



# EXOS®—A Software Development Tool Kit for the 6500 Microprocessor Family



BY DON COLBURN  
EG&G Washington Analytical Services Center, Inc.  
2150 Fields Road  
Rockville, Maryland 20850

The fastest and easiest way to complete any task is with the right tools. This statement is as true with software as it is with hardware. The person that assembles a CPU card with a 60 watt soldering iron, 1/8" solder, and a pair of lineman's pliers is in the same boat as the one who hand assembles an 8K assembly language program. Although it is possible to assemble a CPU card with limited tools or to hand assemble large assembly language programs, few people who are seriously developing software can afford the time and resources required to do the job without the right tools.

This article describes a software development tool kit which provides the "right tools" required to efficiently and effectively both generate and modify 6500 assembly language programs. The level of capability is equal to that of some mini-computer operating systems.

When I bought my JOLT® 6502 based system a couple of years back, I quickly realized that I would need something more powerful than the TIM® monitor if I seriously intended to do more than flash LED's. I had two alternatives. I could write the required software, assemble it, load it, and debug it by hand, or I could purchase someone else's existing software. Faced with the chicken and the egg problem, and feeling somewhat strongly about what such a package should contain, I decided to first buy an existing assembler and editor package: with these basic tools I would develop my own expanded package. The assembler had to be easy to use and compatible with the standard MOS technology mnemonics and the editor had to have the ability to create and save source files which could be directly input to the assembler. The task of finding these basic tools proved more difficult than I first anticipated.

While at a computer show one weekend, I came across a booth offering 6500 software for sale. Their sign advertised a 4K operating system, a 2K single pass Assembler, a 1K Editor, and a Trace/Disassembler package. Skeptically, I approached the counter. Two hours later, I left the counter carrying eight EPROMS and four comprehensive manuals. Not only had I found the assembler and editor I had been looking for, but I had found the software development tools I had intended to write. The name of the package is EXOS®.

EXOS stands for "Extended Operating System." This is because it contains a base-level operating system with 17 standard and 5 extended commands. The extended commands pass control to an optionally resident Assembler, Editor, Disassembler and Trace program, or Disk Operating System. EXOS is compatible with all 6500 systems based on the TIM monitor. Other monitors such as Apple and homebrew versions may be interfaced, because both the program cold and

warm entry points and system I/O vectors are located in the first 27 bytes of each program (Table 1) and initialized into RAM upon cold start. This allows the user to change the I/O calls according to System Requirements.

Table 1

## EXTENDED OPERATING SYSTEM "EXOS" FAMILY VECTORS

1. The following vectors are used as convention in all "EXOS" family software packages and programs.

- This feature should allow "EXOS" software portability or any MOS Technology 65XX based system.
- Vectors are common if used in that particular package. Where a vector is not used, either SEA/SFF will be found.
- All "EXOS" programs will operate under the "TIM" or "DEMON" operating systems without modification. Operation under the "KIM" OS is attained thru change of the address vectors to be described.
- Vectors

HEX PROGRAM ADDRESS	LABEL	OPCODE	LABEL	COMMENTS
XX00*	ENTER	JMP	ENTRY	COLD START initialized regs.
XX03	RENTER	JMP	RENTRY	WARM START re-enters program counter & regs.
XX06*		SPECIAL (See note)		Varies w/EXOS program
XX09	PR1CHR	JMP	PRCHR	Print 1 character to input (i.e. printer, CRT) de- vice from accumu- lator
XX0C	RD1CHR	JMP	RDCHR	Read 1 character from input (i.e., keyboard, tape, disc) device to accumulator
XX0F	PRCRLF	JMP	PREOL	Print CR & LF w/delay in SE3
XX12	PR2SPC	JSR	PR1SPC	Uses PR1SPC
XX15	PR1SPC	JMP	PRSPC	Print 1 space
XX18	PR1BYT	JMP	PRBYTE	Print 1 hex byte
XX1B***		BREAK	JMP	BRKST Test for dynamic escape thru use of "break" key.
XX1E**		SPECIAL		
XX21		3 byte vectors that depend upon particular "EXOS" program		

EXOS® is a Trademark of RCS Associates, NY.  
JOLT® is a Trademark of Microcomputer Associates, CA  
TIM® is a Trademark of MOS Technology, PA.

Table 1 cont'd

NOTES:

- \* All programs use the "ENTER" vector, however the presence of other vectors depend upon the program itself.
- \*\* Varies w/program
- \*\*\* Routine depends upon I/O used w/system some "EXOS" programs have the "BREAK" imbedded in object code, however use of an external break routine is still allowed. Imbedded routines are identified w/the particular program and locations are given to allow the user to alter those routines if desired.

Standard EXOS commands may be divided into four categories; operational, functional diagnostic, and extended. The operational commands provide the user with the capability to ENTER, LOAD, and DUMP data to and from memory and to DISPLAY, FIND, MOVE, calculate branches and addresses (MATH) and EXECUTE that data. The Functional commands allow the user to modify the carriage return delay in all systems, and the baud rate (up to 4800 baud) in TIM systems, under keyboard/software control. The diagnostic commands allow the user to nondestructively TEST or COMPARE any specified blocks of memory on a bit-by-bit basis. Extended commands allow direct access to the optionally resident Assembler, Editor, and Debug firmware. Table 2 contains a brief description of each of the 17 standard EXOS commands.

TABLE 2

The Table contains the name and a brief description of each of the 17 Standard EXOS commands. The short form of the command is capitalized and underlined.

NAME	DESCRIPTION
<u>COMPare</u>	Compares any size block of data to another. All differences are listed along with the Hex Address.
<u>DISPlay</u>	Displays memory contents in hex and ASCII. The display is formatted into 16 lines of 16 bytes to assist in address identification. A header is output every 16 lines (256 byte block) so that a header is always visible. ASCII equivalents are presented along side the display.
<u>DUMP</u>	Dumps the specified range of memory to the I/O port in MOST* format.

\* MOST format is the tape dump format developed by MOS technology for 6500 based systems.

<u>ENTER</u>	Allows entry of hex data into memory in the display format. Pointer is auto-incremented and formatted. Editing is permitted, and ASCII equivalents are displayed similar to the display command.
<u>EXECute</u>	Allows user to jump to any specified address and begin execution.
<u>FILL</u>	Fills a specified range of memory with a specified byte of data. If memory does not accept data, an error diagnostic is printed.
<u>FIND</u>	Lists occurrences of specified hex data or an ASCII string within any specified range.
<u>LOAD</u>	Loads a MOST* format tape from either the serial I/O line or the high-speed reader port. An offset may be applied to the tape specified addresses, and a file number may be searched for.
<u>MATH</u>	Converts decimal to hex, hex to decimal, calculates relative branches, and absolute differences. Answers are displayed in hex and signed decimal.
<u>MOVE</u>	Copies specified range of memory to target address.
<u>NULL</u>	Inserts the requested number of nulls after each carriage return/line feed sequence.
<u>SPEED</u>	When used in systems using the TIM monitor, this command allows keyboard selection of different baud rates under program control. Available baud rates are 110, 300, 1200, and 2400 baud. (4800 baud with 2 MHz systems).
<u>TAPE/IRG</u>	Modifies the print 1 character routine to output an inter-record gap (IRG) between text lines dumped by the editor. This allows any tape cassette to input source to the assembler at speeds up to 2400 baud without any handshake. The IRG command allows the user to specify an inter-record gap timing value different from the "TAPE" default of 2.
<u>TEST</u>	<u>Non-destructively</u> tests each RAM location over a specified range. If an error is found, a diagnostic message specifying the exact word and bit in question is printed on the console.
<u>USER/UTILITY</u>	These commands execute a pre-defined user program pointed at by a Z-page vector. The user may alter these vectors as desired.

The EXOS Assembler (EXASMB) is a very unique piece of software. First and foremost, it is a single pass assembler. This means that forward references are resolved when the referenced label is defined. If never defined, it is so flagged and cross-referenced in the symbol table.

Second, the assembler provides full tape I/O handshake control with both ASCII control characters and physical relay drivers.

Third, when the Assembler is used with EXOS and the EXOS editor (it is available stand alone), it provides audio or digital cassette input capability up to 4800 Baud, without the requirement for incremental start-stop control. This allows complete independence from the mass storage device. No other 6500 assembler that I am aware of provides this capability. All other require an ASR teletype, an incremented digital cassette or use of inefficient memory to memory techniques.

Fourth, a set of MOS Technology compatible pseudo operations is available, and when used with the optional DOS, one may designate multiple disk source files inside a single source file.

Fifth, high and low order address bytes may be designated as immediate operands.

Sixth, the symbol table uses an extremely efficient label packing algorithm which allows two distinct 8 character labels which share the same first six characters to require a total of only 11 bytes in the symbol table. The pseudo-op .END on the EXOS command XREF allow the symbol table to be dumped repeatedly. All labels appear in alpha sorted order when listed.

Finally, and most remarkably, the entire object image of the Assembler requires only 2K. This occupies 1/3 the space of some 6500 assemblers, while providing more capability in a single 2716 or two 2708's.

The EXOS Editor (EXEDIT) was written with the same 'attitude' as the assembler. It provides full line editing capabilities, as well as the ability, when used with EXOS, to output assembler compatible source code to any cassette interface. The line editing capabilities allow one to print any number of lines, to insert a line, or delete a line. A unique feature allows one to automatically generate line numbers when source code is being input, and turn them off when the source is being dumped. Another is a selectable automatic tab feature which advances a variable number of spaces (default is 8) following the label to align all operation fields. Again it is quite remarkable that the total EXEDIT software resides in only 1K of memory (1-2708 EPROM).

The EXOS Disassemble/Trace package (EXTRA) is one of the most powerful software debugging tools available. The disassembler disassembles code from the start to the end of a user defined range. If the user interrupts the program with the BRK key, he is given the option of continuing with the disassembly or exiting back to EXOS. The Trace program requests the user to load the register with the conditions he wants to trace from (includes P.C.). It then disassembles the code at that location, displays the hex code and mnemonic, and awaits user input. If the user strikes a '/', the op code is executed, the resultant registers are displayed and the next line of code is disassembled. If the user strikes a '+' and a hex address, the processor executes the code in real time until the P.C. equals the given address. At this point, the line is disassembled, and the program waits for user input. If the user strikes an 'X', tracing from the current location is

terminated and the user is reprompted for new PC and register contents. Upon receipt of a ':' the software allows the user to modify the current register contents prior to execution of the current line of disassembled code, allowing modification of program flow as desired. Any time the "ESC" key is struck, control is passed back to EXOS. This tool allows tracing through RAM or ROM resident programs in both binary and decimal modes (the 6502 has a unique and easy-to-use decimal operating mode). The object image for both the Trace and Disassembler combined is under 1K bytes (1-2708 EPROM).

A recent addition to the EXOS family is a firmware interface to the PerSci 277 Dual full size floppy disc and intelligent controller. This capability provides EXOS with an extremely powerful Disc Operating System (EXDOS). The interface resides in 2K bytes and adapts the entire EXOS system without modification for Disc I/O, including the Assembler and the Editor. This is typical of all EXOS software in that each higher level of software that is added requires no changes to previously purchased software. This true upward compatibility is provided for through manipulation of the Z-page system I/O vectors so that new capabilities are a super set of the previous level.

I have been using EXOS for more than a year now, and I have yet to come across a software bug. This issue contains two source and object code listings of software developed using EXOS. They include a standalone version of the EXOS Display command and a standalone interface to the Per Sci 1070 controller based on the Interface Age article of August 1977.

EXOS products are available from:

CGRS Microtech  
P.O. Box 368  
Southampton, PA 18966

RCS Associates  
P.O. Box 160  
Miller Place, N.Y. 11764

EXOS software is delivered on prime 450ns 2708 or 2716 EPROMS and each piece of software is accompanied by a comprehensive user manual. Both of the above sources offer the manuals for \$5.00 each or \$15.00 for the entire set (cost is applied towards purchase price). EXOS is now available on AIM/KIM/VIM compatible cassettes for \$30 each from RCS.

