REV: 1 DATE: 3-22-82

PURPOSE:

1) Changes Requested By Marketing

2) Engineering Changes Found To Be Needed During Development

QUALITY ASSURANCE N/A.

PUBLICATIONS:

PRODUCT TEST: 4/2/82

SYSTEMS SOFTWARE: 4/2/82

MARKETING: 4/2/82

REV: 2 DATE: 9-20-82

PURPOSE:

Further Changes Found To Be Needed During Development And Unit Testing.

QUALITY ASSURANCE 9-20

PUBLICATIONS: 9/20/82

PRODUCT TEST: 9-20-82

SYSTEMS SOFTWARE: 9/20/82

MARKETING: 9/20/82

REV: _____ DATE: _____

PURPOSE:

QUALITY ASSURANCE _____

PUBLICATIONS: _____

PRODUCT TEST: _____

SYSTEMS SOFTWARE: _____

MARKETING: _____

REV: _____ DATE: _____

PURPOSE:

QUALITY ASSURANCE _____

PUBLICATIONS: _____

PRODUCT TEST: _____

SYSTEMS SOFTWARE: _____

MARKETING: _____

REV: _____ DATE: _____

PURPOSE:

QUALITY ASSURANCE _____

PUBLICATIONS: _____

PRODUCT TEST: _____

SYSTEMS SOFTWARE: _____

MARKETING: _____

CHAPTER 6

Z800 OPERATING SYSTEM

EXTERNAL REFERENCE SPECIFICATION

HARRY B. STEWART

10-28-81

ATARI COMPANY

PRIVATE TRADE SECRETS ENCLOSE

CONTROLLED VERSION 1
12-4-81

APPROVED BY:

[signature]	12/8/81
Manager, System Software Development	Date
[signature]	12/18/81
Manager, Product Test	Date
[signature]	12/18/81
Manager, Quality Assurance	Date
[signature]	12/11/81
Supervisor, Technical Publications	Date
[signature]	12/8/81
Product Manager, Marketing	Date

REVISION B (12-4-81)

SWEET16 OPERATING SYSTEM
EXTERNAL REFERENCE SPECIFICATION

TABLE OF CONTENTS

1.0 Purpose
    1.1 Introduction and product description
    1.2 Consumer profile
    1.3 Interface with other products
    1.4 Family of products
    1.5 Distribution

2.0 Applicable documents

3.0 Requirements
    3.1 Interfaces
        3.1.1 Physical
        3.1.2 Logical
        3.1.3 Man/machine
    3.2 Functional description
        3.2.1 Deleted items
        3.2.2 Added items
        3.2.3 Enhancements
    3.3 Performance requirements
    3.4 Design requirements
    3.5 Packaging requirements
    3.6 Special requirements
        3.6.1 Serial bus peripheral compatibility
        3.6.2 Application software compatibility

4.0 Applicable standards

Supplement 1 -- SWEET16 self test
Supplement 2 -- SWEET16 relocating loader
Supplement 3 -- SWEET16 power-on device polling
Supplement 4 -- (Not applicable)
Supplement 5 -- SWEET16 international character set

1.0 Purpose

This document provides an external reference specification for the operating system for the Atari SWEET16 home computer.

1.1 Introduction and product description

The SWEET16 is an enhanced version of the Atari 800 Home Computer System. It is reduced in size, lower in price (both for the base unit and the peripheral devices), and upward compatible with the Atari 800 HCS. The SWEET16 and its operating system are designed for ease, simplicity, and friendliness of setup and use, while offering both the power and sophistication of a home computer. The SWEET16 is designed to encourage interest and excitement.

1.2 Consumer profile

The SWEET16's market is the home. The target consumer will buy the SWEET16 primarily for personal reasons and not for professional or business purposes. Those personal reasons may include play, curiosity, adventure, education, status and the hope for increased personal power. As home telecommunications applications become commercially available, convenience will also become a factor.

The average consumer will have had little or no formal exposure to computers or programming and may feel some mixture of pride, excitement and fear at becoming a participant in the home computer revolution.

1.3 Interface with other products

The SWEET16 operating system will interface with all of the Atari A400/800 game controllers and serial bus peripherals. The SWEET16 operating system will execute software written for the A400/800 when configured with corresponding compatible hardware.

1.4 Family of products

The SWEET16 is the bottom end of a family of two computers, one containing 16 Kbytes of RAM and the other containing 64 Kbytes of RAM. A single operating system will be developed to handle both products.

A family of SWEET16 serial bus peripherals are planned to be developed; these will be supported by the operating system.

A series of communications modules are also planned to be developed, and these will be supported by the operating system.

1.5 Distribution

Distribution of the operating system will be via two 8 Kbyte ROMs which will be installed in every base unit at the time of manufacture.

2.0 Applicable documents

Z800 OPERATING SYSTEM FIRMWARE FUNCTIONAL REQUIRENENTS SPECIFICATION, Rev B, October 7, 1981.

Z800 PRODUCT SPECIFICATION, Revision 1, August 26, 1981.

Z800 OPERATING SYSTEM -- CHANGES FROM THE A400/800 REV B O.S., October 15, 1981.

EXTERNAL REFERENCE SPECIFICATION FOR THE APPLICATION AND OPERATING SYSTEM INTERFACES TC THE RACAL-VADIC Z800 DIRECT-CONNECT MODEM MODULE, Revision B, November 23, 1981.

MODEM MODULE INTERFACE CONCEPTUAL DESCRIPTION, September 17, 1981.

ATARI HOME COMPUTER SERIAL INPUT/OUTPUT PORT USER'S HANDBOOK Part I & Part II.

RESEARCH REPORT ON Z800/A800 COMPATIBILITY ISSUES PART 1, Revision A, August 26, 1981.

RESEARCH REPORT ON Z800/A800 COMPATIBILITY ISSUES PART 2, Revision A, September 11, 1981.

Z800 MEETING MINUTES, through February 10, 1982.

ATARI PERSONAL COMPUTER SYSTEM OPERATING SYSTEM USER'S MANUAL.

ATARI PERSONAL COMPUTER SYSTEM HARDWARE MANUAL.

DE RE ATARI.

3.0 Requirements

3.1 Interfaces

The interface between the SWEET16 and the outside world are described in the subsections that follow.

3.1.1 Physical

The SWEET16 operating system requires, as a minimum, a SWEET16 base unit. The operating system shall communicate with the outside world via the console keyboard, a TV (audio and video), the game controller ports, and the serial bus.

3.1.2 Logical

The SWEET16 operating system shall support all Atari Home Computer System language processors, application programs and games.

3.1.3 Man/machine

The SWEET16 operating system's primary function is to facilitate communication between the user and a program, rather than to communicate with the user directly. The operating system shall provide a high level I/O subsystem and a set of data base variables by which a program may monitor and control the system's performance.

3.2 Functional description

The SWEET16 operating system shall support all functions of the A400/A800 Revision B operating system, with the exceptions noted in this functional description. This section describes the differences between the SWEET16 operating system and Revision B of the A400/800 operating system. The A400/800 operating system is defined by the ATARI PERSONAL COMPUTER SYSTEM OPERATING SYSTEM USER'S MANUAL.

The material that follows is divided into three sections: functions that are to be deleted from the Revision B A400/800 operating system, functions that are to be added, and functions that are to be modified.

3.2.1 Deleted items

The SWEET16 operating system shall support all functions of the A400/800 Revision B operating system, with the following

exceptions:

Game controller ports 3 and 4 -- The game controller port 3 and 4 connectors do not exist in the SWEET16. These ports shall not be read by the operating system. The data base variables associated with port 3 shall be set to the values read for port 1, and the data base variables associated with port 4 shall be set to the values read for port 2.

Game cartridge B -- Game cartridge B has been physically and logically eliminated from the SWEET16. No check shall be made for its presence.

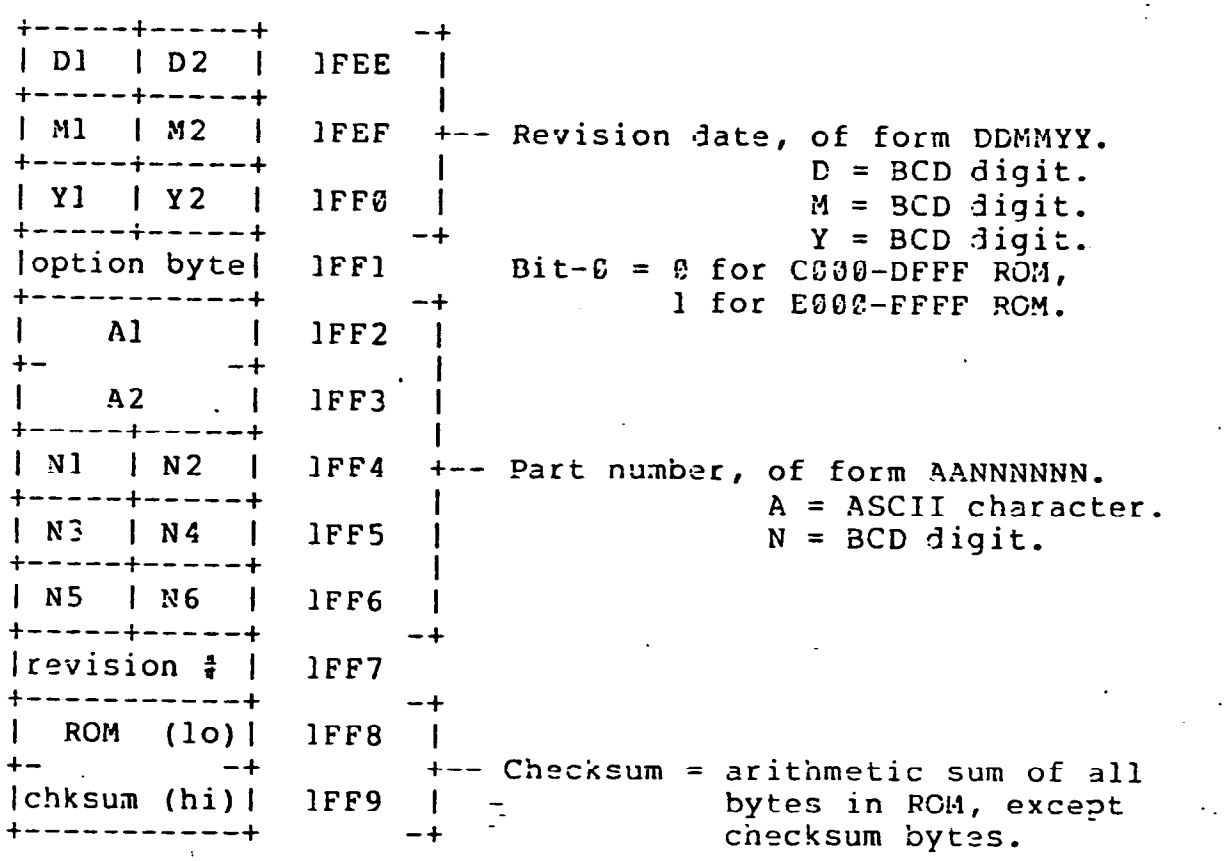
3.2.2 Added items

In addition to those functions supported by the A400/800 Revision B operating system, the SWEET16 shall support the following functions:

NTSC/PAL option -- All operating system provisions for dealing with the timing differences between the NTSC (60 Hz) and the PAL (50 Hz) formats shall be made in a single ROM set. A flag bit is provided in the GTIA chip to identify which format the chip supports. The operating system shall be designed to make certain internal timing values functions of the flag state, the end result being that the associated external timings shall remain constant for NTSC and PAL systems. The timing values relate to the handling of the Model 410 cassette player (and related devices) and the console key auto-repeat rate, as shown below:

Write IRG (long)	3.0 seconds.
Read IRG delay (long)	2.0 seconds.
Write IRG (short)	0.25 seconds.
Read IRG delay (short)	0.16 seconds.
Write file leader	19.2 seconds.
Read leader delay	9.6 seconds.
Beep cue duration	0.5 seconds.
Beep cue separation	0.16 seconds.
Auto-repeat initial delay	0.80 seconds.
Auto-repeat rate	10.0 characters / second.

ROM I.D. and checksum -- Each of the two SWEET16 operating system ROMS shall contain a 12 byte information block in ROM relative locations 1FEE-1FF9. The format of the block shall be as shown below:



Screen DMA control -- The SWEET16 provides explicit control over ANTIC DMA enable/disable. Whenever the system is first powered on or whenever any keyboard key other than CTRL-F2 is pressed, the operating system shall enable the screen DMA. Whenever the CTRL-F2 key combination is pressed, the operating system shall disable the screen DMA.

HELP key -- Whenever the HELP key is pressed, the operating system keyboard interrupt service routine shall set a flag in the operating system data base, without producing an ATASCII code for the key stroke. It will be the responsibility of the application program and/or language processor to recognize and then to clear the flag. The name of this variable is HELPFG and its location in the operating system data base shall be 02DC. The valid flag values shall be as shown below:

$00 = flag cleared.
$11 = HELP key pressed.
$51 = SHIFT-HELP key combination pressed.
$91 = CTRL-HELP key combination pressed.

See the descriptions for the power-on display and the self test functions for additional operating system responses to the HELP key.

Function key decoding -- Whenever one of the four new function keys is pressed, the keyboard interrupt service routine shall store the key code as per any of the standard data keys. The Keyboard Handler shall convert the four unique codes to ATASCII based upon a conversion table. The Screen Editor shall process the codes whenever it is reading from the keyboard.

The application program may override the default table provided by the operating system. The default (power-on) code conversions shall be as shown below (values are in hexadecimal):

KEY (Fn)	unmodified-Fn	SHIFT-Fn	CTRL-Fn	CTRL-SHIFT-Fn
F1	1C (cursor up)	Note 1	Note 5	ignored
F2	1D (" down)	Note 2	Note 6	ignored
F3	1E (" left)	Note 3	Note 7	ignored
F4	1F (" right)	Note 4	Note 8	ignored

Note 1: SHIFT-F1 shall cause the cursor to move to the home position of the screen, as its default assignment; this assignment produces ATASCII code $1C.

Note 2: SHIFT-F2 shall cause the cursor to move to the lower left corner of the screen, as its default assignment; this assignment produces ATASCII code $1D.

Note 3: SHIFT-F3 shall cause the cursor to move to the left margin of the current physical line on which it resides, as its default assignment; this assignment produces ATASCII code $1E.

Note 4: SHIFT-F4 shall cause the cursor to move to the right margin of the current physical line on which it resides, as its default assignment; this assignment produces ATASCII code $1F.

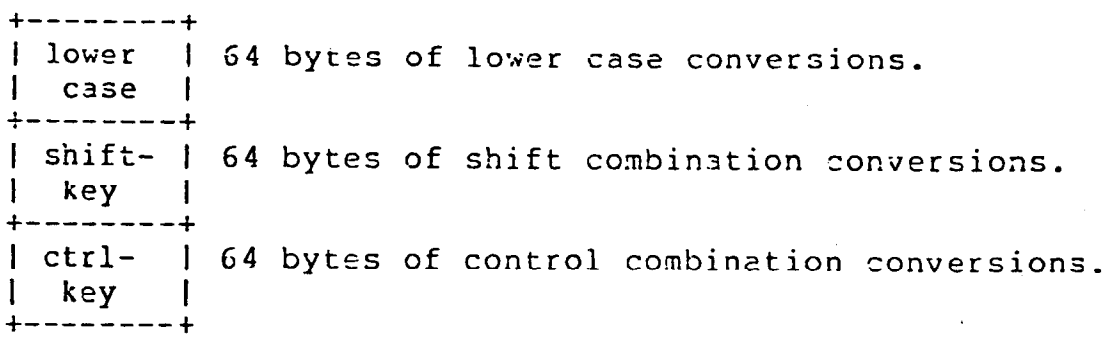
Note 5: CTRL-F1 shall control the keyboard enable/disable function; this assignment produces no ATASCII code. This key combination is not reassignable.

Note 6: CTRL-F2 shall disable the screen DMA; this assignment produces no ATASCII code. This key combination is not reassignable.

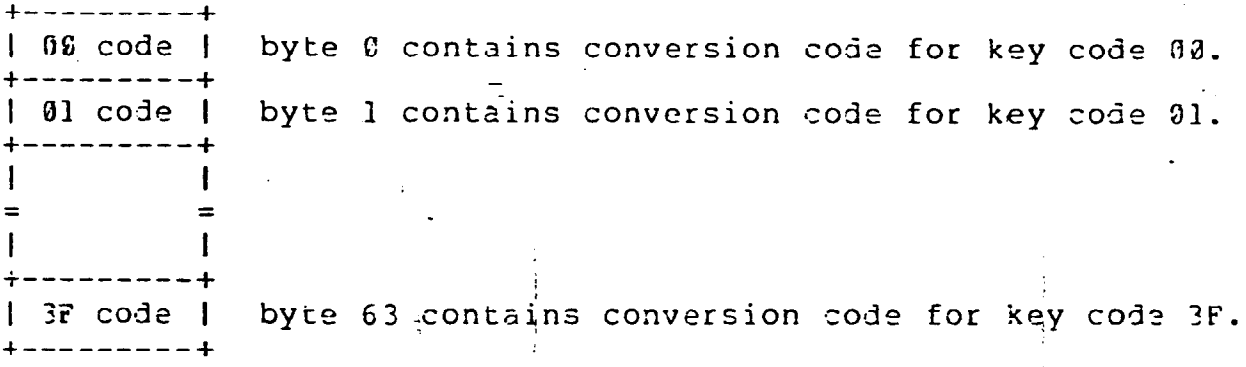
Note 7: CTRL-F3 shall control the key click on/off function as its default assignment; this assignment produces no ATASCII code.

Note 8: CTRL-F4 shall control the domestic/international character set selection; this assignment produces no ATASCII code. This key combination is not reassignable.

Key redefinition -- Most of the SWEET16 console keys shall have the ability to generate any desired ATASCII code or special internal function (the exceptions are listed at the end of this subsection). The mechanism for key reassignment shall be a 192 byte key definition table of the form shown below:



The three subtables shall contain ATASCII codes or control codes to be generated from the hardware key codes (obtained from the POKEY chip). The hardware key codes (0-63) shall be used to index directly into one of the three subtables; the proper subtable shall be selected by the state of the SHIFT and CTRL keys, all CTRL-SHIFT combinations shall be ignored by the Keyboard Handler. Each of the three 64 byte subtables has the form shown below:



The codes in the subtables shall be treated either as ATASCII codes or as internal control codes, as shown in the table below:

$00-7F = ATASCII codes.
$80 = invalid key combination, Keyboard Handler will ignore.
$81 = invert video.
$82 = alpha lock/lower case toggle.
$83 = alpha lock.
$84 = control lock.
$85 = EOF.
$86 = ATASCII code.
$87 = ATASCII code.
$88 = gonzo function.
$89 = key click on/off.
$8A = function 1.
$8B = function 2.
$8C = function 3.
$8D = function 4.
$8E = cursor to home position.
$8F = cursor to bottom of screen.
$90 = cursor to left margin.
$91 = cursor to right margin.
$92-FF = ATASCII codes.

The table below shows the key cap corresponding to each key code, where the label at each row/column intersection is the key cap label for the key code corresponding to the value of the row value + column value:

	0	1	2	3	4	5	6	7
00	L	J	;	F1	F2	K	+	*
08	O		P	U	RET	I	-	=
10	V	HELP	C	F3	F4	B	X	Z
18	4		3	6	ESC	5	2	1
20	,	SPACE	.	N		M	/	)| (
28	R		E	Y	TAB	T	W	Q
30	9		0	7	BACKS	8	<	>
38	F	H	D		CAPS	G	S	A

There shall be a RAM vector which shall be initialized to point to the operating system default keyboard definition table at power-on and RESET. The name of this vector is KEYDEF and its location shall be 0079.

The four function keys shall have two levels of user alterable encoding: 1) as shown in the 192 byte keyboard definition table, and 2) in the function key definition table. This second level is provided so that a program may redefine the function keys with a 8 byte table rather than having to supply the complete 192 byte keyboard definition table. There shall be a RAM vector which shall be initialized to point to the operating system default function key definition table at power-on and RESET. The name of this vector is FKDEF and its location shall be 0060.

The format of the function key definition table is shown below, where each entry in the table is the ATASCII code which is to be generated by the indicated key combination:

```
+-----------+
|    F1     |
+-----------+
|    F2     |
+-----------+
|    F3     |
+-----------+
|    F4     |
+-----------+
| SHIFT-F1  |
+-----------+
| SHIFT-F2  |
+-----------+
| SHIFT-F3  |
+-----------+
| SHIFT-F4  |
+-----------+
```

The keys shown below shall not be reassignable. These keys do not generate ATASCII codes or have other reasons for not being reassignable. The reasons are given for each key.

BREAK -- Fixed function, special case in O.S., special sense in hardware, no requirement for reassignment.
SHIFT -- Integral part of hardware key coding.
CTRL -- Integral part of hardware key coding.
OPTION/SELECT/START -- Directly wired to, and sensed through, the GTIA chip.
RESET -- Directly wired to 6502 processor reset.
HELP -- Fixed function, special case in O.S.
CTRL-1 -- Screen output start/stop, special case in O.S.
CTRL-F1 -- Keyboard enable/disable, special case in O.S.
CTRL-F2 -- Screen DMA control, special case in O.S.
CTRL-F4 -- Domestic/international character set control, special case in O.S.

User alterable key auto-repeat rate -- The keyboard key auto-

repeat rate shall be program alterable. The repeat rate shall be controlled by the value of an operating system data base variable which shall contain the number of VBLANKS to occur between key repeats. The operating system shall set the default (power-on) value of this variable to 6 for NTSC systems (60 Hz) and 5 for PAL systems (50 Hz), thus providing a uniform repeat rate of 10 characters per second for both systems. The name of the control variable is KEYREP and its location shall be 02DA.

The initial auto-repeat delay shall also be program alterable. The delay shall be controlled by the value of of an operating system data base variable which shall contain the number of VBLANKS to occur before the auto-repeat starts. The operating system shall set the default (power-on) value of this variable to 48 for NTSC systems (60 Hz) and 40 for PAL systems (50 Hz), thus providing a uniform delay of 0.8 seconds for both systems. The name of the control variable is KRPDEL and its location shall be 02D9.

Key click enable/disable -- The key click provided by the Keyboard Handler whenever it processes a keystroke shall have the ability to be enabled and disabled by a program. A flag shall be provided in the operating system data base for this purpose. The CTRL-F3 key combination shall toggle the state of this flag when the keyboard is active. The flag shall be initialized to the click enabled state at power-on and RESET. The name of the control flag is NOCLIK and its location shall be 02DB. A value of zero shall indicate that the key click is enabled, and a value of $FF shall indicate that the key click is disabled.

Domestic/international character set selection -- An international version of the display character set shall be provided in the operating system ROM at location C800. The international character set shall be selected whenever a program stores the constant $C8 to data base variable CHBAS at location 02F4. The domestic character set shall be selected whenever a program stores the constant $E0 to that same variable.

The CTRL-F4 key combination shall toggle the character sets as shown in the state table below:

current character set	new character set
domestic	international
international	domestic
user defined	international

Whenever the operating system enables the international character set, it shall light LED 2. Whenever the operating system enables the domestic character set, it shall unlight LED 2. The operating system shall not alter the state of LED 2

when a program alters variable CHBAS.

The international character set is defined in ERS supplement 5.

Keyboard enable/disable -- The keyboard shall have the ability to be enabled and disabled by a program. A flag shall be provided in the operating system data base for this purpose. The CTRL-F1 key combination shall toggle the state of this flag. The flag shall be initialized to the keyboard enabled state at power-on and RESET. The name of the control flag is KEYDIS and its location is 026D. A value of zero shall indicate that the keyboard is enabled, and a value of $FF shall indicate that the keyboard is disabled.

Whenever the operating system disables the keyboard, it shall light LED 1. Whenever the operating system enables the keyboard, it shall unlight LED 1. The operating system shall not alter the state of LED 1 when a program alters variable KEYDIS.

While the keyboard is disabled, the following keys will still perform their normal function:

CTRL-F1 -- keyboard enable.
RESET -- 6502 chip reset and O.S. reset, not disablable.
OPTION/START/SELECT -- not controlled by O.S.

CAPS LOWR key toggle action -- The operator's view of the CAPS LOWR key shall be as shown in the state table below:

Key combination	current state	new state
SHIFT-CAPS	don't care	ALPHA LOCK
CTRL-CAPS	don't care	CONTROL LOCK
CAPS	CONTROL LOCK	LOWER CASE
CAPS	ALPHA LOCK	LOWER CASE
CAPS	LOWER CASE	ALPHA LOCK
CTRL-SHIFT-CAPS	don't care	no change

LED initialization -- The operating system shall initialize the two LEDs to be unlit at power-on and RESET. (The function of LED 1 is to indicate that the keyboard is disabled and the function of LED 2 is to indicate that the international character set is enabled.)

Game cartridge remove/insert interlock -- The operating system Stage 2 VBLANK routine shall detect that a cartridge has been removed or inserted with power on. If a change of state is detected, the operating system shall perform a power-on initialization sequence.

Power-on self-test -- During initial power-on, the operating system shall make the following tests:

Can all RAM locations go to the $FF state?
Can all RAM locations go to the $00 state?
Do the two O.S. ROMs checksum properly?

If any of these tests fail, the operating system shall transfer control to the self-test memory test, where a more complete test of RAM and ROM shall occur. This test shall be selectable based upon the presence or absence of jumper J1, as described in the following paragraph.

Option jumpers -- There are four option jumpers provided in the hardware. The operating system shall read the states of these jumpers during power-on and store the states in data base variable JMPERS in location 030E. The bit assignments are shown below:

Bit	Function	Hardware name
0	0 = power-on self-test enable	J1 (pot 4).
1	unassigned	J2 (pot 5).
2	unassigned	J3 (pot 6).
3	unassigned	J4 (pot 7).
4-7	unused	N/A.

Screen mode 0 split screen option -- The split screen option shall be supported in screen mode 0. With that option enabled, all screen data going to the Display Handler (S:) shall go to the upper 20 lines of the screen with none of the ATASCII control characters being recognized (with the exception of CLEAR and EOL); all screen data going to the Screen Editor (E:) shall go to the lower 4 lines of the screen with the ATASCII control characters being processed normally. This option shall be allowed only when database variable SPLTM0 at location 0247 contains a nonzero value when the OPEN command is issued to the Display Handler.

Text screen fine scrolling -- The Screen Editor (E:) shall support fine scrolling of the text screen data, as an option. The fine scrolling option shall be selected by setting the database variable FINE at location 026E to $FF prior to OPENing the Screen Editor; conversely, the coarse scrolling option shall be selected by setting FINE to $00 prior to the OPEN.

Non-clearing data base variables -- Locations 03ED-03FF shall not be altered during a normal RESET sequence (warmstart). These locations shall all be set to zero during a normal power-on sequence (coldstart).

Additional hardware screen modes -- The table below shows the standard A400/800 software screen mode to hardware screen mode mapping:

Software mode	ANTIC mode	GTIA mode
0 ($00)	2 ($02)	0
1 ($01)	6 ($06)	0
2 ($02)	7 ($07)	0
3 ($03)	8 ($08)	0
4 ($04)	9 ($09)	0
5 ($05)	10 ($0A)	0
6 ($06)	11 ($0B)	0
7 ($07)	13 ($0D)	0
8 ($08)	15 ($0F)	0
9 ($09)	15 ($0F)	1
10 ($0A)	15 ($0F)	2
11 ($0B)	15 ($0F)	3

The following mappings shall be added to the Display Handler for the SWEET16:

Software mode	ANTIC mode	GTIA mode	
12	4 ($04)	0	Note 1
13	5 ($05)	0	Note 1
14	12 ($0C)	0	
15	14 ($0E)	0	

Note 1: Character definitions are not provided for these character modes; the existing ROM character definitions will not produce recognizable characters, so the user will have to provide the character definitions.

User specified disk sector length -- The resident Disk Handler shall have the ability to read and write disk sectors of any length from 1 to 65535 bytes. The default length, which shall be established at power-on and RESET, shall be 128 bytes, but may be altered by a program. The double byte variable named DSCTLN, in the operating system data base, shall be assigned for this function; the location of this variable shall be 02D5.

Disk Handler WRITE SECTOR (NO VERIFY) command -- The resident Disk Handler shall have the ability to recognize and process the WRITE SECTOR (NO VERIFY) command ('P').

Power-on display -- The A400/800 memo pad display shall be replaced by a display which features a dynamic rainbow ATARI. If the HELP key is pressed while this display is being maintained, then the self-test program shall be invoked.

A program which generates this display is resident on a test disk which is included with this specification. In order to execute that program, perform the following steps:

1. Put the disk in drive 1.

2. Power on the A400/800, with no cartridge inserted.

Self test -- See ERS supplement 1.

Relocating loader -- See ERS supplement 2.

Serial bus device polling at power-on time -- See ERS supplement 3.

3.2.3 Enhancements

The following functions that are supported by the A400/800 Rev B Operating System shall be enhanced as indicated:

Printer CLOSE with data in buffer -- The Printer Handler shall insert an EOL to the printer buffer, if one is not present, before sending the buffer to the printer on CLOSE.

Printer unit number handling -- The Printer Handler shall process the unit number in the IOCB, thus allowing separate addressing for devices P1 through P8.

CIO handling of truncated records on read -- CIO shall place an EOL in the user's buffer on the occurrence of either of the two conditions shown below:

Record longer than user buffer (truncated record).
ECF condition.

CIO error handling with zero length buffer -- CIO shall return a buffer length of zero, when there is a handler error while effecting a zero length buffer transfer (6502 A register).

Display Handler cursor handling -- The Display Handler shall accept a clear screen control code, no matter what the values of the cursor coordinates.

Display Handler/Screen Editor memory clearing -- The Display

Handler and Screen Editor shall not clear memory beyond the end of memory as indicated by RAMTOP.

Rework of the Floating Point Package -- The Floating Point Package shall be reworked to produce an error status when an attempt is made to calculate the LOG or LOG10 of 0.

New ROM vectors -- The following fixed entry points shall be present in the operating system ROM:

```
E480  JMP PUPDIS     entry to power-on display.
E483  JMP SLFTST     entry to self-test program.
E486  JMP RELEAD     entry to relocating loader.
E489  JMP PHENTR     entry to uploaded handler enter.
E48C  JMP PHULNK     entry to uploaded handler unlink.
E48F  JMP PHINIS     entry to uploaded handler init.
```

3.3 Performance requirements

The timing of all functions performed by the SWEET16 operating system shall correspond to the timing of the same functions performed by the A400/800 operating system, to the degree required to maintain compatibility with the existing A400/800 software products.

3.4 Design requirements

The total size of the resident operating system, including the Floating Point Package and the screen character definitions, shall not exceed 16 Kbytes.

3.5 Packaging requirements

N/A.

3.6 Special requirements

3.6.1 Serial bus peripheral compatibility

Existing (and future) Atari serial bus peripheral devices shall all be usable on a SWEET16 machine.

3.6.2 Application software compatibility

Atari supported software written to run on the Atari

400 and 800 machines shall operate transparently on a SWEET16 machine of suitable configuration (memory and peripherals). This includes software written in any of the supported languages (assembler, BASIC, PILOT and Pascal).

"Atari supported" shall, in general, be limited to software using published and supported entry points and database variables in the operating system. However, it shall be necessary to guarantee operation of certain software packages which are of significant value to the HCS user. As, in the course of compatibility test, programs are found to be incompatible due to the "illegal" use of operating system code or variables, the effort required to force compatibility into the SWEET16 operating system shall be weighed against the value of the offending application software, and a decision made accordingly, on a case-by-case basis.

This section shall list the offending applications and their variances required in the SWEET16 operating system.

3.6.2.1 VISICALC

A jump vector shall be added to the SWEET16 operating system, in the location corresponding to the Revision B begin address of SIO, which jumps to the entry point of SIO in the SWEET16.

4.0 Applicable standards

Applicable HCD established documentation and programming standards shall be followed throughout the product design and implementation.