DOS/65 OSI NOTES

The DOS/65 package for OSI is somewhat different from the usual DOS/65 package. The following notes describe the unique aspects of that package.

SECTION 1 - DISK FORMAT

1.1 BYTE FORMAT

Each byte on the diskette uses the same bit format as OS65D.

1 Start Bit
8 Data Bits
1 Even Parity Bit
1 Stop Bit

As is the case for OS65D, this format is implemented by writing a $58 to the control port of the controller ACIA at $C010 after issuing an ACIA master reset.

1.2 TRACK FORMAT

The track format for the OSI version of DOS/65 is similar to but not identical to the standard OS65D format. Each track of a DOS/65 diskette is formatted using a format similar to that normally used for all but track zero of an OS65D diskette. Each track has a header consisting of the following bytes:

$43
$57
Track Number in BCD
$58
$76
$01
Track Length in Hex Pages

The track number ranges from $00 to one less than the number of tracks on the diskette and as noted is in BCD. For eight inch systems the tracks are numbered from $00 through $76. For five inch systems the tracks are numbered from $00 through $39.

The number of pages in a track is fixed at 13 ($0D) for eight inch systems and 8 ($08) for five inch systems.

CAUTION

Because of the unique OS65D diskette format for track zero and the other format differences between OS65D and DOS/65, the normal OSI disk copy and initialization routines can not be used on a DOS/65 diskette. Use only the DOS/65 routines COPY.COM and FORMAT.COM respectively.

The data portion of each track consists of a continuous stream of data of the length specified. DOS/65 views each track as 26 (eight inch) or 16 (five inch) logical records of 128 bytes each. DOS/65 reads or writes individual records by buffering a full track at a time in the SYSTEM INTERFACE MODULE (SIM) and then mapping the DOS/65 record from or to the appropriate part of the full track buffer.

1.3 DATA INITIALIZATION

Unlike OS65D, each byte on a DOS/65 diskette must be initialized. The DOS/65 diskette formatting program, FORMAT.COM, accomplishes that by writing $E5s to every data byte in a track after writing the header for that track. Use of an $E5 ensures compatibility with existing DOS/65 and CP/M software which assumes that the diskette is initialized in accordance with the IBM standard.

CAUTION

The OSI version of CP/M may not require all tracks (especially track zero) to be formatted and initialized as described above. When using an OSI CP/M diskette use the program SYSINIT.COM to format the system tracks of the diskette. This process destroys all data on those tracks. If the data on the system tracks must be preserved, make sure that the diskette is copied to on another diskette using OS65D before using SYSINIT.COM under DOS/65.

1.4 SYSTEM TRACKS

The standard OSI SIM for polled keyboard-video based systems requires slightly less than 6 full pages of code. When that requirement is added to the BOOT, CCM, and PEM needs, a minimum of 51 records must be allocated for the system tracks. As the following table indicates, eight inch OSI systems could satisfy that requirement with only two system tracks. Two tracks would however leave only one record for expansion. In addition, the OSI version of CP/M allocates three system tracks. Because of these two factors, three systems tracks have been allocated for the eight inch OSI version of DOS/65 (SIF D). For five inch OSI systems, similar analysis resulted in allocation of four system tracks (SIF E). While this provides less expansion room for the five inch systems than for the eight inch systems, use of four tracks is a reasonable trade between expansion room and lost data area.

TOTAL AVAILABLE RECORDS

	EIGHT INCH	FIVE INCH
TWO TRACKS	52	32
THREE TRACKS	78	48
FOUR TRACKS	104	64

SECTION 2 - SIM

2.1 DISK I/O

2.1.1 FULL TRACK BUFFERING

As delivered, the DOS/65 SIM for OSI (OSIMnn.ASM) is setup for full track buffering. Logic is included which will read or write new tracks only when absolutely required. The logic used to determine when a track should be written is shown in overview form in the following pseudo-code.

```
IF BUFFER EMPTY
     THEN EXIT
     ELSE IF DATA NOT ALTERED
          THEN EXIT
          ELSE IF DRIVE CHANGE AND TRACK CHANGE
               THEN WRITE TRACK
               ELSE EXIT
```

This routine is executed whenever the SIM routines READ, WRITE, & HOME are executed by PEM or by direct user calls.

CAUTION

Some commands (e.g., ERA and SAVE) and programs (e.g., COPY and SYSGEN) may complete execution and show the CCM prompt without having actually written a track to the diskette. Do not remove a diskette until a DIR command is executed or a WARM BOOT is executed by typing a control-c. The polled keyboard/video version of SIM includes code to display a special character on the screen below the data entry line if a track write is pending. If that special character is not displayed then it is safe to remove any diskette. For serial terminal based systems, any attempt to open the door of a drive for which a track write is pending causes a warning message to be printed on the console and causes all execution to be halted until the door is closed.

It is possible to change SIM so that the track would be re-written after every DOS/65 recordr write. While such a change might simplify SIM and eliminate any possibility of lost data due to diskette changes, a penalty would be paid in execution time. Alternatively SIM might be modified to write whole tracks when absolutely needed but to always write after every DOS/65 directory record write. Since that is the most critical operation, the speed and wear & tear impact may be an acceptable trade-off for the assurance provided.

2.1.2 DIRECT SIM CALLS

CAUTION

> *Use of direct calls to SIM for disk I/O is not recommended. Use the file management features provided by PEM.*

When calling SIM disk READ and WRITE routines directly from user programs, the following protocol must be used unless the drive in question has already been accessed by the system (e.g., a DIR command for the drive in question was executed):

<u>FIRST ACCESS TO DRIVE</u>
SELECT DRIVE
HOME DRIVE
SET TRACK
SET RECORD
READ or WRITE RECORD

<u>SUBSEQUENT ACCESSES TO DRIVE</u>
SELECT DRIVE (If Changed)
SET TRACK (If Changed)
SET RECORD (If Changed)
READ or WRITE record

The HOME during the first access to a drive is needed to ensure proper track seeks. Once that is done SIM will maintain a record of the current track and seek a new track as needed. The user need only ensure that the proper drive, track and record are set before calling the READ and WRITE routines in SIM.

2.2 NUMBER OF DISKS

SIM includes code which asks the user to specify the number of drives available. This code is only executed at BOOT time. The user should respond to the question by typing the correct number (1 through 4). SIM includes provisions to allow all DOS/65 programs (including COPY.COM) to be used with a single drive. If the user has a single drive, the prompts "MOUNT A", "MOUNT B", "MOUNT C", and "MOUNT D" will be sent to the console to indicate which "logical" drive is in use. All the user must do is keep track of which diskette is which and insert the correct diskette after the appropriate prompt is displayed. After inserting the correct diskette into the drive, the user must type a RETURN to tell SIM that the requested diskette has been mounted.

2.2 CONSOLE I/O

2.2.1 POLLED KEYBOARD AND VIDEO SYSTEMS

The standard package provided for the C4P and C8P polled keyboard/video systems has the following characteristics:

VIDEO

The video is set for 28 lines of 64 characters each.

The video driver provides those features required for the current version of DOS/65. Additional features can be added by the user, the screen size can be altered, and the control character assignments used can be altered as long as the CONSOLE DEFINITION BLOCK in SIM (see SYSTEM INTERFACE GUIDE) is also altered as needed.

It is assumed that the screen is cleared by RESET (the BREAK key) or by OS65D. If that is not the case or if a stand alone capability is desired, LOADER or SIM could be modified to include a screen clear when a cold BOOT is performed or when otherwise desired.

The video driver includes the special disk write pending flag provisions noted in Section 2.1.1.

KEYBOARD

The routine provided in SIM causes the keyboard to act like a "normal" keyboard. In particular, the SHIFT keys act like "normal" shift keys and the SHIFT LOCK key provides a "CAPS LOCK" for the alphabetic keys. After executing LOADER the SHIFT LOCK key need not be left down and a shift is only needed to type an upper case alphabetic character or the shifted character on a key (e.g., ! instead of 1). Automatic repeat is provided on all keys. The following combinations can also be used:

TYPE to	GENERATE
RPT-P	@
RPT-K	[
RPT-M	]
RPT-L	\
RPT-N	^
SHIFT-RPT-K	{
SHIFT-RPT-M	|
SHIFT-RPT-L	}
SHIFT-RPT-N	

The standard package provided for C1P and SBII systems has the following characteristics:

VIDEO

The video is set for 24 lines of 24 characters each.

The video driver provides those features required for the current version of DOS/65. Data is included in SIM which will allow the user to change to a 12 line display with 48 characters per line. Additional features can be added by the user, the screen size can be altered to non-standard formats, and the control character assignments used can be altered as long as the CONSOLE DEFINITION BLOCK in SIM (see SYSTEM INTERFACE GUIDE) is also altered as needed.

LOADER clears the screen and sets the display mode to 24x24. SIM sets the display mode to the actual mode to be used.

The video driver includes the special disk write pending flag provisions noted in Section 2.1.1.

KEYBOARD

The keyboard routines are identical to those described above for the C4P and C8P.

2.2.2 SERIAL SYSTEMS

The console routines provided for serial systems are based upon the standard OSI routines for a 6850 ACIA at $FC00. They can be altered as necessary. The console status routine includes the door open detection logic discussed in Section 2.1.1.

2.3 LIST DEVICE

For all systems this SIM entry is set to use the ACIA at $FC00 ($F000 for C1P/SBII). A copy of the code for the standard OSI parallel printer is included on the diskette as the file PARALLEL.LIB. If it is desired to use the parallel printer port that file should be included in SIM using the R command under EDIT.COM and the rest of SIM altered to use that routine.

2.4 PUNCH AND READER DEVICE

For all systems these entries in SIM are implemented as NOPs. Users may alter them as needed. The file UART.LIB on the disk shows how to use the 430 board UART for these two devices.

2.5 CPU CLOCK RATE

The standard SIM includes code in the COLD BOOT portion of SIM to adjust SPEED to match the CPU clock rate. The technique used is to measure the number of times a given loop is executed between index pulses on the diskette. For troubleshooting purposes note that SPEED should be 49 or 50 for a 1 MHz clock and proportionally higher for higher clock rates.

SECTION 3 - BOOT AND LOADER

3.1 DESIGN

The design and coding for BOOT (OBnn$2ss.ASM) and LOADER (OL$2ss.ASM) are documented reasonably well by the comments in the two files. It is important to note that LOADER could be located in ROM with only minor changes (i.e., setting ORG of the main code to the ROM address and placement of TRKBUF in RAM). The key requirement for LOADER is that it not change as MSIZE or PAGES (see SIM and BOOT listings) change and that it not use any form of self-modification. As currently implemented both those requirements are met.

3.2 CPU CLOCK RATE

LOADER will work correctly with a CPU clock rate of up to 4 MHz or more since the 1 ms delay constant in LOADER is set to more than five times the value needed for a 1 MHz clock. The only adverse affect is that the step rate in LOADER will be less than ideal. Given the fact that LOADER is only used at COLD BOOT this approach is a reasonable trade between simplicity and performance.

3.3 BOOT PROCEDURE

The actual procedure to be followed to BOOT DOS/65 is as follows:

Step 1- Boot OS65D using your current OS65D system diskette.

Step 2- Exit BASIC to the OS65D command level by entering the command:

```
EXIT
```

Step 3- Insert the diskette labeled LOADER into drive A.

Step 4- Enter the following two commands:

```
Eight Inch            Five Inch
CALL 0200=76,1        CALL 0200=39,1
GO 0200               GO 0200
```

Step 5 - In response to the prompt on your console, insert one of the DOS/65 diskettes in drive A and then type RETURN (SPACE on C1P/SBII).

Step 6- In response to the prompt on your console, enter the number of drives in your system (1 to 4) after ensuring that the SHIFT LOCK key is up.

Step 7- You should now see the A> prompt indicating that DOS/65 is ready for your use. If you have not read the DOS/65 documentation then STOP and read the documentation package.

Step 8- Prepare at least one backup copy of the DOS/65 diskettes. Because the diskettes are nearly full, two backups make sense so that some programs can be deleted from the diskette to provide room for editing, assembly, or other file operations. The procedure to use is:

- Use FORMAT to initialize the new diskette. (Do not forget to take your system diskette out of drive A and insert a blank diskette in drive A when asked if ready!)
- Use COPY (command should be COPY ALL) to duplicate the system diskette. Remember that if you have a single drive, the system will prompt you to "MOUNT A" or "MOUNT B".

Step 9 - Use OS65D to prepare backup copies of the OS65D track containing LOADER on OS65D diskettes (not on DOS/65 diskettes).

SECTION 4 - MISCELLANEOUS

4.1 FORMAT and SYSINIT

FORMAT and SYSINIT include provisions to adjust the primary timing constant, SPEED, for the actual CPU clock rate. No user adjustment is needed.

4.2 BASIC-E/65

Read the manual for BASIC-E/65 carefully and look at the programs included on the diskette to see how to use this part of DOS/65. The set of mail list related programs are a reasonably good illustration of how to use the disk I/O features of BASIC-E/65. Note in particular that the syntax of BASIC-E/65 is not the same as Microsoft/OSI BASIC but is very similar. The file handling capabilities of DOS/65 are much easier to use but take some thought since they are quite different from the Microsoft/OSI methods.

4.3 FILESTAT

The program FILESTAT is a BASIC-E/65 program which displays detailed data about the size of each file on a diskette. In addition to providing necessary data about the files, it also illustrates how use of the CALL capability in BASIC-E/65 can be useful. After typing RUN FILESTAT(cr), have patience as it is not particularly fast.

4.4 OSI Data Extraction

The programs DIROSI, GETASCII, GETASM, and GETBASIC are designed to allow conversion of OSI OS65D files to DOS/65. Each assumes that a DOS/65 diskette is mounted in drive A and that the OS65D diskette is mounted in drive B. Since SIM allows multiple logical drives even in a one drive system, these programs can be used on a single drive system by following the MOUNT A and MOUNT B instructions issued by SIM. When asked to MOUNT B, insert the OS65D diskette into the drive. Insert the DOS/65 diskette when told to MOUNT A.

4.4.1 DIROSI

This program will read the OS65D directory and print the results on the console. The format is similar to the usual OSI approach in that the file name is printed followed by the first and last track numbers.

4.4.2 GETASCII

GETASCII will read user specified tracks and sectors from an OS65D diskette and write the contents to the specified DOS/65 file. The command syntax is:

```
        getascii colors.dat
```

The example shown would write the data to a file named COLORS.DAT. GETASCII assumes that the file being read is a normal ASCII file and that each line is terminated by a carriage return ($0D) without a linefeed ($0A). Conversion will be automatically terminated upon read of a null ($00) from the OS65D file. The operator entries after entering the CCM command are all prompted and are self explanatory.

4.4.3 GETASM

GETASM will read an entire OS65D assembler source file and write the contents to the specified DOS/65 file. The command syntax is identical to that shown for GETASCII. GETASM will expand all characters stored in compressed form and like GETASCII will convert every carriage return to a carriage return-linefeed pair as required by DOS/65. Since line numbers are not needed by the DOS/65 editor or assembler, the line numbers in the OS65D file are ignored. The only required user entry after execution is the track number of the start of the file.

CAUTION

> *If the track specified by the user is not the first track of an OS65D assembler source file, results could be disastrous.*

4.4.4 GETBASIC

This program is very similar to GETASM in that it will read and convert an entire file after being told the track number of the first track in the file. GETBASIC is much more complex in that it assumes that the file being converted is a tokenized OS65D BASIC file and that the line numbers must be preserved. GETBASIC expands all tokens and converts all line numbers to ASCII followed by a tab. In addition, GETBASIC offers the user the option of adding spaces around the expanded keywords in order to ensure that the source file will be easier to convert to BASIC-E/65. While GETBASIC uses a conservative approach when adding spaces, it does add spaces only where the syntax is such that confusion could result under some circumstances (e.g., it adds a space before SIN but not after, it adds a space both before and after GOSUB, it adds no spaces around +, and it adds a space after LET but not before). Extra spaces do no harm and could be removed using EDIT if desired.

CAUTION

> *If the track specified by the user is not the first track of an OS65D BASIC source file, results could be disastrous.*

4.5 DEBUG

The location pointed to by the usual OSI IRQ/BRK vector at $FFFE is $1C0. This location is not explicitly used by DOS/65 so the user can set the byte at $1C0 to $4C so that breakpoints can be used by DEBUG. Since $1C0 is very high in page 1 and might be used by the stack, it is possible that use of breakpoints might cause a system crash.

SECTION 5 - HARDWARE CHANGES

5.1 C4P/C8P

No hardware changes are required to run DOS/65 on C4P or C8P systems.

5.2 C1P/SBII

While no hardware changes are required to run DOS/65 on C1P or SBII systems, it is strongly recommended that the following change be made.

The standard C1P and SBII CPU boards include logic to direct the video RAM address bus to the 6502 whenever the 6502 address bus is 110100xxxxxxxxxx (binary). This normally will only happen when the CPU is actually performing reads or writes to the video RAM beginning at $D000. However there are two I/O devices addressed very close to this area. The keyboard array is at $DCxx and the disk controller is at $C0xx. In both cases it is possible that at some time during the Phase 1 portion of a CPU read or write cycle, the address bus may actually be such that the video address bus is switched from the normal video address generator to the CPU. When that happens the screen will flicker. This flicker is harmless but is objectionable.

The solution is simple. The 6502 really only needs to address the video RAM during the Phase 2 period of each cycle. During Phase 2 the address bus is guaranteed to be stable and hence will not cause flicker. The modification recommended is to cut the trace going to U56 Pin 2 and place a jumper between U56 Pin 2 and the Phase 2 clock signal from the 6502. Note that in some cases, if the 2114 RAMs used for the video RAM are a bit slow, this modification may not work. In that case replace the RAMs with faster devices.

Users who have modified their CPU board for larger display sizes may have already made similar or conflicting changes to the inputs to U56. Those users should adjust the modification to fit their configuration.