

Introducing COMMODORE'S B-SERIES Computer

By Howard Rotenberg
Commodore Business Machines, Canada

The long-awaited B-SERIES computer has finally arrived. It is the computer that many people previously referred to as the P-500, B-700, and various other names. The product is called either the B-128 or the B-256 according to the amount of memory that the computer contains. Most of the information provided here is applicable to both models. Incorporated into the new B-Series are an Extended BASIC, Extended Memory Support, Error Trapping, RS232 Interface, Advanced Editing, Advanced Monitor commands with DOS support, special function keys and many other enhancements. The processor used in this computer, the 6509, lets you address up to one megabyte (one million bytes) of memory. It is accessed as 16 individual banks of 64 bytes. The speed of the computer's processing and I/O has improved since the CBM 8032, which had been considered Commodore's "business" computer. There is also an option to run an extra processor within the computer, as well as CP/M 86 capability.

A NEW STREAMLINED LOOK

When discussing this new computer, a good place to start would be from the outside. The B-Series computer has a slick, streamlined, futuristic look. It has 94 keys, with that rich feel that was previously available only on larger development systems. Some of the functions of these keys have not been offered on Commodore computers before. The numeric keypad is laid out for quick and accurate entry of numbers. It also incorporates arithmetic operators for ease of mathematical operations. It has an ENTER key, and one key that has "00" on it. This key

will literally print two zeros, which is handy when using large numbers. The key that I find most useful is the CE or clear-entry key. This key will delete the last number that you have entered. For example, if you entered 456 + 120 and wanted to change the 120 to another number, you would otherwise have to press the delete key three times. Now with the CE key, you just need to press it once and the 120 is cleared. The parser for the CE key is smart enough that even if you enter a calculation such as 456 + 120 with no spaces and then press the CE key, it will only delete the 120. If you press it once more it deletes the plus sign, and the third time it clears the 456.

Moving to the left we find the main keyboard. It contains all the alpha-numeric and various other symbols such as brackets, percent sign, etc. There is an ESCape key that can be used in conjunction with each of the 26 alphabetic keys for a wide variety of editing functions, as listed below.

- A) Automatic insert
- B) Set bottom
- C) Cancel automatic insert
- D) Delete line
- E) Nonflashing cursor
- F) Flashing cursor
- G) Enable bell (turn it on)
- H) Disable bell (turn it off)
- I) Insert line
- J) Move to start of line
- K) Move to end of line
- L) Enable scrolling
- M) Disable scrolling
- N) Normal screen

- O) Cancel insert, quote and reverse
- P) Erase to start of line
- Q) Erase to end of line
- R) Reverse screen
- S) Solid cursor
- T) Set the top of the page
- U) Underscore cursor
- V) Scroll up
- W) Scroll down
- X) Cancel escape sequence
- Y) Normal character set
- Z) Alternate character set

The keys have graphic symbols printed on them for easy access. Holding down the CTRL key will slow down scrolling, while pressing the Commodore key will stop scrolling completely until another key is pressed. There are some format keys that allow you to switch between normal and graphics mode as well as a reverse-graphics key. The regular editing keys such as a cursor movement and CLR/HOME are all present.

A refreshing and useful new feature is the programmable function keys. There are ten keys altogether, and by using the shift you have access to twenty different functions. The first ten keys are already preset upon power up, but can be re-programmed. To list the contents of these keys you just need to enter the word KEY. The keys are initially programmed to the following:

- F1) "PRINT"
- F2) "LIST"
- F3) "DLOAD" + CHR\$(34)
- F4) "DSAVE" + CHR\$(34)
- F5) "DOPEN"
- F6) "DCLOSE"
- F7) "COPY"

PHONE-DIAL 64

DIALS TOUCH TONE PHONE AUTOMATICALLY

Using your Commodore 64
WITHOUT A MODEM

Access by phone any device
reactive to touch tone.

Save time when dialing long
multi-digit numbers.

Generate/Call a permanent
file of names and numbers.

Automatically generate/dial
sequential/random numbers

AN IDEAL TIME SAVER FOR

SPEED DIALING:

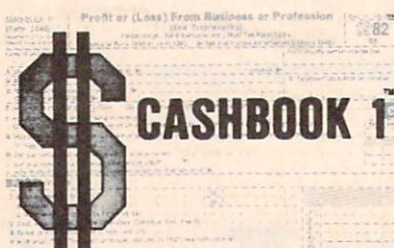
Church/Club, Crime Watch,
Random Sales/Announcements,
and other interesting uses.

DISK: (1541, 4040, 8050)

ONLY — \$24.50

INPUT SYSTEMS, INC.
25101 S. W. 194 Ave., Dept W
Homestead FL 33031,
Phone (305) 245-3141

Commodore 64
is a trademark of Commodore Electronics



CASHBOOKKEEPING.....

for any small business with a Commodore
64™, & 1540/41 Disk Drive. (printer optional)

- 1040 Schedule C Format
- Contractors: Professionals: Small Business
- P & L: Each Month & Year To Date
- Profit Ratios: Owner's Position: Net Currency & Check Book Balance
- All in Just Minutes Each Month

Hire Cashbook 1 as your new book-
keeper.

Send \$65 or call toll free order desk
1-800-321-6927. 24-Hour Service

- Also -



Programs for VIC20™
and Commodore 64™
disk or tape. Write
for listing. OR, try
MOUSE TRAP MATH,
for your VIC20™
(10.95 for tape) and
we'll include listing.

**ROCKY
SOFTWARE
ROCKY
SOFTWARE**

Box 310
Erie, Colorado 80516
(303) 652-2103

- F8) "DIRECTORY"
- F9) "SCRATCH"
- F10) "CHR\$(

The use of these keys is limited only by
your imagination and the 160 characters.
With the proper syntax, you may have a
key defined to do any batch of commands
that may normally be executed in the im-
mediate mode. An example is KEY11,
"DSAVE" + CHR\$(34) + "@:TEST" + CHR
\$(34) + CHR\$(13). This will enable you to
replace the program call test simply by typ-
ing the F11 (SHIFTED F1) key. Finally, we
come to the back of the computer. The
ports available are an IEEE, audio out, car-
tridge, cassette (although not implemen-
ted), video out and a true RS-232
port. There is an internal user port that
may be accessed if you need it.

Well, that should give you a sense of the
new design and keyboard functions of the
B-Series. From here we will go on to ex-
plore the Extended BASIC commands that
are available.

EXTENDED BASIC

There have been a number of exten-
sions to the BASIC interpreter. Some of
these commands will allow a more struc-
tured program while others will do error
trapping and generally allow easier pro-
gramming. All the BASIC 4.0 commands
are present, so I will just deal with the new
ones or any that have changed. The
following is a list of the new commands
and a quick explanation and example of
their use.

BANK

In the B-Series computers, there are
either two or four 64K memory banks for
the 128 and 256 series respectively. I will
discuss the use of these banks a bit later.
The use of the bank command is to set the
bank for the POKE, PEEK, and SYS com-
mand. For example:

```
10 REM SET BANK NUMBER
20 BANK 3
30 REM STORE DECIMAL 20 AT
   LOCATION 1024 IN BANK 3
40 POKE 1024,20
```

BLOAD/BSAVE

Once again this command is used to
reference the different 64K banks in the
computer. This time they are for loading or
saving a binary file to/from any location in
memory.

```
10 REM LOAD TEST INTO BANK 3
   FROM DRIVE 0 STARTING AT
   MEMORY LOCATION 1024
20 BLOAD "TEST",D0 ON B3, P1024
30 REM SAVE TEST FROM BANK 3,
   DRIVE 1 FROM MEMORY LOCATION
   512 TO 1024
```

40 BSAVE "TEST", D1, B3, P512 TO
P1024

POKE/PEEK

This command has only changed in the
respect that you may now use it to look at
or change locations in other banks. The
default is bank 1, which is the BASIC text
bank, or the bank in which your written
program resides. The following is an ex-
ample showing how to use these com-
mands for other banks.

```
10 REM SET TO BANK 3 AND THEN
   ALTER MEMORY LOCATION 1024
20 BANK 3: POKE 1024,52
30 REM SET TO BANK 3 AND DISPLAY
   THE CONTENTS OF MEMORY
   LOCATION 1024
40 BANK 3: PRINT PEEK (1024)
```

SYS/USR

These commands have not changed
much except that the SYS command is
used in conjunction with the bank com-
mand and the USR command calls a ma-
chine language routine with it's starting
address stored at locations 3 and 4 of
bank 15. The expression in brackets is
stored in the floating point accumulator
prior to entering the subroutine.

```
10 REM CALL A MACHINE LANGUAGE
   ROUTINE IN BANK 3 AT DECIMAL
   ADDRESS 512
20 BANK 3: SYS 512
30 REM CALL A MACHINE LANGUAGE
   ROUTINE ACCORDING TO THE
   ABOVE EXPLANATION
40 X = USR(5)
```

FRE

This command will return the number
of free bytes of the bank or segment
specified by the parameter in brackets. An
invalid parameter returns a 0.

```
10 REM RETURN FREE MEMORY
   IN BANK 3
20 PRINT FRE(3)
30 REM RETURN FREE MEMORY
   IN BANK 1
40 PRINT FRE(1)
```

DELETE

This command will delete a range of line
numbers from the user's BASIC program.
The parameters are the same as those
used in LIST.

```
10 REM DELETE LINES 50 TO 150
   IN OUR PROGRAM
20 DELETE 50-150
30 REM DELETE THE ENTIRE
   PROGRAM
40 DELETE-
```

NOTE: When deleting the entire program,
it actually erases all of memory. It does not

PRINT YOUR HEAD OFF

**INCLUDED
FROM I.C.D. AT
NO CHARGE**

- ALL interfacing and cables
- Tractor AND friction feed
- Upper and lower WITH descenders
- Square print wires for SUPERIOR print quality

**with the
NEW**

COMMODORE 1526 PRINTER!

Actual Print Sample
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
the quick brown fox jumped over the lazy dog.

Print Method:
Serial Impact Dot Matrix

Print Speed:
80 CPS

Print Direction:
Bi-directional (logic seeking) in text mode; uni-directional in bit image graphics mode

PRINT CHARACTERISTICS

- Format:**
- Alphanumeric—9x8 full space character matrix
 - Block character graphics—7x8 dot matrix
 - Bit image graphic—Vertical 7 or 8 dots

Character Set:

- 228 ASCII characters—normal and italic alphanumeric fonts, symbols and semi-graphics

Dot Graphics Density:

- Normal—640 dots/203.2 mm (8 in) line
- Horizontal compressed—1,280 dots/208.2 (8 in) line

Graphic Resolution:

- Single mode—80Hx82V DPI
- Double mode—160Hx82V DPI

Print Styles:

- Pica, italic, pica bold and emphasized

PAPER HANDLING

Line Spacing:

- 1/6 in; programmable in increments of 0.35 mm (1/72 in) and 0.118 mm (1/216 in)

Characters Per Line:

- Normal 80
- Double width 40
- Double width, compressed 71
- Compressed print 142

Circle No. 95

Paper Feed:

- Adjustable sprocket and friction feed

Number Of Copies:

- 3 plus original

PAPER

Types:
Fanfold Roll and Single Sheet

Width: 4 to 10 in

RIBBON

Type: Cartridge
Color: Black Film

Life Expectancy:
Up to 1,000,000 characters

DIMENSIONS

Height: 12.5 cm (4.9 in)
Width: 37.7 cm (14.8 in)
Depth: 32.39 cm (12.75 in)
Weight: 5.3 kg (11 lbs)

just reset the pointers and place three zeros at the start of the program. This means that it is absolutely irreversible.

DIRECTORY

This will display the directory of the desired disk. If the screen is about to scroll, it stops the display and prompts you with (more). Pressing any key except stop will resume the display. The directory command will take an optional filename or a * as a wildcard and search for those occurrences.

10 REM DISPLAY PROGRAM TEST IF IT

EXISTS (DEFAULT DRIVE 0)

20 DIRECTORY "TEST"

30 REM DIPLAY PROGRAMS THAT START WITH A DEFAULT DRIVE 0

40 DIRECTORY "D*"

50 REM DISPLAY BOTH DIRECTORIES

60 DIRECTORY

70 REM DISPLAY DIRECTORY OF DRIVE 0

80 DIRECTORY ,D0

90 REM DISPLAY DIRECTORY OF DRIVE 1

95 DIRECTORY ,D1

INSTR:

This command will match the occurrence of B\$ in A\$ with an optional starting position of N. It will return the starting position or 0 if not found.

10 LET A\$ = "ABCDEF"

20 LET B\$ = "CD"

30 REM THIS WILL RETURN THE NUMBER 3

40 PRINT INSTR(A\$,B\$)

50 REM THIS WILL RETURN THE NUMBER 0

60 PRINT INSTR(A\$,B\$,4)

70 PRINT INSTR("COMMODORE", "MOD")

IF-THEN-ELSE

This command structure will allow us to use more structured statements within our programs. If condition #1 is true then the following commands are executed until a branch or the end of the line is reached. If it is false then the commands following the else will be executed.

10 LET A\$ = "CBM": LET C\$ = "PET"

20 REM THIS WILL PRINT YES

30 IF A\$ = "CBM" THEN PRINT "YES":

ELSE PRINT "NO"

40 REM THIS WILL PRINT NO

50 IF A\$ = "HELLO" THEN PRINT

"YES":ELSE PRINT "NO"

60 REM IF A\$ = C\$ THEN GO TO

LINE 10 OR IF NOT EQUAL GO TO LINE 90

70 IF A\$ = C\$ THEN GOTO 10:ELSE 90

For BEST pricing anywhere, call ...



**INSTITUTIONAL
COMPUTER
DEVELOPMENT
CORP.**

Toll Free 1-800/645-4710

in N.Y. outside cont. U.S. (516) 221-3000
2951 MERRICK ROAD, BELLMORE, N.Y. 11710

All Major Credit Cards Accepted

80 PRINT "THIS LINE WILL NOT BE
EXECUTED"
90 PRINT "A\$ WAS NOT EQUAL TO
C\$"

KEY

This is used to define the function keys that I had previously discussed. The contents of the defined keys may be seen by typing KEY.

10 REM DEFINE KEY 11 TO PRINT
GOSUB WHEN PRESSED
20 KEY 11, "GOSUB"
30 REM DISPLAY ALL DEFINED KEYS
40 KEY

TRAP

This command will disable the BASIC error handling routines and let you handle the error yourself. This command is usually used with the reserved variables EL, ER, ERR\$, and RESUME, which will be discussed shortly.

10 REM SET THE LINE THAT BASIC
WILL GO TO IF AN ERROR OCCURS
20 TRAP 95
30 LET A\$ = ""
40 REM THE NEXT LINE IS AN ERROR
50 PRINT ASC(A\$)
60 REM THE NEXT LINE WILL NOT BE
EXECUTED YET
80 PRINT "WE WILL NOW CONTINUE
SINCE THE ERROR HAS BEEN
TRAPPED"
90 STOP
95 PRINT "OUR PROGRAM CAME
HERE INSTEAD OF PRINTING
?ILLEGAL QUANTITY IN 50
99 GOTO 80

RESUME

This statement is used in conjunction with TRAP. It specifies where execution will continue after an error has been trapped. If no options are specified it will try to re-execute the statement in error. You may use the next option which will cause it to resume execution after the statement in error. If a line number is used, then it will proceed from there.

10 TRAP 60
20 LET A\$ = ""
30 PRINT ASC(A\$)
40 PRINT "THE PROGRAM
CONTINUED AT THIS POINT AFTER
RESUME NEXT"
50 STOP
60 PRINT "THE ERROR WAS TRAPPED
HERE"
70 PRINT "WE WILL NOW GO BACK
TO THE STATEMENT AFTER THE
ONE IN ERROR"
80 RESUME NEXT

DISPOSE

This command will allow a user to terminate a FOR-NEXT loop or jump out of a

The Complete Printer Interface



- User's manual with software examples
- 6' of printer cable
- Selectable device address
- LED indicators for printer fault, paper end, or de-select
- Can be used with full 32 K memory expansion - does not require a user port
- More than 6 modes of operation including:
 - a) normal ASCII-only mode; b) graphics-transparent-mode; c) text-listing mode; d) LPRINT/LLIST mode; e) prints Commodore printer graphic characters on most popular printers; f) built-in screen dump

Circle No. 183

Dealer and Distributor Inquiries Welcome

XETEC XETEC, Inc. • 3010 Arnold Rd. • Salina, Ks. 67401 • (913) 827-0685



COMMODORE 64™ - VIC 20™ EDUCATIONAL SOFTWARE TEACHER DEVELOPED - CLASSROOM TESTED

FOR QUICK MASTERY OF BASIC CONCEPTS.
DESIGNED TO IMPROVE THINKING SKILLS.

All programs self-explanatory. Easy to use.
In homes and schools nationwide.

Diskette or Cassette

REES SOFTWARE LABORATORIES

Subject	Ages	Grades	# Programs
Grammar	10 - up	5 - 10	22
Math	8 - up	3 - 8	17
Phonics	8 - adult	—	18
Fun Pack I	12 - adult	—	4
Fun Pack II	10 - adult	—	3
Fun Pack III	8 - adult	—	3

The philosophy behind all RSL software is that learning should be fun and that having fun should also be intellectually stimulating. In each of the educational programs the material begins with fundamentals and progresses to more advanced. Programs combine the color, graphics and sound capabilities of the computer.

Commodore 64 and VIC 20 are registered trademarks of Commodore.

Dealer Inquiries Invited
Telephone (714) 980-9562

Circle No. 77

REES SOFTWARE LABORATORIES, INC.

Post Office Box 763
Cucamonga, CA 91730

subroutine without leaving garbage on the stack. An ideal use would be during an error trapping routine.

```
10 FOR J = 1 TO 10
15 PRINT J
20 IF J = 5 THEN DISPOSE FOR: GOTO 30
30
25 NEXT J
30 PRINT "OUR LOOP WAS
  TERMINATED PREMATURELY AND
  THE STACK LEFT CLEAN"
35 REM EXAMPLE OF A GOSUB
40 GOSUB 60
45 PRINT "A NORMAL RETURN WAS
  EXECUTED":END
50 PRINT "WE HAVE RETURNED WITH
  A CLEAN STACK WITHOUT A
  RETURN"
55 END
60 INPUT "ENTER A NUMBER";N
65 IF N = 10 THEN DISPOSE GOSUB:
  GOTO 50
70 RETURN
```

ERR/EL/ER

These are the new reserved variables that I had mentioned earlier. ERR\$ will return an error message determined by a parameter. The valid parameters are 0 to 42. When the variables EL and ER are used with TRAP, EL will hold the line number with the error, while ER will contain the correct error number used by ERR\$.

```
10 REM DISPLAY A BASIC MESSAGE
20 PRINT ERR$(1)
30 TRAP 70
40 PRINT ASC(A$)
50 PRINT "WE HAVE RETURNED
  FROM THE ERROR ROUTINE"
60 END
70 PRINT "ERROR IN LINE" EL
80 PRINT "THE ERROR IS" ERR$(ER)
90 RESUME NEXT
```

PRINT USING

This is the command that will probably be most appreciated by the majority of users. It will provide formatted printing to the screen or a logical device. The format symbols that may be used are: #, +, -, ., comma, \$, = and the exponential sign. The way to print to a file would be PRINT[#file], using clause; print list. An example of this would be PRINT#4, using "\$####.##";1000. This will result in the figure \$1000.00 sent to that file. The format clause may be put into a string such as A\$ = "\$####.##". In the following examples an exclamation mark has been used instead of an exponential sign. In practice you should use the exponential sign.

```
10 PRINT USING "+###";1: REM
  PRINT UP TO TWO CHARACTERS
20 PRINT USING "#.##";-.01:PRINT
```

```
SIGN IN THE + POSITION
30 PRINT
USING"#,###.##";1000.009:
  REM ROUNDED UP
40 PRINT USING "$###.##";100.00:
  REM ADD $ AND DECIMAL
50 PRINT USING "$###.##";
  10000.00: REM ADD A COMMA
  FOR EASY READING
60 PRINT USING"###.##! ! !";100000:
  REM PRINT IN SCIENTIFIC NOTATION
70 PRINT USING"##### "; "CBM":
  REM RIGHT JUSTIFIED
80 PRINT USING"#####"; "CBM":
  REM LEFT JUSTIFIED
90 PRINT USING "=#####"; "CBM":
  REM CENTERED
```

PUDEF

This will allow the user to redefine certain symbols in the PRINT USING statements.

```
10 REM ASSIGN PRINT USING
  FORMAT TO A$
20 A$ = "$###.##.##"
30 PRINT USING A$;10000.00
40 REM REDEFINE THE PRINT USING
  STATEMENT
50 PUDEF ".,":REM SWITCHES THE
  COMMA AND THE DECIMAL
60 PRINT USING A$;10000.00
```

DCLEAR

This command will simply initialize the drive specified.

```
10 DCLEAR : REM INITIALIZE
  DRIVE 0 BY DEFAULT
20 DCLEAR D0 : REM INITIALIZE
  DRIVE 0
30 DCLEAR D1 : REM INITIALIZE
  DRIVE 1
```

TI\$

Although this is not a command, the timecheck variable has changed in that it now displays the time up to tenths of a second. It may be set by using seven figures, or just six to make it compatible with an older program.

```
10 TI$ = "01000000": REM NEW
  FORMAT
20 TI$ = "01000000": REM OLD
  FORMAT
```

This winds up the new and revamped commands in the Extended BASIC. Having the use of these new commands will definitely allow for easier and more structured programming.

MACHINE LANGUAGE MONITOR

The new monitor has all the old commands plus six new ones. The syntax of some of the old commands has changed to retrofit the new computer.

TIM is the Terminal Interface Monitor program for MOS Technology's 65XX microprocessors. It has been expanded and adapted to function on the B-Series computers. Execution is transferred from the CBM Basic interpreter to TIM by the SYS command. The monitor is incorporated as part of the Kernal.

Commands typed on the CBM keyboard can direct the TIM to start executing a program, display or modify registers and memory locations, load or save binary data, view other segments, send disk commands or read status, set default disk unit and load and execute programs by entering the program name (Segment 15 only). On modifying memory, TIM does not perform automatic read after write verification to insure that the addressed memory exists and is R/W type.

TIM COMMANDS

M Display memory
: Alter memory
R Display registers
; Alter registers
G Begin execution
L Load
S Save
V View Segment
U Set default disk unit

Get More

From Your

PET/CBM!

NEW! • 24K MEMORY EXPANSION (\$129-\$239)

Give your PET/CBM a boost to 32K!

Loaded with nifty features. Low, low power.

• "Real World" SOFTWARE (\$17 - \$25)

Word Processor, Mailing List, Catalog, Ham Radio, Frequency Counter.

"OLD" 8K PETS

• 2114 - TO - 6550 RAM ADAPTER (\$12 - \$25)

Replace 6550 RAMs with low cost 2114s. *Hundreds Sold!*

• 4K MEMORY EXPANSION (\$16 - \$62)

Low cost memory expansion using 2114s for bigger programs.

Professional Products at Personal Prices

OPTIMIZED DATA SYSTEMS

Dept. O, P.O. Box 595 • Placentia, CA 92670

DISK-O-MATE trademark Optimized Data Systems - PET/CBM trademark Commodore

Write for
FREE
Catalog!

FLY Your Computer



FLIGHT SIMULATOR GAMES

NEW COCKPIT 64
For the Commodore 64

- 100% Machine Language
- Windshield View
- 7 Airports

\$30.

Runway 64 (Commodore 64) **\$25**
Runway 20 (VIC-20) **\$25**
Sky Pilot (VIC-20) **\$18**

ADD \$2.00 FOR DISK VERSION

SUSIE SOFTWARE
709 Wilshire Dr. Mt. Prospect, IL 60056
(312) 394-5165 Circle No. 149

Computa-Law

Legal Agreements*

For Your
COMMODORE 64
VIC 20 (16K)
IBM-PC & (Jr.)

Just answer the questions & your computer & printer does the rest!

- ☐ Simple Will
- ☐ Agreement of Sale - Real Estate
- ☐ Agreement of Sale - Goods
- ☐ Lease - Residential
- ☐ Lease - Commercial
- ☐ Power of Attorney
- ☐ Employment Contract
- ☐ Promissory Note
- ☐ Partnership Agreement
- ☐ Computer Software Contract
- ☐ Computer Hardware Contract
- ☐ Pre-Nuptial Agreement
- ☐ Separation Agreement
- ☐ Construction Contract
- ☐ General Release

Circle No. 122

* For informational purposes only not intended as a substitute for legal advice.

Guaranteed to work on your printer.

- ☐ \$19.95 Each Program (Cassette)
- ☐ \$24.95 Each Program (Disk)
- Add \$1.50 postage & handling.
- 65 Other Business & Home Programs also available

FREE CATALOG

LEGAL BYTE SOFTWARE

Box 579, Gwynedd Valley, PA 19437
(215) 643-7666 (609) 424-5485

- @ Send disk command or get disk status
- X Exit to basic
- Z Transfer to second microprocessor
- <file name> load and execute

SAMPLE USES OF MONITOR COMMANDS

M DISPLAY MEMORY

M 0000 0010
: 0000 0F 0F 4C D9 9A 00 00 00
00 00 00 00 22 22 9E 00
: 0010 00 00 00 00 00 00 00 D4
FB 04 00 04 00 00 C4 FB

In a "display memory" command, the start and ending addresses must be completely specified as 4-digit hex numbers. To alter a memory location, move the cursor up in the display, type the correction and press RETURN to enter the change. When you move the cursor to a line and press RETURN, the colon tells the monitor that you are re-entering data.

R DISPLAY REGISTERS

R
PC IRQ SR AC XR YR SP
: 0007 FBFB B0 DD 71 04 71

The registers are saved and restored upon each entry or exit from the TIM. They may be modified or preloaded as in the display memory example above. The semicolon tells the monitor you are modifying the registers.

G BEGIN EXECUTION

G 0200
The Go command may have an optional address for the target. If none is specified, the PC from the R command is taken as the target.

L LORD

L "filename", 08

No defaults are allowed on a load command. The device number and the file name must be completely specified. Operating system prompts for operator intervention are the same as for BASIC. Memory addresses are loaded as specified in the file header which is set up by the SAVE command. Machine language subroutines may be loaded from BASIC but care must be taken not to use BASIC variables as the variable pointer is set to the last byte loaded + 1. The machine language subroutine will be loaded into the segment that you are currently in as determined by the V command. After the load, the system will be initialized back to segment 15.

S SAVE

S "filename", 08, 010200, 010300

As in the load command, no defaults are allowed in the Save command. The

device number, file name and a six byte start and end address must be given. The above example will save a program to device 8 from segment #1 starting at 0200 hex and ending at 0300 hex. The first two bytes are the segment number followed by the address. Valid segment bytes may be 00 to 0F depending on your memory. After a save, the system will be initialized back to segment 15.

V VIEW

V 01

This will change the segment to the one that you wish to view, save, load, or change memory from. The valid segments are 00 to 0F.

U UNIT ADDRESS

U 09

This command will allow you to set the disk unit default address while you are in the monitor. When leaving, the original address is reset. Valid unit addresses are 8 to 1F. These must be entered in HEX.

@ READ ERROR CHANNEL AND PROCESS DISK COMMANDS

- @ :Display error message and clear channel
- @ S1:filename :Scratch specific file from drive 1
- @ I0 :Initialize disk in drive 0
- @ R0:newname :Rename file on drive 0
= oldname
- @ C1:filename :Copy file from drive 0 to drive 1
= oldname
- @ V0 :Validate or collect disk in drive 0
- @ N1:file-name, id :New or Header disk in drive 1

The above examples use the same syntax as the wedge program supplied with the disk drives.

file name LOAD AND EXECUTE FILE IN SEGMENT 15

This will load a machine language program from the disk and execute it. Its use is restricted to segment 15.

Z TRANSFER TO SECOND MICROPROCESSOR

This command will allow you to utilize the auxiliary processor when applicable.

X EXIT TO BASIC

X

This will cause a warm start to BASIC. In a warm start, memory is not altered in any way and BASIC resumes operation the way it was before the call to the monitor was made.