,prmt1        ;if resident command
FA95    11 FC55                         ld      de,cloc         ;move transient command to RAM area
FA98    D5                              push    de
FA99    01 0299                         ld      bc,tpamax       ;set length of largest transient
FA9C    F3                              di
FA9D    DB 1C                           in      a,(syspio)
FA9F    B7                              or      a
FAA0    F5                              push    af
FAA1    F4 F29C                         call    p,crton         ;enable rom if disabled
FAA4    ED B0                           ldir
FAA6    F1                              pop     af
FAA7    F4 F293                         call    p,crtoff        ;disable rom if enabled
FAAA    FB                              ei
FAAB    E1                              pop     hl              ;set execution address
FAAC    E5                      prmt1:  push    hl
FAAD    CD FC36                         call    crlf
FAB0    FD 21 FF5D                      ld      iy,linbuf+1
FAB4    CD FB5F                         call    params          ;input numeric parameters from
FAB7    DD E1                           pop     ix              ;line buffer and test if error
FAB9    2A FFB5                         ld      hl,(param1)
FABC    ED 5B FFB7                      ld      de,(param2)
FAC0    ED 4B FFB9                      ld      bc,(param3)
FAC4    CD FAD6                         call    jpix            ;call subroutine @ ix
FAC7    30 99                           jr      nc,prompt       ;go back to prompt if no errors
```

```
3291
3292  FAC9  CD FC3D              what:  call  pnext
3293  FACC  20 77 68 61                 defm  ' what?'
3294  FAD0  74 3F
3295  FAD2  07                           defb  'G'-64          ;say 'what ?' and squeal
3296  FAD3  04                           defb  eot
3297  FAD4  18 8C                        jr    prompt
3298
3299  FAD6  DD E9                jpix:  jp    (ix)            ;call subroutine @ ix
3300
3301  FAD8  177B                 cmdtab: defw  help            ;@ - Help user
3302  FADA  11B8                         defw  boot            ;a - boot cp/m
3303  FADC  1353                         defw  baud            ;b - bit rate
3304  FADE  1436                         defw  block           ;c - memory block move
3305  FAE0  12F2                         defw  memdmp          ;d - dump memory in hex/ascii
3306  FAE2  1315                         defw  view            ;e - enter memory
3307  FAE4  1428                         defw  fill            ;f - fill memory
3308  FAE6  12DB                         defw  goto            ;g - goto program
3309  FAE8  14E2                         defw  term            ;h - host terminal
3310  FAEA  13CA                         defw  incmd           ;i - read from input port
3311  FAEC  FAC9                         defw  what            ;j - not used
3312  FAEE  FAC9                         defw  what            ;k - not used
3313  FAF0  11B8                         defw  boot            ;l - load system
3314  FAF2  1315                         defw  view            ;m - memory examine/change
3315  FAF4  FAC9                         defw  what            ;n - not used
3316  FAF6  13F1                         defw  outcmd          ;o - write to output port
3317  FAF8  1459                         defw  proto           ;p - printer protocol
3318  FAFA  FAC9                         defw  what            ;q - not used
3319  FAFC  1367                         defw  dskcmd          ;r - display disk sector data
3320  FAFE  FAC9                         defw  what            ;s - not used
3321  FB00  1477                         defw  type            ;t - typewriter mode
3322  FB02  FAC9                         defw  what            ;u - not used
3323  FB04  1443                         defw  vercmd          ;v - memory block compare
3324  FB06  1367                         defw  dskcmd          ;w - disc sector write command
3325  FB08  13FB                         defw  test            ;x - ram diagnostic
3326  FB0A  FAC9                         defw  what            ;y - not used
3327  FB0C  FAC9                         defw  what            ;z - not used
3328  0036                       cmdsiz  equ   $-cmdtab
3329
3330  FB0E  BE                   check:  cp    (hl)
3331  FB0F  C8                           ret   z               ;return if (hl)=a
3332  FB10  F5                           push  af
3333  FB11  CD FB22                      call  mdata           ;print what was actually read
3334  FB14  CD FC3D                      call  pnext
3335  FB17  73 68 6F 75                  defm  'should='
3336  FB1B  6C 64 3D
3337  FB1E  04                           defb  eot
3338  FB1F  F1                           pop   af
3339  FB20  18 07                        jr    put2j
3340
3341  FB22  CD FC36              mdata:  call  crlf
3342  FB25  CD FC16                      call  put4hs
3343  FB28  7E                           ld    a,(hl)
3344  FB29  C3 FC1B              put2j:  jp    put2hs
3345
3346                                     subttl  Console support routines
```

```
3348
3349                                    ;;      getlin - read line into buffer.
3350                                    ;
3351    FB2C    3E 40           gethlp: ld      a,'@'
3352    FB2E    32 FF5C                 ld      (linbuf),a
3353    FB31    3E 0D                   ld      a,cr
3354    FB33    32 FF5D                 ld      (linbuf+1),a
3355    FB36    C9                      ret
3356    FB37    41              getlin: ld      b,c             ;save max line length parameter in b
3357    FB38    CD FC27         glin1:  call    echo            ;get a character from the console
3358    FB3B    FE 1E                   cp      Helpkey
3359    FB3D    28 ED                   jr      z,gethlp        ;if user needs help
3360    FB3F    77                      ld      (hl),a
3361    FB40    FE 0D                   cp      cr              ;check for carriage return
3362    FB42    C8                      ret     z               ;if end of line
3363    FB43    FE 08                   cp      'H'-64          ;check for ctl-h backspace
3364    FB45    28 09                   jr      z,glin4
3365    FB47    FE 20                   cp      ' '
3366    FB49    D8                      ret     c               ;other control characters are illegal
3367    FB4A    23                      inc     hl              ;store character in buffer
3368    FB4B    0D                      dec     c
3369    FB4C    20 EA                   jr      nz,glin1        ;get another if there's more room
3370    FB4E    37                      scf
3371    FB4F    C9                      ret                     ;return with carry=1 if too
3372                                                            ;many characters are entered
3373    FB50    2B              glin4:  dec     hl              ;delete last character from buffer
3374    FB51    CD FC3D                 call    pnext
3375    FB54    20 08                   defb    ' ','H'-64      ;delete character from screen
3376    FB56    04                      defb    eot
3377    FB57    0C                      inc     c
3378    FB58    78                      ld      a,b             ;set max line length
3379    FB59    91                      sub     c
3380    FB5A    30 DC                   jr      nc,glin1        ;if backspace not past the start of the line
3381    FB5C    C9                      ret
3382
3383    FB5D    FD 23           para0:  inc     iy              ;advance character scan
3384    FB5F    01 00FF         params: ld      bc,low -1       ;set parameter index
3385    FB62    FD 7E 00                ld      a,(iy+0)        ;fetch character
3386    FB65    D6 0D                   sub     cr
3387    FB67    C8                      ret     z               ;if no parameters
3388    FB68    D6 13                   sub     ' '-cr
3389    FB6A    28 F1                   jr      z,para0         ;if leading blanks
3390    FB6C    0C              para1:  inc     c               ;advance parameter index
3391    FB6D    CB 51                   bit     2,c
3392    FB6F    37                      scf
3393    FB70    C0                      ret     nz              ;error if > 4 numbers entered
3394    FB71    C5              para2:  push    bc              ;save parameter count
3395    FB72    CD FBDA                 call    gethex          ;read a number from line buffer
3396    FB75    C1                      pop     bc
3397    FB76    DD 21 FFB5      para4:  ld      ix,param1       ;point to parameter storage area
3398    FB7A    DD 09                   add     ix,bc           ;add parameter count in bc
3399    FB7C    DD 09                   add     ix,bc
3400    FB7E    DD 75 00                ld      (ix+0),l
3401    FB81    DD 74 01                ld      (ix+1),h        ;store data returned from 'GETHEX'
3402    FB84    FE 20                   cp      ' '
```

```
 3403  FB86  28 E4                    jr      z,para1         ;get another item if space
 3404  FB88  FE 2C                    cp      ','
 3405  FB8A  28 E0                    jr      z,para1         ;get another item if comma
 3406  FB8C  79                       ld      a,c             ;set parameter count
 3407  FB8D  3C                       inc     a
 3408  FB8E  C9                       ret
 3409
 3410                          ;;      dump - dump memory.
 3411                          ;
 3412  FB8F  E5                dump:   push    hl              ;save starting address
 3413  FB90  CD FC16                   call    put4hs          ;print starting address in hex
 3414  FB93  CD FC1E                   call    space
 3415  FB96  06 10                     ld      b,16
 3416  FB98  3E 0F             dump2:  ld      a,16-1          ;skip 3 columns on 16 byte boundry
 3417  FB9A  CD FC23                   call    dmpfmt
 3418  FB9D  3E 07                     ld      a,8-1           ;skip 2 columns on 8 byte boundry
 3419  FB9F  CD FC23                   call    dmpfmt
 3420  FBA2  3E 03                     ld      a,4-1           ;skip 1 column  on 4 byte boundry
 3421  FBA4  CD FC23                   call    dmpfmt
 3422  FBA7  7E                        ld      a,(hl)          ;get a data byte @ hl
 3423  FBA8  23                        inc     hl
 3424  FBA9  CD FC1B                   call    put2hs          ;print the data in hex
 3425  FBAC  10 EA                     djnz    dump2           ;repeat 16 times
 3426  FBAE  CD FC1E                   call    space
 3427  FBB1  E1                        pop     hl              ;restore starting address
 3428  FBB2  06 10                     ld      b,16
 3429  FBB4  3E 1F             dump3:  ld      a,1fh           ;force next character
 3430  FBB6  CD F00C                   call    conout
 3431  FBB9  7E                        ld      a,(hl)          ;get back data byte @ hl
 3432  FBBA  23                        inc     hl
 3433  FBBB  CD F00C                   call    conout          ;print ascii character in a
 3434  FBBE  10 F4                     djnz    dump3
 3435  FBC0  CD F006                   call    const           ;check console status
 3436  FBC3  28 0C                     jr      z,dump4         ;if char not ready
 3437  FBC5  CD F009                   call    conin           ;read char
 3438  FBC8  FE 0D                     cp      cr
 3439  FBCA  C8                        ret     z               ;if user abort
 3440  FBCB  CD F009                   call    conin           ;pause while user examines display
 3441  FBCE  FE 0D                     cp      cr
 3442  FBD0  C8                        ret     z               ;if user found it
 3443  FBD1  CD FC36           dump4:  call    crlf            ;send end of line
 3444  FBD4  1B                        dec     de
 3445  FBD5  7A                        ld      a,d
 3446  FBD6  B3                        or      e
 3447  FBD7  20 B6                     jr      nz,dump         ;if dump not complete
 3448  FBD9  C9                        ret
 3449
 3450                          ;;      gethex converts ascii to binary.
 3451                          ;
 3452                          ;       carry set on illegal conversion result
 3453                          ;       terminating character returns in a.
 3454                          ;       hl returns with 16 bit binary integer
 3455                          ;
 3456  FBDA  21 0000           gethex: ld      hl,0            ;preset result
 3457  FBDD  54                        ld      d,h
 3458  FBDE  5D                        ld      e,l
```

```
3459    FBDF    29                      gnum1:  add     hl,hl           ;multiply result by 16
3460    FBE0    29                              add     hl,hl
3461    FBE1    29                              add     hl,hl
3462    FBE2    29                              add     hl,hl
3463    FBE3    19                              add     hl,de           ;append next digit
3464    FBE4    FD 7E 00                gnum3:  ld      a,(iy+0)        ;get next character from line buffer
3465    FBE7    4F                              ld      c,a
3466    FBE8    FD 23                           inc     iy              ;advance buffer address
3467    FBEA    CD FBF3                         call    hexbin          ;convert one ascii hex to binary
3468    FBED    5F                              ld      e,a
3469    FBEE    30 EF                           jr      nc,gnum1
3470    FBF0    79                              ld      a,c             ;return first non hex digit
3471    FBF1    B7                              or      a
3472    FBF2    C9                              ret
3473
3474                                    ;;      hexbin - convert hex to binary.
3475                                    ;
3476    FBF3    D6 30                   hexbin: sub     '0'
3477    FBF5    D8                              ret     c
3478    FBF6    FE 0A                           cp      10
3479    FBF8    3F                              ccf
3480    FBF9    D0                              ret     nc
3481    FBFA    D6 07                           sub     7
3482    FBFC    FE 0A                           cp      10
3483    FBFE    D8                              ret     c
3484    FBFF    FE 10                           cp      16
3485    FC01    3F                              ccf
3486    FC02    C9                              ret
3487
3488    FC03    F5                      put2hx: push    af
3489    FC04    1F                              rra
3490    FC05    1F                              rra
3491    FC06    1F                              rra
3492    FC07    1F                              rra
3493    FC08    CD FC0C                         call    putnib
3494    FC0B    F1                              pop     af
3495    FC0C    E6 0F                   putnib: and     00001111b
3496    FC0E    C6 90                           add     a,90h
3497    FC10    27                              daa
3498    FC11    CE 40                           adc     a,40h
3499    FC13    27                              daa
3500    FC14    18 0A                           jr      output
3501
3502    FC16    7C                      put4hs: ld      a,h
3503    FC17    CD FC03                         call    put2hx
3504    FC1A    7D                              ld      a,l
3505    FC1B    CD FC03                 put2hs: call    put2hx
3506
3507                                    ;;      space - output space.
3508                                    ;
3509    FC1E    3E 20                   space:  ld      a,' '           ;fall through to output space
3510
3511    FC20    C3 F00C                 output: jp      conout          ;display character
3512
3513                                    ;;      dmpfmt - Dump Command Output Formatter.
3514                                    ;
```

```
3515    FC23    A5                  dmpfmt: and     1               ;check address boundry
3516    FC24    C0                          ret     nz              ;if not on boundry
3517    FC25    18 F7                       jr      space           ;skip one column
3518
3519                                ;;      echo - read and echo console character.
3520                                ;
3521                                ;       Echo inputs one character from the console
3522                                ;       device, prints it on the console output and
3523                                ;       then returns it in the A register in upper case.
3524                                ;
3525    FC27    CD F009             echo:   call    conin           ;input a character and echo it
3526    FC2A    FE 1E                       cp      Helpkey
3527    FC2C    C8                          ret     z               ;do not echo help key
3528    FC2D    CD F00C                     call    conout
3529    FC30    FE 61                       cp      'a'
3530    FC32    D8                          ret     c               ;if not lower case
3531    FC33    D6 20                       sub     'a'-'A'         ;convert lower case to upper case
3532    FC35    C9                          ret
3533
3534                                ;;      crlf - carriage return-linefeed.
3535                                ;
3536    FC36    CD FC3D             crlf:   call    pnext           ;print next message
3537    FC39    0D 0A 04                    defb    cr,lf,eot
3538    FC3C    C9                          ret
3539
3540                                ;;      pnext - print message after call.
3541                                ;
3542    FC3D    E3                  pnext:  ex      (sp),hl         ;set message address
3543    FC3E    7E                          ld      a,(hl)
3544    FC3F    23                          inc     hl
3545    FC40    E3                          ex      (sp),hl         ;set return address
3546    FC41    FE 04                       cp      eot
3547    FC43    C8                          ret     z
3548    FC44    CD F00C                     call    conout
3549    FC47    18 F4                       jr      pnext
3550
3551                                        if      options and (o.move or o.verf)
3552                                ;;      set block address for move and verify.
3553                                ;
3554    FC49    EB                  blocad: ex      de,hl
3555    FC4A    B7                          or      a               ;clear carry
3556    FC4B    ED 52                       sbc     hl,de           ;get diffrence between
3557    FC4D    EB                          ex      de,hl           ;hl & de for bytecount
3558    FC4E    D5                          push    de              ;exchange de,bc
3559    FC4F    50                          ld      d,b
3560    FC50    59                          ld      e,c
3561    FC51    C1                          pop     bc
3562    FC52    03                          inc     bc              ;get count+1 into bc
3563    FC53    C9                          ret
3564                                        endif
3565    FC54    C9                          ret
3566
3567                                        subttl  Transient Command Area
3568                                        page
```

```
3569
3570    FC55                            cloc    defl    $               ;establish overlay execution address
3571                                            overlay start
3572    0C55"                   +               c&seg
3573                                    ;;      signon - Announce System Ready.
3574                                    ;
3575    FC55    08                      signon: ex      af,af'
3576    FC56    CD F293                         call    crtoff          ;disable rom/ram
3577    FC59    08                              ex      af,af'
3578    FC5A    28 4A                           jr      z,sign4         ;if Rx1984 loaded disk driver
3579    FC5C    08                              ex      af,af'          ;get syspio data
3580    FC5D    21 F091                         ld      hl,confg        ;point to configuration byte
3581    FC60    CB 47                           bit     0,a
3582    FC62    28 24                           jr      z,sign3         ;if SASI interface present
3583    FC64    F3                              di
3584    FC65    3E CF                           ld      a,11001111b     ;set Pio B in Bit Mode
3585    FC67    D3 1D                           out     (sysctl),a
3586    FC69    3E 38                           ld      a,00111000b     ;turn around d0,1,2
3587    FC6B    D3 1D                           out     (sysctl),a
3588    FC6D    D3 1C                           out     (syspio),a      ;drop all drive selects
3589    FC6F    3E D0                           ld      a,0d0h          ;reset wd-1797-02
3590    FC71    D3 10                           out     (wd1797),a
3591    FC73    10 FE                   sign1:  djnz    sign1           ;wait 1797 not busy
3592    FC75    DB 1C                           in      a,(syspio)
3593    FC77    CB 67                           bit     c.five,a
3594    FC79    3E 02                           ld      a,2             ;preset 10 msec step rate
3595    FC7B    20 04                           jr      nz,sign2        ;if not 5"
3596    FC7D    CB E6                           set     c.five,(hl)
3597    FC7F    3E 03                           ld      a,3             ;set long step
3598    FC81    D3 10                   sign2:  out     (wd1797),a      ;restore / unload heads
3599    FC83    32 FF54                         ld      (steprt),a
3600    FC86    18 1E                           jr      sign4
3601    FC88    CB F6                   sign3:  set     c.sasi,(hl)     ;set Sasi card installed
3602    FC8A    21 F708                         ld      hl,Rigdpb       ;set address of rigid dpb
3603    FC8D    11 F470                         ld      de,dpb5s        ;set address of 5.25" floppy dpb
3604    FC90    01 0300                         ld      bc,Sasidl       ;set sasi driver length
3605    FC93    ED B0                           ldir                    ;Move driver down
3606    FC95    E6 02                           and     2
3607    FC97    20 0D                           jr      nz,sign4        ;if not A/E swap
3608    FC99    21 F361                         ld      hl,Seltab+1
3609    FC9C    06 08                           ld      b,8
3610    FC9E    7E                      sign3a: ld      a,(hl)
3611    FC9F    EE 04                           xor     4
3612    FCA1    77                              ld      (hl),a
3613    FCA2    23                              inc     hl
3614    FCA3    23                              inc     hl
3615    FCA4    10 F8                           djnz    sign3a
3616
3617    FCA6    CD FC3D                 sign4:  call    pnext
3618    FCA9    1A                              defb    clrs            ;clear screen
3619    FCAA    1B 38                           defb    esc,'8'         ;set low light as default mode
3620    FCAC    38 32 30 2D                     defm    '820-II v '
3621    FCB0    49 49 20 76
3622    FCB4    20
3623    FCB5    34 2E 30 31                     defb    rev/100+'0','.',(rev mod 100)/10+'0',(rev mod 10)+'0'
```

```
3624    FCB9    20 1F 1C 20                     defm    ' ',31,28;' 1982 Xerox Corp'
3625    FCBD    31 39 38 32
3626    FCC1    20 58 65 72
3627    FCC5    6F 78 20 43
3628    FCC9    6F 72 70
3629    FCCC    0D 0A                           defb    cr,lf
3630    FCCE    0A                              defb    lf
3631    FCCF    4C 20 2D 20                     defm    'L - Load System'
3632    FCD3    4C 6F 61 64
3633    FCD7    20 53 79 73
3634    FCDB    74 65 6D
3635    FCDE    0D 0A                           defb    cr,lf
3636
3637                                            if      options and o.term
3638    FCE0    48 20 2D 20                     defm    'H - Host Terminal'
3639    FCE4    48 6F 73 74
3640    FCE8    20 54 65 72
3641    FCEC    6D 69 6E 61
3642    FCF0    6C
3643    FCF1    0D 0A                           defb    cr,lf
3644                                            endif
3645                                            if      options and o.type
3646    FCF3    54 20 2D 20                     defm    'T - Typewriter'
3647    FCF7    54 79 70 65
3648    FCFB    77 72 69 74
3649    FCFF    65 72
3650    FD01    0D 0A                           defb    cr,lf
3651                                            endif
3652    FD03    07 04                           defb    7,eot
3653
3654    FD05    CD F006             eatkey:     call    const
3655    FD08    CA F003                         jp      z,warm          ;go enter monitor
3656    FD0B    CD F009                         call    conin
3657    FD0E    18 F5                           jr      eatkey
3658
3659                                            subttl  I/O byte Drivers
3660                                            page
```

```
                                        overlay iobdvr
00BB'                           +       c&seg

                                        .dephase
                                        .phase  iobloc

                                ;;      comins - Communications input status.
                                ;
F770    DB 06                   comins: in      a,(siocpa)
F772    0F                              rrca
F773    9F                              sbc     a,a
F774    C9                              ret

                                ;;      cominp - Communications input data.
                                ;
F775    DB 06                   cominp: in      a,(siocpa)
F777    0F                              rrca
F778    30 FB                           jr      nc,cominp
F77A    DB 04                           in      a,(siodpa)
F77C    C3 F0E2                         jp      kbmask

                                ;;      comout - Communications output.
                                ;
F77F    CD F788                 comout: call    comots
F782    28 FB                           jr      z,comout
F784    79                              ld      a,c
F785    D3 04                           out     (siodpa),a
F787    C9                              ret

                                ;;      comots - Communications output status.
                                ;
F788    DB 06                   comots: in      a,(siocpa)
F78A    E6 04                           and     4
F78C    C8                              ret     z
F78D    F6 FF                           or      -1
F78F    C9                              ret

                                ;;      coniob - get console i/o byte.
                                ;
F790    3A 0003                 coniob: ld      a,(iobyte)
F793    E6 03                           and     00000011b
F795    C9                              ret

                                ;;      iocono - Console output through iobyte.
                                ;
F796    CD F790                 iocono: call    coniob
F799    28 E4                           jr      z,comout
F79B    3D                              dec     a
F79C    CA F2FE                         jp      z,fastcrt
F79F    79                              ld      a,c
F7A0    C3 F0F8                         jp      sioout

                                ;;      iocons - Console status through iobyte.
                                ;
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
I/O byte Drivers

 3716     F7A3    CD F790              iocons: call    coniob
 3717     F7A6    28 C8                        jr      z,comins
 3718     F7A8    3D                           dec     a
 3719     F7A9    CA F0CD                      jp      z,kbdst
 3720     F7AC    C3 F0E5                      jp      siost
 3721
 3722                                  ;;      ioconi - Console input through iobyte.
 3723                                  ;
 3724     F7AF    CD F790              ioconi: call    coniob
 3725     F7B2    28 C1                        jr      z,cominp
 3726     F7B4    3D                           dec     a
 3727     F7B5    CA F0D8                      jp      z,kbdin
 3728     F7B8    C3 F0F0                      jp      sioin
 3729
 3730                                  ;;      lstout - List output through iobyte.
 3731                                  ;
 3732     F7BB    3A 0003              iolist: ld      a,(iobyte)
 3733     F7BE    E6 C0                        and     11000000b
 3734     F7C0    28 BD                        jr      z,comout
 3735     F7C2    EA F7DC                      jp      pe,pioout
 3736     F7C5    79                           ld      a,c
 3737     F7C6    FA F0F8                      jp      m,sioout
 3738     F7C9    C3 F2FE                      jp      fastcrt
 3739
 3740                                  ;;      List output through iobyte
 3741                                  ;
 3742     F7CC    3A 0003              iolsts: ld      a,(iobyte)
 3743     F7CF    E6 C0                        and     11000000b
 3744     F7D1    28 B5                        jr      z,comots
 3745     F7D3    EA F7F4                      jp      pe,piosto
 3746     F7D6    FA F105                      jp      m,siordy
 3747     F7D9    F6 FF                        or      -1
 3748     F7DB    C9                           ret
 3749
 3750                                  ;;      Parallel Output Driver.
 3751                                  ;
 3752     F7DC    CD F7F4              pioout: call    piosto
 3753     F7DF    28 FB                        jr      z,pioout        ;if printer not ready
 3754     F7E1    79                           ld      a,c
 3755     F7E2    D3 08                        out     (gpioda),a      ;load character data
 3756     F7E4    DB 0A                        in      a,(gpiodb)
 3757     F7E6    CB 97                        res     p.strb,a        ;assert strobe
 3758     F7E8    D3 0A                        out     (gpiodb),a
 3759     F7EA    CB D7                        set     p.strb,a        ;release stobe
 3760     F7EC    D3 0A                        out     (gpiodb),a
 3761     F7EE    3E 0A                        ld      a,10            ;delay for ACK
 3762     F7F0    3D                   piol:   dec     a
 3763     F7F1    20 FD                        jr      nz,piol
 3764     F7F3    C9                           ret
 3765
 3766                                  ;;      Parallel Output Status.
 3767                                  ;
 3768     F7F4    DB 0A                piosto: in      a,(gpiodb)      ;read status
 3769     F7F6    2F                           cpl
 3770     F7F7    E6 10                        and     1 shl p.rdyo
 3771     F7F9    C8                           ret     z               ;if ready
```

```
3772    F7FA    F6 FF                       or      -1
3773    F7FC    C9                          ret
3774
3775    008D                        iobdvs  equ     $-iobloc
3776
3777                                        .dephase
3778                                        .phase  cloc+iobdvs
3779
3780                                        subttl  Transient Command Processors
3781                                        page
```

```
3782
3783                                    if      options and o.ddvr
3784                            ;       -- disk boot loader command --
3785                            ;
3786                                    overlay boot
3787    0148'                   +       c&seg
3788
3789    FC55    21 FF5D                 ld      hl,linbuf+1
3790    FC58    7E              boot1:  ld      a,(hl)          ;scan command line
3791    FC59    2C                      inc     l
3792    FC5A    D6 0D                   sub     cr
3793    FC5C    28 0B                   jr      z,boot2         ;if no parameter, boot from A:
3794    FC5E    FE 13                   cp      ' '-cr
3795    FC60    28 F6                   jr      z,boot1         ;skip leading blanks
3796    FC62    D6 34                   sub     'A'-cr
3797    FC64    D8                      ret     c               ;if invalid drive
3798    FC65    FE 10                   cp      16
3799    FC67    3F                      ccf
3800    FC68    D8                      ret     c               ;if bad drive
3801    FC69    4F              boot2:  ld      c,a             ;set boot drive selected
3802    FC6A    C6 41                   add     a,'A'
3803    FC6C    32 FCDD                 ld      (bootd),a       ;set up error message
3804    FC6F    2E 00                   ld      l,0             ;set A:
3805    FC71    C5                      push    bc
3806    FC72    E5                      push    hl
3807    FC73    CD FCEE                 call    swap            ;switch boot drive with A:
3808    FC76    21 FCD9                 ld      hl,booter       ;set boot error return
3809    FC79    E5                      push    hl
3810    FC7A    0E 00                   ld      c,0             ;then boot from A:
3811    FC7C    CD FA17                 call    select
3812    FC7F    C0                      ret     nz              ;if drive not configured or density error
3813    FC80    3E FF                   ld      a,-1
3814    FC82    12                      ld      (de),a
3815    FC83    11 000A                 ld      de,10           ;set dpb address offset within dph
3816    FC86    19                      add     hl,de
3817    FC87    5E                      ld      e,(hl)          ;set dpb address
3818    FC88    23                      inc     hl
3819    FC89    56                      ld      d,(hl)
3820    FC8A    CD FA3C                 call    home
3821    FC8D    0E 01                   ld      c,1             ;set sector 1
3822    FC8F    1A                      ld      a,(de)          ;get low sectors per track
3823    FC90    32 FCD4                 ld      (boots),a       ;inform boot loader
3824    FC93    B7                      or      a
3825    FC94    20 0D                   jr      nz,boot3        ;if not rigid
3826    FC96    21 000D                 ld      hl,13           ;set reserved track offset within dpb
3827    FC99    19                      add     hl,de
3828    FC9A    4E                      ld      c,(hl)          ;get reserved tracks
3829    FC9B    23                      inc     hl
3830    FC9C    46                      ld      b,(hl)
3831    FC9D    0B                      dec     bc              ;point behind directory
3832    FC9E    ED 43 FA11              ld      (phytrk),bc     ;do implied seek
3833    FCA2    4F                      ld      c,a             ;set sector zero for rigid
3834    FCA3    21 ED80         boot3:  ld      hl,bootbf       ;point to boot load buffer
3835    FCA6    CD FA48                 call    read            ;read cold start loader
3836    FCA9    C0                      ret     nz              ;if read error
```

```
3837    FCAA  11 00FF                    ld      de,bootld+128-1 ;set boot loader address
3838    FCAD  21 EDFF                    ld      hl,bootbf+128-1
3839    FCB0  01 0080                    ld      bc,128
3840    FCB3  ED B8                      lddr                    ;move front half of boot loader to 80h
3841    FCB5  13                         inc     de
3842    FCB6  1A                         ld      a,(de)          ;verify instructions read in
3843    FCB7  FE E5                      cp      0e5h
3844    FCB9  C8                         ret     z               ;if disk has no system
3845    FCBA  21 F000                    ld      hl,Monitr
3846    FCBD  22 F004                    ld      (warm+1),hl     ;set warm start to reload monitor
3847    FCC0  CD FD05                    call    lcp             ;load configuration parameters
3848    FCC3  21 112B                    ld      hl,iobdvr       ;load iobyte driver
3849    FCC6  11 F770                    ld      de,iobloc
3850    FCC9  01 008D                    ld      bc,iobdvs
3851    FCCC  AF                         xor     a
3852    FCCD  CD F2A3                    call    crtldir
3853    FCD0  21 0080                    ld      hl,bootld       ;set start address
3854    FCD3  3E 00                      ld      a,0
3855    FCD4                    boots    equ     $-1             ;sectors per track
3856    FCD5  11 FA0E                    ld      de,phycmd       ;tell boot loader from whence he came
3857    FCD8  E9                         jp      (hl)            ;execute Boot Loader with return to booter
3858
3859                            ;;       Booter - Boot Error Processor.
3860                            ;
3861    FCD9  CD FC3D           booter:  call    pnext
3862    FCDC  07                         defb    7
3863    FCDD  64 3A 20 4C       bootd:   defm    'd: Load error.'
3864    FCE1  6F 61 64 20
3865    FCE5  65 72 72 6F
3866    FCE9  72 2E
3867    FCEB  04                         defb    eot
3868    FCEC  C1                         pop     bc              ;switch drives back
3869    FCED  E1                         pop     hl
3870
3871                            ;;       swap - swap logical drives.
3872                            ;
3873                            ;        Entry:  C = first  drive index, 0-15
3874                            ;                L = second drive index, 0-15
3875                            ;
3876    FCEE  06 00             swap:    ld      b,0             ;clear upper indices
3877    FCF0  60                         ld      h,b
3878    FCF1  11 F360                    ld      de,seltab       ;set select table address
3879    FCF4  29                         add     hl,hl
3880    FCF5  19                         add     hl,de
3881    FCF6  EB                         ex      de,hl           ;set second address to DE, get seltab to HL
3882    FCF7  09                         add     hl,bc
3883    FCF8  09                         add     hl,bc           ;set first address to HL
3884    FCF9  06 02                      ld      b,2
3885    FCFB  4E                swap1:   ld      c,(hl)          ;swap two bytes
3886    FCFC  1A                         ld      a,(de)
3887    FCFD  77                         ld      (hl),a
3888    FCFE  79                         ld      a,c
3889    FCFF  12                         ld      (de),a
3890    FD00  23                         inc     hl
3891    FD01  13                         inc     de
3892    FD02  10 F7                      djnz    swap1           ;if swap not complete
```

```
3893    FD04    C9                      ret
3894
3895                            ;;      lcp - load configuration parameters.
3896                            ;
3897    FD05    3E 81           lcp:    ld      a,10000001b     ;default i/o byte to CRT: and LPT:
3898    FD07    32 0003                 ld      (iobyte),a
3899    FD0A    3A FCD4                 ld      a,(boots)       ;get boot diskette type
3900    FD0D    B7                      or      a
3901    FD0E    0E 20                   ld      c,32
3902    FD10    21 ED80                 ld      hl,bootbf       ;use boot loader buffer
3903    FD13    28 08                   jr      z,lcp1          ;if rigid, use system track, sector 32
3904    FD15    FE 1B                   cp      26+1
3905    FD17    D8                      ret     c               ;no parameters from single density boots
3906    FD18    0E 03                   ld      c,3             ;dd configuration comes from track 0, sector 3
3907    FD1A    21 EE00                 ld      hl,bootbf+128   ;use second half of boot loader buffer
3908    FD1D    CD FA48         lcp1:   call    read
3909    FD20    C0                      ret     nz              ;if can't read configuration
3910    FD21    3A EE00                 ld      a,(bootbf+128)
3911    FD24    D6 E5                   sub     0e5h
3912    FD26    C8                      ret     z
3913    FD27    3A EE7B                 ld      a,(z.xonp)      ;configure Xon-Xoff
3914    FD2A    B7                      or      a
3915    FD2B    28 03                   jr      z,lcp2
3916    FD2D    FE C9                   cp      0c9h
3917    FD2F    C0                      ret     nz
3918    FD30    32 F115         lcp2:   ld      (Xonenb),a
3919    FD33    3A EE60                 ld      a,(z.stpr)      ;configure step rate
3920    FD36    32 FF54                 ld      (steprt),a
3921    FD39    3A EE5F                 ld      a,(z.scra)      ;configure screen attribute
3922    FD3C    32 FD49                 ld      (lcpa),a
3923    FD3F    3A EE62                 ld      a,(z.keym)      ;configure keyboard mask
3924    FD42    32 FD4B                 ld      (lcpb),a
3925    FD45    CD FC3D                 call    pnext
3926    FD48    1B                      defb    esc
3927    FD49    00              lcpa:   defb    0
3928    FD4A    1B                      defb    esc
3929    FD4B    00              lcpb:   defb    0
3930    FD4C    04                      defb    eot
3931    FD4D    21 EE63                 ld      hl,z.sioA       ;configure Sio channels
3932    FD50    3E 02                   ld      a,2
3933    FD52    46              lcp3:   ld      b,(hl)          ;get number of bytes
3934    FD53    23                      inc     hl
3935    FD54    4E                      ld      c,(hl)          ;get port address
3936    FD55    23                      inc     hl
3937    FD56    ED B3                   otir
3938    FD58    3D                      dec     a
3939    FD59    20 F7                   jr      nz,lcp3
3940    FD5B    3A EE7D                 ld      a,(z.baua)      ;configure channel A bit rate
3941    FD5E    D3 00                   out     (bauda),a
3942    FD60    3A EE7E                 ld      a,(z.baub)      ;configure channel B bit rate
3943    FD63    D3 0C                   out     (baudb),a
3944    FD65    3A EE77                 ld      a,(z.siom)      ;configure printer ready mask
3945    FD68    32 F10C                 ld      (siomsk),a
3946    FD6B    3A EE79                 ld      a,(z.siov)      ;configure printer ready value
3947    FD6E    32 F10E                 ld      (sioval),a
3948    FD71    3A EE7F                 ld      a,(z.iobt)      ;configure I/O byte
```

```
3949     FD74    32 0003                    ld      (iobyte),a
3950     FD77    C9                         ret
3951                                        else
3952                                boot    equ     what
3953                                        endif
3954  \
3955                                ;;      -- goto to memory location command --
3956                                ;
3957                                        overlay goto
3958     026B'                  +            c&seg
3959
3960     FC55    B7                         or      a
3961     FC56    37                         scf
3962     FC57    C8                         ret     z               ;if no parameters
3963     FC58    E5                         push    hl              ;set goto address
3964     FC59    DD E1                      pop     ix              ;ld     ix,hl
3965     FC5B    EB                         ex      de,hl           ;set second arg to HL
3966     FC5C    7D                         ld      a,l             ;and A
3967     FC5D    50                         ld      d,b             ;set third arg to DE
3968     FC5E    5D                         ld      e,l
3969     FC5F    ED 4B FFBB                 ld      bc,(param4)     ;set fourth arg to BC
3970     FC63    CD FAD6                    call    jpix
3971     FC66    CD FC1B                    call    put2hs          ;print A reg
3972     FC69    C3 FC16                    jp      put4hs
3973
3974                                ;;      -- memory dump çommand --
3975                                ;
3976                                        overlay memdmp
3977     0282'                  +            c&seg
3978
3979     FC55    3D                         dec     a               ;check parameter count
3980     FC56    28 06                      jr      z,mdmp2
3981     FC58    3D                         dec     a
3982     FC59    28 08                      jr      z,mdmp3
3983     FC5B    2A FFBD            mdmp1:  ld      hl,(last)
3984     FC5E    11 0010            mdmp2:  ld      de,16
3985     FC61    18 0E                      jr      mdmp3b
3986
3987     FC63    EB                 mdmp3:  ex      de,hl
3988     FC64    ED 52                      sbc     hl,de           ;derive bytecount for dump range
3989     FC66    D8                         ret     c               ;if addresses backwards
3990     FC67    06 04                      ld      b,4
3991     FC69    CB 3C              mdmp3a: srl     h               ;divide bytecount by 16
3992     FC6B    CB 1D                      rr      l
3993     FC6D    10 FA                      djnz    mdmp3a
3994     FC6F    23                         inc     hl
3995     FC70    EB                         ex      de,hl
3996     FC71    CD FB8F            mdmp3b: call    dump            ;dump de*16 bytes strting at hl
3997     FC74    22 FFBD                    ld      (last),hl
3998     FC77    C9                         ret
3999
4000                                ;       -- memory examine command --
4001                                ;
4002                                        overlay view
4003     02A5'                  +            c&seg
4004
```

```
4005    FC55    CD FB22             view0:  call    mdata
4006    FC58    CD FC27                     call    echo
4007    FC5B    FE 0D                       cp      cr
4008    FC5D    28 2F                       jr      z,view4
4009    FC5F    FE 2D                       cp      '-'
4010    FC61    28 2D                       jr      z,view5
4011    FC63    FE 22                       cp      '"'
4012    FC65    20 08                       jr      nz,view1
4013    FC67    CD F009                     call    conin
4014    FC6A    CD F00C                     call    conout
4015    FC6D    18 1B                       jr      view3
4016    FC6F    CD FBF3             view1:  call    hexbin
4017    FC72    3F                          ccf
4018    FC73    D0                          ret     nc
4019    FC74    4F                          ld      c,a
4020    FC75    87                          add     a,a
4021    FC76    87                          add     a,a
4022    FC77    87                          add     a,a
4023    FC78    87                          add     a,a
4024    FC79    47                          ld      b,a
4025    FC7A    CD FC27                     call    echo
4026    FC7D    D6 0D                       sub     cr
4027    FC7F    28 08                       jr      z,view2
4028    FC81    C6 0D                       add     a,cr
4029    FC83    CD FBF3                     call    hexbin
4030    FC86    3F                          ccf
4031    FC87    D0                          ret     nc
4032    FC88    48                          ld      c,b
4033    FC89    81                  view2:  add     a,c
4034    FC8A    77                  view3:  ld      (hl),a
4035    FC8B    CD FB0E                     call    check
4036    FC8E    23                  view4:  inc     hl
4037    FC8F    23                          inc     hl
4038    FC90    2B                  view5:  dec     hl
4039    FC91    18 C2                       jr      view0
4040
4041                                        if      options and o.baud
4042                                ;       -- Baud Rate Command --
4043                                ;
4044                                ;       * B<rate> [channel]     ;channel may be 0,1 or A/B
4045                                ;
4046                                        overlay baud
4047    02E3'                       +       c&seg
4048
4049    FC55    B7                          or      a
4050    FC56    37                          scf
4051    FC57    C8                          ret     z               ;if no parameters
4052    FC58    3D                          dec     a
4053    FC59    20 02                       jr      nz,baud1        ;if channel specified
4054    FC5B    1E 01                       ld      e,1             ;set channel 1 (B)
4055    FC5D    CB 43               baud1:  bit     0,e             ;check port
4056    FC5F    0E 00                       ld      c,bauda         ;set communications port
4057    FC61    28 02                       jr      z,baud2
4058    FC63    0E 0C                       ld      c,baudb         ;set printer port
4059    FC65    ED 69               baud2:  out     (c),l           ;set baud rate
4060    FC67    AF                          xor     a
```

```
4061  FC68  C9                        ret
4062                                  else
4063                        baud      equ     what
4064                                  endif
4065
4066                                  if      options and o.disk
4067                        ;;        -- disk sector read/write command --
4068                        ;
4069                        ;         * R <unit> <track> <sector> <address>
4070                        ;         * W <unit> <track> <sector> <address>
4071                        ;
4072                                  overlay dskcmd
4073  02F7                +           c&seg
4074
4075  FC55  47                        ld      b,a
4076  FC56  3A FF5C                   ld      a,(linbuf)
4077  FC59  D6 57                     sub     'W'
4078  FC5B  20 1E                     jr      nz,dsk1
4079  FC5D  B0                        or      b
4080  FC5E  20 16                     jr      nz,dsk0
4081  FC60  4F                        ld      c,a
4082  FC61  CD F2FE                   call    Fastcrt
4083  FC64  23                        inc     hl
4084  FC65  11 0011                   ld      de,17
4085  FC68  01 0015                   ld      bc,21
4086  FC6B  EB                        ex      de,hl
4087  FC6C  AF                        xor     a
4088  FC6D  32 FC54                   ld      ($-25),a
4089  FC70  CD F2A3                   call    crtldir
4090  FC73  C3 FC36                   jp      crlf
4091
4092  FC76  3A FC54           dsk0:   ld      a,($-34)
4093  FC79  B7                        or      a
4094  FC7A  C0                        ret     nz
4095  FC7B  78                dsk1:   ld      a,b
4096  FC7C  FE 04                     cp      4               ;check parameter count
4097  FC7E  37                        scf
4098  FC7F  C0                        ret     nz
4099  FC80  21 FFB5                   ld      hl,param1       ;move parameters to disk command
4100  FC83  11 FA10                   ld      de,phydrv
4101  FC86  01 0007                   ld      bc,3*2+1
4102  FC89  ED A0                     ldi
4103  FC8B  23                        inc     hl              ;skip upper unit
4104  FC8C  ED B0                     ldir
4105  FC8E  05                        dec     b               ;set select operation
4106  FC8F  CD FA51                   call    xqphys          ;execute physical select
4107  FC92  7D                        ld      a,l
4108  FC93  B4                        or      h
4109  FC94  28 16                     jr      z,dskerr        ;if select error
4110  FC96  06 00                     ld      b,0             ;preset write command
4111  FC98  3A FF5C                   ld      a,(linbuf)      ;get command
4112  FC9B  FE 57                     cp      'W'
4113  FC9D  28 01                     jr      z,dsk3          ;if write
4114  FC9F  04                        inc     b
4115  FCA0  CD FA51           dsk3:   call    xqphys          ;execute driver
4116  FCA3  2A FFBB                   ld      hl,(param4)
```

```
4117    FCA6    11 0010                 ld      de,16           ;assume 256-byte sector
4118    FCA9    CA FB8F                 jp      z,dump          ;dump disk read buffer if no error
4119
4120    FCAC    CD FC3D         dskerr: call    pnext
4121    FCAF    44 73 6B 20             defm    'Dsk Err'
4122    FCB3    45 72 72
4123    FCB6    04                      defb    eot
4124    FCB7    C9                      ret
4125                                    else    ;(not disk options)
4126                            dskcmd  equ     what
4127                                    endif
4128
4129                                    if      options and o.inpc
4130                            ;;      -- read input port command --
4131                            ;
4132                            ;       * I <16-bit port address>
4133                            ;
4134                                    overlay incmd
4135    035A'                   +               c&seg
4136
4137    FC55    3D                      dec     a
4138    FC56    37                      scf
4139    FC57    C0                      ret     nz              ;if not one parameter
4140    FC58    4D                      ld      c,l             ;set input port
4141    FC59    44                      ld      b,h
4142    FC5A    CD FC36         in1:    call    crlf
4143    FC5D    79                      ld      a,c             ;display port address
4144    FC5E    CD FC1B                 call    put2hs
4145    FC61    ED 78           in2:    in      a,(c)
4146    FC63    CD FC1B                 call    put2hs
4147    FC66    CD F009                 call    conin           ;read character
4148    FC69    FE 20                   cp      ' '
4149    FC6B    28 F4                   jr      z,in2           ;read same port again
4150    FC6D    FE 0D                   cp      cr
4151    FC6F    28 06                   jr      z,in3           ;if read next
4152    FC71    FE 2D                   cp      '-'
4153    FC73    28 04                   jr      z,in4           ;if read previous
4154    FC75    B7                      or      a               ;clear carry
4155    FC76    C9                      ret
4156    FC77    03              in3:    inc     bc              ;advance to next port
4157    FC78    03                      inc     bc
4158    FC79    0B              in4:    dec     bc
4159    FC7A    18 DE                   jr      in1
4160                                    else
4161                            incmd   equ     what
4162                                    endif
4163
4164                                    if      options and o.outc
4165                            ;;      -- write to output port command --
4166                            ;
4167                            ;       * O <16-bit port address> <8-bit value>
4168                            ;
4169                                    overlay outcmd
4170    0381'                   +               c&seg
4171
4172    FC55    FE 02                   cp      2               ;require two parameters
```

```
4173    FC57    37                              scf
4174    FC58    C0                              ret     nz              ;if not 2 parameters
4175    FC59    4D                              ld      c,l             ;set 16 bit output port address
4176    FC5A    44                              ld      b,h
4177    FC5B    ED 59                           out     (c),e           ;output to d0-d7 and address to a0-a17
4178    FC5D    B7                              or      a
4179    FC5E    C9                              ret
4180                                            else
4181                                outcmd      equ     what
4182                                            endif
4183
4184                                            if      options and o.ramt
4185                                ;;          -- memory read/write diagnostic command --
4186                                ;
4187                                ;           * X <first addr> <last addr>
4188                                ;
4189                                            overlay test
4190    038B'                       +           c&seg
4191
4192    FC55    FE 02                           cp      2               ;check parameter count
4193    FC57    37                              scf
4194    FC58    C0                              ret     nz
4195    FC59    13                              inc     de
4196    FC5A    5A                              ld      e,d             ;get ending page address into e
4197    FC5B    54                              ld      d,h             ;get starting page address into d
4198    FC5C    06 00                           ld      b,0             ;initialize pass counter
4199    FC5E    62                  test1:      ld      h,d             ;point hl to start of block
4200    FC5F    2E 00                           ld      l,0
4201    FC61    7D                  test2:      ld      a,l
4202    FC62    AC                              xor     h               ;generate test byte
4203    FC63    A8                              xor     b
4204    FC64    77                              ld      (hl),a          ;store byte in ram
4205    FC65    23                              inc     hl
4206    FC66    7C                              ld      a,h
4207    FC67    BB                              cp      e               ;check for end of test block
4208    FC68    20 F7                           jr      nz,test2
4209    FC6A    62                              ld      h,d             ;now read back each byte & compare
4210    FC6B    2E 00                           ld      l,0             ;point hl back to start
4211    FC6D    7D                  test3:      ld      a,l
4212    FC6E    AC                              xor     h               ;re-generate test byte data
4213    FC6F    A8                              xor     b
4214    FC70    CD FB0E                         call    check           ;verify memory data still good
4215    FC73    C0                              ret     nz              ;exit if escape request is indicated
4216    FC74    23                              inc     hl              ; else go on to next byte
4217    FC75    7C                              ld      a,h
4218    FC76    BB                              cp      e               ;check for end of block
4219    FC77    20 F4                           jr      nz,test3
4220    FC79    04                              inc     b               ;bump pass count
4221    FC7A    3E 2B                           ld      a,'+'
4222    FC7C    CD FC20                         call    output
4223    FC7F    28 DD                           jr      z,test1         ;do another pass if user not unhappy
4224    FC81    C9                              ret
4225                                            else
4226                                test        equ     what
4227                                            endif
4228
```

```
4229                                    if      options and o.fill
4230                            ;;      -- fill memory with constant command --
4231                            ;
4232                                    overlay fill
4233    03BB'                   +       c&seg
4234
4235    FC55    FE 03                   cp      3               ;check if parameter count=3
4236    FC57    37                      scf
4237    FC58    C0                      ret     nz
4238    FC59    71              fill1:  ld      (hl),c
4239    FC5A    E5                      push    hl
4240    FC5B    B7                      or      a
4241    FC5C    ED 52                   sbc     hl,de           ;compare hl to end address in de
4242    FC5E    E1                      pop     hl
4243    FC5F    23                      inc     hl              ;advance pointer after comparison
4244    FC60    38 F7                   jr      c,fill1
4245    FC62    C9                      ret
4246                                    else
4247                            fill    equ     what
4248                                    endif
4249
4250                                    if      options and o.move
4251                            ;;      -- memory block move command --
4252                            ;
4253                                    overlay block
4254    03C6'                   +       c&seg
4255
4256    FC55    FE 03                   cp      3               ;check if parameter count=3
4257    FC57    37                      scf
4258    FC58    C0                      ret     nz
4259    FC59    CD FC49                 call    blocad
4260    FC5C    79                      ld      a,c
4261    FC5D    B0                      or      b
4262    FC5E    C8                      ret     z               ;exit now if bc=0
4263    FC5F    ED B0                   ldir
4264    FC61    C9                      ret
4265                                    else
4266                            block   equ     what
4267                                    endif
4268
4269                                    if      options and o.verf
4270                            ;;      -- memory block compare command --
4271                            ;
4272                                    overlay vercmd
4273    03D3'                   +       c&seg
4274
4275    FC55    FE 03                   cp      3               ;check if parameter count=3
4276    FC57    37                      scf
4277    FC58    C0                      ret     nz
4278    FC59    CD FC49                 call    blocad
4279    FC5C    18 08                   jr      verf2
4280
4281    FC5E    1A              verf1:  ld      a,(de)
4282    FC5F    CD FB0E                 call    check           ;compare data @ de and @ hl
4283    FC62    C0                      ret     nz              ;exit if escape request is indicated
4284    FC63    23                      inc     hl
```

```
4285    FC64    13                          inc     de
4286    FC65    0B                          dec     bc
4287    FC66    78                  verf2:  ld      a,b
4288    FC67    B1                          or      c
4289    FC68    20 F4                       jr      nz,verf1
4290    FC6A    C9                          ret
4291                                        else
4292                                vercmd  equ     what
4293                                        endif
4294
4295                                        if      options and o.prot
4296                                ;;      Printer Protocol.
4297                                ;
4298                                        overlay proto
4299    03E9'                   +           c&seg
4300
4301    FC55    3D                          dec     a
4302    FC56    28 10                       jr      z,prot1         ;if one parameter
4303    FC58    D6 02                       sub     2
4304    FC5A    37                          scf
4305    FC5B    C0                          ret     nz
4306    FC5C    7B                          ld      a,e
4307    FC5D    F6 04                       or      4
4308    FC5F    32 F10C                     ld      (siomsk),a
4309    FC62    79                          ld      a,c
4310    FC63    F6 04                       or      4
4311    FC65    32 F10E                     ld      (sioval),a
4312    FC68    7D                  prot1:  ld      a,l
4313    FC69    B7                          or      a
4314    FC6A    3E C9                       ld      a,0c9h
4315    FC6C    28 01                       jr      z,prot2
4316    FC6E    AF                          xor     a
4317    FC6F    32 F115             prot2:  ld      (Xonenb),a
4318    FC72    C9                          ret
4319                                        else
4320                                proto   equ     what
4321                                        endif
4322
4323                                        if      (options and o.type) ne 0
4324                                ;;      Type - Simple Typewriter.
4325                                ;
4326                                        overlay type
4327    0407'                   +           c&seg
4328
4329    FC55    B7                          or      a
4330    FC56    28 06                       jr      z,typ0          ;if no baud rate
4331    FC58    3D                          dec     a
4332    FC59    37                          scf
4333    FC5A    C0                          ret     nz              ;if more than one parameter
4334    FC5B    7D                          ld      a,l
4335    FC5C    D3 0C                       out     (baudb),a       ;set printer baud rate
4336    FC5E    CD FC3D             typ0:   call    pnext
4337    FC61    1A                          defb    clrs
4338                                        if      (options and o.esct) ne 0
4339    FC62    1B 31                       defb    esc,'1'         ;set 8 bit keyboard mode
4340                                        else
```

```
4341                                            defb    '0'-'@'         ;set 8 bit keyboard mode
4342                                            endif
4343    FC64    54 79 70 65                     defm    'Typewriter mode.  Touch CTRL+ESC to exit.'
4344    FC68    77 72 69 74
4345    FC6C    65 72 20 6D
4346    FC70    6F 64 65 2E
4347    FC74    20 20 54 6F
4348    FC78    75 63 68 20
4349    FC7C    43 54 52 4C
4350    FC80    2B 45 53 43
4351    FC84    20 74 6F 20
4352    FC88    65 78 69 74
4353    FC8C    2E
4354    FC8D    0D 0A 04                        defb    cr,lf,eot
4355
4356    FC90    CD F0E5             typ1:       call    siost           ;status printer
4357    FC93    28 0F                           jr      z,typ2          ;if char not ready
4358    FC95    CD F0F0                         call    sioin           ;read character
4359    FC98    E6 7F                           and     7fh             ;strip parity bit
4360    FC9A    CD F00C                         call    conout
4361    FC9D    FE 0D                           cp      cr
4362    FC9F    3E 0A                           ld      a,lf
4363    FCA1    CC F00C                         call    z,conout
4364    FCA4    CD F006             typ2:       call    const           ;status console
4365    FCA7    28 E7                           jr      z,typ1          ;if user not active
4366    FCA9    CD F009                         call    conin           ;read keyboard
4367    FCAC    CD F0F8                         call    sioout          ;send character to printer
4368    FCAF    CD F00C                         call    conout          ;and screen
4369    FCB2    FE 0D                           cp      cr
4370    FCB4    20 DA                           jr      nz,typ1         ;if not CR
4371    FCB6    3E 0A                           ld      a,lf            ;send line feed to screen and printer.
4372    FCB8    CD F0F8                         call    sioout
4373    FCBB    CD F00C                         call    conout
4374    FCBE    18 D0                           jr      typ1
4375                                            else
4376                                type        equ     what
4377                                            endif
4378
4379                                            if      options and 0.term
4380                                ;;          Terminal / Scroll Driver.
4381                                ;
4382                                            subttl  Terminal / Screen Manager
4383                                            page
```

```
4384
4385
4386                                        overlay term
4387    0472'                               c&seg
4388
4389    000F                        pass8   equ     15
4390    0016                        inslin  equ     22
4391    0017                        dellin  equ     23
4392    001A                        clrchr  equ     26
4393    001E                        homscr  equ     30
4394    001F                        force   equ     31
4395
4396    0081                        kuplin  equ     81h             ;Move top line off screen to buffer
4397    0082                        kdnlin  equ     82h             ;Move bottom line off screen to buffer
4398    00B1                        Rmttog  equ     80h+'1'         ;Toggle Remote Echo
4399    00B2                        Rmtalf  equ     80h+'2'         ;Toggle Remote Auto LF after CR
4400    008A                        Localf  equ     80h+lf          ;Toggle Local Auto LF after CR
4401    00FF                        Typtog  equ     80h+7fh         ;Toggle Local Echo
4402    00AE                        Brkkey  equ     80h+'.'         ;Hardware BREAK function
4403
4404    0007                        s.lecho equ     7               ;local echo
4405    0006                        s.recho equ     6               ;remote echo
4406    0005                        s.autol equ     5               ;local auto lf after cr
4407    0004                        s.autor equ     4               ;remote auto lf after cr
4408
4409    0100                        Trmbuf  equ     100h
4410    EE80                        Buftop  equ     Trmbuf+760*80
4411    EF00                        Siobuf  equ     Monitr-100h
4412    EF00                        Trmstk  equ     Siobuf
4413
4414    FC55    FE 02                       cp      2               ;check number of arguments
4415    FC57    3F                          ccf
4416    FC58    D8                          ret     c               ;if more than 1
4417    FC59    B7                          or      a
4418    FC5A    20 02                       jr      nz,term1        ;if port specifed
4419    FC5C    2E 00                       ld      l,0
4420    FC5E    01 0406             term1:  ld      bc,siocpa+siodpa*256    ;preset A channel ports
4421    FC61    CB 45                       bit     0,l
4422    FC63    28 03                       jr      z,term2         ;if 0/1 or A/B
4423    FC65    01 0507                     ld      bc,siocpb+siodpb*256    ;set B channel ports
4424    FC68    ED 43 FE78          term2:  ld      (ports),bc
4425    FC6C    31 EF00                     ld      sp,trmstk
4426    FC6F    CD FC3D                     call    pnext
4427    FC72    1A                          db      clrs
4428                                        if      options and o.esct
4429    FC73    1B 31                       db      esc,'1'
4430                                        else
4431                                        db      pass8
4432                                        endif
4433    FC75    54 65 72 6D                 db      'Terminal mode.  Touch CTRL+ESC to exit.'
4434    FC79    69 6E 61 6C
4435    FC7D    20 6D 6F 64
4436    FC81    65 2E 20 20
4437    FC85    54 6F 75 63
4438    FC89    68 20 43 54
```

```
4439    FC8D    52 4C 2B 45
4440    FC91    53 43 20 74
4441    FC95    6F 20 65 78
4442    FC99    69 74 2E
4443    FC9C    0D 0A                   db      cr,lf
4444    FC9E    04                      db      eot
4445
4446    FC9F    FD 21 FEE5              ld      iy,status       ;set pointer to status byte
4447
4448    FCA3    CD F006         term3:  call    const           ;status keyboard
4449    FCA6    C4 FCB1                 call    nz,pki          ;process keyboard input
4450    FCA9    CD FEC8                 call    sioist          ;status sio
4451    FCAC    C4 FD40                 call    nz,prc          ;process remote character
4452    FCAF    18 F2                   jr      term3           ;until user escapes
4453
4454                            ;;      pki - Process Keyboard Input.
4455                            ;
4456    FCB1    3E 00           pki:    ld      a,0
4457    FCB2                    brkflg  equ     $-1
4458    FCB3    B7                      or      a
4459    FCB4    C4 FD21                 call    nz,clrbrk       ;terminate pending break
4460    FCB7    CD F009                 call    conin           ;read input
4461    FCBA    CB 7F                   bit     7,a
4462    FCBC    20 0B                   jr      nz,pki1         ;if function key
4463    FCBE    CD FD5F                 call    sndrmt          ;send it to remote
4464    FCC1    FD CB 00 7E             bit     s.lecho,(iy)
4465    FCC5    C4 FD4C                 call    nz,sndloc       ;if local echo enabled, display console input
4466    FCC8    C9                      ret
4467    FCC9    FE 81           pki1:   cp      kuplin
4468    FCCB    20 12                   jr      nz,pki2         ;if not scroll up
4469    FCCD    CD FE50                 call    gcp             ;get cursor position
4470    FCD0    CD FC3D                 call    pnext           ;home screen, then delete top line
4471                                    if      options and o.esct
4472    FCD3    1E 1B 52 04             defb    homscr,esc,'R',eot
4473                                    else
4474                                    defb    homscr,dellin,eot
4475                                    endif
4476    FCD7    CD FD89                 call    ltl             ;link top line
4477    FCDA    CD FDEB                 call    dbl             ;display bottom line
4478    FCDD    18 14                   jr      pki3
4479    FCDF    FE 82           pki2:   cp      kdnlin
4480    FCE1    20 13                   jr      nz,pki4         ;if not scroll down
4481    FCE3    CD FE50                 call    gcp             ;get cursor position
4482    FCE6    CD FC3D                 call    pnext           ;home screen, insert blank line
4483                                    if      options and o.esct
4484    FCE9    1E 1B 45 04             defb    homscr,esc,'E',eot
4485                                    else
4486                                    defb    homscr,dellin,eot
4487                                    endif
4488    FCED    CD FE0B                 call    lbl             ;link bottom line
4489    FCF0    CD FDB1                 call    dtl             ;display top line
4490    FCF3    C3 FE6E         pki3:   jp      rcp             ;restore cursor position
4491    FCF6    FE FF           pki4:   cp      Typtog
4492    FCF8    20 04                   jr      nz,pki5
4493    FCFA    3E 80                   ld      a,1 shl s.lecho
4494    FCFC    18 16                   jr      pki8
```

```
4495    FCFE    FE B1               pki5:   cp      Rmttog
4496    FD00    20 04                       jr      nz,pki6
4497    FD02    3E 40                       ld      a,1 shl s.recho
4498    FD04    18 0E                       jr      pki8
4499    FD06    FE 8A               pki6:   cp      Localf
4500    FD08    20 04                       jr      nz,pki7
4501    FD0A    3E 20                       ld      a,1 shl s.autol
4502    FD0C    18 06                       jr      pki8
4503    FD0E    FE B2               pki7:   cp      Rmtalf
4504    FD10    20 09                       jr      nz,pki9
4505    FD12    3E 10                       ld      a,1 shl s.autor
4506    FD14    FD AE 00            pki8:   xor     (iy)
4507    FD17    FD 77 00                    ld      (iy),a
4508    FD1A    C9                          ret
4509    FD1B    FE AE               pki9:   cp      Brkkey
4510    FD1D    C0                          ret     nz
4511    FD1E    3A FCB2                     ld      a,(brkflg)
4512    FD21    EE FF               clrbrk: xor     -1
4513    FD23    32 FCB2                     ld      (brkflg),a
4514    FD26    16 10                       ld      d,10h           ;set line SPACING
4515    FD28    20 02                       jr      nz,setbrk
4516    FD2A    16 00                       ld      d,0             ;set line MARKING
4517    FD2C    ED 4B FE78          setbrk: ld      bc,(ports)
4518    FD30    3E 05                       ld      a,5             ;set up WR5
4519    FD32    F3                          di
4520    FD33    ED 79                       out     (c),a
4521    FD35    3E AA                       ld      a,10101010b     ;assert DTR, 7 bpc, RTS, Tx Enb
4522    FD37    B2                          or      d
4523    FD38    ED 79                       out     (c),a
4524    FD3A    FB                          ei
4525    FD3B    3E FF                       ld      a,0ffh
4526    FD3D    C3 FE90                     jp      sioot           ;send RUBOUT to allow MARKING
4527
4528                                ;;      prc - Process Remote Character.
4529                                ;
4530    FD40    CD FED6             prc:    call    sioinc          ;read remote character
4531    FD43    FD CB 00 76                 bit     s.recho,(iy)
4532    FD47    C4 FD5F                     call    nz,sndrmt       ;echo it back
4533    FD4A    18 26                       jr      doc             ;display it locally
4534
4535                                ;;      sndloc - send character to screen.
4536                                ;
4537    FD4C    CD FD72             sndloc: call    doc
4538    FD4F    FE 0D                       cp      cr
4539    FD51    C0                          ret     nz
4540    FD52    FD CB 00 6E                 bit     s.autol,(iy)
4541    FD56    C8                          ret     z
4542    FD57    3E 0A                       ld      a,lf
4543    FD59    CD FD72                     call    doc
4544    FD5C    3E 0D                       ld      a,cr
4545    FD5E    C9                          ret
4546
4547                                ;;      sndrmt - send character to remote.
4548                                ;
4549    FD5F    CD FE90             sndrmt: call    sioot
4550    FD62    FE 0D                       cp      cr
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Terminal / Screen Manager

 4551   FD64   C0                        ret     nz
 4552   FD65   FD CB 00 66                bit     s.autor,(iy)
 4553   FD69   C8                        ret     z
 4554   FD6A   3E 0A                     ld      a,lf
 4555   FD6C   CD FE90                   call    sioot
 4556   FD6F   3E 0D                     ld      a,cr
 4557   FD71   C9                        ret
 4558
 4559                          ;;        doc - Display One Character.
 4560                          ;
 4561   FD72   FE 7F           doc:       cp      7fh
 4562   FD74   C8                        ret     z               ;don't display RUBOUT
 4563   FD75   4F                        ld      c,a             ;send it to screen
 4564   FD76   C5                        push    bc
 4565   FD77   CD FE9B                   call    outcrt          ;display character
 4566   FD7A   C1                        pop     bc
 4567   FD7B   47                        ld      b,a
 4568   FD7C   79                        ld      a,c
 4569   FD7D   FE 0A                     cp      lf
 4570   FD7F   C0                        ret     nz
 4571   FD80   04                        inc     b
 4572   FD81   05                        dec     b
 4573   FD82   C0                        ret     nz              ;if line feed did not scroll
 4574   FD83   CD FD89                   call    ltl             ;link top line
 4575   FD86   3E 0A                     ld      a,lf
 4576   FD88   C9                        ret
 4577
 4578                          ;;        ltl - link top line.
 4579                          ;
 4580   FD89   21 FF5C         ltl:       ld      hl,linbuf
 4581   FD8C   ED 5B FEE8                ld      de,(topptr)     ;set address of line above screen
 4582   FD90   01 0050                   ld      bc,80
 4583   FD93   ED B0                     ldir                    ;move line
 4584   FD95   CD FE34                   call    wup             ;wrap upper pointer
 4585   FD98   ED 53 FEE8                ld      (topptr),de     ;set new top line address
 4586   FD9C   2A FEE6                   ld      hl,(botptr)
 4587   FD9F   EB                        ex      de,hl
 4588   FDA0   B7                        or      a
 4589   FDA1   ED 52                     sbc     hl,de
 4590   FDA3   C0                        ret     nz
 4591   FDA4   11 0050                   ld      de,80
 4592   FDA7   19                        add     hl,de           ;advance bottom pointer
 4593   FDA8   EB                        ex      de,hl
 4594   FDA9   CD FE34                   call    wup             ;wrap upper pointer
 4595   FDAC   ED 53 FEE6                ld      (botptr),de
 4596   FDB0   C9                        ret
 4597
 4598                          ;;        dtl - Display Top Line.
 4599                          ;
 4600   FDB1   ED 5B FEE8      dtl:       ld      de,(topptr)     ;get line above screen
 4601   FDB5   21 FFB0                   ld      hl,-80
 4602   FDB8   19                        add     hl,de
 4603   FDB9   CD FE41                   call    wlp             ;wrap lower pointer
 4604   FDBC   22 FEE8                   ld      (topptr),hl
 4605   FDBF   01 0050                   ld      bc,80
 4606
```

```
4607                                    ;;      dln - display line.
4608                                    ;
4609    FDC2    E5                      dln:    push    hl
4610    FDC3    09                              add     hl,bc
4611    FDC4    2B                              dec     hl
4612    FDC5    3E 20                           ld      a,' '
4613    FDC7    ED A9                   dln1:   cpd
4614    FDC9    20 03                           jr      nz,dln2         ;if not trailing blank
4615    FDCB    EA FDC7                         jp      pe,dln1
4616    FDCE    E1                      dln2:   pop     hl
4617    FDCF    E0                              ret     po              ;if entire line blank
4618    FDD0    41                              ld      b,c
4619    FDD1    04                              inc     b
4620    FDD2    C5                      dln3:   push    bc
4621    FDD3    7E                              ld      a,(hl)
4622    FDD4    4F                              ld      c,a
4623    FDD5    FE 20                           cp      ' '
4624    FDD7    30 08                           jr      nc,dln4
4625    FDD9    E5                              push    hl
4626    FDDA    0E 1F                           ld      c,force         ;force next character out
4627    FDDC    CD FE9B                         call    outcrt
4628    FDDF    E1                              pop     hl
4629    FDE0    4E                              ld      c,(hl)
4630    FDE1    23                      dln4:   inc     hl              ;advance address
4631    FDE2    E5                              push    hl
4632    FDE3    CD FE9B                         call    outcrt          ;display character
4633    FDE6    E1                              pop     hl
4634    FDE7    C1                              pop     bc
4635    FDE8    10 E8                           djnz    dln3            ;if not entire line
4636    FDEA    C9                              ret
4637
4638                                    ;;      dbl - Display bottom line.
4639                                    ;
4640    FDEB    CD FC3D                 dbl:    call    pnext           ;plant cursor on bottom line
4641    FDEE    1B 3D 37 20                     db      esc,'=',' '+23,' ',eot
4642    FDF2    04
4643    FDF3    2A FEE6                         ld      hl,(botptr)
4644    FDF6    E5                              push    hl
4645    FDF7    01 004F                         ld      bc,80-1
4646    FDFA    CD FDC2                         call    dln             ;display bottom line
4647    FDFD    E1                              pop     hl
4648    FDFE    01 0050                         ld      bc,80
4649    FE01    09                              add     hl,bc
4650    FE02    EB                              ex      de,hl
4651    FE03    CD FE34                         call    wup
4652    FE06    ED 53 FEE6                      ld      (botptr),de.
4653    FE0A    C9                              ret
4654
4655                                    ;;      lbl - link bottom line.
4656                                    ;
4657    FE0B    01 0050                 lbl:    ld      bc,80
4658    FE0E    2A FEE6                         ld      hl,(botptr)
4659    FE11    B7                              or      a
4660    FE12    ED 42                           sbc     hl,bc
4661    FE14    CD FE41                         call    wlp
4662    FE17    22 FEE6                         ld      (botptr),hl
```

```
4663    FE1A    EB                      ex      de,hl
4664    FE1B    2A FEE8                 ld      hl,(topptr)
4665    FE1E    B7                      or      a
4666    FE1F    ED 52                   sbc     hl,de
4667    FE21    20 0B                   jr      nz,lbl1
4668    FE23    2A FEE8                 ld      hl,(topptr)
4669    FE26    ED 42                   sbc     hl,bc
4670    FE28    CD FE41                 call    wlp
4671    FE2B    22 FEE8                 ld      (topptr),hl
4672    FE2E    21 FF5C         lbl1:   ld      hl,linbuf
4673    FE31    ED B0                   ldir                    ;move gold mine to buffer
4674    FE33    C9                      ret
4675
4676  \                         ;;      wup - Wrap upper pointer.
4677                              ;
4678    FE34    E5              wup:    push    hl
4679    FE35    21 EE80                 ld      hl,buftop       ;set end of buffer address
4680    FE38    37                      scf
4681    FE39    ED 52                   sbc     hl,de
4682    FE3B    E1                      pop     hl
4683    FE3C    D0                      ret     nc              ;if not past end of buffer
4684    FE3D    11 0100                 ld      de,Trmbuf       ;start over at beggining of buffer
4685    FE40    C9                      ret
4686
4687                              ;;      wlp - Wrap lower pointer.
4688                              ;
4689    FE41    E5              wlp:    push    hl
4690    FE42    D5                      push    de
4691    FE43    11 0100                 ld      de,Trmbuf       ;set start of buffer address
4692    FE46    B7                      or      a
4693    FE47    ED 52                   sbc     hl,de
4694    FE49    D1                      pop     de
4695    FE4A    E1                      pop     hl
4696    FE4B    D0                      ret     nc              ;if not below start of buffer
4697    FE4C    21 EE30                 ld      hl,buftop-80    ;start over at end of buffer
4698    FE4F    C9                      ret
4699
4700                              ;;      gcp - get cursor position.
4701                              ;
4702    FE50    ED 4B FFB1      gcp:    ld      bc,(base)
4703    FE54    2A FFAC                 ld      hl,(cursor)
4704    FE57    45                      ld      b,l             ;get column to B
4705    FE58    CB B8                   res     7,b
4706    FE5A    29                      add     hl,hl           ;set screen row to H
4707    FE5B    7C                      ld      a,h
4708    FE5C    D6 60                   sub     crtbas*2
4709    FE5E    91                      sub     c               ;row = 23-(base-curh)
4710    FE5F    C6 17                   add     a,23
4711    FE61    D6 18           gcp1:   sub     24
4712    FE63    30 FC                   jr      nc,gcp1
4713    FE65    4F                      ld      c,a             ;set row
4714    FE66    21 1F38                 ld      hl,'  '+24-100h ;offset for <esc>=
4715    FE69    09                      add     hl,bc
4716    FE6A    22 FE73                 ld      (rcpa),hl
4717    FE6D    C9                      ret
4718
```

```
4719                                    ;;      rcp - Restore cursor position.
4720                                    ;
4721    FE6E    CD FC3D                 rcp:    call    pnext
4722    FE71    1B 3D                           db      esc,'='
4723    FE73    20 20                   rcpa:   db      ' '
4724    FE75    04                              db      eot
4725    FE76    C9                              ret
4726
4727                                    ;;      sio drivers.
4728                                    ;
4729    FE77    01 FE78                 sioins: ld      bc,ports        ;set status port to c
4730    FE78                            ports   equ     $-2
4731    FE7A    ED 78                           in      a,(c)
4732    FE7C    CB 47                           bit     0,a             ;test rca
4733    FE7E    C9                              ret
4734
4735    FE7F    CD FE77                 sioinp: call    sioins          ;get status
4736    FE82    28 FB                           jr      z,sioinp        ;if not ready
4737    FE84    48                              ld      c,b             ;set data port address
4738    FE85    ED 78                           in      a,(c)
4739    FE87    CB BF                           res     7,a             ;pitch parity bit
4740    FE89    C9                              ret
4741
4742    FE8A    CD FE77                 siordt: call    sioins          ;get sio status
4743    FE8D    CB 57                           bit     2,a             ;test TX empty
4744    FE8F    C9                              ret
4745
4746    FE90    08                      sioot:  ex      af,af'          ;save char to send
4747    FE91    CD FE8A                 siootl: call    siordt          ;test transmit ready status
4748    FE94    28 FB                           jr      z,siootl        ;if not ready
4749    FE96    48                              ld      c,b
4750    FE97    08                              ex      af,af'
4751    FE98    ED 79                           out     (c),a
4752    FE9A    C9                              ret
4753
4754    FE9B    CD FEA7                 outcrt: call    siopl           ;poll for input before & after
4755    FE9E    CD F2FE                         call    fastcrt
4756    FEA1    F5                              push    af              ;save balcones gold
4757    FEA2    CD FEA7                         call    siopl
4758    FEA5    F1                              pop     af
4759    FEA6    C9                              ret
4760
4761                                    ;;      Siopl - Sio Poll Input Characters.
4762                                    ;
4763    FEA7    C5                      Siopl:  push    bc
4764    FEA8    CD FE77                         call    sioins          ;input Sio status
4765    FEAB    28 19                           jr      z,siopl3        ;if input not ready
4766    FEAD    48                              ld      c,b             ;set data port address
4767    FEAE    ED 78                           in      a,(c)
4768    FEB0    CB BF                           res     7,a             ;pitch parity bit
4769    FEB2    E5                              push    hl
4770    FEB3    2A FEEA                         ld      hl,(ipoint)     ;set in pointer
4771    FEB6    77                              ld      (hl),a          ;store character in fifo
4772    FEB7    2C                              inc     l
4773    FEB8    20 02                           jr      nz,siopl1
4774    FEBA    2E 00                           ld      l,low siobuf
```

```
4775    FEBC    3A FEEC             siopl1: ld      a,(opoint)
4776    FEBF    95                          sub     l
4777    FEC0    28 03                       jr      z,siopl2        ;if buffer full
4778    FEC2    22 FEEA                     ld      (ipoint),hl
4779    FEC5    E1                  siopl2: pop     hl
4780    FEC6    C1                  siopl3: pop     bc
4781    FEC7    C9                          ret
4782
4783                                ;;      Sioist - Sio Input Status.
4784                                ;
4785    FEC8    CD FEA7             Sioist: call    Siopl           ;poll for input
4786    FECB    2A FEEC                     ld      hl,(opoint)     ;set out pointer
4787    FECE    3A FEEA                     ld      a,(ipoint)
4788    FED1    95                          sub     l
4789    FED2    C8                          ret     z               ;if data not ready
4790    FED3    F6 FF                       or      -1
4791    FED5    C9                          ret
4792
4793                                ;;      Sioin - Sio Input Character.
4794                                ;
4795    FED6    CD FEC8             Sioinc: call    Sioist          ;set input ready status
4796    FED9    28 FB                       jr      z,Sioinc
4797    FEDB    7E                          ld      a,(hl)
4798    FEDC    2C                          inc     l               ;advance out
4799    FEDD    20 02                       jr      nz,Sioil
4800    FEDF    2E 00                       ld      l,low siobuf
4801    FEE1    22 FEEC             Sioil:  ld      (opoint),hl
4802    FEE4    C9                          ret
4803
4804    FEE5    00                  status: db      0
4805
4806    FEE6    0100                botptr: dw      Trmbuf
4807    FEE8    0100                topptr: dw      Trmbuf
4808
4809    FEEA    EF00                ipoint: dw      siobuf
4810    FEEC    EF00                opoint: dw      siobuf
4811                                        else
4812                                term    equ     what
4813                                        endif
4814
4815                                        if      options and o.help
4816
4817                                ;;      Help Key Command.
4818                                ;
4819                                        overlay help
4820    070B'                       +       c&seg
4821
4822    FC55    CD FC3D                     call    pnext
4823    FC58    42 61 75 64                 defb    'Baud           <rate> [B/A]',cr,lf
4824    FC5C    09 09 3C 72
4825    FC60    61 74 65 3E
4826    FC64    20 5B 42 2F
4827    FC68    41 5D 0D 0A
4828    FC6C    44 75 6D 70                 defb    'Dump           [start] [end]',cr,lf
4829    FC70    09 09 5B 73
4830    FC74    74 61 72 74
```

```
 4831     FC78    5D 20 5B 65
 4832     FC7C    6E 64 5D 0D
 4833     FC80    0A
 4834     FC81    47 6F 74 6F                    defb    'Goto           <addr>',cr,lf
 4835     FC85    09 09 3C 61
 4836     FC89    64 64 72 3E
 4837     FC8D    0D 0A
 4838     FC8F    4D 6F 64 69                    defb    'Modify         <addr>',cr,lf
 4839     FC93    66 79 09 09
 4840     FC97    3C 61 64 64
 4841     FC9B    72 3E 0D 0A
 4842     FC9F    50 72 6F 74                    defb    'Protocol       <xon> [msk val]',cr,lf
 4843     FCA3    6F 63 6F 6C
 4844     FCA7    09 3C 78 6F
 4845     FCAB    6E 3E 20 5B
 4846     FCAF    6D 73 6B 20
 4847     FCB3    76 61 6C 5D
 4848     FCB7    0D 0A
 4849     FCB9    04                             defb    eot
 4850     FCBA    C9                             ret
 4851                                            else
 4852                                    help    equ     what
 4853                                            endif
 4854
 4855                                            subttl  Segment Size Information
 4856                                            page
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
Segment Size Information

 4857
 4858                                   ;;      Top of Overlay Area.
 4859                                   ;
 4860                                           overlay stop
 4861     0771'                    +            c&seg
 4862     0299                           tpamax  equ     tpal            ;set length of transient move
 4863
 4864                                   ;;      Top of Resident Monitor.
 4865                                   ;
 4866                                           below
 4867     0000!                    +            defs    comres
 4868     041B                           rbase   equ     $
 4869
 4870                                   ;;      Top of Non Resident Monitor.
 4871                                   ;
 4872                                           above
 4873     0518!                    +            d&seg
 4874     FC55                           restop  equ     $               ;resident top
 4875     0C55                           reslen  equ     $-monitr        ;length of resident monitor
 4876
 4877                                           update                  ;clear active segment
 4878
 4879                                   ;;      Top of Burned Rom Set.
 4880                                   ;
 4881     17E1                          romtop   equ     bloc+dloc+tloc-monitr
 4882
 4883                                   ;;      Fill Out Unused Rom Space.
 4884                                   ;
 4885     0C55"                                 cseg
 4886
 4887                                           if      (rom+romsiz-romtop) gt 0
 4888     0771'                                 defs    (rom+romsiz-romtop),-1
 4889                                           endif
 4890
 4891                                           subttl  Resident Monitor System Ram
 4892                                           page
```

```
                                .phase  ram

                        ;;      Start of Documented Storage Locations.
                        ;
FF00                    vectab:                         ;interrupt vector table starts here
FF00                    siovec: defs    16              ;space for 8 vectors for sio
FF10                    ctcvec: defs    8               ;space for 4 vectors for ctc
FF18                    sysvec: defs    4               ;space for 2 vectors for system pio
FF1C                    genvec: defs    4               ;space for 2 vectors for general pio

                        ;;      keyboard data input fifo variables
                        ;
FF20                    fifo:   defs    16              ;console input fifo
FF30                    fifcnt: defs    1               ;fifo data counter
FF31                    fifin:  defs    1               ;fifi input pointer
FF32                    fifout: defs    1               ;fifo output pointer

FF33                            defs    1               ;round address

                        ;;      More interrupt vectors
                        ;
FF34                    expvec: defs    8               ;space for 4 vectors for expansion slot

                        ;;      Available memory pointers.
                        ;
FF3C                    availb: defs    2               ;bottom of available memory
FF3E                    availt: defs    2               ;top of available memory

                        ;
                        ;;      End of documented storage locations.

FF40                            defs    16              ;local stack for interrupts
FF50                    intstk:

                        ;;      clock-timer interrupt variables
                        ;
FF50                    Milsec: defs    2               ;One   Millisecond timer, Enable int on ctc1
FF52                    tikcnt: defs    2               ;16 bit seconds counter (18 hr, 12 min, 16 sec)
FF54                    steprt: defs    1               ;WD 1797 step rate
FF55                    timout: defs    1               ;time-out, decrements once per second
                        ;
                        ;       Getime entry returns the address of DAY
                        ;
FF56                    day:    defs    1               ;calendar day           (01-31)
FF57                    month:  defs    1               ;       month           (01-12)
FF58                    year:   defs    1               ;       year-1970       (1970-2225)
FF59                    hrs:    defs    1               ;clock hours            (00-23)
FF5A                    mins:   defs    1               ;       minutes         (00-59)
FF5B                    secs:   defs    1               ;       seconds         (00-59)

                        ;;      crt output driver variables
                        ;
FF5C                    linbuf: defs    80              ;line buffer & Bcc gold mine
FFAC                    cursor: defs    2               ;cursor pointer
```

```
4948    FFAE                    csrchr: defs    1               ;character used for a cursor
4949    FFAF                    dircur: defs    2               ;cursor pointer for direct crt display
4950    FFB1                    base:   defs    1               ;current contents of scroll register
4951    FFB2                    leadin: defs    1               ;state of lead-in sequence handler
4952    FFB3                    attrib: defs    1               ;attribute enable
4953    FFB4                    chrsav: defs    1               ;character under cusror
4954
4955                            ;;      console monitor program variables
4956                            ;
4957    FFB5                    param1: defs    2               ;storage for numbers read
4958    FFB7                    param2: defs    2               ; from line input buffer
4959    FFB9                    param3: defs    2               ; by 'PARAMS' subroutine
4960    FFBB                    param4: defs    2               ; for command processors
4961    FFBD                    last:   defs    2               ;last address used by 'MEMDMP'
4962
4963                            ;;      Configurable parameter address table
4964                            ;
4965    FFBF                    contbl: defs    2*numcon
4966
4967    FFCB                    spare1: defs    1               ;spare configuration byte
4968    FFCC                    spare2: defs    1               ;another spare byte
4969
4970    FFCD                    spare:  defs    (ram+100h-48)-$ ;spare ram space
4971
4972    FFD0                    sparnd:                         ;end of spare ram
4973
4974    FFD0                            defs    16              ;crt stack
4975    FFE0                    crtstk:
4976
4977    FFE0                    rstsp:  defs    2               ;sp register on reset
4978    FFE2                    rsthl:  defs    2               ;hl register on reset
4979    FFE4                    rstpc:  defs    2               ;possible pc from top of stack
4980
4981    FFE6                            defs    26              ;monitor stack
4982    0000                    stack:
4983
4984                                    .dephase
4985
4986                                    subttl  Console Messages
4987                                    page
```

```
4988
4989                            ;;      Console Message Macros.
4990                            ;
4991                            message macro   text,h1,h2
4992                                    if1
4993                                    .radix  16
4994                                    printx  <text>,%(h1),%(h2-1),%((h2)-(h1))
4995                                    .radix  10
4996                                    endif
4997                                    endm
4998
4999                            printx  macro   text,h1,h2,h3
5000                                    .printx + text h1 - h2 = h3 +
5001                                    endm
5002
5003                                    if      romtop ge (rom+romsiz)
5004                                    message <* The ROM set is Too big *>,rom+romsiz,romtop
5005                                    endif
5006
5007                                    if      cloc+tpal gt ram
5008                                    message <* The TPA set is Too big *>,ram,cloc+tpal
5009                                    endif
5010
5011                                    message <Non-resident executes >,rom,bloc
5012                                    message <Rom is burned up from >,rom,romtop
5013                                    message <Unused Rom Space from >,romtop,rom+romsiz-1
5014                                    message <Resident Monitor needs>,monitr,rqtop
5015                                    message <Space Wasted to Driver>,Wasted,Seltab
5016                                    message <Physical Disk Drivers >,Seltab,Dvrlmt
5017                                    message <Driver Offset for ZSID>,200h+bloc+Seltab-Monitr,0
5018                                    message <I/O Byte Drivers from >,iobloc,iobloc+iobdvs
5019                                    message <Command Processor Area>,rqtop,restop
5020                                    message <Transient Overlay ROM >,start,stop
5021                                    message <Transient Command Area>,cloc,cloc+tpal
5022                                    message <Spare Locations in Ram>,spare,sparnd
5023
5024                                    subttl  The*End
5025                                    end     entry
```

```
Balcones Operating System for the XEROX 820-II  MACRO-80 3.44   09-Dec-81
The*End

Macros:
ABOVE           BELOW           BSEG            MESSAGE         OVERLAY
PHEX            PMSG            PRINTX          SEGMENT         SERVICE
UPDATE

Symbols:
0080    ..              F360    .A              F362    .B
F364    .C              F366    .D              F368    .E
F36A    .F              F36C    .G              F36E    .H
F370    .I              F372    .J              F374    .K
F376    .L              F378    .M              F37A    .N
F37C    .O              F37E    .P              009B    ABORT
F6F2    ADDRH           F6F3    ADDRL           FD80    ALL00
FDA0    ALL01           FDC0    ALL02           FDE0    ALL03
FE00    ALL04           FE80    ALL05           FEC0    ALL06
FEE0    ALL07           0068    ASYNC           FFB3    ATTRIB
FA78    AUTOBT          FF3C    AVAILB          FF3E    AVAILT
0000    B.BSY           0002    B.CD            0004    B.IO
0001    B.MSG           0006    B.PAR           0003    B.REQ
0007    B.RST           0005    B.SEL           02CE    BAKSPC
FFB1    BASE            1353    BAUD            FC5D    BAUD1
FC65    BAUD2           0000    BAUDA           000C    BAUDB
0018!   BBASE           040F    BBG             032F    BELL
0331    BELL1           0028    BELLOF          0029    BELLON
041B    BLOC            FC49    BLOCAD          1436    BLOCK
0061    BLOFC           0061    BLONC           0035    BLTIM
000F    BNDRY           11B8    BOOT            FC58    BOOT1
FC69    BOOT2           FCA3    BOOT3           ED80    BOOTBF
FCDD    BOOTD           FCD9    BOOTER          0080    BOOTLD
FCD4    BOOTS           FEE6    BOTPTR          FCB2    BRKFLG
00AE    BRKKEY          0000    BSPACE          EE80    BUFTOP
0004    C.8IN           0004    C.FIVE          0007    C.FLAW
00C0    C.FLPY          0004    C.FMAT          0006    C.FTRK
000C    C.INIT          0007    C.KEYM          0008    C.READ
0001    C.RECAL         0003    C.RQSN          0002    C.RSYN
0006    C.SASI          000B    C.SEEK          00E0    C.TRAM
0000    C.TRDY          0005    C.TWO           0005    C.VTRK
000A    C.WRIT          0009    C.WRPR          F31E    CCA
F323    CCA1            F324    CCA2            00AF    CCS
00B1    CCS1            F57A    CDD             F584    CDDO
F592    CDD1            F59B    CDD2            0108    CFINIT
F6A5    CFT             F6A9I   CFTA            FB0E    CHECK
FD00    CHK00           FD20    CHK01           FD40    CHK02
FD60    CHK03           0000    CHK04           0000    CHK05
0000    CHK06           0000    CHK07           03F5    CHRDEL
03EA    CHRIN1          03F0    CHRIN2          03DC    CHRINS
0034    CHROM1          0035    CHROM2          FFB4    CHRSAV
FC55    CLOC            0350    CLR1            FD21    CLRBRK
001A    CLRCHR          0344    CLREOL          0361    CLREOS
0341    CLRLIN          001A    CLRS            0365    CLRS1
037A    CLRS2           0357    CLRSCN          0036    CMDSIZ
FAD8    CMDTAB          F767    CNFDPB          F000    COLD
F775    COMINP          F770    COMINS          F788    COMOTS
F77F    COMOUT          0518    COMRES          0518C   COMROM
F091    CONFG           F08B    CONFIG          F009    CONIN
F790    CONIOB          F00C    CONOUT          F006    CONST
```

FFBF	CONTBL	023D	CONTRL	F723	CPB
F731	CPB1	F736	CPB2	F759	CPB3
000D	CR	FC36	CRLF	0030	CRTBAS
0182	CRTD1	0196	CRTD2	0199	CRTD3
01B2	CRTD4	0169	CRTDVR	F2A3	CRTLDIR
3C00	CRTMAX	3000	CRTMEM	F2E7	CRTMV
F2E9	CRTMV0	F296	CRTOF1	F293	CRTOFF
F29C	CRTON	F299	CRTON1	F2F1	CRTOUT
FFE0	CRTSTK	003C	CRTTOP	0000	CSPACE
FFAE	CSRCHR	0018	CTC	0018	CTC0
0019	CTC1	001A	CTC2	001B	CTC3
FF10	CTCVEC	0036	CTLSIZ	0254	CTLTAB
FFAC	CURSOR	F6D7	CWP	FF56	DAY
F039	DAYTI	F086	DAYTIM	FDEB	DBL
F6F5	DCTRL	0000	DEBUG	02C5	DEFCUR
F6EA	DEFLPY	F6EB	DEFLUN	0017	DELLIN
FC80	DIRBUF	FFAF	DIRCUR	01F3	DIS1
01F2	DISATR	FDC2	DLN	FDC7	DLN1
FDCE	DLN2	FDD2	DLN3	FDE1	DLN4
FC55	DLOC	FC23	DMPFMT	02E7	DNCSR
FD72	DOC	F490	DPB5D	F470	DPB5S
F450	DPB8D	F430	DPB8S	F390	DPBASE
000A	DPBOFS	F470	DPBRG4	F480	DPBRG5
F490	DPBRG6	F4A0	DPBRG7	F1F1	DPM
F380	DRVTAB	FC76	DSK0	FC7B	DSK1
FCA0	DSK3	1367	DSKCMD	F4B0	DSKDVR
FCAC	DSKERR	03EF	DSM4	01EF	DSM5
00EF	DSM6	00EF	DSM7	0000	DSPACE
F632	DSW	FDB1	DTL	F6D5	DTYPE
FB8F	DUMP	FB98	DUMP2	FBB4	DUMP3
FBD1	DUMP4	FA08	DVRLMT	FD05	EATKEY
FC27	ECHO	F69B	ECR	F0A2	EIRET
01EF	ENATR	0000!	ENTRY	0004	EOT
00C4	ERR	00BC	ERR1	00C1	ERR2
00CC	ERR3	00D4	ERRM1	00DD	ERRM2
0009	ERRML	001B	ESC	029B	ESCADR
02BD	ESCAPE	028A	ESCTAB	0011	ESCTBL
FF34	EXPVEC	0000	FALSE	0000	FALUN
F2FE	FASTCRT	0001	FBLUN	0000	FCLUN
0002	FDLUN	FF30	FIFCNT	FF31	FIFIN
FF20	FIFO	FF32	FIFOUT	1428	FILL
FC59	FILL1	F6F6	FIRST	F708	FIRST1
F721	FIRST2	F4BD	FLOP1	F4DE	FLOP2
F4F2	FLOP3	F4FC	FLOP4	F506	FLOP5
F6EF	FLPFRM	0005	FM.DD	0007	FM.DDDS
00A0	FM.DDSS	0004	FM.DS	0006	FM.FV
0080	FM.HARD	0001	FM.SDDS	0000	FM.SDSS
0002	FM.SZ	0007	FM.UN	0003	FM.WR
0001	FMDD	0006	FMDDSS	0000	FMDS
001F	FORCE	02D4	FORSPC	F61A	GCA
F626	GCA0	F634	GCA1	F63F	GCA2
F627	GCAA	FE50	GCP	FE61	GCP1
FF1C	GENVEC	FBDA	GETHEX	FB2C	GETHLP
FB37	GETLIN	F097	GETSEL	FB38	GLIN1
FB50	GLIN4	FBDF	GNUM1	FBE4	GNUM3
F319	GOLD	12DB	GOTO	0009	GPIOCA

Balcones Operating System for the XEROX 820-II MACRO-80 3.44 09-Dec-81
The*End

000B	GPIOCB	0008	GPIODA	000A	GPIODB
177B	HELP	001E	HELPKEY	FBF3	HEXBIN
FA3C	HOME	02C9	HOMEUP	001E	HOMSCR
FF59	HRS	F649	ICC	F643	ICCS
F64D	ICCS1	F066	IDLE	FC5A	IN1
FC61	IN2	FC77	IN3	FC79	IN4
13CA	INCMD	F137	INDEX	0016	INSLIN
00E6	INTAB	FF50	INTSTK	112B	IOBDVR
008D	IOBDVS	F770	IOBLOC	0003	IOBYTE
F7AF	IOCONI	F796	IOCONO	F7A3	IOCONS
F7BB	IOLIST	F7CC	IOLSTS	FEEA	IPOINT
F643	ISC	FAD6	JPIX	001F	KBDCTL
001E	KBDDAT	F0D8	KBDIN	F0D5	KBDIN1
F0CD	KBDST	F0E2	KBMASK	0082	KDNLIN
F162	KEY1	F167	KEY2	F17F	KEY3
F184	KEY4	F18F	KEY5	F140	KEYSRV
0081	KUPLIN	FFBD	LAST	F5F0	LASTFM
FE0B	LBL	FE2E	LBL1	FD05	LCP
FD1D	LCP1	FD30	LCP2	FD52	LCP3
FD49	LCPA	FD4B	LCPB	F2B3	LDIR1
F2B5	LDIR2	F2C4	LDIR3	0418	LDIRX
FFB2	LEADIN	000A	LF	02F7	LFEED
FF5C	LINBUF	0388	LIND1	039E	LIND2
039F	LIND3	037C	LINDEL	03B7	LINI1
03C0	LINI2	03CD	LINI3	03A4	LININS
008A	LOCALF	0036	LOWLITE	F333	LSTATT
FD89	LTL	F6F1	LUN	0800	LX1984
021C	M3TST	022B	M4TST	0232	M5TST
F0E3	MASK	FB22	MDATA	FC5B	MDMP1
FC5E	MDMP2	FC63	MDMP3	FC69	MDMP3A
FC71	MDMP3B	12F2	MEMDMP	F236	MILO
F246	MILO1	F250	MILO2	F258	MILL1
F265	MILL2	F278	MILL3	F27C	MILL4
F27E	MILL5	F281	MILL6	F1FD	MILLI
FF50	MILSEC	FF5A	MINS	F5AF	MLU
01DF	MODE	01E1	MODE1	F000	MONITR
FF57	MONTH	00E5	MOVLN	F5C2	MPA
F5D7	MPA1	F5DA	MPA2	F5E6	MPA21
F5E9	MPA22	F606	MPA3	F610	MPA4
F613	MPA5	F559	MTRADR	01B4	MULTI
F6F4	NBLK	0066	NMI	02C0	NONO
0040	NT4	0020	NT5	0010	NT6
0010	NT7	0028	NTRK5	004D	NTRK8
F13F	NULINT	0006	NUMCON	F502	NUMUNT
4000	O.AUTO	0400	O.BAUD	0800	O.DDVR
0020	O.DISK	0010	O.ESCT	0004	O.FILL
2000	O.HELP	0200	O.INPC	0002	O.MOVE
0100	O.OUTC	1000	O.PROT	0040	O.RAMT
8000	O.RESV	0001	O.TERM	0008	O.TYPE
0080	O.VERF	F6F0	OPCODE	FEEC	OPOINT
BFFF	OPTIONS	13F1	OUTCMD	FE9B	OUTCRT
F28B	OUTCUR	FC20	OUTPUT	0007	P.ACKN
0000	P.AUTO	0006	P.ONLN	0005	P.RDYI
0004	P.RDYO	0002	P.STRB	F59F	P2L
FB5D	PARA0	FB6C	PARA1	FB71	PARA2
FB76	PARA4	FFB5	PARAM1	FFB7	PARAM2

FFB9	PARAM3	FFBB	PARAM4	FB5F	PARAMS
000F	PASS8	FA0E	PHYCMD	FA15	PHYDMA
FA10	PHYDRV	FA13	PHYSEC	FA11	PHYTRK
FA0F	PHYUNT	F7F0	PIO1	0010	PIOAD
0011	PIOAS	0012	PIOBD	0013	PIOBS
F7DC	PIOOUT	F7F4	PIOSTO	FCB1	PKI
FCC9	PKI1	FCDF	PKI2	FCF3	PKI3
FCF6	PKI4	FCFE	PKI5	FD06	PKI6
FD0E	PKI7	FD14	PKI8	FD1B	PKI9
FC3D	PNEXT	FE78	PORTS	FD40	PRC
FAAC	PRMT1	FA62	PROMPT	FC68	PROT1
FC6F	PROT2	1459	PROTO	0000	PRS
0002	PRS1	003B	PRS2	0061	PRS3
0070	PRS4	00AC	PRS5	F339	PRVATT
FC1B	PUT2HS	FC03	PUT2HX	FB29	PUT2J
FC16	PUT4HS	FC0C	PUTNIB	FF00	RAM
041B	RBASE	FE6E	RCP	FE73	RCPA
F647	RDC	F605	RDID	F61B	RDID1
F6DC	RDONLY	F4E7	RDOP	FA4A	RDWR
F4FE	RDWRA	F4F6	RDWRS	FA48	READ
F641	RECAL	F6E9	RECLUN	F130	REMOVE
F6AD	RESET	0C55	RESLEN	F07C	RESTART
FC55	RESTOP	F1EF	RETINS	02F2	RETURN
F0A1	RETV1	F09A	RETVAL	F5F6	RETZR
0191	REV	F1E9	RFI	EE00	RGDBUF
0003	RGLUN	F6E8	RGRECAL	F708	RIGDPB
00B2	RMTALF	00B1	RMTTOG	0000	ROM
1800	ROMSIZ	17E1	ROMTOP	FA08	RQTOP
F5F8	RSE	F603	RSE1	F332	RSTATT
FFE2	RSTHL	FFE4	RSTPC	FFE0	RSTSP
0001	RTK4	0041	RTK5	0061	RTK6
0071	RTK7	1800	RX1984	0005	S.AUTOL
0004	S.AUTOR	0007	S.LECHO	0006	S.RECHO
F4B0	SA1403	F4C0	SAS0	F4B2	SAS0A
F4F3	SAS1	F4F6	SAS2	0012	SASIC
0010	\SASID	0300	SASIDL	0012	SASIS
F470	SASSTR	F1EC	SAVSTK	0014	SCROLL
009E	SCRPRT	01C3	SEARCH	F4E6	SECLEN
FF5B	SECS	FA3E	SEEK	F5DE	SEEK0
F5EB	SEEK1	F5ED	SEEK2	F5EF	SEEK3
F5A3	SEEKX	0518!	SEGA	F5B9	SEK0
F5C4	SEK1	FA39	SEL1	F578	SEL1W
F585	SEL2	F591	SEL3	F598	SELDEN
F596	SELDNS	F510	SELEC	FA17	SELECT
F50B	SELER1	F42A	SELERR	F360	SELTAB
FA5A	SELTBL	F544	SELUNT	01D4	SETBLI
FD2C	SETBRK	021F	SETCOL	F0A4	SETCON
F284	SETCUR	01DD	SETGRA	01D9	SETINV
01CF	SETLOW	01F7	SETMSK	F337	SETPRV
0210	SETROW	0205	SETXY	0209	SETXY1
FC73	SIGN1	FC81	SIGN2	FC88	SIGN3
FC9E	SIGN3A	FCA6	SIGN4	FC55	SIGNON
F6CE	SIM	EF00	SIOBUF	0006	SIOCPA
0007	SIOCPB	0004	SIODPA	0005	SIODPB
FEE1	SIOI1	F0F0	SIOIN	F0ED	SIOIN1
FED6	SIOINC	FE7F	SIOINP	FE77	SIOINS

Balcones Operating System for the XEROX 820-II MACRO-80 3.44 09-Dec-81
The*End

FEC8	SIOIST	F10C	SIOMSK	FE90	SIOOT
FE91	SIOOT1	F0F8	SIOOUT	FEA7	SIOPL
FEBC	SIOPL1	FEC5	SIOPL2	FEC6	SIOPL3
F113	SIORD1	F129	SIORD2	F12C	SIORD3
FE8A	SIORDT	F105	SIORDY	F0E5	SIOST
F10E	SIOVAL	FF00	SIOVEC	F0F9	SIOX1
0031	SLDDEN	FA08	SLERR	0030	SLSDEN
F65A	SMF	F66D	SMF0	F675	SMF0A
F693	SMF1	F6A3	SMF1A	F6BC	SMF1B
F6BE	SMF2	F6D4	SMF4	F6D1	SMFA
F50E	SMFS	F530	SMFS1	F548	SMFS1A
F550	SMFS2	F559	SMFS3	F573	SMFS4
F53D	SMFSA	03D1	SMP	FD4C	SNDLOC
FD5F	SNDRMT	F0D2	SOFT	F069	SOFTV
F6D2	SOM	F6D4	SOM1	FC1E	SPACE
F20E	SPACT	F22D	SPADDR	FFCD	SPARE
FFCB	SPARE1	FFCC	SPARE2	FFD0	SPARND
F224	SPCNT	F500	SSELEC	F0BF	SSP
0000	STACK	1070	START	FEE5	STATUS
F61D	STC	F644	STEPR	FF54	STEPRT
17E1	STOP	F66E	STPADR	02C1	STUFF
FCEE	SWAP	FCFB	SWAP1	0069	SYNC
001D	SYSCTL	001C	SYSPIO	FF18	SYSVEC
031F	TAB	F65F	TDI	F656	TDO
14E2	TERM	FC5E	TERM1	FC68	TERM2
FCA3	TERM3	13FB	TEST	FC5E	TEST1
FC61	TEST2	FC6D	TEST3	FF52	TIKCNT
F192	TIMER	F1B1	TIMER1	F1B9	TIMER2
F1E8	TIMER3	0006	TIMOU	FF55	TIMOUT
0771	TLOC	FEE8	TOPPTR	0299	TPAL
0299	TPAMAX	F6FF	TRKTBL	0100	TRMBUF
EF00	TRMSTK	F6ED	TRN5	F410	TRN6
FFFF	TRUE	F639	TTC	F63A	TTCA
FC5E	TYPO	FC90	TYP1	FCA4	TYP2
1477	TYPE	00FF	TYPTOG	02DC	UPCSR
F048	USRSEC	F31B	USRSTK	FF00	VECTAB
1443	VERCMD	FC5E	VERF1	FC66	VERF2
1315	VIEW	FC55	VIEW0	FC6F	VIEW1
FC89	VIEW2	FC8A	VIEW3	FC8E	VIEW4
FC90	VIEW5	F003	WARM	F35F	WASTED
F669	WCC	0010	WD1797	0010	WDCR
0031	WDDD	0013	WDDT	0030	WDSD
001C	WDSL	0012	WDSN	0010	WDSR
0011	WDTR	F687	WFR	F69A	WFR1
F696	WFRA	FAC9	WHAT	FE41	WLP
F64B	WOC	F64D	WOC1	F650	WOC2
FA44	WRITE	FE34	WUP	0000!	XCKS
0008!	XCKS1	0013	XOFF	F12D	XOFFLG
0011	XON	F115	XONENB	F344	XQDVR
FA51	XQPHYS	F508	XSELERR	FF58	YEAR
EE7D	Z.BAUA	EE7E	Z.BAUB	EE7F	Z.IOBT
EE62	Z.KEYM	EE5F	Z.SCRA	EE63	Z.SIOA
EE6D	Z.SIOB	EE77	Z.SIOM	EE79	Z.SIOV
EE60	Z.STPR	EE7B	Z.XONP		

No Fatal error(s)

Notes

Quick, Fast Cold Start Loader MACRO-80 3.44 09-Dec-81

```
                                        Title   Quick, Fast Cold Start Loader
                                ;;;     Quick, Fast Cold Start Loader.
                                ;
                                ;       Copyright (C) 1982, Balcones Computer Corporation.
                                ;
                                        .z80
F02A                            Xqdvr   equ     0f02ah          ;Physical Driver Executioner
0004                            cdisk   equ     00004h          ;current user/disk

0000'                                   cseg
0000'                           bios    equ     $               ;origin of bios above ccp & bdos

                                        .phase  80h

0080                            phycmd  equ     $
0081                            phyunt  equ     $+1
0082                            phydrv  equ     $+2
0083                            phytrk  equ     $+3
0085                            physec  equ     $+5
0087                            phydma  equ     $+7

                                ;;      quick, fast loader.
                                ;
                                ;       Entry:  A =  Sectors per Track
                                ;               DE = Address of Physical Command Block that loaded QFD
                                ;
0080    EB                      qfs:    ex      de,hl           ;transfer command block
0081    01 0005                         ld      bc,physec-phycmd
0084    ED B0                           ldir                    ;woe be unto he who changes qfs
0086    21 00D5                         ld      hl,ldrtbl-5     ;set loader control table address
0089    0E 05                           ld      c,5             ;set table entry size
008B    09                      qfs1:   add     hl,bc           ;advance table address
008C    BE                              cp      (hl)            ;match with loader control table entry
008D    38 FC                           jr      c,qfs1          ;if match not found yet
008F    C0                              ret     nz              ;if entry not in table
0090    23                              inc     hl
0091    4E                              ld      c,(hl)          ;set track offset+1
0092    23                              inc     hl
0093    ED A0                           ldi                     ;move starting sector, adjust track offset
0095    AF                              xor     a
0096    12                              ld      (de),a          ;clear upper sector
0097    7E                              ld      a,(hl)          ;set number of sectors
0098    23                              inc     hl
0099    6E                              ld      l,(hl)          ;set (sector size)/4-1
009A    23                              inc     hl              ;sector size / 4
009B    29                              add     hl,hl
009C    29                              add     hl,hl
009D    EB                              ex      de,hl           ;set sector size in DE
009E    2A 0083                         ld      hl,(phytrk)     ;add track offset
00A1    09                              add     hl,bc
00A2    22 0083                         ld      (phytrk),hl
00A5    47                              ld      b,a             ;set number of sectors
00A6    7A                              ld      a,d             ;check sector size
00A7    21 FF80'                        ld      hl,bios-80h     ;set starting address
```

```
00AA  B7                        or      a
00AB  20 01                     jr      nz,qfs3         ;if big sectors
00AD  19                qfs2:   add     hl,de           ;advance transfer address
00AE  22 0087           qfs3:   ld      (phydma),hl
00B1  D9                        exx                     ;switch register sets
00B2  21 0080                   ld      hl,phycmd or qfs;set physical command address
00B5  CD F02A                   call    Xqdvr           ;execute driver read
00B8  21 0085                   ld      hl,physec       ;advance sector
00BB  34                        inc     (hl)
00BC  D9                        exx                     ;switch back
00BD  B7                        or      a
00BE  C0                        ret     nz              ;if boot error
00BF  10 EC                     djnz    qfs2            ;if cold start not complete
00C1  32 0004                   ld      (cdisk),a       ;start out in A: user 0
00C4  21 0003'                  ld      hl,bios+3       ;warm start after signon
00C7  E5                        push    hl
00C8  21 00F6                   ld      hl,sernum
00CB  11 00DA                   ld      de,ldrtbl
00CE  0E 0A                     ld      c,sernml
00D0  ED B0                     ldir
00D2  12                        ld      (de),a
00D3  13                        inc     de
00D4  3E C9                     ld      a,0c9h
00D6  12                        ld      (de),a
00D7  CD 0000'                  call    bios            ;execute cold start loader

                        ;;      Loader Control Table.
                        ;
                        ;       Entries Must be in sort order
                        ;
                        ;       db      sectors per track
                        ;               offset+1 from boot track
                        ;               starting bios sector
                        ;               number of bios sectors
                        ;               sector size/4-1
                        ;
00DA  34 02 16 04       ldrtbl: db      52,1+1,22,4,256/4-1     ;8" Double Density
00DE  3F
00DF  22 03 05 04               db      34,2+1,05,4,256/4-1     ;5" Double Density
00E3  3F
00E4  1A 02 14 07               db      26,1+1,20,7,128/4-1     ;8" Single Density
00E8  1F
00E9  12 03 0A 07               db      18,2+1,10,7,128/4-1     ;5" Single Density
00ED  1F
00EE  00 01 16 04               db      00,0+1,22,4,256/4-1     ;Any SASI Rigid Disk
00F2  3F
00F3  00                        db      00                      ;   End of table

                                if      $ gt 100h-10
                                .printx * Too Big *
                                else
                                if      $ eq 100h-10
                                .printx + Perfect Fit +
                                else
00F4                            ds      100h-10-$,-1
                                endif
```

```
                                        endif
                                ;;      Serialization.
                                ;
00F6    20 44 43 2A             sernum: db      ' DC*******'            ;Manufacturing serial number here
00FA    2A 2A 2A 2A
00FE    2A 2A
000A                            sernml  equ     $-sernum

                                        if      $ ne 100h
                                        .printx * Serial Number Out of Place *
                                        endif
                                        .dephase

                                        end
```

```
Quick, Fast Cold Start Loader   MACRO-80 3.44   09-Dec-81

Macros:

Symbols:
0000'   BIOS            0004    CDISK           00DA    LDRTBL
0080    PHYCMD          0087    PHYDMA          0082    PHYDRV
0085    PHYSEC          0083    PHYTRK          0081    PHYUNT
0080    QFS             008B    QFS1            00AD    QFS2
00AE    QFS3            000A    SERNML          00F6    SERNUM
F02A    XQDVR

No Fatal error(s)
```

BIOS	13#	56	71	81
CDISK	10#	70		
LDRTBL	32	74	93#	
PHYCMD	17#	30	62	
PHYDMA	22#	60		
PHYDRV	19#			
PHYSEC	21#	30	64	
PHYTRK	20#	51	53	
PHYUNT	18#			
QFS	29#	62		
QFS1	34#	36		
QFS2	59#	69		
QFS3	58	60#		
SERNML	75	120#		
SERNUM	73	117#	120	
XQDVR	9#	63		

Notes

```
XEROX 820-II BIOS        MACRO-80 3.44   09-Dec-81
Bios Jump Table

                                        subttl  Bios Jump Table
                                        title   XEROX 820-II BIOS

                                ;;      XEROX 820+ Rom Resident Bios Jump Table.
                                ;
                                ;       Copyright 1981, Balcones Computer Corporation.
                                ;
                                        .z80

0000' C3 00F1'                  bios:   jp      cboot           ;cold start
0003' C3 0069'                  bwboot: jp      wboot           ;warm start

0006' C3 F04B                   bconst: jp      const           ;console status
0009' C3 F04E                   bconin: jp      conin           ;console character in
000C' C3 F051                   bconot: jp      conout          ;console character out
000F' C3 F054                   bprint: jp      list            ;list character on printer
0012' C3 F060                   bpunch: jp      punch           ;punch
0015' C3 F05D                   breadr: jp      reader          ;reader

0018' C3 01B6'                  bhome:  jp      home            ;move head to home position
001B' C3 0154'                  bseld:  jp      seldsk          ;select disk
001E' C3 01B9'                  bsett:  jp      settrk          ;set track number
0021' C3 01BE'                  bsets:  jp      setsec          ;set sector number
0024' C3 01C3'                  bsetd:  jp      setdma          ;set dma address
0027' C3 01EB'                  bread:  jp      read            ;read a record
002A' C3 01F3'                  bwrit:  jp      write           ;write a record

002D' C3 F057                   bprnts: jp      listst          ;printer ready status
0030' C3 01C8'                  bsctrn: jp      sectrn          ;sector translate

0033' 81                        initio: db      10000001b       ;Initial I/O Byte

                                Subttl  Cold and Warm Start Module
                                page
```

```
XEROX 820-II BIOS       MACRO-80 3.44   09-Dec-81
Cold and Warm Start Module

                                ;
                                ;
0E00                    bdosln  equ     0e00h           ;Length of CP/M v 2.2 BDOS
0800                    ccplen  equ     0800h           ;Length of CP/M v 2.2 CCP
0004                    cdisk   equ     4               ;Current user/disk
002C                    nsects  equ     (ccplen+bdosln)/128     ;number of sectors for ccp + bdos
0062                    rev     equ     'b'
                                ;
                        ;;      Wboot - Warm Start CP/M.
                                ;
0034' 3E C3             wbt5:   ld      a,0c3h          ;plant jumps
0036' 21 F206'                  ld      hl,bios-bdosln+6
0039' BE                        cp      (hl)
003A' 20 1C                     jr      nz,wbterr       ;if no jump to bdos
003C' 32 0000 -                 ld      (0),a
003F' 32 0005                   ld      (5),a
0042' 22 0006                   ld      (6),hl          ;set address of jump to bdos
0045' 21 0003'                  ld      hl,bwboot       ;set warm boot address
0048' 22 0001                   ld      (1),hl
004B' ED 4B 0004                ld      bc,(cdisk)      ;set current disk / user
004F' 21 EA00'                  ld      hl,bios-bdosln-ccplen ;Enter CCP
0050'                   wbtcom  equ     $-2             ;patch to "03" to disable warm boot command
0052' 3E 03                     ld      a,3
0054' 32 0050'                  ld      (wbtcom),a
0057' E9                        jp      (hl)
                                ;
0058' CD 0115'          wbterr: call    pmsg            ;display error message
005B' 0D 0A 42 6F               db      13,10,'Boot Err',0
005F' 6F 74 20 45
0063' 72 72 00
0066' CD 0009'                  call    bconin  ;wait for key
0069' 31 0100           wboot:  ld      sp,100h         ;use external stack
006C' CD 013F'                  call    dboot           ;inform deblocker
006F' 4F                        ld      c,a             ;(zero) select A:
0070' 3E 2C                     ld      a,nsects        ;set number of sectors to read
0072' 32 00D0'                  ld      (seccnt),a      ;set sector counter
0075' 21 E980'                  ld      hl,bios-bdosln-ccplen-128
0078' 22 013B'                  ld      (dmabas),hl     ;set base track dma address
007B' CD 001B'                  call    bseld           ;select boot drive (A:)
007E' 7C                        ld      a,h
007F' B5                        or      l
0080' 28 D6                     jr      z,wbterr
0082' 23                        inc     hl              ;point to high translate address
0083' 7E                        ld      a,(hl)
0084' 32 0122'                  ld      (xlate),a
0087' E5                        push    hl
0088' 0E 00                     ld      c,0             ;translate sector zero
008A' CD 0121'                  call    mls
008D' 79                        ld      a,c
008E' 32 0133'                  ld      (transz),a      ;set sector zero translate value
0091' E1                        pop     hl
0092' 11 0009                   ld      de,10-1         ;offset to dpb
0095' 19                        add     hl,de
0096' 4E                        ld      c,(hl)          ;get dpb address
0097' 23                        inc     hl
```

```
XEROX 820-II BIOS        MACRO-80 3.44   09-Dec-81
Cold and Warm Start Module

   90     0098' 46                        ld      b,(hl)
   91     0099' 0A                        ld      a,(bc)          ;get low sectors per track
   92     009A' 32 00DC'                  ld      (spt),a
   93     009D' 21 000D                   ld      hl,13
   94     00A0' 09                        add     hl,bc
   95     00A1' 4E                        ld      c,(hl)          ;get reserved tracks
   96     00A2' 23                        inc     hl
   97     00A3' 46                        ld      b,(hl)
   98     00A4' 0B                        dec     bc
   99     00A5' 1E 01                     ld      e,1             ;set sector 1
  100     00A7' B7                        or      a               ;test low sectors per track
  101     00A8' 28 06                     jr      z,wbt1          ;if rigid disk
  102     00AA' 4A                        ld      c,d             ;set track 0
  103     00AB' FE 1B                     cp      26+1
  104     00AD' 38 01                     jr      c,wbt1          ;if single density 8" or 5"
  105     00AF' 4B                        ld      c,e             ;double density starts on track 1, sector 1
  106     00B0' C5                wbt1:   push    bc              ;save track
  107     00B1  D5                        push    de              ;save starting sector
  108     00B2' CD 001E'                  call    bsett           ;position disk
  109     00B5' C1                        pop     bc
  110     00B6' C5                wbt2:   push    bc              ;save sector
  111     00B7' CD 0121'                  call    mls             ;map logical sector
  112     00BA' E5                        push    hl              ;save address
  113     00BB' CD 0021'                  call    bsets           ;set sector
  114     00BE' C1                        pop     bc
  115     00BF' 21 FFFE'                  ld      hl,bios-2
  116     00C2' ED 42                     sbc     hl,bc
  117     00C4' 38 12                     jr      c,wbt3          ;if within bios
  118     00C6' CD 0024'                  call    bsetd           ;set dma address
  119     00C9' CD 01EB'                  call    read            ;read next sector
  120     00CC' B7                        or      a
  121     00CD' 20 89                     jr      nz,wbterr       ;if load error
  122     00CF' 3E 00                     ld      a,0             ;update sectors read counter
  123     00D0'                   seccnt  equ     $-1
  124     00D1' 3D                        dec     a
  125     00D2' 32 00D0'                  ld      (seccnt),a
  126     00D5' CA 0034'                  jp      z,wbt5          ;if end of load
  127     00D8' C1                wbt3:   pop     bc              ;advance sector
  128     00D9' 0C                        inc     c
  129     00DA' 79                        ld      a,c
  130     00DB' FE FF                     cp      -1
  131     00DC'                   spt     equ     $-1
  132     00DD' 20 D7                     jr      nz,wbt2         ;if not end of track
  133     00DF' 41                        ld      b,c
  134     00E0' 2A 013B'                  ld      hl,(dmabas)     ;advance base dma address
  135     00E3' 11 0080                   ld      de,128
  136     00E6' 19                wbt4:   add     hl,de           ;by spt*128
  137     00E7' 10 FD                     djnz    wbt4
  138     00E9' 22 013B'                  ld      (dmabas),hl
  139     00EC' C1                        pop     bc              ;advance track
  140     00ED' 0C                        inc     c
  141     00EE' 58                        ld      e,b             ;and restart on sector 0
  142     00EF' 18 BF                     jr      wbt1
  143
  144                             ;;      Cold Start CP/M.
  145                             ;
```

```
                00F1' CD 0115'          cboot:  call    pmsg            ;Announce CP/M size and version
                00F4' 1A 58 65 72               db      26,'Xerox 60k CP/M vers 2.2',rev
                00F8' 6F 78 20 36
                00FC' 30 6B 20 43
                0100' 50 2F 4D 20
                0104' 76 65 72 73
                0108' 20 32 2E 32
                010C' 62
                010D' 20 23 32 2D               db      ' #2-294'
                0111' 32 39 34
                0114' 00                        db      0

                                        ;;      pmsg - print message at return address.
                                        ;
                0115' E1                pmsg:   pop     hl              ;print message after call
                0116' 7E                        ld      a,(hl)
                0117' 23                        inc     hl
                0118' B7                        or      a
                0119' E5                        push    hl
                011A' C8                        ret     z               ;if end of message
                011B' 4F                        ld      c,a
                011C' CD 000C'                  call    bconot          ;display message at current console
                011F' 18 F4                     jr      pmsg

                                        ;;      mls - map logical sector.
                                        ;
                0121' 3E 00             mls:    ld      a,0             ;set translate address
                0122'                   xlate   equ     $-1
                0123' B7                        or      a
                0124' 79                        ld      a,c
                0125' 28 0B                     jr      z,mls2          ;if not single density
                0127' 87                        add     a,a             ;read by half tracks
                0128' 2A 00DC'                  ld      hl,(spt)        ;get sectors per track
                012B' BD                        cp      l
                012C' 38 02                     jr      c,mls1          ;if not past end of track
                012E' 95                        sub     l               ;offset back to beginning of track
                012F' 3C                        inc     a
                0130' 3C                mls1:   inc     a               ;map sector 0->1
                0131' 4F                        ld      c,a
                0132' D6 00             mls2:   sub     0               ;offset by translate of sector zero
                0133'                   transz  equ     $-1
                0134' 1F                        rra
                0135' 67                        ld      h,a
                0136' 2E 00                     ld      l,0
                0138' CB 1D                     rr      l
                013A' 11 0000                   ld      de,0            ;set base dma for this track
                013B'                   dmabas  equ     $-2
                013D' 19                        add     hl,de           ;compute address for this sector
                013E' C9                        ret

                                                Subttl  CHARIO - Character I/O Module
                                                page
```

```
XEROX 820-II BIOS        MACRO-80 3.44   09-Dec-81
CHARIO - Character I/O Module

  198
  199
  200     F000                          monitr  equ     0f000h          ;820+ Resident Monitor Address
  201
  202     F04B                          const   equ     monitr+4bh
  203     F04E                          conin   equ     monitr+4eh
  204     F051                          conout  equ     monitr+51h
  205     F054                          list    equ     monitr+54h
  206     F057                          listst  equ     monitr+57h
  207     F05D                          reader  equ     monitr+5dh
  208     F060                          punch   equ     monitr+60h
  209
  210                                   Subttl  CP/M Deblocking Driver
  211                                   page
```

```
                                ;;      CP/M Deblocking Driver.
                                ;
013F'                           cseg

                                ;;      Ascii.
                                ;
000A                            lf      equ     10
000B                            up      equ     11
000D                            cr      equ     13
001B                            esc     equ     27

451B                            inslin  equ     ('E' shl 8) + esc
521B                            dellin  equ     ('R' shl 8) + esc

                                ;;      Absolute Machine Addresses.
                                ;
F02A                            xqdvr   equ     0f02ah          ;Resident Monitor Driver Executioner
0004                            cdisk   equ     4               ;CCP active user/disk

                                ;;      CP/M Write Types.
                                ;
0000                            wrall   equ     0               ;normal write to allocated sector
0001                            wrdir   equ     1               ;write to directory sector
0002                            wrual   equ     2               ;first write to unallocated block

                                ;;      skip - skip next instruction.
                                ;
                                ;       Uses HL to perform very short jumps
                                ;
                                skip    macro   n
                                        if      ((n)-$) eq 2
                                        db      26h             ;;;set PC = $+2  (ld h,...)
                                        endif
                                        if      ((n)-$) eq 3
                                        db      21h             ;;;set PC = $+3  (ld hl,...)
                                        endif
                                        endm

                                ;;      Dboot - Deblocking Bootstrap.
                                ;
                                ;       Entry:  Called prior to Warm Start reload.
                                ;
013F'  21 0000"                 dboot:  ld      hl,hstbuf       ;initialize host buffer address
0142'  22 021A"                         ld      (hstdma),hl
0145'  21 021C"                         ld      hl,dphtab       ;clear internal DPH table of addresses
0148'  01 2000                          ld      bc,16*2*256     ;set table length, zero
014B'  71                       dbt2:   ld      (hl),c          ;clear next byte
014C'  23                               inc     hl
014D'  10 FC                            djnz    dbt2            ;if table not clear

                                ;;      clract - Clear host buffer active.
                                ;
014F'  AF                       clract: xor     a
```

```
0150' 32 026E'                  ld      (hstact),a      ;clear host buffer active
0153' C9                        ret

                        ;;      select - select CP/M disk.
                        ;
                        ;       Entry:  C = CP/M Logical Drive, 0-15.
                        ;               E = 2*n+0 if media identification required
                        ;               E = 2*n+1 if media previously identified
                        ;
0154' 79                seldsk: ld      a,c             ;remember disk to seek
0155' 32 0205"                  ld      (sekdsk),a
0158' 06 00                     ld      b,0
015A' 21 021C"          sel1:   ld      hl,dphtab       ;set table of remembered dph's
015D' 09                        add     hl,bc           ;index by words
015E' 09                        add     hl,bc
015F' CB 43                     bit     0,e
0161' 28 0A                     jr      z,sel2          ;if drive not previously selected
0163' 7E                        ld      a,(hl)          ;set disk parameter header address in hl
0164' 23                        inc     hl
0165' 66                        ld      h,(hl)
0166' 6F                        ld      l,a
0167' B4                        or      h
0168' 20 19                     jr      nz,sel3         ;if previous select succesful
016A' 5F                        ld      e,a             ;force media identification
016B' 18 ED                     jr      sel1
016D' E5                sel2:   push    hl              ;save dph table address
016E' 21 0202"                  ld      hl,selcmd+2
0171' 71                        ld      (hl),c          ;set CP/M Logical drive
0172' 2B                        dec     hl
0173' 2B                        dec     hl              ;point to select command
0174' 36 FF                     ld      (hl),-1         ;set driver select operation
0176' CD 02DF'                  call    xdr             ;execute driver request
0179' EB                        ex      de,hl
017A' E1                        pop     hl
017B' 73                        ld      (hl),e          ;remember disk parameter header address
017C' 23                        inc     hl
017D' 72                        ld      (hl),d
017E' EB                        ex      de,hl
017F' 7D                        ld      a,l
0180' B4                        or      h
0181' 28 22                     jr      z,sel4          ;if drive not succesfully selected
0183' E5                sel3:   push    hl              ;save dph address
0184' 01 000A                   ld      bc,10           ;set dpb offset in dph
0187' 09                        add     hl,bc
0188' 5E                        ld      e,(hl)          ;set disk parameter block address
0189' 23                        inc     hl
018A' 56                        ld      d,(hl)
018B' EB                        ex      de,hl
018C' 22 0241'                  ld      (dpbadr),hl
018F' 0E 03                     ld      c,3
0191' 09                        add     hl,bc
0192' 7E                        ld      a,(hl)          ;set block shift factor
0193' 3C                        inc     a               ;form 128 byte records per block
0194' 32 021B'                  ld      (rpb),a
0197' 0E 0C                     ld      c,15-3          ;point to end of dpb
0199' 09                        add     hl,bc
```

```
                                ld      a,(hl)          ;set internal parameter
                                ld      (trkzfl),a      ;set track zero single density flag
                                and     7
                                ld      (secmsk),a      ;set sector mask
                                pop     hl
                                ret
                        sel4:                           ;fall into clear active disk

                        ;;      cad - Clear Active Disk.
                        ;
                        cad:    ld      hl,cdisk        ;get disk that CCP will log in
                                ld      a,(sekdsk)      ;get disk that failed
                                xor     (hl)
                                and     not 11110000b   ;clear active user
                                jr      nz,cad1         ;if selected disk is not default disk
                                ld      a,(hl)          ;cause CCP to log in A:
                                and     not 1111b       ;retain active user area
                                ld      (hl),a
                        cad1:   ld      l,h             ;indicate select failure
                                ret

                        ;;      Home - Set Track Zero.
                        ;
                        home:   ld      bc,0            ;seek track zero

                        ;;      Settrk - Set Track.
                        ;
                        ;       Entry:  BC = Track number
                        ;
                        settrk: ld      (sektrk),bc     ;set track to seek
                                ret

                        ;;      Setsec - Set Sector.
                        ;
                        ;       Entry:  BC = Sector number
                        ;
                        setsec: ld      (seksec),bc     ;set sector to seek
                                ret

                        ;;      Setdma - Set Direct Memory Address.
                        ;
                        ;       Entry:  BC = DMA address
                        ;
                        setdma: ld      (sekdma),bc
                                ret

                        ;;      Sectran - Sector Translate.
                        ;
                        ;       Entry:  BC = Sector number, 0 <= BC <  Sectors per Track
                        ;               DE = Single byte skew table address
                        ;
                        ;       Exit:   HL = BC         if DE =  0
                        ;                L = (DE+BC)    if DE <> 0
                        ;                H = B          which better be zero
                        ;
                        sectrn: ld      l,c             ;set untranslated sector
```

```
XEROX 820-II BIOS        MACRO-80 3.44   09-Dec-81
CP/M Deblocking Driver

  379     01C9'   60                            ld      h,b
  380     01CA'   7A                            ld      a,d
  381     01CB'   B3                            or      e
  382     01CC'   C8                            ret     z               ;if no translate table
  383     01CD'   EB                            ex      de,hl
  384     01CE'   09                            add     hl,bc
  385     01CF'   6E                            ld      l,(hl)          ;single byte translate
  386     01D0'   60                            ld      h,b
  387     01D1'   C9                            ret
  388
  389                                   ;;      Rdwrs - Read or Write Single Density.
  390                                   ;
  391     01D2'   3A 02AE'              rdwrs:  ld      a,(readop)      ;set read/write operation
  392     01D5'   21 0203"                      ld      hl,sekcmd       ;set seek request
  393     01D8'   18 07                         jr      rdwrhs          ;enter read/write dispatcher
  394
  395                                   ;;      Readhs - Read Host Sector.
  396                                   ;
  397     01DA'   3E 01                 readhs: ld      a,1             ;set read operation
  398                                           skip    $+2             ;jump over write entry point
  399     01DC'   26                    +       db      26h
  400
  401                                   ;;      Wriths - Write Host Sector.
  402                                   ;
  403     01DD'   AF                    wriths: xor     a               ;set write operation
  404     01DE'   21 0213"                      ld      hl,hstcmd
  405
  406                                   ;;      Rdwrhs - Read or Write Host Sector.
  407                                   ;
  408                                   ;       Entry:  HL = Physical command request address
  409                                   ;                A = 0 to write
  410                                   ;                A = 1 to read
  411                                   ;
  412                                   ;       Exit:   A  =  0, if no errors
  413                                   ;               A  = -1, if errors
  414                                   ;               Z  = condition of A reg
  415                                   ;
  416     01E1'   77                    rdwrhs: ld      (hl),a          ;set driver operation
  417     01E2'   CD 02DF'                      call    xdr             ;execute driver read or write
  418     01E5'   21 02BE'                      ld      hl,erflag       ;merge error flag for directory protection
  419     01E8'   B6                            or      (hl)
  420     01E9'   77                            ld      (hl),a
  421     01EA'   C9                            ret
  422
  423                                   ;;      Read - Read CP/M Sector.
  424                                   ;
  425                                   ;       Entry:  Seldsk, Settrk, Setsec, Setdma previously called
  426                                   ;
  427                                   ;       Exit:   A =  0 if no errors
  428                                   ;               A = -1 if errors
  429                                   ;
  430     01EB'   AF                    read:   xor     a               ;clear unalloc processing
  431     01EC'   32 0226'                      ld      (unacnt),a
  432     01EF'   0E 00                         ld      c,wrall         ;inhibit buffer flush after read
  433     01F1'   3C                            inc     a               ;set read operation
  434                                           skip    $+2
```

```
XEROX 820-II BIOS       MACRO-80 3.44   09-Dec-81
CP/M Deblocking Driver

01F2' 26                +               db      26h

                        ;;      Write - Write CP/M Sector.
                        ;
                        ;       Entry:  Seldsk, Settrk, Setsec, Setdma previously called
                        ;
                        ;       Exit:   A =  0 if no errors
                        ;               A = -1 if errors
                        ;               C = Write type
                        ;
01F3' AF                write:  xor     a               ;set write operation

                        ;;      Rdwr - Read or Write.
                        ;
                        ;       Entry:   A = 0 to write
                        ;                A = 1 to read
                        ;
01F4' 32 02AE'          rdwr:   ld      (readop),a      ;set read/write switch
01F7' AF                        xor     a               ;reset error flag
01F8' 32 02BE'                  ld      (erflag),a
01FB' 2A 020C"                  ld      hl,(seksec)     ;set seek host sector
01FE' 22 0208"                  ld      (sekhst),hl
0201' 3A 029B'                  ld      a,(secmsk)      ;set sector size
0204' B7                        or      a
0205' 28 CB                     jr      z,rdwrs         ;if deblocking not required
0207' F6 00                     or      0               ;check track zero single density flag
0208'                   trkzfl  equ     $-1
0209' F2 0212'                  jp      p,rdwr1         ;if track zero not single density
020C' 3A 0206"                  ld      a,(sektrk)      ;set seek track
020F' B7                        or      a
0210' 28 C0                     jr      z,rdwrs         ;if track 0, read or write without deblocking
0212' 79                rdwr1:  ld      a,c             ;save write type
0213' 32 02BA'                  ld      (wrtype),a
0216' FE 02                     cp      wrual
0218' 20 0B                     jr      nz,writ1        ;if not write to unallocated group
021A' 3E 00                     ld      a,0             ;set records per block
021B'                   rpb     equ     $-1
021C' 32 0226'                  ld      (unacnt),a      ;start counting unallocated writes
021F' 11 020E"                  ld      de,unadsk       ;set unallocated parameter block address
0222' CD 02D6'                  call    cpb             ;copy parameter block
0225' 3E 00             writ1:  ld      a,0             ;set remaining unallocated sectors
0226'                   unacnt  equ     $-1
0227' B7                        or      a
0228' 28 2E                     jr      z,writ4         ;if not processing unallocated group
022A' 3D                        dec     a
022B' 32 0226'                  ld      (unacnt),a      ;update unallocated sectors remaining
022E' 21 0205"                  ld      hl,sekdsk       ;set seek parameters
0231' 11 020E"                  ld      de,unadsk       ;set unallocated parameters
0234' CD 02CC'                  call    cmp             ;compare parameter blocks
0237' 20 1B                     jr      nz,writ3        ;if not seek to unallocated sector
0239' 2A 0211"                  ld      hl,(unasec)     ;advance unallocated sector
023C' 23                        inc     hl
023D' 22 0211"                  ld      (unasec),hl
0240' 11 0000                   ld      de,0            ;set sectors per track
0241'                   dpbadr  equ     $-2
0243' ED 52                     sbc     hl,de
```

```
  491  0245'  20 0A                    jr      nz,writ2        ;if not end of track
  492  0247'  22 0211"                 ld      (unasec),hl     ;reset to sector zero
  493  024A'  2A 020F"                 ld      hl,(unatrk)     ;advance unallocated track
  494  024D'  23                       inc     hl
  495  024E'  22 020F"                 ld      (unatrk),hl
  496  0251'  AF               writ2:  xor     a               ;mark pre-read not required
  497  0252'  18 05                    jr      rwoper
  498  0254'  AF               writ3:  xor     a               ;clear unallocated processing
  499  0255'  32 0226'                 ld      (unacnt),a
  500  0258'  3C               writ4:  inc     a               ;mark pre-read required
  501
  502                          ;;      Rwoper - Read or Write Operation Proper.
  503                          ;
  504  0259'  32 028B'          rwoper: ld      (rsflag),a      ;set pre-read block flag
  505  025C'  3A 029B'                 ld      a,(secmsk)      ;set shift counter
  506  025F'  2A 020C"                 ld      hl,(seksec)
  507  0262'  CB 3C             rwop1:  srl     h               ;compute host sector = cpmsec/(2**sekmsk)
  508  0264'  CB 1D                    rr      l
  509  0266'  CB 3F                    srl     a
  510  0268'  20 F8                    jr      nz,rwop1        ;if shift incomplete
  511  026A'  22 0208"                 ld      (sekhst),hl     ;set seek host sector
  512  026D'  F6 00                    or      0               ;check host active flag
  513  026E'                    hstact  equ     $-1
  514  026F'  3E 01                    ld      a,1
  515  0271'  32 026E'                 ld      (hstact),a      ;host buffer always becoms active
  516  0274'  28 0E                    jr      z,rwop2         ;if host buffer was not active
  517  0276'  21 0215"                 ld      hl,hstdsk       ;set active host buffer identification
  518  0279'  11 0205"                 ld      de,sekdsk       ;set seek identification
  519  027C'  CD 02CC'                 call    cmp             ;compare seek request with active host sector
  520  027F'  28 16                    jr      z,rwop3         ;if host buffer contains seek sector
  521  0281'  CD 02C2'                 call    flush           ;flush buffer if previously written
  522  0284'  11 0215"          rwop2:  ld      de,hstdsk       ;set host request block address
  523  0287'  CD 02D6'                 call    cpb             ;copy seek parameter block to host
  524  028A'  3E 00                    ld      a,0             ;check pre-read required
  525  028B'                    rsflag  equ     $-1
  526  028C'  B7                       or      a
  527  028D'  C4 01DA'                 call    nz,readhs       ;read host sector if preread required
  528  0290'  B7                       or      a
  529  0291'  C4 014F'                 call    nz,clract       ;clear host buffer active if read errors
  530  0294'  32 02C3'                 ld      (hstwrt),a      ;mark buffer not written into
  531  0297'  3A 020C"          rwop3:  ld      a,(seksec)      ;set seek sector
  532  029A'  E6 00                    and     0               ;form host buffer index from sector mask
  533  029B'                    secmsk  equ     $-1
  534  029C'  1F                       rra
  535  029D'  57                       ld      d,a             ;multiply index by 128 bytes/sector
  536  029E'  3E 00                    ld      a,0
  537  02A0'  1F                       rra
  538  02A1'  5F                       ld      e,a
  539  02A2'  2A 021A"                 ld      hl,(hstdma)     ;set host buffer address
  540  02A5'  19                       add     hl,de           ;form seek buffer address
  541  02A6'  ED 5B 020A"              ld      de,(sekdma)     ;set user transfer address
  542  02AA'  01 0080                  ld      bc,128          ;set CP/M sector length
  543  02AD'  3E 00                    ld      a,0             ;set transfer direction
  544  02AE'                    readop  equ     $-1
  545  02AF'  B7                       or      a
  546  02B0'  20 05                    jr      nz,rwop4        ;if read operation
```

```
02B2' EB                        ex      de,hl           ;switch directions
02B3' 3C                        inc     a
02B4' 32 02C3'                  ld      (hstwrt),a      ;mark buffer written into
02B7' ED B0                rwop4: ldir                    ;move sector to/from user buffer
02B9' 3E 00                     ld      a,0             ;set write type
02BA'                      wrtype equ     $-1
02BB' FE 01                     cp      wrdir
02BD' 3E 00                     ld      a,0             ;set error flag
02BE'                      erflag equ     $-1
02BF' C0                        ret     nz
02C0' B7                        or      a
02C1' C0                        ret     nz              ;if errors, do not clobber directory

                           ;;   Flush - Flush buffer to disk.
                           ;
02C2' 3E 00                flush: ld       a,0             ;check host written flag
02C3'                      hstwrt equ     $-1
02C4' B7                        or      a
02C5' C4 01DD'                  call    nz,wriths       ;if buffer written into, write host sector
02C8' 32 02C3'                  ld      (hstwrt),a      ;clear host written flag if no errors
02CB' C9                        ret

                           ;;   Cmp - Compare Paramater Blocks.
                           ;
                           ;    Entry:  HL = Parameter block
                           ;            DL = Parameter block
                           ;
                           ;    Exit:   Z  = Set   if parameters identical
                           ;            Z  = Clear if parameters different
                           ;
02CC' 06 05                cmp:   ld       b,5             ;set length of parameter block
02CE' 1A                   cmp1:  ld       a,(de)          ;compare next byte
02CF' AE                        xor     (hl)
02D0' C0                        ret     nz              ;if parameters different
02D1' 13                        inc     de
02D2' 23                        inc     hl
02D3' 10 F9                     djnz    cmp1            ;if more bytes
02D5' C9                        ret

                           ;;   Cpb - Copy Parameter Block.
                           ;
                           ;    Entry:  DE = Address of Unallocated or Host parameter block
                           ;
                           ;    Exit:   Seek parameter block copied into block at DE
                           ;
02D6' 21 0205"             cpb:   ld       hl,sekdsk       ;set source parameters
02D9' 01 0005                   ld      bc,5            ;set block length
02DC' ED B0                     ldir                    ;copy parameter block
02DE' C9                        ret

                           ;;   Xdr - Execute Driver Request.
                           ;
                           ;    Entry:  HL = pointer to Physical Driver Request Block
                           ;
                           ;    Exit:   Physical Driver exit condition are maintained if
                           ;            no errors or user did not request warm start.
```

```
XEROX 820-II BIOS        MACRO-80 3.44   09-Dec-81
CP/M Deblocking Driver

  603                                   ;
  604     02DF'   22 02EA'              xdr:    ld      (xdra),hl       ;save request for retrys
  605     02E2'   2A 02EA'              xdr1:   ld      hl,(xdra)       ;restore request address
  606     02E5'   CD F02A                       call    Xqdvr           ;execute physical driver
  607     02E8'   47                            ld      b,a             ;save read/write error status
  608     02E9'   3A 02EA'                      ld      a,(xdra)        ;get driver operation
  609     02EA'                         xdra    equ     $-2
  610     02EC'   4F                            ld      c,a             ;set message index
  611     02ED'   3C                            inc     a
  612     02EE'   20 05                         jr      nz,xdr2         ;if not select request
  613     02F0'   B5                            or      l
  614     02F1'   B4                            or      h
  615     02F2'   C0                            ret     nz              ;if dph address returned by driver
  616     02F3'   18 03                         jr      xdr3
  617     02F5'   78                    xdr2:   ld      a,b             ;set read/write error status
  618     02F6'   B7                            or      a
  619     C2F7'   C8                            ret     z               ;if no read/write errors
  620     02F8'   2A 02EA'              xdr3:   ld      hl,(xdra)       ;put drive name in message
  621     02FB'   23                            inc     hl
  622     02FC'   23                            inc     hl
  623     02FD'   7E                            ld      a,(hl)
  624     02FE'   C6 41                         add     a,'A'
  625     0300'   32 0330'                      ld      (xdrb),a
  626     0303'   0C                            inc     c
  627     0304'   20 0C                         jr      nz,xdr4         ;if not select request
  628     0306'   CD 0375'                      call    pmsgi
  629     0309'   53 65 6C 65                   db      'Select',0
  630     030D'   63 74 00
  631     0310'   18 16                         jr      xdr6
  632     0312'   0D                    xdr4:   dec     c
  633     0313'   20 0B                         jr      nz,xdr5         ;if not write request
  634     0315'   CD 0375'                      call    pmsgi
  635     0318'   57 72 69 74                   db      'Write',0
  636     031C'   65 00
  637     031E'   18 08                         jr      xdr6
  638     0320'   CD 0375'              xdr5:   call    pmsgi           ;must be read request
  639     0323'   52 65 61 64                   db      'Read',0
  640     0327'   00
  641     0328'   CD 0115'              xdr6:   call    pmsg
  642     032B'   20 45 72 72                   db      ' Err '
  643     032F'   20
  644     0330'   64 3A 20              xdrb:   db      'd: '
  645     0333'   41 28 63 63                   db      'A(ccept), '
  646     0337'   65 70 74 29
  647     033B'   2C 20
  648     033D'   49 28 67 6E                   db      'I(gnore), '
  649     0341'   6F 72 65 29
  650     0345'   2C 20
  651     0347'   52 28 65 74                   db      'R(etry) '
  652     034B'   72 79 29 20
  653     034F'   00                            db      0
  654     0350'   CD 0009'                      call    bconin          ;read character from console
  655     0353'   F5                            push    af
  656     0354'   CD 0115'                      call    pmsg
  657     0357'   0D                            db      cr
  658     0358'   521B                          dw      dellin
```

```
035A' 0B 00                         db      up,0
035C' 67                            ld      h,a             ;zero dph for accepted or ignored select errors
035D' 6F                            ld      l,a
035E' F1                            pop     af
035F' E6 5F                         and     5fh             ;ignore parity, case
0361' FE 03                         cp      3
0363' 28 0A                         jr      z,xdr7          ;if warm start requested
0365' D6 49                         sub     'I'
0367' C8                            ret     z               ;if user ignored error, don't tell BDOS
0368' D6 F8                         sub     'A'-'I'
036A' C2 02E2'                      jp      nz,xdrl         ;retry request
036D' 2F                            cpl
036E' C9                            ret

036F' CD 01A5'              xdr7:   call    cad             ;clear active disk
0372' C3 0003'                      jp      bwboot

0375' CD 0115'              pmsgi:  call    pmsg
0378' 0D 0A                         db      cr,lf
037A' 451B                          dw      inslin
037C  00                            db      0
037D' C3 0115'                      jp      pmsg

                                    subttl  Deblocker Storage Area
                                    page
```

```
XEROX 820-II BIOS        MACRO-80 3.44   09-Dec-81
Deblocker Storage Area

                                reserve macro   s,n
                                s       equ     $+.
                                .       aset    .+n
                                        endm
                        /
                        0000    .       aset    0
                        0380'           dseg

                                ;;      Host Sector Deblocking Buffer.
                                ;
                                reserve hstbuf,512

                                ;;      Physical Driver Select Command.
                                ;
                                reserve selcmd,3                ;select command, unit, drive

                                ;;      Seek Sector Parameter Block.
                                ;
                                reserve sekcmd,1                ;kindly
                                reserve sekunt,1                ; leave
                                reserve sekdsk,1                ;  these
                                reserve sektrk,2                ;   bytes
                                reserve sekhst,2                ;    alone
                                reserve sekdma,2
                                reserve seksec,2

                                ;;      Unallocated Sector Parameter Block.
                                ;
                                reserve unadsk,1                ;kindly
                                reserve unatrk,2                ; leave
                                reserve unasec,2                ;  these

                                ;;      Host Sector Parameter Block.
                                ;
                                reserve hstcmd,1                ;kindly
                                reserve hstunt,1                ; leave
                                reserve hstdsk,1                ;  these
                                reserve hsttrk,2                ;   bytes
                                reserve hstsec,2                ;    alone
                                reserve hstdma,2

                                ;;      Disk Parameter Header Addresses.
                                ;
                                reserve dphtab,('P'-'A'+1)*2

                        0000"           cseg

                                        end
```

```
XEROX 820-II BIOS        MACRO-80 3.44   09-Dec-81
Deblocker Storage Area

Macros:
RESERVE         SKIP

Symbols:
023C            0009'   BCONIN          000C'   BCONOT
0006'   BCONST   0E00    BDOSLN          0018'   BHOME
0000'   BIOS     000F'   BPRINT          002D'   BPRNTS
0012'   BPUNCH   0027'   BREAD           0015'   BREADR
0030'   BSCTRN   001B'   BSELD           0024'   BSETD
0021'   BSETS    001E'   BSETT           0003'   BWBOOT
002A'   BWRIT    01A5'   CAD             01B4'   CAD1
00F1'   CBOOT    0800    CCPLEN          0004    CDISK
014F'   CLRACT   02CC'   CMP             02CE'   CMP1
F04E    CONIN    F051    CONOUT          F04B    CONST
02D6'   CPB      000D    CR              013F'   DBOOT
014B'   DBT2     521B    DELLIN          013B'   DMABAS
0241'   DPBADR   021C"   DPHTAB          02BE'   ERFLAG
001B    ESC      02C2'   FLUSH           01B6'   HOME
026E'   HSTACT   0000"   HSTBUF          0213"   HSTCMD
021A"   HSTDMA   0215"   HSTDSK          0218"   HSTSEC
0216"   HSTTRK   0214"   HSTUNT          02C3'   HSTWRT
0033'   INITIO   451B    INSLIN          000A    LF
F054    LIST     F057    LISTST          0121'   MLS
0130'   MLS1     0132'   MLS2            F000    MONITR
002C    NSECTS   0115'   PMSG            0375'   PMSGI
F060    PUNCH    01F4'   RDWR            0212'   RDWR1
01E1'   RDWRHS   01D2'   RDWRS           01EB'   READ
F05D    READER   01DA'   READHS          02AE'   READOP
0062    REV      021B'   RPB             028B'   RSFLAG
0262'   RWOP1    0284'   RWOP2           0297'   RWOP3
02B7'   RWOP4    0259'   RWOPER          00D0'   SECCNT
029B'   SECMSK   01C8'   SECTRN          0203"   SEKCMD
020A"   SEKDMA   0205"   SEKDSK          0208"   SEKHST
020C"   SEKSEC   0206"   SEKTRK          0204"   SEKUNT
015A'   SEL1     016D'   SEL2            0183'   SEL3
01A5'   SEL4     0200"   SELCMD          0154'   SELDSK
01C3'   SETDMA   01BE'   SETSEC          01B9'   SETTRK
00DC'   SPT      0133'   TRANSZ          0208'   TRKZFL
0226'   UNACNT   020E"   UNADSK          0211"   UNASEC
020F"   UNATRK   000B    UP              0069'   WBOOT
00B0'   WBT1     00B6'   WBT2            00D8'   WBT3
00E6'   WBT4     0034'   WBT5            0050'   WBTCOM
0058'   WBTERR   0000    WRALL           0001    WRDIR
0225'   WRIT1    0251'   WRIT2           0254'   WRIT3
0258'   WRIT4    01F3'   WRITE           01DD'   WRITHS
02BA'   WRTYPE   0002    WRUAL           02DF'   XDR
02E2'   XDR1     02F5'   XDR2            02F8'   XDR3
0312'   XDR4     0320'   XDR5            0328'   XDR6
036F'   XDR7     02EA'   XDRA            0330'   XDRB
0122'   XLATE    F02A    XQDVR


No Fatal error(s)
```

	689#	695	695#	695	699	699#	699	703	703#	703	704	704#
	704	705	705#	705	706	706#	706	707	707#	707	708	708#
	708	709	709#	709	713	713#	713	714	714#	714	715	715#
	715	719	719#	719	720	720#	720	721	721#	721	722	722#
	722	723	723#	723	724	724#	724	728	728#	728		
BCONIN	14#	65	654									
BCONOT	15#	167										
BCONST	13#											
BDOSLN	37#	40	46	55	71							
BHOME	20#											
BIOS	10#	46	55	71	115							
BPRINT	16#											
BPRNTS	28#											
BPUNCH	17#											
BREAD	25#											
BREADR	18#											
BSCTRN	29#											
BSELD	21#	73										
BSETD	24#	118										
BSETS	23#	113										
BSETT	22#	108										
BWBOOT	11#	52	674									
BWRIT	26#											
CAD	333#	673										
CAD1	337	341#										
CBOOT	10	146#										
CCPLEN	38#	40	55	71								
CDISK	39#	54	231#	333								
CLRACT	266#	529										
CMP	483	519	577#									
CMP1	578#	583										
CONIN	14	203#										
CONOUT	15	204#										
CONST	13	202#										
CPB	474	523	592#									
CR	222#	657	677									
DBOOT	67	256#										
DBT2	260#	262										
DELLIN	226#	658										
DMABAS	72	134	138	192#								
DPBADR	315	489#										
DPHTAB	258	279	728#									
ERFLAG	418	454	555#									
ESC	223#	225	226									
FLUSH	521	562#										
HOME	20	346#										
HSTACT	267	513#	515									
HSTBUF	256	695#										

Notes

```
                        Title   Banked Physical Driver
                ;;      Banked Physical Driver
                ;
                ;       Copyright (C) 1982, Balcones Computer Corporation
                ;
                ;       Transferred to Public Domain - (PD) 1983
                ;
                ;;      After executing this program by entering BANK x:(where x is any valid
                ;       CP/M disk drive A-P).  The BANK program will load a physical disk
                ;       driver into memory.  This physical driver is executed when drive x: is
                ;       accessed by CP/M.  This particular disk driver will map normal CP/M
                ;       files onto the address space of the alternate memory bank
                ;       (bank 0) in the 820-II.
                ;
                ;       This utility demonstrates the flexibility of the logical to
                ;       physical disk mapping in the 820-II.  The BANK program
                ;       moves the physical disk driver up to high memory.
                ;       It then updates the entry for drive x: in the logical to physical
                ;       disk drive mapping table telling the system to use physical
                ;       disk driver #3 when CP/M requests service from drive x:.
                ;
                ;       The execution address of the BANK driver is then placed in
                ;       entry #3 of the physical disk driver address table.
                ;
                ;       If BANK is executed by entering:  A>BANK P:
                ;
                ;       Then doing a A>DIR P: would display the following directory:
                ;
                ;       BOOT    .ROM  :  OPTION  .ROM  :  SCREEN  .MEM  :  EXPAND  .RAM
                ;
                ;       Entering: A>STAT P:*.* will display the following:
                ;
                ;                       Recs  Bytes  Ext Acc
                ;                         64    12k     1 R/O P:BOOT.ROM
                ;                        256    32k     1 R/W P:EXPAND.RAM
                ;                         16     2k     1 R/W P:OPTION.ROM
                ;                         24     4k     1 R/W P:SCREEN.MEM
                ;                       Bytes Remaining On P: 0k
                ;
                ;       The files map to the following memory addresses in bank 0:
                ;
                ;                       BOOT.ROM        0000h-2fffh
                ;                       EXPAND.RAM      4000h-bfffh
                ;                       OPTION.ROM      17ffh-1fffh
                ;                       SCREEN.MEM      3000h-3bffh
                ;
                ;       The BANK program can also be a very useful tool in that after
                ;       it has been executed a high level language program can access
                ;       items in the alternate memory bank as disk files on drive x:
                ;
                ;       Of particular interest is the file SCREEN.MEM, notice that it
                ;       is 24 records long.  Each record (128 bytes) corresponds to a
                ;       line on the CRT (only the first 80 bytes of each record are in
                ;       the display window).  The first record of the file corresponds
                ;       to the first line of the CRT only if the CRT has not been
```

```
                ;       permitted to scroll since the last clear screen command was sent
                ;       to it.
                ;
                        Subttl  Constants & Program Mover
                        page
```

```
Banked Physical Driver                          MACRO-80 3.44   09-Dec-81
Constants & Program Mover

                                        Monitr  equ     0f000h          ;Base address of resident monitor
                                        Xcrtmv  equ     monitr+33h      ;Crt <-> Ram Move LDIR Simulator
                                        Xgetsl  equ     monitr+36h      ;Get driver select table address to hl

                                        Bavail  equ     0ff3ch          ;Pointer to beginning of available memory
                                        Eavail  equ     0ff3eh          ;Pointer to end of available memory

                                        bdos    equ     5
                                        dfcb    equ     5ch

                                        drvadr  equ     0fa80h          ;address for Bank driver
                                        stack   equ     0

                                                .z80

0000                                            Aseg
                                                Org     100h
0100    18 5A                                   jr      loadit

0102    43 6F 70 79                             db      'Copyright (C) 1982 Balcones Computer Corporation'
0106    72 69 67 68
010A    74 20 28 43
010E    29 20 31 39
0112    38 32 20 42
0116    61 6C 63 6F
011A    6E 65 73 20
011E    43 6F 6D 70
0122    75 74 65 72
0126    20 43 6F 72
012A    70 6F 72 61
012E    74 69 6F 6E
0132    20 54 72 61                             db      ' Transferred to Public Domain - (PD) 1983',26
0136    6E 73 66 65
013A    72 72 65 64
013E    20 74 6F 20
0142    50 75 62 6C
0146    69 63 20 44
014A    6F 6D 61 69
014E    6E 20 2D 20
0152    28 50 44 29
0156    20 31 39 38
015A    33 1A

015C    CD 03D8                 loadit: call    req822          ;see if machine is 820-II
015F    3A 005C                         ld      a,(dfcb)
0162    B7                              or      a
0163    28 30                           jr      z,bnkusg
0165    F5                              push    af
0166    CD 03F8                         call    ckspac          ;see if room for driver
0169    F1                              pop     af
016A    3D                              dec     a
016B    4F                              ld      c,a
```

```
  117   016C   21 0259                    ld      hl,driver
  118   016F   11 FA80                    ld      de,banked
  119   0172   D5                         push    de
  120   0173   C5                         push    bc
  121   0174   01 017F                    ld      bc,drvlen
  122   0177   ED B0                      ldir
  123   0179   26 00                      ld      h,0             ;indicate register return
  124   017B   CD F036                    call    XGetsl          ;get select table address
  125   017E   C1                         pop     bc
  126   017F   E5                         push    hl
  127   0180   09                         add     hl,bc
  128   0181   09                         add     hl,bc
  129   0182   36 03                      ld      (hl),3
  130   0184   23                         inc     hl
  131   0185   36 00                      ld      (hl),0
  132   0187   E1                         pop     hl
  133   0188   11 0026                    ld      de,2*16+3*2
  134   018B   19                         add     hl,de
  135   018C   D1                         pop     de
  136   018D   73                         ld      (hl),e
  137   018E   23                         inc     hl
  138   018F   72                         ld      (hl),d
  139   0190   0E 0D                      ld      c,13
  140   0192   C3 0005                    jp      bdos
  141
  142   0195   11 019D            bnkusg: ld      de,bnkmsg
  143   0198   0E 09                      ld      c,9
  144   019A   C3 0005                    jp      bdos
  145
  146   019D   55 73 61 67        bnkmsg: db      'Usage: BANK x:$'
  147   01A1   65 3A 20 42
  148   01A5   41 4E 4B 20
  149   01A9   78 3A 24
  150
  151   01AC                              ds      200h-103h-($-loadit),-1
  152   0259                      driver:
  153                                     .phase  Drvadr
  154
  155                                     Subttl  Bank Driver
  156                                     page
```

```
                                banked: ld      a,(hl)          ;get driver op
FA80    7E                              ld      c,a
FA81    4F                              inc     hl
FA82    23                              inc     a
FA83    3C                              jr      z,selbnk        ;if select op
FA84    28 51                           inc     hl
FA86    23                              inc     hl
FA87    23                              ld      d,(hl)          ;set track
FA88    56                              inc     hl
FA89    23                              inc     hl
FA8A    23                              ld      a,(hl)          ;set sector
FA8B    7E                              rrca
FA8C    0F                              ld      e,a
FA8D    5F                              inc     hl
FA8E    23                              inc     hl
FA8F    23                              ld      a,(hl)          ;set transfer address
FA90    7E                              inc     hl
FA91    23                              ld      h,(hl)
FA92    66                              ld      l,a
FA93    6F                              ld      b,0             ;preset crtldir op
FA94    06 00                           ld      a,h
FA96    7C                              cp      0c0h
FA97    FE C0                           jr      nc,bank1        ;if transfer outside banked area
FA99    30 01                           dec     b               ;set ram->crt
FA9B    05                      bank1:  ld      a,c             ;set read/write op
FA9C    79                              or      a
FA9D    B7                              jr      z,bank2         ;if write
FA9E    28 02                           ld      b,1             ;set crt->ram
FAA0    06 01                   bank2:  push    bc              ;save direction op
FAA2    C5                              or      d               ;check directory track
FAA3    B2                              jp      m,bank6         ;if directory operation
FAA4    FA FAC8                         cp      030h
FAA7    FE 30                           ld      a,c             ;set read/write switch
FAA9    79                              ld      bc,128
FAAA    01 0080                         jr      c,bank3         ;if not within screen memory
FAAD    38 05                           or      a
FAAF    B7                              jr      nz,bank4        ;if read
FAB0    20 05                           ld      c,80            ;only write one line
FAB2    0E 50                   bank3:  or      a               ;test read/write
FAB4    B7                              jr      z,bank5         ;if write
FAB5    28 01                   bank4:  ex      de,hl           ;set read
FAB7    EB                      bank5:  pop     af              ;get mover op to A
FAB8    F1                              ld      (stksav),sp     ;use high stack
FAB9    ED 73 FAC4                      ld      sp,stack
FABD    31 0000                         call    Xcrtmv          ;move it to/from crt bank
FAC0    CD F033                         ld      sp,0
FAC3    31 0000                 stksav  equ     $-2
FAC4                                    xor     a               ;always succeeds
FAC6    AF                              ret
FAC7    C9                      bank6:  ld      de,Direct       ;set directory address
FAC8    11 FADB                         dec     c
FACB    0D                              jr      nz,bank7        ;if directory write
FACC    20 01                           ex      de,hl
FACE    EB
```

```
  212     FACF    01 0080             bank7:  ld      bc,80h
  213     FAD2    ED B0                       ldir                    ;read or re-write directory
  214     FAD4    F1                          pop     af
  215     FAD5    AF                          xor     a
  216     FAD6    C9                          ret
  217
  218     FAD7    21 FB5B             selbnk: ld      hl,dph
  219     FADA    C9                          ret
  220
  221                                         Subttl  Directory Sector, Dpb & Dph
  222                                         page
```

```
                                Direct: db      0
FADC    42 4F 4F 54                     dc      'BOOT    R'
FAE0    20 20 20 20
FAE4    D2
FAE5    4F 4D                           db      'OM'
FAE7    00 00 00 40                     db      00,00,00,64
FAEB    01 02 03 04                     db      01,02,03,04     ;Bank 0 Memory locations        0000h-1fffh
FAEF    05 06 00 00                     db      05,06,00,00     ;                               2000h-2fffh
FAF3    00 00 00 00                     db      00,00,00,00
FAF7    00 00 00 00                     db      00,00,00,00

FAFB    00                              db      0
FAFC    4F 50 54 49                     db      'OPTION  ROM'
FB00    4F 4E 20 20
FB04    52 4F 4D
FB07    00 00 00 10                     db      00,00,00,16     ;Bank 0 Memory locations        17ffh-1fffh
FB0B    04 00 00 00                     db      04,00,00,00
FB0F    00 00 00 00                     db      00,00,00,00
FB13    00 00 00 00                     db      00,00,00,00
FB17    00 00 00 00                     db      00,00,00,00

FB1B    00                              db      0
FB1C    53 43 52 45                     db      'SCREEN  MEM'
FB20    45 4E 20 20
FB24    4D 45 4D
FB27    00 00 00 18                     db      0,0,0,24
FB2B    07 08 00 00                     db      07,08,00,00     ;Bank 0, Memory locations       3000h-3bffh
FB2F    00 00 00 00                     db      00,00,00,00
FB33    00 00 00 00                     db      00,00,00,00
FB37    00 00 00 00                     db      00,00,00,00

FB3B    00                              db      0
FB3C    45 58 50 41                     db      'EXPAND  RAM'
FB40    4E 44 20 20
FB44    52 41 4D
FB47    01 00 00 80                     db      01,00,00,80h
FB4B    09 0A 0B 0C                     db      09,10,11,12     ;Bank 0, Memory locations       4000h-5fffh
FB4F    0D 0E 0F 10                     db      13,14,15,16     ;                               6000h-7fffh
FB53    11 12 13 14                     db      17,18,19,20     ;                               8000h-9fffh
FB57    15 16 17 18                     db      21,22,23,24     ;                               a000h-bfffh

FB5B    0000 0000       dph:    dw      0,0,0,0
FB5F    0000 0000
FB63    FB7F FB6B               dw      dirbuf,dpb
FB67    0000 FB7B               dw      0,alloc

FB6B    0002            dpb:    dw      2               ;spt
FB6D    04 0F 01                db      4,15,1          ;blkshf, blkmsk, nullmsk
FB70    0018 0003               dw      24,3,128,0,-8   ;dsw,dirm,alloc01,chksiz,trk off
FB74    0080 0000
FB78    FFF8
FB7A    00                      db      0               ;128 byte sectors
```

```
  278      FB7B                alloc:  ds      4               ;allocation vector
  279      FB7F                dirbuf: ds      128             ;directory buffer
  280
  281                                  .dephase
  282      017F                drvlen  equ     $-driver
  283
  284                                  Subttl  System Identification
  285                                  page
```

```
                                ;;      Verify The machine this program is being run by Murphy or
                                ;       a Xerox 820-II
                                ;
03D8    3A F000         Req822: ld      a,(monitr)      ;make certain system is an 820-II
03DB    FE C3                   cp      0c3h            ;should be a jump instruction if 820
03DD    20 0D                   jr      nz,notii        ;if not give error message
03DF    2A F001                 ld      hl,(monitr+1)   ;follow reload monitor jump
03E2    7E                      ld      a,(hl)
03E3    FE F3                   cp      0f3h
03E5    20 05                   jr      nz,notii        ;if interrupts not disabled
03E7    23                      inc     hl
03E8    7E                      ld      a,(hl)
03E9    FE DB                   cp      0dbh
03EB    C8                      ret     z
03EC    E1              Notii:  pop     hl              ;pitch return address
03ED    11 0434                 ld      de,msg
03F0    0E 09           pmsg:   ld      c,9
03F2    CD 0005                 call    bdos
03F5    C3 0000                 jp      0

                                ;;      The pointer at Bavail points to the start of free memory, Eavail
                                ;       points to the end of free memory.  This test verifies that there
                                ;       is enough space for this program to fit in this un-allocated memory
                                ;       space.  If so the Eavail pointer is updated to the start of the driver -1.
                                ;       If not an error message is sent to the console.
                                ;
03F8    ED 5B FF3C      Ckspac: ld      de,(bavail)     ;get pointer to start of free address space
03FC    21 FA80                 ld      hl,drvadr       ;start of driver
03FF    B7                      or      a
0400    ED 52                   sbc     hl,de
0402    38 11                   jr      c,nroom         ;if drvadr < bavail then no space
0404    2A FF3E                 ld      hl,(eavail)     ;get pointer to end of available space
0407    11 FBFF                 ld      de,drvadr+drvlen
040A    ED 52                   sbc     hl,de
040C    38 07                   jr      c,nroom         ;if driver end > end of eavail then no space
040E    21 FA80                 ld      hl,drvadr       ;else update end pointer
0411    22 FF3E                 ld      (eavail),hl
0414    C9                      ret
0415    11 041A         nroom:  ld      de,nspace
0418    18 D6                   jr      pmsg

041A    46 72 65 65     Nspace: db      'Free memory space in use.$'
041E    20 6D 65 6D
0422    6F 72 79 20
0426    73 70 61 63
042A    65 20 69 6E
042E    20 75 73 65
0432    2E 24

0434    54 68 69 73     Msg:    db      'This program requires a Xerox 820-II Information Processor.$'
0438    20 70 72 6F
043C    67 72 61 6D
0440    20 72 65 71
```

```
  341     0444    75 69 72 65
  342     0448    73 20 61 20
  343     044C    58 65 72 6F
  344     0450    78 20 38 32
  345     0454    30 2D 49 49
  346     0458    20 49 6E 66
  347     045C    6F 72 6D 61
  348     0460    74 69 6F 6E
  349     0464    20 50 72 6F
  350     0468    63 65 73 73
  351     046C    6F 72 2E 24
  352
  353                         Subttl  Symbol Table
  354                         end
```

```
Banked Physical Driver                          MACRO-80 3.44   09-Dec-81
Symbol Table

Macros:

Symbols:
FB7B    ALLOC           FA9C    BANK1           FAA2    BANK2
FAB4    BANK3           FAB7    BANK4           FAB8    BANK5
FAC8    BANK6           FACF    BANK7           FA80    BANKED
FF3C    BAVAIL          0005    BDOS            019D    BNKMSG
0195    BNKUSG          03F8    CKSPAC          005C    DFCB
FB7F    DIRBUF          FADB    DIRECT          FB6B    DPB
FB5B    DPH             0259    DRIVER          FA80    DRVADR
017F    DRVLEN          FF3E    EAVAIL          015C    LOADIT
F000    MONITR          0434    MSG             03EC    NOTII
0415    NROOM           041A    NSPACE          03F0    PMSG
03D8    REQ822          FAD7    SELBNK          0000    STACK
FAC4    STKSAV          F033    XCRTMV          F036    XGETSL


No Fatal error(s)
```

```
                                        Title   Position encoded keyboard handler
                                ;;      Position encoded keyboard handler for the 820-II & 16/8
                                ;       professional computer.
                                ;
                                ;       Copyright 1983 (C) XEROX Corporation
                                ;
                                ;;              This is the stand alone rom addition to the Xerox
                                ;               820-II monitor. It is called once during monitor restart
                                ;               and at that time patches the monitor in ram to
                                ;               call the modified k/b,crt,Screenprint and printer
                                ;               routines.  It then moves in its own SIGNON overlay
                                ;               and jumps into it.
                                ;
                                ;               This SIGNON in addition to selecting the disk driver also
                                ;               moves into ram (in the spare driver area) translation
                                ;               tables and code for k/b and printer routines (crt is run
                                ;               out of rom).
                                ;
                                ;               There is also a RX BOOT overlay which is selected instead
                                ;               of the Xerox one. This loads the national translation
                                ;               tables from disk and then calls the Xerox BOOT.

                                        .z80

0000                            ver     defl    013

                                        subttl  Xerox ROM dependant equates
                                        page
```

```
                                ;;      The following equates are dependant on the revision of the Xerox ROM
                                ;       These are compatible with Ver. 4.02
                                ;
1800                            rx1984  equ     1800h           ;start of rx1984
0800                            romsiz  equ     800h            ;size of eprom
F000                            monitr  equ     0f000h          ;start of monitor ad jump table
F1EC                            savstk  equ     x'f1ec'         ;stack save address
FF10                            ctcvec  equ     x'ff10'         ;counter timer interrupt vector
FF18                            sysvec  equ     0ff18h          ;vector page
FF1A                            kbvec   equ     sysvec+2        ;keyboard vector
FC5D                            tca     equ     0fc5dh          ;start of 4.02 transient command area
0002                            boff1   equ     ('A'-'@')*2     ;A command vector in command table
0018                            boff2   equ     ('L'-'@')*2     ;L command vector in command table
0019                            sioff   equ     19h             ;sioout vector in monitor table
000A                            kboff   equ     0ah             ;offset in k/b int service for patch
0010                            fcrtof, equ     10h             ;fast crt out vector in monitor table
0012                            crtcall equ     12h             ;offset in crt driver for patch
F006                            const   equ     monitr+6
F009                            conin   equ     monitr+9
F003                            warm    equ     monitr+3
F01B                            select  equ     monitr+1bh
F01E                            home    equ     monitr+1eh
F024                            read    equ     monitr+24h
F03C                            config  equ     monitr+x'3c'    ;monitor configure routine
F03F                            siordy  equ     monitr+x'3f'    ;sio channel b output ready status
F066                            idle    equ     monitr+x'66'    ;idle while i/o pending
F06C                            mntrex  equ     monitr+x'6c'    ;monitor jump table expansion area
F06C                            kybdlp  equ     monitr+x'6c'    ;low profile keyboard entry address
F06F                            key2    equ     monitr+x'6f'    ;keyboard xlat char entry address
F072                            key5    equ     monitr+x'72'    ;keyboard without xlat char entry address
F075                            pnext   equ     monitr+x'75'    ;print message after call
F078                            prboff  equ     monitr+x'78'    ;promt boot entry
0182                            crtd1   equ     0182h
0196                            crtd2   equ     0196h
01DD                            grpad   equ     1ddh            ;address of set graphics attribute
1078                            xrsign  equ     1078h           ;adress of xr signon overlay
0060                            sigoff  equ     60h             ;offset of ver value in signon
11C0                            xrboot  equ     11c0h           ;boot o/l address
F167                            mkey2   equ     x'f167'         ;keyboard handler entry address
F18F                            mkey5   equ     x'f18f'         ;return from keyboard and timer interrupt add
F22F                            sprnt1  equ     0f22fh          ;patch address for screen print
F232                            sprnt2  equ     0f232h          ;return address from RX screenprint code
F293                            crtoff  equ     x'f293'         ;switch to ram side
F339                            prvatt  equ     0f339h          ;contains address of current set attribute
FA62                            prompt  equ     0fa62h          ;4.02 PROMPT
FA95                            mprmt0  equ     x'fa95'         ;4.01 PRMT0
FC3D                            mpnext  equ     x'fc3d'         ;4.01 PNEXT

                                ;       Data Addreses

ED80                            bootbf  equ     0ed80h
F0E3                            mask    equ     0f0e3h
F091                            confg   equ     0f091h
```

```
Position encoded keyboard handler       MACRO-80 3.44   09-Dec-81
Xerox ROM dependant equates

   87     F20E                          spact   equ     x'f20e'
   88     F319                          gold    equ     0f319h
   89     F360                          seltab  equ     0f360h
   90     F470                          fivdpb  equ     0f470h
   91     F708                          rigdpb  equ     0f708h
   92     F800                          tabled  equ     0f800h          ;space for rx code
   93     FA11                          phytrk  equ     0fa11h
   94     FF3C                          availb  equ     x'ff3c'         ;bottom available ram memory
   95     FF50                          intstk  equ     x'ff50'         ;tempory stack address
   96     FF54                          steprt  equ     0ff54h
   97     FF5C                          linbuf  equ     0ff5ch
   98     FFAC                          cursor  equ     0ffach
   99     FFB2                          leadin  equ     0ffb2h
  100     FFB3                          attrib  equ     0ffb3h          ;address of attributes enabled flag
  101     FFB4                          chrsav  equ     0ffb4h
  102
  103                                   ;       Port addressess
  104                                   ;
  105     001D                          sysctl  equ     1dh
  106     001C                          syspio  equ     1ch
  107     0005                          siodpb  equ     05h
  108     0010                          wd1797  equ     10h
  109     001E                          kbdat   equ     1eh
  110     0019                          ctc1    equ     x'19'           ;ctc1 port address
  111
  112                                   ;       Other Equates
  113                                   ;
  114     0081                          encntr  equ     x'81'           ;enable ctc command
  115     0001                          stcntr  equ     x'01'           ;stop ctc command
  116     0000                          rev0    equ     x'00'           ;4.00 Revision Level
  117     0001                          rev1    equ     x'01'           ;4.01 Revision Level
  118     0064                          rev50   equ     5*100-400       ;5.00 Revision level
  119     003C                          cnfgoff equ     x'3c'           ;monitor configuration offset
  120     0006                          cnfbyte equ     x'06'           ;configuration subroutine byte offset
  121     0008                          kblp    equ     x'08'           ;configuration bit id for LPKYBD
  122     0008                          romofs  equ     x'08'           ;PROMPT offset between 4.02 & 4.01 monitor
  123     0001                          lpkofs  equ     x'01'           ;additonal sector required for table storage
  124     007B                          upper   equ     'z'+1           ;upper limit for alpha test
  125     0061                          lower   equ     'a'             ;lower limit for alpha test
  126     0020                          upascii equ     'a'-'A'         ;set to upper case ASCII mask
  127     0000                          zero    equ     0               ;zero
  128     00FF                          setflg  equ     x'ff'           ;set flag
  129
  130                                   ;       Equates
  131                                   ;
  132     0004                          c.five  equ     04
  133     0006                          c.sasi  equ     06
  134     0001                          o.term  equ     0001h
  135     0300                          sasidl  equ     300h
  136
  137                                   ;       Internal equates
  138                                   ;
  139     001D                          rtab1   equ     29              ;rigid disk tables sector 1
  140     001E                          rtab2   equ     30              ;  "     "       "     "    2
  141     0004                          ftab1   equ     04              ;floppy "      "     "    1
  142     0005                          ftab2   equ     05              ;  "     "      "     "    2
```

```
  143     0006                          ftab3   equ     06              ;  "    "       "       "    3
  144     0000                          lang    equ     00              ;offset of language no. in index table
  145     0001                          kbrd    equ     01              ;offset of k/b flag in index table
  146     0002                          font    equ     02              ;offset of font flag in index table
  147     0003                          prnt    equ     03              ;offset of printer flag in index table
  148     0004                          kbrdtb  equ     04              ;offset of k/b tables in first sector
  149     001A                          clrs    equ     1ah             ;clear screen
  150     001B                          esc     equ     1bh             ;escape key
  151     0004                          eot     equ     04h             ;end of text
  152     000D                          cr      equ     0dh             ;carriage return
  153     000A                          lf      equ     0ah             ;line feed
  154
  155                                   subttl  RX1984 Restart
  156                                   page
```

```
                                start:
                                        .phase  rx1984
                                ;;      RX1984
                                ;       Entry here from Xerox monitor bfore entering SIGNON.
                                ;
                                ;       Input:-
                                ;               hl - cmdtab
                                ;               de - seltab
                                ;               bc - cloc
                                ;
                                ;
1800    C5                              push    bc
1801    D5                              push    de
1802    E5                              push    hl
1803    21 0000                         ld      hl,0
1806    CD F03C                         call    config          ;get monitor configuration
1809    7C                              ld      a,h
180A    FE 00                           cp      rev0
180C    CA 187E                         jp      z,noload        ;skip if below 4.01
180F    FE 64                           cp      rev50
1811    D2 187E                         jp      nc,noload       ;skip if 5.00 or above
1814    21 1B1A                         ld      hl,rv1tbl       ;4.01 spring board table
1817    FE 01                           cp      rev1
1819    28 03                           jr      z,tbxfer        ;skip if 4.01
181B    21 1B29                         ld      hl,rv2tbl       ;4.02+ spring board table
181E    11 F06C                 tbxfer: ld      de,mntrex
1821    01 000F                         ld      bc,jtblsz
1824    F5                              push    af              ;save monitor level
1825    ED B0                           ldir                    ;append monitor table with lpkybd jmp vectors
1827    DD 2A F03D                      ld      ix,(monitr+cnfgoff+1)   ;set address at monitor config:
182B    DD 7E 06                        ld      a,(ix+cnfbyte)
182E    F6 08                           or      kblp            ;set low profile bit flag
1830    DD 77 06                        ld      (ix+cnfbyte),a
1833    F1                              pop     af              ;recover monitor level

                                ;;      Alter BOOT commnd vectors
                                ;
1834    DD E1                           pop     ix              ;cmdtab address
1836    DD E5                           push    ix
1838    DD 36 02 3D'                    ld      (ix+boff1),low rxboot           ;assume 4.01 monitor
183C    DD 36 03 06'                    ld      (ix+boff1+1),high rxboot
1840    DD 36 18 3D'                    ld      (ix+boff2),low rxboot
1844    DD 36 19 06'                    ld      (ix+boff2+1),high rxboot
1848    FE 01                           cp      rev1            ;monitor check
184A    28 10                           jr      z,soout.        ;skip if 4.01 monitor
184C    DD 36 02 45'                    ld      (ix+boff1),low (rxboot+romofs)  ;4.02+ monitor boot over addr
1850    DD 36 03 06'                    ld      (ix+boff1+1),high (rxboot+romofs)
1854    DD 36 18 45'                    ld      (ix+boff2),low (rxboot+romofs)
1858    DD 36 19 06'                    ld      (ix+boff2+1),high (rxboot+romofs)

                                ;;      Alter keyboard interrupt service
                                ;
```

```
 212     185C    DD 2A FF1A                  Soout:  ld      ix,(kbvec)      ;k/b int vector
 213     1860    DD 36 0A CD                         ld      (ix+kboff),0cdh ;CALL operation
 214     1864    DD 36 0B 6C                         ld      (ix+kboff+1),low kybdlp
 215     1868    DD 36 0C F0                         ld      (ix+kboff+2),high kybdlp
 216
 217                                         ;;      Move in RX SIGNON to o/l area and execute it
 218                                         ;
 219     186C    E1                                  pop     hl
 220     186D    D1                                  pop     de
 221     186E    C1                                  pop     bc
 222     186F    C1                                  pop     bc              ;throw away  return address
 223     1870    21 0552'                            ld      hl,rxsign       ;rom address
 224     1873    11 FC5D                             ld      de,tca          ;o/l area
 225     1876    01 00EB                             ld      bc,rxsigl       ;length
 226     1879    ED B0                               ldir
 227     187B    C3 FC5D                             jp      tca             ;GO SIGN ON
 228     187E    E1                          noload: pop     hl
 229     187F    D1                                  pop     de
 230     1880    C1                                  pop     bc
 231     1881    3E FF                               ld      a,x'ff'         ;wrong monitor
 232     1883    A7                                  and     a               ;load signon from monitor
 233     1884    C9                                  ret
 234
 235                                                 subttl  ROM resident CRT Driver
 236                                                 page
```

```
                                        ;;      Crtdvr - Crt Driver RX Addition.
                                        ;
1885    2A FFAC                 Rxcrt:  ld      hl,(cursor)     ;set cursor address
1888    3A FFB4                         ld      a,(chrsav)      ;retrieve character under cursor
188B    77                              ld      (hl),a          ;replace character under cursor
188C    32 F319                         ld      (gold),a        ;bury balcones gold
188F    3A FFB2                         ld      a,(leadin)      ;set leadin state
1892    B7                              or      a
1893    C2 0196                         jp      nz,crtd2        ;if processing escape sequence
1896    3A F0E3                         ld      a,(mask)        ;get keyboard mask
1899    A1                              and     c
189A    4F                              ld      c,a
189B    FE 20                           cp      ' '
189D    DA 0196                         jp      c,crtd2         ;if control code
18A0    CD 18A6                         call    fonchk          ;do font translation
18A3    C3 0182                         jp      crtd1           ;go to XR code

                                        ;;      Subroutine fonchk does the font translation for national
                                        ;       character sets.
                                        ;       entry:  C contains the character
                                        ;       exit:   C contains the translation
                                        ;
18A6    E5                      Fonchk: push    hl              ;save cursor posn.
18A7    79                              ld      a,c             ;get char in a
18A8    E6 80                           and     10000000b       ;preserve attribute bit
18AA    F5                              push    af
18AB    21 FFB3                         ld      hl,attrib       ;point to attribute enabled flag
18AE    B6                              or      (hl)            ;test if set
18AF    28 0A                           jr      z,fon1          ;no attribute bit - go do translation
18B1    11 01DD                         ld      de,grpad        ;check if graphics mode
18B4    2A F339                         ld      hl,(prvatt)     ;current attribute mode
18B7    ED 52                           sbc     hl,de
18B9    28 0F                           jr      z,fon2          ;grahics mode - no translate
18BB    79                      fon1:   ld      a,c             ;here to do translate
18BC    CB BF                           res     7,a             ;clear attribute bit
18BE    21 F960                         ld      hl,fontbl       ;address of exceptions table
18C1    01 000D                         ld      bc,fontsz       ;size of exceptions table
18C4    ED B1                           cpir                    ;search for char. in exceptions
18C6    4F                              ld      c,a             ;restore char to c
18C7    CC 18CF                         call    z,fntran        ;if found do translation
18CA    F1                      fon2:   pop     af              ;retrieve attribute bit
18CB    B1                              or      c               ;or it in
18CC    4F                              ld      c,a
18CD    E1                              pop     hl              ;retrieve cursor
18CE    C9                              ret

                                        ;;      s/r fntran translates font characters
                                        ;       entry: (HL) - address+1 of char to be translated in fontbl
                                        ;       exit:  (c)  - translated character
                                        ;
18CF    2B                      Fntran: dec     hl              ;back to byte to be translated
18D0    01 000D                         ld      bc,fontsz       ;size of table
18D3    09                              add     hl,bc           ;add to address of char. to be translated
```

```
  292   18D4  4E                        ld      c,(hl)          ;get translated value
  293   18D5  C9                        ret
  294
  295                           ;;      This routine is called from the Xerox screenprint handler. It
  296                           ;       picks up the character to be printed from the crt ram, does a
  297                           ;       reverse font translate,replaces any control codes with a space,
  298                           ;       does a printer translate and outputs the character to the printer.
  299                           ;       entry:- HL - address of byte to be printed
  300                           ;
  301   18D6  E5                scrprt: push    hl
  302   18D7  C5                        push    bc
  303   18D8  7E                        ld      a,(hl)          ;byte for printing
  304   18D9  CB BF                     res     7,a             ;ignore attribute bit
  305                                                           ;do reverse font translate
  306   18DB  21 F96D                   ld      hl,fontbl+fontsz        ;point to translates
  307   18DE  01 000D                   ld      bc,fontsz
  308   18E1  ED B1                     cpir                    ;search for char.
  309   18E3  20 07                     jr      nz,scr01        ;not in table
  310                                                           ;in table convert to media code
  311   18E5  01 000D                   ld      bc,fontsz       ;offset back to media code
  312   18E8  37                        scf
  313   18E9  ED 42                     sbc     hl,bc           ;points to media code
  314   18EB  7E                        ld      a,(hl)
  315   18EC                  scr01:                            ;here with media code
  316   18EC  FE 20                     cp      20h             ;is it a control code
  317   18EE  30 02                     jr      nc,scr02        ;no
  318   18F0  3E 20                     ld      a,20h           ;yes. substitute a space
  319   18F2  CD 1966         scr02:    call    potran          ;do printer translation
  320   18F5  20 0D                     jr      nz,scr03        ;no translation done. go output char.
  321                                                           ;translation done.check escape bit
  322   18F7  CB 7F                     bit     7,a             ;escape bit
  323   18F9  28 09                     jr      z,scr03         ;not set go output char
  324   18FB  4F                        ld      c,a             ;set. save char.
  325   18FC  3E 1B                     ld      a,esc           ;output an escape
  326   18FE  CD 1959                   call    posout          ;output routine
  327   1901  79                        ld      a,c             ;restore char.
  328   1902  CB BF                     res     7,a             ;clear escape bit
  329   1904  CD 1959         scr03:    call    posout          ;print char
  330   1907  C1                        pop     bc
  331   1908  E1                        pop     hl
  332   1909  C3 F232                   jp      sprnt2          ;return to Xerox code
  333
  334                           ;;      Exception print driver - ROM entry point
  335                           ;
  336   190C  C5                Rmposend:push   bc
  337   190D  E5                        push    hl
  338   190E  CD 1914                   call    posend
  339   1911  E1                        pop     hl
  340   1912  C1                        pop     bc
  341   1913  C9                        ret
  342
  343                           ;;      Posend - deals with character translation and escape
  344                           ;               sequences for the diablo 630
  345                           ;               input--- a contains char for output to channel b
  346   1914  4F                Posend: ld      c,a
  347   1915  3A F9A6                   ld      a,(escsq)       ;in an escape sequence?
```

```
 348  1918  B7                        or      a
 349  1919  20 22                     jr      nz,pos04        ;yes
 350  191B  79                        ld      a,c
 351  191C  FE 1B                     cp      esc             ;escape char?
 352  191E  20 07                     jr      nz,pos01        ;no
 353  1920  CD 1959                   call    posout          ;output char
 354  1923  32 F9A6                   ld      (escsq),a       ;set escape sequence flag
 355  1926  C9                        ret
 356  1927                    pos01:                          ;not escape char
 357  1927  CD 1966                   call    potran          ;do translation if neccessary
 358  192A  20 0D                     jr      nz,pos03        ;wasn't neccessary
 359  192C  CB 7F                     bit     7,a             ;escape marker set?
 360  192E  28 09                     jr      z,pos02         ;no
 361  1930  4F                        ld      c,a
 362  1931  3E 1B                     ld      a,esc
 363  1933  CD 1959                   call    posout          ;output escape char
 364  1936  79                        ld      a,c
 365  1937  CB BF                     res     7,a             ;clear escape marker
 366  1939                    pos02:                          ;escape marker not set
 367  1939                    pos03:                          ;no translation
 368  1939  CD 1959                   call posout             ;output char
 369  193C  C9                        ret
 370  193D                    pos04:                          ;escape sequence
 371  193D  FE FF                     cp      0ffh            ;3rd byte?
 372  193F  20 06                     jr      nz,pos05        ;no 2nd
 373  1941  79                        ld      a,c
 374  1942  CD 1959                   call    posout          ;output char
 375  1945  18 0D                     jr      pos06
 376  1947  79                pos05:  ld      a,c             ;2nd byte of escape sequence
 377  1948  CD 1959                   call    posout          ;output byte
 378  194B  CD 1986                   call    poesc           ;search escape table for char
 379  194E  20 04                     jr      nz,pos06        ;not present---2 byte sequence
 380  1950  3E FF                     ld      a,0ffh          ;set sequence for 3rd byte
 381  1952  18 01                     jr      pos07
 382  1954  AF                pos06:  xor     a               ;end of 2 byte sequence
 383  1955  32 F9A6            pos07:  ld      (escsq),a       ;toggle escape sequence flag
 384  1958  C9                        ret
 385  1959  47                posout: ld      b,a
 386  195A  CD F03F            siox1:  call    siordy
 387  195D  CC F066                    call    z,idle
 388  1960  28 F8                     jr      z,siox1
 389  1962  78                        ld      a,b
 390  1963  D3 05                     out     (siodpb),a
 391  1965  C9                        ret
 392
 393                            ;;      Potran - does printer translation if neccessary and returns
 394                            ;               a flag to indicate if translation has been done.
 395                            ;               input- a  char for translaation
 396                            ;
 397                            ;               output- a  (translated) char
 398                            ;                       z  set if char is translated (otherwise reset)
 399                            ;
 400  1966  21 F97A            Potran: ld      hl,prntbl       ;print exceptions table
 401  1969  01 0016                    ld      bc,prntsz       ;size of table
 402  196C  ED B1                      cpir
 403  196E  C0                        ret     nz              ;no match - don't translate
```

```
 404   196F  2B                        dec     hl              ;translate
 405   1970  01 0016                   ld      bc,prntsz
 406   1973  09                        add     hl,bc           ;offset to translation
 407   1974  7E                        ld      a,(hl)          ;translation byte
 408   1975  B7                        or      a               ;if zero, requires overstriking sequence
 409   1976  20 0B                     jr      nz,ptr01        ;non-zero - go output char
 410                                                           ;zero - use next 2 bytes in table as sequence
 411   1978  23                        inc     hl
 412   1979  7E                        ld      a,(hl)          ;first byte
 413   197A  CD 1959                   call    posout
 414   197D  3E 08                     ld      a,08h           ;backspace
 415   197F  CD 1959                   call    posout
 416   1982  23                        inc     hl              ;second byte
 417   1983  AF                ptr01:  xor     a               ;set z for return flag
 418   1984  7E                        ld      a,(hl)          ;get translation
 419   1985  C9                        ret
 420
 421                           ;;      Poesc - searches the escape table for a match with the char
 422                           ;               passed in a. if found returns with z set otherwise
 423                           ;               z is clear
 424                           ;
 425   1986  21 198F           Poesc:  ld      hl,pesctb       ;table of escape sequences
 426   1989  01 0007                   ld      bc,esctsz       ;size of table
 427   198C  ED B1                     cpir
 428   198E  C9                        ret
 429
 430   198F  09 0B 0C 1E       pesctb: defb    09h,0bh,0ch,1eh,1fh,16h,11h     ;630 daisy printer
 431   1993  1F 16 11
 432   0007                    esctsz  equ     $-pesctb
 433
 434
 435                           ;;      Function:- to deal with characters form a position
 436                           ;               encoded keyboard.
 437                           ;       input:-     A  character read from PIO
 438                           ;                   CMD/STATUS byte
 439                           ;                        bit 7 -CMD/STATUS byte if set
 440                           ;                        bit 6 -upstroke flag
 441                           ;                        bit 5 -y axis negative (mouse)
 442                           ;                        bit 4 -x axis negative (mouse)
 443                           ;                        bit 3 -mouse active
 444                           ;                        bit 2 -ctrl key station active
 445                           ;                        bit 1 -shift key station active
 446                           ;                        bit 0 -lock key station active
 447                           ;                   First data byte
 448                           ;                        bit 7 -Always reset
 449                           ;                        bits(6-0) -key station or x mouse displacement
 450                           ;                                   Second mouse data byte
 451                           ;                        bit 7 -Always reset
 452                           ;                        bits(6-0) -y mouse displacement
 453                           ;
 454                           ;       output:-    1) Carry set -- command byte or sequence error
 455                           ;                   2) Carry clear -- translated character returned in A
 456                           ;
 457   1996  2F                Pekhdl: cpl                     ;complement keyboard byte
 458   1997  D5                        push    de              ;save registers
 459   1998  16 00                     ld      d,zero          ;get flags
```

```
                                        bit     cmd,a           ;command byte?
                                        jr      z,kypos         ;skip to position byte handler
                                        jr      cmdb            ;skip to command byte handler
                                nochar  equ     $
                                peknoc2:ld      a,d             ;clear command byte (non-valid position byte)
                                cmdb:   ld      (cmdstat),a     ;save command-status byte
                                        bit     mouse,a         ;mouse cmd?
                                        jr      z,peknoc        ;skip if not mouse
                                        ld      hl,mstbl
                                        res     xy,(hl)
                                peknoc: call    stpctcl         ;reset repeat flag
                                peknoc1:scf                     ;this no the position byte
                                pekex:  pop     de              ;recover registers
                                        jp      lpkext          ;jmp instead of ret - small interrupt stack
                                kypos:  ld      e,a             ;save position code
                                        ld      hl,cmdstat      ;fetch command byte
                                        bit     cmd,(hl)        ;out of sync check
                                        jr      z,peknoc        ;quit if no command byte
                                        bit     mouse,(hl)      ;test for mouse movement
                                        jp      nz,mice         ;skip if mouse moved
                                        bit     ustrk,(hl)      ;test key posistion
                                        jr      nz,upstrk       ;jump if special upstroke
                                        call    ctrtst          ;test for control codes
                                        jr      z,peknoc2       ;quit if non printable control character
                                        call    tblsel          ;select translation table
                                        call    alphtst         ;test for alpha lock char
                                        call    rptst           ;test for repeat keys
                                charout:ld      hl,cmdstat      ;fetch command byte
                                        ld      (hl),d          ;clear command byte (valid position byte)
                                        and     a               ;clear carry
                                        jr      pekex

                                ;;      Character is tested for the lock, shift, and ctrl key station.
                                ;
                                ;       input   a - key station code
                                ;
                                ;       output z - set if lock,shift, or ctrl key station
                                ;
                                Ctrtst: push    hl              ;save registers
                                        ld      hl,ctrltb       ;non printable char table
                                        ld      bc,cntctr       ;byte count of table
                                        cpir                    ;search table
                                        pop     hl
                                        ret

                                ;;      The up-stroke is tested for special up-stroke key-stations.
                                ;
                                ;       input   a - key station code
                                ;               strkup - user enable flag
                                ;
                                ;       output  a - translated up-stroke key-station code
                                ;
                                Upstrk: ld.     hl,mstbl        ;user enable up-stroke flag
                                        bit     strkup,(hl)
                                        jr      z,nochar        ;quit if user inhibited
                                        ld      hl,ups          ;exception key-station table
```

```
                                        ld      bc,upssz        ;byte count of table
                                        cpir                    ;search table
                                        jr      nz,nochar       ;quit if no match
                                        dec     hl              ;get exception
                                        ld      bc,upssz
                                        add     hl,bc
                                        ld      a,(hl)
                                        jr      charout         ;return translated character

                                ;;      The appropriate keyboard translation table is selected
                                ;
                                ;       input   hl - command-status address
                                ;               de - key station code
                                ;
                                ;       output  a  - translated key station code
                                ;
                                Tblsel: ld      a,(hl)          ;move cmd-status byte
                                        push    hl              ;save command-status ptr
                                        ld      hl,shtab        ;preset to shift table
                                        bit     shift,a         ;shift bit set?
                                        jr      nz,cmdbl        ;skip if set
                                        ld      hl,ctdtab       ;preset to control table
                                        bit     ctrl,a          ;control bit set?
                                        jr      nz,cmdbl        ;skip if set
                                        ld      hl,tabl         ;preset to un shifted table
                                        bit     lock,a          ;lock key set
                                        jr      z,cmdbl         ;skip if reset
                                        ld      a,(shftlck)     ;lock key set
                                        and     a               ;test for shift lock (not alpha lock)
                                        jr      z,cmdbl         ;skip if reset
                                        ld      hl,shtab        ;preset to shift table
                                cmdbl:                          ;here with translation table address in hl
                                        add     hl,de           ;index into table
                                        ld      a,(hl)          ;get translated char
                                        pop     hl              ;recover command-status ptr
                                        ret

                                ;;      If the lock key is depressed, the translated character is
                                ;       tested to see if it is an alphabet.  If it is lower case,
                                ;       then it is forced upper case.
                                ;
                                ;       input   hl - command-status address
                                ;               a  - translated character
                                ;
                                ;       output a  - translated character(upper chase if alpha+lock)
                                ;
                                Alphtst:bit     lock,(hl)       ;test alpha lock flag
                                        ret     z               ;quit if not alpha lock
                                        cp      upper           ;test for upper alpha range
                                        jr      nc,alphexc      ;skip if non alpha range
                                        cp      lower           ;test for lower case alpha range
                                        jr      c,alphexc       ;skip if not lower alpha case
                                        sub     upascii         ;set upper case ASCII alpha character

                                ;;      Three additional caracters are allowed for the alpha lock key
                                ;
```

```
                        ;       input   hl - command-status address
                        ;               a  - translated character
                        ;
                        ;       output  a  - upper case exception
                        ;
1A30    21 F935         Alphexc:ld      hl,captab               ;lock exception table
1A33    01 0003                 ld      bc,cptbsz               ;table size
1A36    ED B1                   cpir                            ;search
1A38    C0                      ret     nz                      ;quit if not found
1A39    23                      inc     hl                      ;get exception
1A3A    23                      inc     hl
1A3B    7E                      ld      a,(hl)
1A3C    C9                      ret

                        ;;      Checks for repeat character.  If repeat character, the millisec
                        ;       timer is vector address is modified and the timer is set up
                        ;       for 0.5 second.  The timer is kicked off.
                        ;
                        ;       input   a - translated character
                        ;
1A3D    21 F940         Rptst:  ld      hl,rptbl        ;repeat char table
1A40    01 0013                 ld      bc,cntrp        ;number of repeat chars
1A43    ED B1                   cpir                    ;test for repeat chars
1A45    C0                      ret     nz              ;quit if not repeat char
1A46    2A F93C                 ld      hl,(tick)       ;millisec count
1A49    22 F9A8                 ld      (millcnt),hl    ;save it in table
1A4C    21 F9AA                 ld      hl,rptchar      ;repeat char save address
1A4F    77                      ld      (hl),a          ;save repeat char
1A50    F5                      push    af
1A51    23                      inc     hl              ;repeat flag address
1A52    36 FF                   ld      (hl),setflg     ;set repeat flag
1A54    2A FF12                 ld      hl,(ctcvec+2)   ;get 1 millisec interrupt vector
1A57    22 F9AC                 ld      (save),hl       ;save it
1A5A    21 F9D0                 ld      hl,rptclk       ;kybd repeat key timer
1A5D    22 FF12                 ld      (ctcvec+2),hl   ;substitute it
1A60    3E 81                   ld      a,encntr        ;enable millisec timmer
1A62    D3 19                   out     (ctc1),a        ;do it
1A64    F1                      pop     af              ;recover character
1A65    C9                      ret

                        ;;      This routine stops the millisecond timer and restores the
                        ;       original timer vector
                        ;
1A66    21 F9AB         Stpctc1:ld      hl,rptflg       ;fetch repeat char flag
1A69    7E                      ld      a,(hl)
1A6A    A7                      and     a               ;set flags
1A6B    C8                      ret     z               ;quit if no repeat keys
1A6C    72                      ld      (hl),d          ;clear repeat char flag
1A6D    2A F9AC                 ld      hl,(save)       ;original 1 millisec interrupt address
1A70    22 FF12                 ld      (ctcvec+2),hl   ;restore it
1A73    3A F20E                 ld      a,(spact)       ;fetch screen print flag
1A76    A7                      and     a
1A77    C0                      ret     nz              ;don't kill timer, if screen printing
1A78    3E 01                   ld      a,stcntr        ;stop timer
1A7A    D3 19                   out     (ctc1),a
1A7C    C9                      ret
```

```
                                ;;      This is the optical mouse handler.  The delta x and y mouse
                                ;       movement is accumulate and stored in memory locations.
                                ;
                                ;       input   a=      mouse delta either x or y
                                ;               xy=     zero for x mouse delta
                                ;                       x'ff' for y mouse delta
                                ;               hl=     cmdstat address
                                ;               cmdstat=bit 3   mouse moved
                                ;                       bit 4   x delta negative
                                ;                       bit 5   y delta negative
                                ;               mstbl   =bit 7   mouse enabled user flag
                                ;                       bit 6   intrp/npol
                                ;                       bit 1   y/nx byte
                                ;                       bit 0   mouse table is updated
                                ;               mbyte   =interrupt return byte
                                ;               xmax    =maximum horizontal display units
                                ;               ymax    =maximum vertical display units
                                ;
                                ;       output mstbl=   bit 0, mouse table updated
                                ;               xloc=   x position of mouse
                                ;               yloc=   y position of mouse
                                ;               dxmv=   prior x signed displacement
                                ;               dymv=   prior y signed displacement
                                ;
                                ;       special requirements
                                ;               The majority of the mouse-pointer table is require memory
                                ;               resident in the in the user address space above X'BFFF'
                                ;               since this handle is ROM resident.  Variables MSTBL AND MSPTR
                                ;               reside in keyboard RAM space.  Variable MSPTR pointers to
                                ;               where the user mouse table resides.  These variable are
                                ;               update by both the handler and the user.  The following
                                ;               data structure resides in the user's RAM space only if
                                ;               the two-button mouse pointer is required for the applicaton:
                                ;               (MSPTR+0)       =MBYTE
                                ;               (MSPTR+1)       =XMAX-LSB
                                ;               (MSPTR+2)       =    -MSB
                                ;               (MSPTR+3)       =YMAX-LSB
                                ;               (MSPTR+4)       =    -MSB
                                ;               (MSPTR+5)       =XLOC-LSB
                                ;               (MSPTR+6)       =    -MSB
                                ;               (MSPTR+7)       =YLOC-LSB
                                ;               (MSPTR+8)       =    -MSB
                                ;               (MSPTR+9)       =DXMV-LSB
                                ;               (MSPTR+A)       =    -MSB
                                ;               (MSPTR+B)       =DYMV-LSB
                                ;               (MSPTR+C)       =    -MSB
                                ;
1A7D    5F                      Mice:   ld      e,a             ;save mouse delta
1A7E    7E                              ld      a,(hl)
1A7F    47                              ld      b,a             ;save cmd/status byte
1A80    21 F95D                         ld      hl,mstbl        ;mouse table
1A83    CB 7E                           bit     msflg,(hl)      ;test for user enabled
1A85    CA 19A0                         jp      z,nochar        ;quit if mouse handler is not enabled
1A88    DD E5                           push    ix              ;save register
1A8A    DD 2A F95E                      ld      ix,(msptr)      ;fetch user's table
```

```
1A8E    CB 4E                       bit     xy,(hl)         ;test y/nx mouse byte flag
1A90    20 26                       jr      nz,micey        ;skip if y axis delta
1A92    DD 6E 05                    ld      l,(ix+5)        ;fetch current x position
1A95    DD 66 06                    ld      h,(ix+6)
1A98    7B                          ld      a,e             ;save unsigned mouse delta byte
1A99    CB 60                       bit     xneg,b          ;test polarity of x delta
1A9B    CD 1AEB                     call    mice1           ;add delta & do min value check
1A9E    DD 73 09                    ld      (ix+x'9'),e     ;save signed mouse delta word
1AA1    DD 72 0A                    ld      (ix+x'a'),d
1AA4    DD 5E 01                    ld      e,(ix+1)        ;fetch max position value
1AA7    DD 56 02                    ld      d,(ix+2)
1AAA    CD 1B00                     call    mice2           ;do max value check
1AAD    DD 75 05                    ld      (ix+5),l        ;save position
1AB0    DD 74 06                    ld      (ix+6),h
1AB3    DD E1               micex1: pop     ix              ;restore regiser
1AB5    C3 19B0                     jp      peknocl         ;return to wait for y mouse byte
1AB8    DD 6E 07            micey:  ld      l,(ix+7)        ;fetch current y position
1ABB    DD 66 08                    ld      h,(ix+8)
1ABE    7B                          ld      a,e             ;save unsigned mouse delta byte
1ABF    CB 68                       bit     yneg,b          ;test polarity of y delta
1AC1    CD 1AEB                     call    mice1           ;add delta & do min value check
1AC4    DD 73 0B                    ld      (ix+x'b'),e     ;save signed mouse delta word
1AC7    DD 72 0C                    ld      (ix+x'c'),d
1ACA    DD 5E 03                    ld      e,(ix+3)        ;fetch max position value
1ACD    DD 56 04                    ld      d,(ix+4)
1AD0    CD 1B00                     call    mice2           ;do max value check
1AD3    DD 75 07                    ld      (ix+7),l        ;save position
1AD6    DD 74 08                    ld      (ix+8),h
1AD9    21 F95D                     ld      hl,mstbl        ;update mouse status
1ADC    CB C6                       set     msmov,(hl)      ;set the mouse update flag
1ADE    CB 76                       bit     mintrp,(hl)     ;test for interrupt mode
1AE0    CA 1AB3                     jp      z,micex1        ;bye bye if polling mode
1AE3    DD 7E 00                    ld      a,(ix)          ;user interrupt byte
1AE6    DD E1                       pop     ix              ;restore regiser
1AE8    C3 19D4                     jp      charout         ;bye bye

                            ;;      This routine adds the delta to either the x or y position and
                            ;       does minimum position check
                            ;       input   e=      mouse delta (absolute)
                            ;               zero    set if positive mouse delta
                            ;                       reset if negative mouse delta
                            ;
                            ;       output  hl=     update position
                            ;
1AEB    20 04               Mice1:  jr      nz,mice11       ;skip if delta negative
1AED    16 00                       ld      d,zero          ;set msb positive
1AEF    18 07                       jr      mice12
1AF1    16 FF               mice11: ld      d,-1            ;set msb negative
1AF3    7B                          ld      a,e             ;recover delta
1AF4    2F                          cpl                     ;2's complement
1AF5    C6 01                       add     a,1
1AF7    5F                          ld      e,a             ;put negative value back
1AF8    19                  mice12: add     hl,de           ;add delta to mouse position
1AF9    7C                          ld      a,h             ;get msb
1AFA    A7                          and     a
1AFB    F0                          ret     p               ;skip if msb is positive
```

```
                                        ld      hl,zero         ;set to minimum boundary
                                        ret
                                ;;      This routine does a maximum x or y position check
                                ;       input   hl=     x or y position
                                ;               de=     x or y max value
                                ;
                                ;       output  a =     mstbl
                                ;
                                Mice2:  ld      a,d             ;msb position test
                                        cp      h
                                        jr      c,mice21        ;skip if msb too big
                                        ld      a,e             ;lsb position test
                                        cp      l
                                        jr      nc,mice22       ;skip if lsb is not too big
                                mice21: ex      de,hl           ;force maximum limit
                                mice22: ld      a,(mstbl)       ;mouse table
                                        bit     xy,a            ;complement xy flag
                                        jr      z,mice23
                                        res     xy,a
                                        jr      mice24
                                mice23: set     xy,a
                                mice24: ld      (mstbl),a       ;update table
                                        ret

                                ;;      Jump table for keyboard translator and interrupt handler.
                                ;       Exit points and monitor adjustment points for the SIGNON
                                ;       overlay and boot overlay
                                ;
                                Rv1tbl: jp      lpkybd          ;4.01   monitor lpkybd jump table
                                        jp      mkey2
                                        jp      mkey5
                                        jp      mpnext
                                        jp      mprmt0
                                jtblsz  equ     $-rv1tbl

                                rv2tbl: jp      lpkybd          ;4.02   monitor lpkybd jump table
                                        jp      mkey2
                                        jp      mkey5
                                        jp      mpnext+romofs
                                        jp      mprmt0+romofs

                                ;;      The keyboard tables are restored to the original default values
                                ;       that are stored in rom
                                ;
                                Movtbl: push    bc
                                        push    de
                                        push    hl
                                        ld      hl,tables
                                        ld      de,tabled
                                        ld      bc,tablex
                                        ldir
                                        pop     hl
                                        pop     de
                                        pop     bc
                                        ret
```

```
796
797                                             .dephase
798                                             subttl  RAM resident (Tables)
799                                             page
```

```
                                tables:                         ;rom address
                                        .phase  tabled          ;ram address

                                        .radix  16

                                ;;      k/b unshifted table
                                ;
F800    00 1B 31 32             Tabl:   defb    00h,1bh,31h,32h,33h,34h,35h,36h         ;nul,esc,1,2,3,4,5,6
F804    33 34 35 36
F808    37 38 39 30                     defb    37h,38h,39h,30h,2dh,3dh,08h,09h         ;7,8,9,0,-,=,bs,tab
F80C    2D 3D 08 09
F810    71 77 65 72                     defb    71h,77h,65h,72h,74h,79h,75h,69h         ;q,w,e,r,t,y,u,i
F814    74 79 75 69
F818    6F 70 5B 5D                     defb    6fh,70h,5bh,5dh,0dh,0ee,61h,73h         ;o,p,[,],cr,lctrl,a,s
F81C    0D EE 61 73
F820    64 66 67 68                     defb    64h,66h,67h,68h,6ah,6bh,6ch,3bh         ;d,f,g,h,j,k,l,;
F824    6A 6B 6C 3B
F828    27 0A EC 2E                     defb    27h,0ah,0ec,2eh,7ah,78h,63h,76h         ;',lf,lshift,.,z,x,c,v
F82C    7A 78 63 76
F830    62 6E 6D 2C                     defb    62h,6eh,6dh,2ch,2eh,2fh,0ed,1eh         ;b,n,m,,,.,/,rshift,help
F834    2E 2F ED 1E
F838    EF 20 EB F1                     defb    0ef,20h,0eb,0f1,0f2,0f3,0f4,0f5         ;rctrl,sp,f1,f2,f3,f4,f5
F83C    F2 F3 F4 F5
F840    F6 F7 F8 F9                     defb    0f6,0f7,0f8,0f9,0fa,0fb,0fc,37h         ;f6,f7,f8,f9,f10,f11,f12,7
F844    FA FB FC 37
F848    38 39 2C 34                     defb    38h,39h,2ch,34h,35h,36h,0bd,31h         ;8,9,,,4,5,6,=enter,1
F84C    35 36 BD 31
F850    32 33 30 E7                     defb    32h,33h,30h,0e7,82h,84h,83h,80h         ;2,3,0,next,darr,larr,rarr,h
F854    82 84 83 80
F858    81 E6 FD 7F                     defb    81h,0e6,0fd,7fh,2bh,2dh,2ah,2fh         ;uarr,prev,acc,del,+,-,mul,d
F85C    2B 2D 2A 2F
F860    F0 18 8E 8F                     defb    0f0,18h,8eh,8fh,0a0,0a2,0a4             ;ins,can,msw1,msw2,rx1,rx2,r
F864    A0 A2 A4
                                ;;      k/b shifted
                                ;
F867    00 1B 21 40             Shtab:  defb    00h,1bh,21h,40h,23h,24h,25h,5eh         ;nul,esc,!,@,#,$,%,^
F86B    23 24 25 5E
F86F    26 2A 28 29                     defb    26h,2ah,28h,29h,5fh,2bh,08h,09h         ;&,*,(,),_,+,bs,tab
F873    5F 2B 08 09
F877    51 57 45 52                     defb    51h,57h,45h,52h,54h,59h,55h,49h         ;Q,W,E,R,T,Y,U,I
F87B    54 59 55 49
F87F    4F 50 7B 7D                     defb    4fh,50h,7bh,7dh,0dh,0ee,41h,53h         ;O,P,{,},cr,lcrtl,A,S
F883    0D EE 41 53
F887    44 46 47 48                     defb    44h,46h,47h,48h,4ah,4bh,4ch,3ah         ;D,F,G,H,J,K,L,:
F88B    4A 4B 4C 3A
F88F    22 0A EC 2E                     defb    22h,0ah,0ec,2eh,5ah,58h,43h,56h         ;",lf,lshift,.,Z,X,C,V
F893    5A 58 43 56
F897    42 4E 4D 3C                     defb    42h,4eh,4dh,3ch,3eh,3fh,0ed,1eh         ;B,N,M,<,>,?,rshift,help
F89B    3E 3F ED 1E
F89F    EF 20 EB F1                     defb    0ef,20h,0eb,0f1,0f2,0f3,0f4,0f5         ;rctrl,sp,lock,f1,f2,f3,f4,f
F8A3    F2 F3 F4 F5
F8A7    F6 F7 F8 F9                     defb    0f6,0f7,0f8,0f9,0fa,0fb,0fc,37h         ;f6,f7,f8,f9,f10,f11,f12,7
F8AB    FA FB FC 37
```

```
                                        defb    38h,39h,2ch,34h,35h,36h,0bd,31h         ;8,9,,,4,5,6,=enter,1
                                        defb    32h,33h,30h,0e7,82h,84h,83h,80h         ;2,3,0,next,darr,larr,rarr,h
                                        defb    81h,0e6,0fd,7fh,2bh,2dh,2ah,2fh         ;uarr,prev,acc,del,+,-,mul,d
                                        defb    0f0,18h,8eh,8fh,0a1,0a3,0a5             ;ins,can,msw1,msw2,rx1,rx2,r

                                ;;      k/b coded
                                ;
                                Cdtab:  defb    00h,9bh,91h,92h,93h,94h,95h,96h         ;nul,esc,1,2,3,4,5,6
                                        defb    97h,98h,99h,90h,1fh,9ah,88h,89h         ;7,8,9,0,-,=,bs,tab
                                        defb    11h,17h,05h,12h,14h,19h,15h,09h         ;q,w,e,r,t,y,u,i
                                        defb    0fh,10h,1bh,1dh,8dh,0ee,01h,13h         ;o,p,[,],cr,lctrl,a,s
                                        defb    04h,06h,07h,08h,0ah,0bh,0ch,7eh         ;d,f,g,h,j,k,l,~
                                        defb    60h,08a,0ec,0ae,1ah,18h,03h,16h         ;`,lf,lshift,,,z,s,c,v
                                        defb    02h,0eh,0dh,1ch,7ch,5ch,0ed,9eh         ;b,n,m,,,|,\,rshift,help
                                        defb    0ef,00h,0eb,0d1,0d2,0d3,0d4,0d5         ;rctrl,sp,lock,f1,f2,f3,f4,f
                                        defb    0d6,0d7,0d8,0d9,0da,0db,0dc,0b7         ;f6,f7,f8,f9,f10,f11,f12,7
                                        defb    0b8,0b9,0ac,0b4,0b5,0b6,0fe,0b1         ;8,9,,,4,5,6,=enter,1
                                        defb    0b2,0b3,0b0,0c7,32h,04h,03h,1eh         ;2,3,0,next,darr,larr,rarr,h
                                        defb    01h,0c6,0dd,0ff,0ab,0ad,0aa,0af         ;uarr,prev,acc,del,+,-,mul,d
                                        defb    0d0,0de,8eh,8fh,0c8,0c9,0ca             ;ins,can,msw1,msw2,rx1,rx2,r

                                        .radix  10

                                captab: defb    0,0,0           ;table of exceptions requiring shifting for
                                cptex:  defb    0,0,0           ;caps lock key.(3 excepts then 3 translates)
                                cptbsz  equ     ($-captab)/2    ;size of table
                                shftlck: defb   0               ;if set, locks all keys to shift table if lock set
                                tick:   defb    low hlfsec      ;lsb - repeat char speed
                                        defb    high hlfsec     ;msb
                                tock:   defb    low tenths      ;lsb
                                        defb    high tenths     ;msb
                                hlfsec  equ     500             ;0.5 second count
                                tenths  equ     63              ;16 chars/sec

                                rptbl:  defb    x'08',x'0a',x'0d',x'20'         ;bs,lf,cr,sp
                                        defb    x'2d',x'2e',x'2f'               ;-,.,/
                                        defb    x'3d',x'58',x'78',x'7f'         ;=,X,x,del
                                        defb    x'81',x'82',x'83',x'84'         ;ucur,dcur,rcur,lcur
                                rptex:  defb    x'e0',x'e0',x'e0',x'e0'         ;16 TBD repeat keys
```

```
  911     0013                          cntrp   equ     $-nptbl
  912
  913     F953    1D 2A 36 38           ctrltb: db      x'1d',x'2a',x'36',x'38',x'3a'   ;lctr,lshft,rshft,rctr,lck
  914     F957    3A
  915     F958    E0                    ctrlex: db      x'e0'                           ;19 TBD additional key stations
  916     0006                          cntctr  equ     $-ctrltb
  917     0159                          tablex  equ     $-tabl
  918
  919     F959    E0 E0                 ups:    defb    x'e0',x'e0'     ;upstroke exception key stations
  920     F95B    E0 E0                 upsx:   defb    x'e0',x'e0'     ;upstroke char or code translations
  921     0002                          upssz   equ     ($-ups)/2       ;size
  922     F95D    00                    mstbl:  defb    0               ;mouse table
  923     0007                          msflg   equ     7               ;mouse translator enabled if set
  924     0006                          mintrp  equ     6               ;user interrupt with mbyte else user polls
  925     0003                          strkup  equ     3               ;upstroke user enable flag
  926     0001                          xy      equ     1               ;set for x delta else y delta
  927     0000                          msmov   equ     0               ;mouse table contains new data
  928     F95E    0000                  msptr:  defw    0               ;user address containing the following table
  929     0160                          ktabsz  equ     $-tabl          ;size of k/b tables
  930
  931                                   ;;      Font translation table.
  932                                   ;       first the exception codes
  933                                   ;
  934     F960    23 3C 3E 40           Fontbl: defb    23h,3ch,3eh,40h,5bh,5ch,5dh,5eh         ;#,<,>,@,[,\,],^
  935     F964    5B 5C 5D 5E
  936     F968    60 7B 7C 7D                   defb    60h,7bh,7ch,7dh,7eh                     ;`,{,|,},~
  937     F96C    7E
  938
  939                                   ;;      Now their translations
  940                                   ;
  941     F96D    23 3C 3E 40                   defb    23h,3ch,3eh,40h,5bh,5ch,5dh,5eh
  942     F971    5B 5C 5D 5E
  943     F975    60 7B 7C 7D                   defb    60h,7bh,7ch,7dh,7eh
  944     F979    7E
  945     000D                          fontsz  equ     ($-fontbl)/2    ;size of font tables
  946
  947                                   ;;      Printer translation table
  948                                   ;       first the exception codes
  949                                   ;
  950     F97A    21 23 2A 2B           Prntbl: defb    21h,23h,2ah,2bh,2ch,2eh,3ch,3eh         ;!,#,*,+,,,<,>
  951     F97E    2C 2E 3C 3E
  952     F982    40 5B 5C 5D                   defb    40h,5bh,5ch,5dh,5eh,60h,7bh,7ch         ;@,[,\,],^,`,{,|
  953     F986    5E 60 7B 7C
  954     F98A    7D 7E FF FF                   defb    7dh,7eh,x'ff',x'ff',x'ff',x'ff'         ;},~,TBD,TBD,TBD,TBD
  955     F98E    FF FF
  956
  957                                   ;;      Now the translations
  958                                   ;
  959     F990    21 23 2A 2B                   defb    21h,23h,2ah,2bh,2ch,2eh,3ch,3eh
  960     F994    2C 2E 3C 3E
  961     F998    40 5B 5C 5D                   defb    40h,5bh,5ch,5dh,5eh,60h,7bh,7ch
  962     F99C    5E 60 7B 7C
  963     F9A0    7D 7E FF FF                   defb    7dh,7eh,x'ff',x'ff',x'ff',x'ff'
  964     F9A4    FF FF
  965     0016                          prntsz  equ     ($-prntbl)/2    ;size of printer table
  966
```

```
Position encoded keyboard handler       MACRO-80 3.44   09-Dec-81
RAM resident (Tables)

  967     F9A6    00                      escsq:  defb    0               ;escape sequence flag
  968                                                                     ;0=> no sequence in progress
  969                                                                     ;1b=> expecting 2nd byte
  970                                                                     ;ff=> expecting 3rd byte
  971     F9A7    00                      cmdstat:defb    00              ;comand-status byte
  972     0007                            cmd     equ     7               ;command-status flag
  973     0006                            ustrk   equ     6               ;up stroke
  974     0005                            yneg    equ     5               ;mouse y axis negative
  975     0004                            xneg    equ     4               ;mouse x axis negative
  976     0003                            mouse   equ     3               ;mouse active
  977     0002                            ctrl    equ     2               ;control key active
  978     0001                            shift   equ     1               ;shift key active
  979     0000                            lock    equ     0               ;lock key active
  980     F9A8    0000                    millcnt:defw    0               ;current millisecond count
  981     F9AA    00                      rptchar:defb    0               ;repeat character
  982     F9AB    00                      rptflg: defb    0               ;repeat flag
  983     F9AC    0000                    save:   defw    0               ;save address of the interrupt vector
  984     F9AE    00                      romram: defb    0               ;rom/ram memory bank status
  985     0007                            siderom equ     7               ;ram side
  986     01AF                            olsiz   equ     $-tabled
  987
  988                                     ;;      k/b interrupt handler for the low profile position encoded k/b.
  989                                     ;       The interrupt interrupt service routine inputs two or three
  990                                     ;       bytes from the keyboard port.  The first byte is the cmd/status
  991                                     ;       byte.  Appropriate information is saved in memory and the return
  992                                     ;       from interrupt code is invoked.  The second and third byte(mouse)
  993                                     ;       is position encoded data or mouse displacement is analyzed,
  994                                     ;       translated, and return to the existing keyboard handler if the
  995                                     ;       information is valid; otherwise it is truncated and a return
  996                                     ;       from interrupt is invoked.  All registers saved except for the
  997                                     ;       A register.
  998                                     ;
  999                                     ;       input-  keyboard input port (data complemented) kbdat
 1000                                     ;
 1001                                     ;       output- Carry flag set - Command byte or truncated character
 1002                                     ;               Carry flag reset - Translated character in A register
 1003                                     ;
 1004     F9AF    DB 1E                  Lpkybd: in      a,(kbdat)       ;read k/b port
 1005     F9B1    CD F9BE                        call    romside         ;switch to romside
 1006     F9B4    C3 1996                        jp      pekhdl          ;decode posn. enc. k/b
 1007                                                                     ;jp instead of call - interrupt stack small
 1008     F9B7    CD F9C8                lpkext: call    ramside         ;restore original memory side
 1009     F9BA    D0                             ret     nc              ;return to xerox code with char.
 1010                                                                     ;here if command byte
 1011     F9BB    C1                             pop     bc              ;throw away return address
 1012     F9BC    18 3B                          jr      rptclk2         ;return from interrupt (via xerox)
 1013
 1014                                     ;;      Romside saves the status of the current side of memory
 1015                                     ;       and switches to romside.
 1016                                     ;
 1017                                     ;       output  romram= status of syspio
 1018                                     ;
 1019     F9BE    F5                     Romside:push    af              ;save register
 1020     F9BF    DB 1C                          in      a,(syspio)      ;read ram/rom status
 1021     F9C1    32 F9AE                        ld      (romram),a      ;save it
 1022     F9C4    CB FF                          set     siderom,a       ;force rom side
```

```
1023    F9C6    18 04                           jr      rmside2
1024
1025                                    ;;      Ramside restores the ram/rom memory back to where it was
1026                                    ;
1027                                    ;       input   romram= ;status of prior ram/rom
1028                                    ;
1029    F9C8    F5                      Ramside:push    af              ;save register
1030    F9C9    3A F9AE                         ld      a,(romram)      ;recover prior ram/rom status
1031    F9CC    D3 1C                   rmside2:out     (syspio),a      ;switch it
1032    F9CE    F1                              pop     af
1033    F9CF    C9                              ret
1034
1035                                    ;;      This routine is the repeat key timer interrupt handler.  If the
1036                                    ;       the repeat flag or the count is not zero, then the return from
1037                                    ;       interrupt is invoked.  If the count is zero, then the repeat
1038                                    ;       character is passed to the ASCII keyboard handler.
1039                                    ;
1040                                    ;       input   rptflg  -repeat flag
1041                                    ;               rptchar -repeat char
1042                                    ;               millcnt -timer table
1043                                    ;
1044                                    ;       output  a       -repeat char if count is zero
1045                                    ;
1046    F9D0    ED 73 F1EC              Rptclk: ld      (savstk),sp     ;save current stack ptr
1047    F9D4    31 FF50                         ld      sp,intstk       ;temporary interrupt stack
1048    F9D7    E5                              push    hl              ;save registers
1049    F9D8    F5                              push    af
1050    F9D9    C5                              push    bc
1051    F9DA    3A F9AB                         ld      a,(rptflg)      ;repeat key flag
1052    F9DD    A7                              and     a               ;set flags
1053    F9DE    28 19                           jr      z,rptclk2       ;quit if not in repeat key mode
1054    F9E0    2A F9A8                         ld      hl,(millcnt)    ;current millisec count
1055    F9E3    7C                              ld      a,h
1056    F9E4    B5                              or      l               ;test count
1057    F9E5    28 06                           jr      z,rptclk1       ;skip if time elasped
1058    F9E7    2B                              dec     hl              ;dcr count
1059    F9E8    22 F9A8                         ld      (millcnt),hl    ;save  millisec count
1060    F9EB    18 0C                           jr      rptclk2         ;quit if not time
1061    F9ED    2A F93E                 rptclk1:ld      hl,(tock)       ;reset millisec count
1062    F9F0    22 F9A8                         ld      (millcnt),hl
1063    F9F3    3A F9AA                         ld      a,(rptchar)     ;fetch repeat char
1064    F9F6    C3 F06F                         jp      key2            ;give char to keyboard key
1065    F9F9    C3 F072                 rptclk2:jp      key5
1066
1067                                    ;;      Siout - output to channel b after translation and
1068                                    ;               escape sequence handling
1069    F9FC    F3                      Rxsioo: di                      ;char in a
1070    F9FD    CD F9BE                         call    romside         ;switch to romside
1071    FA00    CD 190C                         call    rmposend        ;does real work
1072    FA03    CD F9C8                         call    ramside         ;restore original memory side
1073    FA06    FB                              ei
1074    FA07    C9                              ret
1075    FA07                            kbramend equ    $-1             ;last location of code in ram
1076    0208                            olsiz3  equ     $-tabled        ;size of relocatable code
1077
1078                                            .dephase
```

```
                                        subttl  Overlay (signon)
                                        page
```

```
1081
1082
1083                                    ;;      signon - Announce System Ready.
1084                                    ;
1085    0552'                           rxsign:                         ;source address in rom
1086                                            .phase  tca             ;execution in transient command area
1087    FC5D    21 F091                 Signon: ld      hl,confg        ;point to configuration byte
1088    FC60    DB 1C                           in      a,(syspio)      ;check configuration
1089    FC62    CB 47                           bit     0,a
1090    FC64    28 26                           jr      z,sign3         ;if SASI interface present
1091    FC66    F3                              di
1092    FC67    3E CF                           ld      a,11001111b     ;set Pio B in Bit Mode
1093    FC69    D3 1D                           out     (sysctl),a
1094    FC6B    3E 38                           ld      a,00111000b     ;turn around d0,1,2
1095    FC6D    D3 1D                           out     (sysctl),a
1096    FC6F    3E 80                           ld      a,10000000b     ;ensure rom switched on
1097    FC71    D3 1C                           out     (syspio),a      ;drop all drive selects
1098    FC73    3E D0                           ld      a,0d0h          ;reset wd-1797-02
1099    FC75    D3 10                           out     (wd1797),a
1100    FC77    10 FE                   sign1:  djnz    sign1           ;wait 1797 not busy
1101    FC79    DB 1C                           in      a,(syspio)
1102    FC7B    CB 67                           bit     c.five,a
1103    FC7D    3E 02                           ld      a,2             ;preset 10 msec step rate
1104    FC7F    20 04                           jr      nz,sign2        ;if not 5"
1105    FC81    CB E6                           set     c.five,(hl)
1106    FC83    3E 03                           ld      a,3             ;set long step
1107    FC85    D3 10                   sign2:  out     (wd1797),a      ;restore / unload heads
1108    FC87    32 FF54                         ld      (steprt),a
1109    FC8A    18 1E                           jr      sign4
1110    FC8C    CB F6                   sign3:  set     c.sasi,(hl)     ;set Sasi card installed
1111    FC8E    21 F708                         ld      hl,Rigdpb       ;set address of rigid dpb
1112    FC91    11 F470                         ld      de,Fivdpb       ;set address of 5.25" floppy dpb
1113    FC94    01 0300                         ld      bc,Sasidl       ;set sasi driver length
1114    FC97    ED B0                           ldir                    ;Move driver down
1115    FC99    E6 02                           and     2
1116    FC9B    20 0D                           jr      nz,sign4        ;if not A/E swap
1117    FC9D    21 F361                         ld      hl,Seltab+1
1118    FCA0    06 08                           ld      b,8
1119    FCA2    7E                      sign3a: ld      a,(hl)
1120    FCA3    EE 04                           xor     4
1121    FCA5    77                              ld      (hl),a
1122    FCA6    23                              inc     hl
1123    FCA7    23                              inc     hl
1124    FCA8    10 F8                           djnz    sign3a
1125    FCAA    21 034A'                sign4:  ld      hl,tables       ;move rx resident code to ram
1126    FCAD    11 F800                         ld      de,tabled
1127    FCB0    01 0208                         ld      bc,olsiz3
1128    FCB3    ED B0                           ldir                    ;move on top of GETHLP
1129    FCB5    21 FA08                         ld      hl,kbramend+1   ;next available ram loc
1130    FCB8    22 FF3C                         ld      (availb),hl     ;tell the world
1131    FCBB    21 0000                         ld      hl,0
1132    FCBE    CD F03C                         call    config          ;get monitor configuration
1133    FCC1    7C                              ld      a,h             ;monitor level
1134    FCC2    21 10D0                         ld      hl,xrsign+sigoff-romofs ;assume 4.01 level location
1135    FCC5    FE 01                           cp      rev1
```

```
1136    FCC7    28 03                           jr      z,sign7         ;skip if 4.01
1137    FCC9    21 10D8                         ld      hl,xrsign+sigoff ;4.02+ level location
1138    FCCC    11 FCE6                 sign7:  ld      de,sign6        ;put it in our signon message
1139    FCCF    01 0004                         ld      bc,4
1140    FCD2    ED B0                           ldir
1141    FCD4    CD F293                         call    crtoff          ;disable rom bank
1142    FCD7    CD F075                         call    pnext
1143    FCDA    1A                              defb    clrs            ;clear screen
1144    FCDB    1B 38                           defb    esc,'8'         ;set low light as default mode
1145    FCDD    38 32 30 2D                     defm    '820-II v '
1146    FCE1    49 49 20 76
1147    FCE5    20
1148    FCE6    00 00 00 00             sign6:  defb    0,0,0,0         ;********** space for the XR rev value
1149    FCEA    20 1F 1C 20                     defm    ' ',31,28,' 1983 Xerox Corp'
1150    FCEE    31 39 38 33
1151    FCF2    20 58 65 72
1152    FCF6    6F 78 20 43
1153    FCFA    6F 72 70
1154    FCFD    20 28 76                        defm    ' (v'
1155    FD00    30 31 33                        defm    ver/100+'0',(ver mod 100)/10+'0',(ver mod 10)+'0'
1156    FD03    29 0D 0A                        defb    ')',cr,lf
1157    FD06    0A                              defb    lf
1158    FD07    4C 20 2D 20                     defm    'L - Load System'
1159    FD0B    4C 6F 61 64
1160    FD0F    20 53 79 73
1161    FD13    74 65 6D
1162    FD16    0D 0A                           defb    cr,lf
1163
1164                                            if      o.term
1165    FD18    48 20 2D 20                     defm    'H - Host Terminal'
1166    FD1C    48 6F 73 74
1167    FD20    20 54 65 72
1168    FD24    6D 69 6E 61
1169    FD28    6C
1170    FD29    0D 0A                           defb    cr,lf
1171                                            endif
1172
1173                                            if      o.term
1174    FD2B    54 20 2D 20                     defb    'T - Typewriter'
1175    FD2F    54 79 70 65
1176    FD33    77 72 69 74
1177    FD37    65 72
1178    FD39    0D 0A                           defb    cr,lf
1179                                            endif
1180
1181    FD3B    07 04                           defb    7,eot
1182
1183    FD3D    CD F006                 devour: call    const
1184    FD40    CA F003                         jp      z,warm          ;go enter monitor
1185    FD43    CD F009                         call    conin
1186    FD46    18 F5                           jr      devour
1187    00EB                            rxsigl  equ     $-signon
1188
1189                                            .dephase
1190                                            subttl  Overlay (boot)
1191                                            page
```

```
1192
1193
1194  063D'                          rxboot:                 ;rom source address
1195                                         .phase  tca-romofs      ;execution address o/l area
1196  FC55                                   ds      romofs,0        ;4.01 overlay start address
1197  FC5D  21 FF5D                          ld      hl,linbuf+1     ;4.02 overlay start address
1198  FC60  7E                       boot1:  ld      a,(hl)          ;scan command line
1199  FC61  2C                               inc     l
1200  FC62  D6 0D                            sub     cr
1201  FC64  28 0B                            jr      z,boot2         ;if no parameter, boot from A:
1202  FC66  FE 13                            cp      ' '-cr
1203  FC68  28 F6                            jr      z,boot1         ;skip leading blanks
1204  FC6A  D6 34                            sub     'A'-cr
1205  FC6C  D8                               ret     c               ;if invalid drive
1206  FC6D  FE 10                            cp      16
1207  FC6F  3F                               ccf
1208  FC70  D8                               ret     c               ;if bad drive
1209  FC71  4F                       boot2:  ld      c,a             ;set boot drive selected
1210  FC72  C6 41                            add     a,'A'
1211  FC74  32 FD72                          ld      (bootd),a       ;set up error message
1212  FC77  2E 00                            ld      l,0             ;set A:
1213  FC79  C5                               push    bc
1214  FC7A  E5                               push    hl
1215  FC7B  CD FD89                          call    swap            ;switch boot drive with A:
1216  FC7E  21 FD6E                          ld      hl,booter       ;set boot error return
1217  FC81  E5                               push    hl
1218  FC82  0E 00                            ld      c,0             ;then boot from A:
1219  FC84  CD F01B                          call    select
1220  FC87  C0                               ret     nz              ;if drive not configured or density error
1221  FC88  3E FF                            ld      a,-1
1222  FC8A  12                               ld      (de),a
1223  FC8B  11 000A                          ld      de,10           ;set dpb address offset within dph
1224  FC8E  19                               add     hl,de
1225  FC8F  5E                               ld      e,(hl)          ;set dpb address
1226  FC90  23                               inc     hl
1227  FC91  56                               ld      d,(hl)
1228  FC92  CD F01E                          call    home
1229  FC95  1A                               ld      a,(de)          ;get low sectors per track
1230  FC96  32 FD6D                          ld      (boots),a       ;inform boot loader
1231  FC99  B7                               or      a
1232  FC9A  20 20                            jr      nz,boot3        ;if not rigid
1233  FC9C  21 000D                          ld      hl,13           ;set reserved track offset within dpb
1234  FC9F  19                               add     hl,de
1235  FCA0  4E                               ld      c,(hl)          ;get reserved tracks
1236  FCA1  23                               inc     hl
1237  FCA2  46                               ld      b,(hl)
1238  FCA3  0B                               dec     bc              ;point behind directory
1239  FCA4  ED 43 FA11                       ld      (phytrk),bc     ;do implied seek
1240                                                                 ;here for rigid
1241  FCA8  0E 1D                            ld      c,rtabl         ;first rigid sector
1242  FCAA  21 ED80                          ld      hl,bootbf       ;buffer
1243  FCAD  CD F024                          call    read            ;layout and k/b tables
1244  FCB0  C0                               ret     nz
1245  FCB1  0E 1E                            ld      c,rtabl+lpkofs  ;2nd rigid sector
1246  FCB3  21 EE80                          ld      hl,bootbf+x'100' ;buffer
```

```
Position encoded keyboard handler       MACRO-80 3.44   09-Dec-81
Overlay (boot)

 1247    FCB6    CD F024                        call    read            ;layout and k/b tables
 1248    FCB9    C0                             ret     nz
 1249    FCBA    18 20                          jr      rxb01
 1250    FCBC                           boot3:                          ;here for floppy
 1251    FCBC    FE 1B                          cp      27              ;double density?
 1252    FCBE    DA FD52                        jp      c,boot4         ;no - exit
 1253    FCC1    0E 04                          ld      c,ftab1         ;first floppy sector
 1254    FCC3    21 ED80                        ld      hl,bootbf       ;buffer
 1255    FCC6    CD F024                        call    read            ;layout table and half of k/b
 1256    FCC9    C0                             ret     nz
 1257    FCCA    0E 05                          ld      c,ftab2         ;second floppy sector
 1258    FCCC    21 EE00                        ld      hl,bootbf+128
 1259    FCCF    CD F024                        call    read            ;midle third of k/b tables
 1260    FCD2    C0                             ret     nz
 1261    FCD3    0E 06                          ld      c,ftab2+lpkofs  ;third floppy sector
 1262    FCD5    21 EE80                        ld      hl,bootbf+128+128
 1263    FCD8    CD F024                        call    read            ;last third of k/b tables
 1264    FCDB    C0                             ret     nz
 1265    FCDC                           rxb01:                          ;check tables are present
 1266    FCDC    3A ED80                        ld      a,(bootbf+lang) ;language no. set?
 1267    FCDF    FE E5                          cp      0e5h
 1268    FCE1    28 6F                          jr      z,boot4         ;no - exit
 1269    FCE3    3A ED81                        ld      a,(bootbf+kbrd) ;k/b tables present?
 1270    FCE6    FE 6B                          cp      'k'
 1271    FCE8    20 68                          jr      nz,boot4        ;no - exit
 1272    FCEA    3A ED82                        ld      a,(bootbf+font) ;font tables present?
 1273    FCED    FE 66                          cp      'f'
 1274    FCEF    20 61                          jr      nz,boot4        ;no - exit
 1275    FCF1    3A ED83                1       ld      a,(bootbf+prnt) ;printer tables prsent?
 1276    FCF4    FE 70                          cp      'p'
 1277    FCF6    20 5A                          jr      nz,boot4        ;no - exit
 1278    FCF8    21 ED84                        ld      hl,bootbf+kbrdtb ;move in k/b tables
 1279    FCFB    11 F800                        ld      de,tabled
 1280    FCFE    01 0160                        ld      bc,ktabsz
 1281    FD01    ED B0                          ldir
 1282    FD03    0E 1F                          ld      c,rtab2+lpkofs  ;3rd rigid sector
 1283    FD05    3A FD6D                        ld      a,(boots)       ;rigid or floppy?
 1284    FD08    B7                             or      a
 1285    FD09    28 02                          jr      z,boot5         ;rigid
 1286    FD0B    0E 07                          ld      c,ftab3+lpkofs  ;floppy - 4th sector
 1287    FD0D    21 ED80                boot5:  ld      hl,bootbf       ;buffer
 1288    FD10    CD F024                        call    read            ;font and print tables
 1289    FD13    C0                             ret     nz
 1290    FD14    3A ED80                        ld      a,(bootbf)      ;configured?
 1291    FD17    FE E5                          cp      0e5h
 1292    FD19    28 37                          jr      z,boot4         ;no -exit
 1293    FD1B    21 ED80                        ld      hl,bootbf       ;move font & print tables in
 1294    FD1E    11 F960                        ld      de,fontbl
 1295    FD21    01 0046                        ld      bc,fontsz*2+prntsz*2
 1296    FD24    ED B0                          ldir
 1297
 1298                                   ;;      alter SIOOUT
 1299                                   ;
 1300    FD26    DD 2A F019                     ld      ix,(monitr+sioff);sioout address
 1301    FD2A    DD 36 00 C3                    ld      (ix),0c3h       ;jump instruction
 1302    FD2E    DD 36 01 FC                    ld      (ix+1),low rxsioo
```

```
1303    FD32    DD 36 02 F9                     ld      (ix+2),high rxsioo
1304
1305                                    ;       alter crt driver
1306                                    ;
1307    FD36    DD 2A F010                      ld      ix,(monitr+fcrtof)      ;address of fast crt handler
1308    FD3A    DD 36 12 85                     ld      (ix+crtcall),low rxcrt
1309    FD3E    DD 36 13 18                     ld      (ix+crtcall+1),high rxcrt
1310    FD42    DD 21 F22F                      ld      ix,sprnt1       ;address of screenprint patch
1311    FD46    DD 36 00 C3                     ld      (ix),0c3h       ;jump instruction
1312    FD4A    DD 36 01 D6                     ld      (ix+1),low scrprt
1313    FD4E    DD 36 02 18                     ld      (ix+2),high scrprt
1314    FD52                            boot4:                          ;here to exit
1315    FD52    C1                              pop     bc              ;throw away booter return
1316    FD53    C1                              pop     bc              ;get disk swap parameters
1317    FD54    E1                              pop     hl
1318    FD55    CD FD89                         call    swap            ;swap them back for xerox boot
1319    FD58    C1                              pop     bc              ;throw away return address
1320    FD59    21 0000                         ld      hl,0
1321    FD5C    CD F03C                         call    config          ;get monitor configuration
1322    FD5F    7C                              ld      a,h
1323    FD60    21 1188                         ld      hl,xrboot-romofs ;assumed 4.01 monitor boot overlay address
1324    FD63    FE 01                           cp      rev1
1325    FD65    28 03                           jr      z,boot6         ;skip if 4.01
1326    FD67    21 11C0                         ld      hl,xrboot       ;address of 4.02+ monitor boot overlay
1327    FD6A    C3 F078                 boot6:  jp      prboff          ;enter xerox code to execute boot
1328    FD6D    00                      boots:  defb    0               ;workbyte to save disk type
1329
1330                                    ;;      Booter - Boot Error Processor.
1331                                    ;
1332    FD6E    CD F075                 Booter: call    pnext
1333    FD71    07                              defb    7
1334    FD72    64 3A 54 61             bootd:  defm    'd:Tables Load error.'
1335    FD76    62 6C 65 73
1336    FD7A    20 4C 6F 61
1337    FD7E    64 20 65 72
1338    FD82    72 6F 72 2E
1339    FD86    04                              defb    eot
1340    FD87    C1                              pop     bc              ;switch drives back
1341    FD88    E1                              pop     hl
1342
1343                                    ;;      Swap - swap logical drives.
1344                                    ;
1345                                    ;       Entry:  C = first  drive index, 0-15
1346                                    ;               L = second drive index, 0-15
1347                                    ;
1348    FD89    06 00                   Swap:   ld      b,0             ;clear upper indices
1349    FD8B    60                              ld      h,b
1350    FD8C    11 F360                         ld      de,seltab       ;set select table address
1351    FD8F    29                              add     hl,hl
1352    FD90    19                              add     hl,de
1353    FD91    EB                              ex      de,hl           ;set second address to DE, get seltab to HL
1354    FD92    09                              add     hl,bc
1355    FD93    09                              add     hl,bc           ;set first address to HL
1356    FD94    06 02                           ld      b,2
1357    FD96    4E                      swap1:  ld      c,(hl)          ;swap two bytes
1358    FD97    1A                              ld      a,(de)
```

```
Position encoded keyboard handler       MACRO-80 3.44   09-Dec-81
Overlay (boot)

 1359    FD98    77                             ld      (hl),a
 1360    FD99    79                             ld      a,c
 1361    FD9A    12                             ld      (de),a
 1362    FD9B    23                             inc     hl
 1363    FD9C    13                             inc     de
 1364    FD9D    10 F7                          djnz    swap1           ;if swap not complete
 1365    FD9F    C9                             ret
 1366                                           .dephase
 1367
 1368    0788'                           romtop:
 1369    0788'                                  defs    (romsiz-x'24')-(romtop-start),-1
 1370
 1371                                    ;;     Drctry is a table containing the RAM addresses of the keyboard
 1372                                    ;      tables.  This table is located on ROM side of memory.  The
 1373                                    ;      ROM address must be offset by x'1800' since resides in
 1374                                    ;      the fourth 2kx8 ROM slot.  This directory is helpful if future
 1375                                    ;      release require the RAM tables to reside in a different RAM
 1376                                    ;      location
 1377                                    ;
 1378    07DC'   F97A                    Drctry: defw    prntbl          ;print exception table
 1379    07DE'   F960                            defw    fontbl          ;font exception table
 1380    07E0'   F95D                            defw    mstbl           ;mouse table
 1381    07E2'   F959                            defw    ups             ;upstroke table
 1382    07E4'   F958                            defw    ctrlex          ;function key inhibit expansion table
 1383    07E6'   F953                            defw    ctrltb          ;function key inhibit table
 1384    07E8'   F94F                            defw    rptex           ;repeat key expansion table
 1385    07EA'   F940                            defw    rptbl           ;repeat key table
 1386    07EC'   F93C                            defw    tick            ;repeat speed table
 1387    07EE'   F93B                            defw    shftlck         ;shift lock flag
 1388    07F0'   F938                            defw    cptex           ;alpha lock expansion table
 1389    07F2'   F935                            defw    captab          ;alpha lock table
 1390    07F4'   F8CE                            defw    cdtab           ;code + table
 1391    07F6'   F867                            defw    shtab           ;shift table
 1392    07F8'   F800                            defw    tabl            ;unshifted table
 1393
 1394    07FA'   0D                              defb    ver             ;revision level
 1395    07FB'   00 FF                   lpid:   defb    x'00',x'ff'     ;low profile kybd id
 1396
 1397                                    ;;     define checkword to let xerox know
 1398                                    ;      that we are present
 1399                                    ;
 1400    07FD'   AA 55                           defb    0aah,55h        ;id
 1401    07FF'   00                              defb    0               ;space for checksum
 1402
 1403                                           Subttl  Symbol Table
 1404                                           end
```

Position encoded keyboard handler MACRO-80 3.44 09-Dec-81
Symbol Table

Macros:

Symbols:

1A30	ALPHEXC	1A23	ALPHTST	FFB3	ATTRIB
FF3C	AVAILB	0002	BOFF1	0018	BOFF2
FC60	BOOT1	FC71	BOOT2	FCBC	BOOT3
FD52	BOOT4	FD0D	BOOT5	FD6A	BOOT6
ED80	BOOTBF	FD72	BOOTD	FD6E	BOOTER
FD6D	BOOTS	0004	C.FIVE	0006	C.SASI
F935	CAPTAB	F8CE	CDTAB	19D4	CHAROUT
FFB4	CHRSAV	001A	CLRS	0007	CMD
19A1	CMDB	1A1F	CMDB1	F9A7	CMDSTAT
0006	CNFBYTE	003C	CNFGOFF	0006	CNTCTR
0013	CNTRP	F091	CONFG	F03C	CONFIG
F009	CONIN	F006	CONST	0003	CPTBSZ
F938	CPTEX	000D	CR	0012	CRTCALL
0182	CRTD1	0196	CRTD2	F293	CRTOFF
0019	CTC1	FF10	CTCVEC	0002	CTRL
F958	CTRLEX	F953	CTRLTB	19DB	CTRTST
FFAC	CURSOR	FD3D	DEVOUR	07DC	DRCTRY
0081	ENCNTR	0004	EOT	001B	ESC
F9A6	ESCSQ	0007	ESCTSZ	0010	FCRTOF
F470	FIVDPB	18CF	FNTRAN	18BB	FON1
18CA	FON2	18A6	FONCHK	0002	FONT
F960	FONTBL	000D	FONTSZ	0004	FTAB1
0005	FTAB2	0006	FTAB3	F319	GOLD
01DD	GRPAD	01F4	HLFSEC	F01E	HOME
F066	IDLE	FF50	INTSTK	000F	JTBLSZ
001E	KBDAT	0008	KBLP	000A	KBOFF
FA07	KBRAMEND	0001	KBRD	0004	KBRDTB
FF1A	KBVEC	F06F	KEY2	F072	KEY5
0160	KTABSZ	F06C	KYBDLP	19B5	KYPOS
0000	LANG	FFB2	LEADIN	000A	LF
FF5C	LINBUF	0000	LOCK	0061	LOWER
07FB	LPID	F9B7	LPKEXT	0001	LPKOFS
F9AF	LPKYBD	F0E3	MASK	1A7D	MICE
1AEB	MICE1	1AF1	MICE11	1AF8	MICE12
1B00	MICE2	1B08	MICE21	1B09	MICE22
1B14	MICE23	1B16	MICE24	1AB3	MICEX1
1AB8	MICEY	F9A8	MILLCNT	0006	MINTRP
F167	MKEY2	F18F	MKEY5	F06C	MNTREX
F000	MONITR	0003	MOUSE	1B38	MOVTBL
FC3D	MPNEXT	FA95	MPRMT0	0007	MSFLG
0000	MSMOV	F95E	MSPTR	F95D	MSTBL
19A0	NOCHAR	187E	NOLOAD	0001	O.TERM
01AF	OLSIZ	0208	OLSIZ3	19B1	PEKEX
1996	PEKHDL	19AD	PEKNOC	19B0	PEKNOC1
19A0	PEKNOC2	198F	PESCTB	FA11	PHYTRK
F075	PNEXT	1986	POESC	1927	POS01
1939	POS02	1939	POS03	193D	POS04
1947	POS05	1954	POS06	1955	POS07
1914	POSEND	1959	POSOUT	1966	POTRAN
F078	PRBOFF	0003	PRNT	F97A	PRNTBL
0016	PRNTSZ	FA62	PROMPT	F339	PRVATT
1983	PTR01	F9C8	RAMSIDE	F024	READ

Position encoded keyboard handler MACRO-80 3.44 09-Dec-81
Symbol Table

0000	REV0	0001	REV1	0064	REV50
F708	RIGDPB	190C	RMPOSEND	F9CC	RMSIDE2
0008	ROMOFS	F9AE	ROMRAM	F9BE	ROMSIDE
0800	ROMSIZ	0788'	ROMTOP	F940	RPTBL
F9AA	RPTCHAR	F9DC	RPTCLK	F9ED	RPTCLK1
F9F9	RPTCLK2	F94F	RPTEX	F9AB	RPTFLG
1A3D	RPTST	001C	RTAB1	001E	RTAB2
1B1A	RV1TBL	1B29	RV2TBL	1800	RX1984
FCDC	RXB01	063D'	RXBOOT	1885	RXCRT
00EB	RXSIGL	0552'	RXSIGN	F9FC	RXSIO0
0300	SASIDL	F9AC	SAVE	F1EC	SAVSTK
18EC	SCR01	18F2	SCR02	1804	SCR03
18D6	SCRPRT	F01B	SELECT	F360	SELTAB
00FF	SETFLG	F936	SHFTLCK	0001	SHIFT
F867	SHTAB	0007	SIDEROM	FC77	SIGN1
FC85	SIGN2	FC8C	SIGN3	FCA2	SIGN3A
FCAA	SIGN4	FCE6	SIGN6	FCCC	SIGN7
FC5D	SIGNON	0060	SIGOFF	0005	SIODPB
0019	SIOFF	F03F	SIORDY	195A	SIOX1
185C	SOOUT	F20E	SPACT	F22F	SPRNT1
F232	SPRNT2	0000'	START	0001	STCNTR
FF54	STEPRT	1A66	STPCTC1	0003	STRKUP
FD89	SWAP	FD96	SWAP1	001D	SYSCTL
001C	SYSPIO	FF18	SYSVEC	F800	TABL
F800	TABLED	034A'	TABLES	0159	TABLEX
19FF	TBLSEL	181E	TBXFER	FC5D	TCA
003F	TENTHS	F93C	TICK	F93E	TOCK
0020	UPASCII	007B	UPPER	F959	UPS
0002	UPSS2	19E6	UPSTRK	F95B	UPSX
0006	USTRK	000D	VER	F003	WARM
0010	WD1797	0004	XNEG	11C0	XRBOOT
1078	XRSIGN	0001	XY	0005	YNEG
0000	ZERO				

No Fatal error(s)

UPSX	920#					
USTRK	480	973#				
VER	28#	1155	1155	1155	1394	
WARM	53#	1184				
WD1797	108#	1099	1107			
XNEG	689	975#				
XRBOOT	71#	1323	1326			
XRSIGN	69#	1134	1137			
XY	469	684	757	759	761	926#
YNEG	703	974#				
ZERO	127#	459	729	740		

Notes

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

V

W

X

Z