NO. 60 MAY 1983

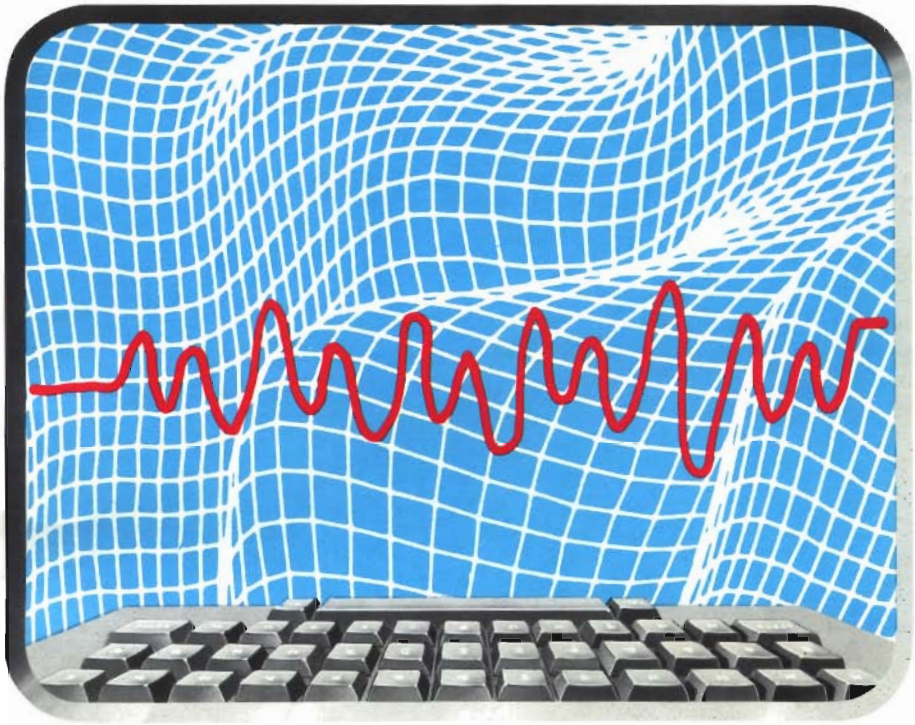
U.S. Edition: $2.50
International Edition: $3.00

MICRO™

Advancing Computer Knowledge

New Wave of Computers

- ATARI 1200XL
- APPLE IIe
- C64...



Color Computer Clock

Graphics on the APPLE and PET

05
0 74470 11690

In this month's Learning Center:

File Management for Commodore Computers

Breakup: An Animation Game for APPLE and Commodore...

GUARANTEED
TOUGH.
LEADING
EDGE.
225 Turnpike Street, Canton, Massachusetts 02021
Toll-free: 1-800-343-6833; in Massachusetts (617) 828-8150. Telex 951-624
Circle No. 83

MAGIC MEMORY

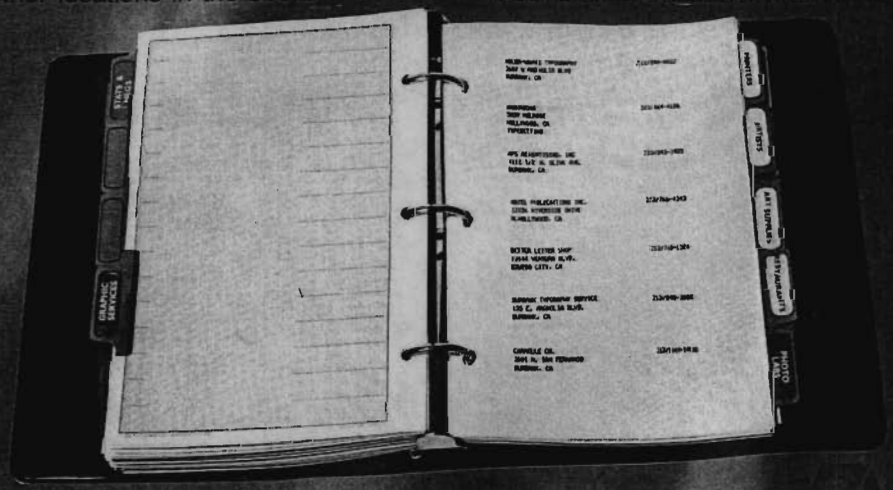
Will Remember Everything For You

Imagine a system that would record all the wonderous, valuable information you have assimilated onto a single tiny disk. (No more scattered bits of paper, business cards, etc.) Imagine the same system giving you a typed sheet you could put into a notebook or print out for a party and instantly change, or add to, at a moments notice. Imagine cross-referencing to suit both your business needs and personal desires so that all your data was organized into one little black book! On top of all this — imagine having fun putting it together.

"Imagine" no more. ARTSCI, INC. proudly announces **MAGIC MEMORY**™, the complete advanced database system. **MAGIC MEMORY** is made to simplify the information storage process.

The typical first-time computer user has no *human* experience that will help him to relate to a *computer's* methods of handling information. Therefore, his learning ability is hampered and remains so for many hours of use, UNLESS the computer has been designed to run simply.

MAGIC MEMORY™ is built for the computer rookie. Everyone can relate to **MAGIC MEMORY**™ because its form is familiar. It looks like an address book but its not. Its *more*. Like the address book, **MAGIC MEMORY**™ presents an A thru Z index tabulation on the right edge of the video display. The user simply selects a tab and the *book* is opened to the proper page(s). A second set of tabs are available that can be labeled by the user (i.e., companies you deal with, birthdays, lists, xmas shops, etc.) Yet **MAGIC MEMORY**™ is **much** more versatile than the old paper address book. This system can instantly add or delete information, sort alphabetically, and transfer data to other locations in the book.



One of the features in the print section of **MAGIC MEMORY**™ is the ability to print out any or all of the *address* book.

nor any of its special se[illegible]s. A "vi[illegible] binder is included in the [illegible]ge for [illegible] address or special sec[illegible]ges. [illegible] are also available.

MAGIC MEMORY s[illegible] presente[illegible] screen will [illegible] any [illegible] of information [illegible] It is only lim[illegible] your need[illegible]

MAGIC MEMORY [illegible] designed to [illegible] and [illegible] commercially. [illegible] APPLE [illegible] The system [illegible] 40 columns [illegible] 80 columns [illegible] 70-column display [illegible] additional hard[illegible]

MAGIC MEMORY will retail for $89.95. Call 213/985-5763. Use your imagination to outwit your organizational wives.

MAGIC MEMORY is a trademark of ARTSCI, INC.

213/985-5763
5547 Satsuma Avenue
North Hollywood, Ca 91601

Circle No. 1

SAGE TECHNICAL BRIEFING 1

SYSTEM DESIGN, SAGE IV

Meet The New Giant Of The Microcomputer Industry. All 4.8 x 12.5 x 16.75" Of It.

The challenge was to create a computer having room for a megabyte of RAM, a built-in Winchester with floppy backup, and the ability to perform 2,000,000 instructions per second.

A small miracle, in other words.

And small is exactly what it turned out to be. In fact, the 16-bit Sage™ IV, including all of the above attributes, takes up less than ½ cubic foot.

What makes such a breakthrough possible? System design.

It took the latest 64K dynamic RAMs and the Motorola 68000 processor technology, plus Winchester technology. And it took a highly integrated, closely packed, low power, high speed design incorporating a proprietary bus.

Now the Sage™ IV is ready for you. Actually, you can choose from three different Sage™ IV models to meet your exact needs—configurations with a 5 megabyte Winchester plus 640K floppy right on up to a combination of four fixed or removable Winchesters plus one or two floppies (200 megabytes of disk capacity in all).

Because of the Sage™ IV's no-compromise system design you can load a 16K program in 1/10 second from Winchester disk.

What's more, there are over 120 sources for existing popular programs for the Sage™ IV. The incredible p-System operating system, standard on every Sage™ IV converts software that was originally written for 8-bit computers in Pascal, BASIC and Fortran. Optionally, CP/M-68K Modula, and Hyper-Forth are also available.

Better yet, our small miracles come with prices to match.

So give us a call or write today for more Sage™ IV information and the name of your nearest dealer.

Western United States
Sage Computer Technology,
35 North Edison Way, #4, Reno,
NV 89502 (702) 322-6868.

Eastern United States
Sage Computer Technology,
15 New England Executive Park
Suite 120, Burlington, MA 01803
(617) 229-6868

In UK
TDI LTD, 29 Alma Vale Road,
Clifton, Bristol BS8-2HL
Tel: (0272) 742796.
Tx: 444 653 Advice G

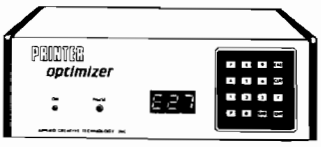
In Germany
MM Computer, GmbH,
Hallwanger Str. 59, 8210 Prien
Tel: 08051/3074
Tx: 525 400 mmco-d

SAGE™ COMPUTER TECHNOLOGY

Sage trademark of Sage Computer Technology
CP/M trademark of Digital Research
Modula trademark of Volition Systems
Hyper-Forth trademark of Forthright Engineering
p-System trademark of UC Regents

Circle No. 4

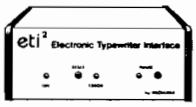
PRINTER OPTIMIZER®



Why buy a mere "spooler" when you can have THE PRINTER OPTIMIZER?

* 64k to 256k spooling buffer
* adapts different brands - can mix Serial and Parallel
* character conversion: 1 to 1, 1 to many, many to 1, ignore
* pushbutton automated access to your printer's various type styles and printing modes
* several ways to PAUSE printing
* access any character, graphic design or printer "trick" from any program at any time
* special features for use as a MODEM buffer

ETI²®



The cost effective alternative. Converts IBM, Adler/Royal, Olympia and other Electronic Typewriters into letter quality printers.

* 2K memory buffer
* access all typewriter characters and automated features
* Serial or Parallel versions
* Many proprietary features and commands insure compatibility with your system and software
* Typesetting capability!

PETI²®

Super low cost adaptor for new inexpensive portable Electronic Typewriters can yield typewriter and letter quality printer combination for around $500 total!

* perfect for "personal use"
* easy "plug-in" Parallel connection to most computers
* compatible with popular word processing programs

APPLIED CREATIVE TECHNOLOGY INC.

2723 Avenue E East, Suite 717
Arlington, Texas 76011
(817)-261-6905
(800)-433-5373

Circle No. 3

MICRO™ Highlights

The New Wave of Computers

"Great waves looked over others coming in...." Robert Frost was speaking of the Pacific Ocean; we are speaking of the many new computer products that are flooding the market. A few years ago it was a simple matter to go out and buy a home computer — you didn't have much selection. Today you must choose from computers that fit nearly any situation, any pocketbook, and any kind of computerist — from novice to highly technical.

Once you have decided which computer is right for you (taking into consideration memory requirements, load and save capabilities, editing functions, etc.), then you should become knowledgeable about add-on boards, modems, printers, hardware, and software that is available. The May issue of MICRO is designed to provide you with information about the tide of new products flowing into the market today and what you can expect in the near future.

Paul Swanson opens our feature section with a comparison of the Atari 1200XL, 400, and 800 in "The Atari 1200XL" (pg. 20). In "Microprocessors for Your Apple II" (pg. 26), Phil Daley and John Hedderman compare add-on boards for the Apple II that increase capabilities, provide new features, and teach new languages. "New Commodores" (pg. 30) by Loren Wright is a discussion of Commodore's new product line, including the VIC-20, Commodore 64, and a computer soon to appear on the market — the C 128/80. Keith Roberson asks "Is 16 Bits the Solution?" (pg. 32). The 16-bit microprocessor has made a significant impact on the imagination of microprocessor users. However, Keith believes 8-bit technology can be expected to have a long and active life in the small computer marketplace.

Programming Aids

This month we provide you with seven programming aids. Werner Kolbe shows you how to write up to four times more numerical data on a CBM disk using the computer's internal binary format instead of ASCII strings. Read "Get More Data on Your CBM Disk" (pg. 38). Randall Hyde discusses several methods for passing data to and from assembly-language subroutines. He uses examples for the 6502, 6809, 68000, and 16032 microprocessors. See "Parameter Passing in Assembly Language" (pg. 40). John Steiner has written a subroutine that adds a real-time clock and date function to your programs. Learn how to use TIME$/SUB in "Real-time Clock for the Color Computer" (pg. 50).

"&GET" by David Dice (pg. 52) is a machine-language program to allow customized input routines; and "Modifying and Using MAE" (pg. 56), by F. Arthur Cochrane, describes how to modify the MAE assembler to enable output to an ASCII printer and listing output to a disk file. Joe Hootman continues his series with a discussion of miscellaneous instructions implemented by the 68000. See "68000 Instructions" (pg. 58). And finally, Jim Strasma continues his series "It's All Relative, Part 5: Printing Commodore File Data" (pg. 62) with a discussion on formatting and printing data from Commodore relative disk files.

Graphics and Printers

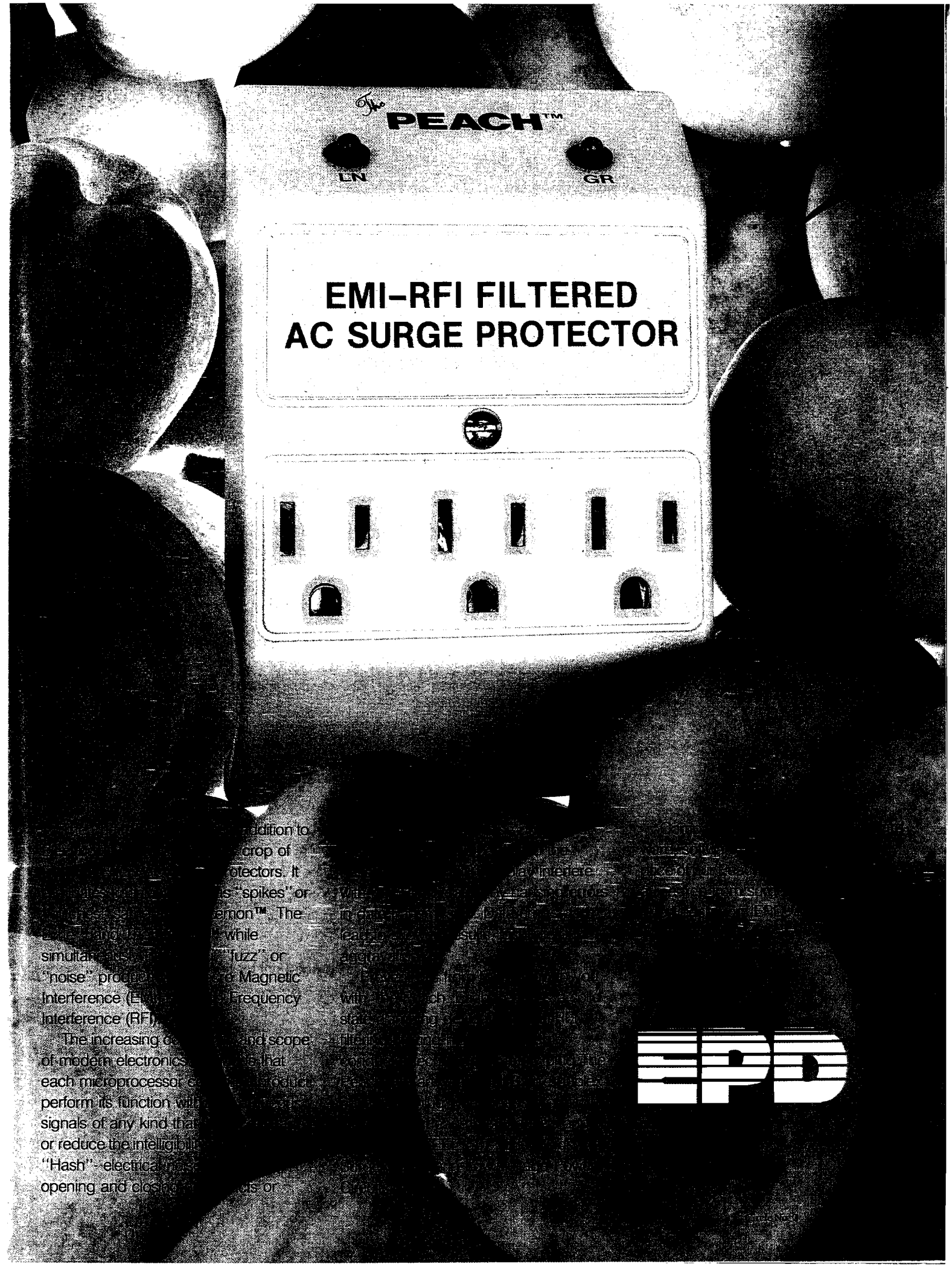
The graphics section includes several informative articles regarding printers. You can learn how to print Apple II's hi-res graphics screens on Okidata Microline 80's printer; use an assembly-language program for the Apple II that automatically produces top and bottom margins on each page of printer output, address a second PET as an IEEE device 4. Also learn to print AIM listings the full width of your printer with a machine-language printer driver, and to interface an OSI Superboard II to a Radio Shack Quick Printer II.

The Learning Center

In the Learning Center this month you will study PEEKs and POKEs in "Breakup" (pg. 71) by Loren Wright and Phil Daley. "Breakup" is a simple animation display game that includes a ball and brick wall and tests for collisions. Brian Zupke's "Addressfile" (pg. 76) is an easy tape-based file-management system for the VIC-20 and other Commodore computers. You can store names, addresses, phone numbers, and memos in a cassette file.

We continue to provide you with a wealth of information in our on-going columns. Be sure to read PET Vet, From Here to Atari, Coco Bits, and Interface Clinic, as well as our regular departments. MICRO keeps you on the crest of the wave of new computers.

MICRO



dition to
crop of
tectors. It
s "spikes" or
emon™. The
while
"fuzz" or
Magnetic
Frequency
Interference (RF)
The increasing
of modern electronics
each microprocessor
perform its function wit
signals of any kind tha
or reduce the intelligibi
"Hash"--electrical
opening and clos
ts or

simultane
"noise" prod
Interference (E
Interference (RF)
and scope
s that
product
interfere
with
in
fea
and
Ex
with
sta
file
con
Re
Ch
Ch
Ch

MICRO™

Advancing Computer Knowledge

MICRO
P.O. Box 6502
Amherst, NH 03031
(603) 889-4330

Managing Editor
Marjorie Morse
Technical Editors
Phil Daley
Loren Wright
Programming
John Hedderman
Editorial
Emmalyn H. Bentley
Maureen Dube

Contributing Editors
Cornelis Bongers
Dave Malmberg
John Steiner
Jim Strasma
Paul Swanson
Richard Vile

Production
Helen Bergeron
Advertising
Sales Manager—Bob Mackintosh
Paula Kramer
Admin. Assist.—Dawn Blute
Dealer Sales
Sales Manager—Kathie Maloof
Linda Hensdill
Circulation
Carol A. Stark
Cindy Schalk
Accounting
Donna M. Tripp

President
Robert M. Tripp
Publisher
John G. Grow
Associate Publisher/ Circulation Manager
Nancy Lapointe

MICRO is published monthly by: MICRO, Amherst, NH 03031. Second Class postage paid at: Amherst, NH 03031 and additional mailing offices. USPS Publication Number: 483470. ISSN: 0271-9002. Send subscriptions, change of address, USPS Form 3579, requests for back issues and all other fulfillment questions to MICRO, 10 Northern Blvd., P.O. Box 6502, Amherst, NH 03031, or call (603) 889-4330, Telex: 955329 TLX SRVC, 800-227-1617. *Subscription rates (per year):* U.S. $24.00, $42.00 / 2 yr. Foreign surface mail $27.00. Air mail: Europe $42.00; Mexico, Central America, Middle East, North Africa, Central Africa $48.00; South America, South Africa, Far East, Australasia, New Zealand $72.00.
Copyright © 1982 by MICRO.
All Rights Reserved.

New Wave of Computers

20 **The ATARI 1200XL** . . . . . *Paul S. Swanson*
A comparison of the ATARI 1200XL, 800, and 400 computers

24 **Microprocessors for Your APPLE II** . . . . . *Phil Daley/John Hedderman*
A comparison of add-on boards for the APPLE II

30 **COMMODORE New Wave** . . . . . *Loren Wright*
A discussion of COMMODORE's new product line

32 **Is 16 Bits the Solution?** . . . . . *Keith Roberson*
Can 8-bit technology survive in the small computer market?

PROGRAMMING AIDS

38 **Get More Data on Your CBM Disk** . . . . *Werner Kolbe*
Write up to four times more numerical data on disk using binary format

40 **Parameter Passing in Assembly Language, Part I** . . . . . *Randall Hyde*
Pass data to and from assembly-language subroutines

50 **Real-time Clock for the Color Computer** . . . . . *John Steiner*
Add a real-time clock and date function to your programs

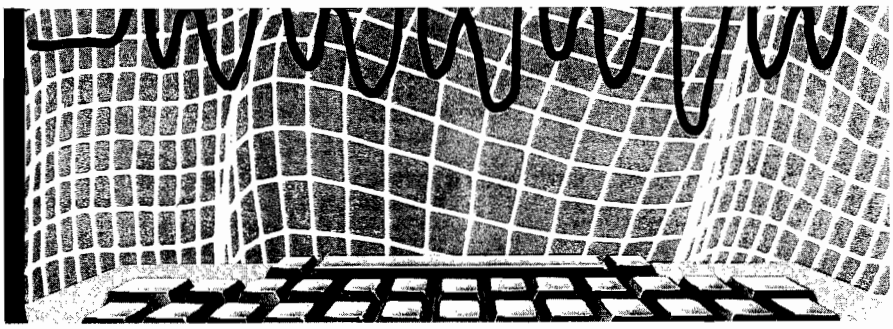
52 **&GET on the APPLE** . . . . . *David Dice*
A machine-language program to allow customized input routines

56 **Modifying and Using MAE on the PET** . . . . . *F. Arthur Cochrane*
Modifications allow output to an ASCII printer and listing output to a disk file

58 **68000 Instructions** . . . . . *Joe Hootman*
Miscellaneous instructions implemented on the 68000

61 **It's All Relative, Part 5: Printing COMMODORE File Data** . . . . . *Jim Strasma*
How to format and print data from COMMODORE relative disk files

New computers change the nature of the microcomputer industry.



THE LEARNING CENTER

73 **Breakup** ..... *Loren Wright/Phil Daley*
A simple animation game that demonstrates PEEKs and POKEs

76 **Addressfile** ..... *Brian Zupke*
A file-management system for the VIC-20 and other COMMODORE computers

GRAPHICS

80 **APPLE Graphics for Okidata Microline 80: Revisited** ..... *Gary Little*
Print APPLE II hi-res graphics screens on an Okidata Microline 80 printer

84 **A Universal Printer Pager for APPLE** ..... *C. Anthony Ray*
Automatically produce top and bottom margins on each page of printer output

87 **Second PET as Soft Printer** ..... *Hans Hoogstraat*
Address a second PET as IEEE device 4

91 **MULTIC — A Multi-Column Print for AIM** ..... *Joel Swank*
Print AIM listings the full width of your printer

99 **Superboard II/Quick Printer II** ..... *Ken Benson*
An inexpensive way to interface the OSI Superboard II to a RS Quick Printer II

COLUMNS

12 **PET VET** ..... *Loren Wright*
LOAD-and-RUN programming

16 **From Here to Atari** ..... *Paul S. Swanson*
Drawing geometric shapes

17 **CoCo Bits** ..... *John Steiner*
A look at the TDP 100, modems, and service manuals

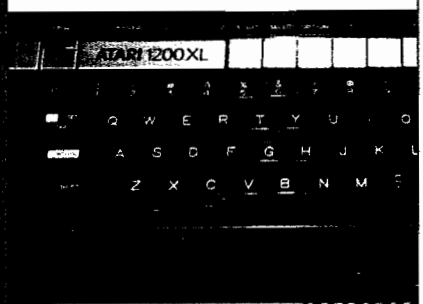
102 **Interface Clinic** ..... *Ralph Tenny*
Experiments in interfacing

DEPARTMENTS

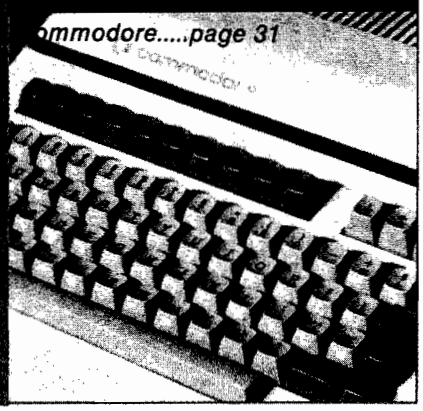
2 May Highlights
6 Letters/Updates/Microbes
10 Editorial
105 Annual Index
110 Reviews in Brief
115 Software Catalog
119 Data Sheet
122 Hardware Catalog
124 6809 Bibliography
127 Next Month in MICRO
128 Advertiser's Index

Apple II.....page 24

Atari 1200XL.....page 21



Commodore.....page 31



NEW LOWER PRICES

ASHTON TATE
dBASE II $489.00
BOTTOM LINE STRATEGIST 279.00

C.ITOH
PROWRITER PARALLEL 399.00
F-10 55 . . . . . 1799.00
F-10 PARRALLEL/SERIAL 1359.00
GRAPPLER INTERFACE 125.00
PROWRITER II 789.00

CALIFORNIA COMPUTER SYSTEMS
ASYNCHRONOUS INTERFACE $129.00
SYNCHRONOUS INTERFACE 149.00
CALENDAR CLOCK 105.00
RS232 INTERFACE 124.00
PROGRAMMABLE TIMER FOR APPLE...99.00

COMSHARE TARGET MARKETING
PLANNER CALC $79.00
TARGET FINANCIAL MODELING . . . 249.00

CONTINENTAL SOFTWARE
HOME ACCOUNTANT FOR APPLE...$59.00
HOME ACCOUNTANT FOR IBM 109.00

DICTRONICS, INC.
RANDOM HOUSE ELECT. THESAURUS$129.00
PROOF READER 50.00

EAGLE
MONEY DECISIONS $99.00

FORCE II
MATH ★ $89.00

FOX & GELLER
dUTIL $68.00
QUICKCODE FOR dBASE II 219.00
QUICKSCREEN 129.00

CALL TOLL•FREE
1•800•523•9511
IN PENNSYLVANIA
1•215•868•8219

T/MAKER COMPANY
T/MAKER III $199.00

VIDEX
ENHANCER II 129.00
VIDEX KEYBOARD ENHANCER 129.00
VIDEX VIDEOTERM FOR APPLE II . . . 269.00

VISICORP
DESKTOP PLAN APPLE II $179.00
DESKTOP PLAN IBM 228.00
VISICALC 179.00
VISICALC ADVANCED VERSION . . . . . CALL
VISICALC BUSINESS FORECASTING . 89.00
VISIFILE APPLE II/IBM . . . . . CALL
VISILINK 184.00
VISIPLOT FOR APPLE 159.00
VISITREND/VISIPLOT 228.00

IBM IS A TRADEMARK OF INTERNATIONAL BUSINESS MACHINE, APPLE IS A REGISTERED TRADEMARK OF APPLE COMPUTERS, INC., CROSSTALK IS A TRADEMARK OF MICROSTUFF. INFOSTAR IS A TRADEMARK OF MICROPRO. PRICES, SPECIFICATIONS AND AVAILIBILITY SUBJECT TO CHANGE WITHOUT NOTICE. NOT ALL PROGRAMS AVAILABLE IN ALL FORMATS. PLEASE CALL FOR ADDITIONAL PRODUCT INFORMATION.

MICROHOUSE
P.O. Box 499/1444 Linden Street
Department 302
Bethlehem, PA 18016
Dealers inquiries welcome!

INTEGRAL DATA SYSTEMS
IDS MICROPRISM 480 PRINTER . . . $589.00
IDS PRISM-132 PRINTER 1279.00
IDS PRISM-80 PRINTER 999.00

INTERACTIVE STRUCTURES
PKASSO $135.00

INNOVATIVE SOFTWARE, INC.
TIM III $359.00
GRAPHMAGIC 69.00
MATHEMAGIC 69.00

INFORMATION UNLIMITED SYSTEMS
EASYFILER $349.00
EASYSPELLER 159.00
EASYWRITER II 279.00

MAXELL
FD-1 or FH-1-32 (8" SINGLE SIDED) . . . $39.00
FD-2 (8" DOUBLE SIDED) 49.00
MD-1 or MH-1 (5¼" SINGLE SIDED) 31.00
MD-2 or MH-2 (5¼" DOUBLE SIDED) 48.00

MICROPRO
CALCSTAR $99.00
DATASTAR 165.00
MAILMERGE 149.00
WORDSTAR 279.00
WORDSTAR/MAILMERGE 349.00

MICROSOFT
128K RAM FOR IBM PC $389.00
BASIC COMPILER FOR APPLE II 289.00
128K RAMCARD 389.00
192K RAMCARD 519.00
256K RAMCARD 659.00
64K RAMCARD 259.00
MICROSOFT Z80 PREMIUM PACK . . 489.00
MICROSOFT Z80 SOFTCARD 249.00
MULTIPLAN 199.00
TASC APPLESOFT COMPILER 129.00

NORTH AMERICAN BUSINESS SYSTEMS
THE ANSWER $229.00

PEACHTREE
ACCOUNTS PAYABLE . . . . . CALL
ACCOUNTS RECEIVABLE . . . . . CALL
GENERAL LEDGER . . . . . CALL
INVENTORY . . . . . CALL
PAYROLL . . . . . CALL
SALES INVOICING . . . . . CALL

PERFECT SOFTWARE
PERFECT CALC $139.00
PERFECT FILER 199.00
PERFECT SPELLER 139.00
PERFECT WRITER 199.00

QUADRAM
128K MEMORY EXPANSION $380.00
192K MEMORY EXPANSION 475.00
64K MEMORY EXPANSION 280.00
64K MEMOY UPGRADE 129.00
MICROFAZERS ALL MODELS . . . . . CALL
QUADBOARD 64K 459.00
QUADBOARD 128K 599.00
QUADBOARD 192K 699.00
QUADBOARD 256K 779.00

CONTROLLER FOR ELITE I $99.00
RANA ELITE I 299.00
RANA ELITE II 465.00
RANA ELITE III 599.00

SMITH-CORONA
SMITH-CORONA TP-1 . . . . . $599.00

SORCIM
SUPERCALC BY SORCIM $219.00
SPELLGUARD 189.00

MICROHOUSE™

YOUR MICRO-COMPUTER PEOPLE

Circle No. 4

PUT PRICES IN CHECK

CARTRIDGE RIBBONS FOR

APPLE PRINTERS

NEC 8023A

C. ITOH PROWRITER

$9.95 EA. $107.46 DOZ.

CARTRIDGE RIBBONS FOR

EPSON

MX-80 MX-100

$6.99 EA $11.95 EA

$86.29 EA $129.06 DOZ

MAXELL DISKETTES

5¼" SINGLE SIDE
DUAL DENSITY
MD-1

$29.90 10 PACK

INNOVATIVE CONCEPTS

FLIP'N'FILE

DISC STORAGE BOX
HOLDS UP TO 60 DISKETTES

5¼" 8"

$24.95 EA $29.95 EA

RIBBONS FOR

IDS PRINTERS

	EA	DOZ.
440	$2.77	$29.92
PAPER TIGER	$6.95	$75.06
MICROPRISM	$7.99	$86.29
PRISM	10.95	118.25

DISKETTE STORAGE BOXES

5¼" — BLUE OR BEIGE

$2.49 EA.

DUAL SPOOL RIBBONS FOR

OKIDATA PRINTERS

	EA	DOZ
80, 82, 83 92, 93	$2.77	$29.92
84	$5.99	$64.69

MEMOREX DISKETTES

5¼ SINGLE SIDE - DUAL DENSITY

$24.99 10 PACK

COLOR-CODER

LIBRARY CASE SET
CONTAINS 5 BRIGHT COLORS

5¼" 8"

$19.95 SET OF 5 $23.95 SET OF 5

ANTI-STATIC SPRAY

FULL QUART SIZE
WITH DISPENSER

$6.95 QT.

1 GALLON REFILL $19.95

CARTRIDGE RIBBONS FOR

COMREX

DAISYWRITER 2000

$2.49 EA $26.89 DOZ

LABEL SPECIAL

$2.99 /K
(5K MIN)

1 ACROSS 3 x 15/16 CONTINUOUS LABELS

MOST RIBBONS AVAILABLE IN COLORS TOO!
CALL OR WRITE FOR OUR SUPPLIES CATALOGUE
ON ORDERS UNDER $14.00 PLEASE ADD $3.00 FOR SHIPPING
MINIMUM RIBBON ORDER $30.00 OR 1 DOZEN

MasterCard VISA

Check-Mate™

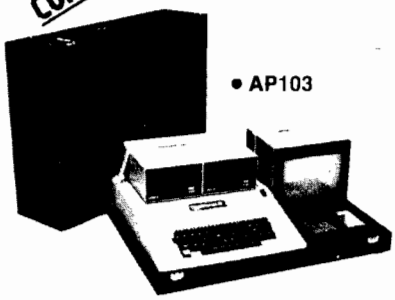
TOLL FREE 800-343-7706
IN MASS 617-963-7694
PHONES OPEN 9AM-7PM EASTERN TIME

51 DIAUTO DR P O BOX 103
RANDOLPH, MA 02368

MASS RESIDENTS
ADD 5% SALES TAX

Circle No. 6

computer case company



Attache-style cases for carrying and protecting your complete computer set-up. Accommodates equipment in a fully operational configuration. Never a need to remove equipment from case. Simply remove lid, connect power, and operate.

AP101	Apple II with Single Drive	$109
AP102	Apple II with Two Disk Drives	119
AP103	Apple II, 9 Inch Monitor & Two Drives	129
AP104	Apple III, Two Drives & Silentype Printer	139
AP105	13" Monitor with Accessories	99
AP106	AMDEK Color Monitor	119
RS201	TRS-80 Model I, Expansion Unit & Drives	109
RS204	TRS-80 Model III	129
AT301	ATARI Computers with Peripherals	109
P402	Centronics 730/737 & Radio Shack Printer	89
P403	Epson MX70/80 or Microline 82A	89
P404	Epson MX100 Printer	99
P405	IDS 560 or Prism 132 Printer	109
P406	Starwriter/Printmaster F-10 Printer	119
P407	Okidata Microline 83A or 84 Printer	99
P408	Prowriter 2 Printer	99
P409	Prowriter (Apple Dot Matrix) Printer	89
IB501	IBM Personal Computer	129
IB502	IBM Monitor	99
HP601	HP41 with Accessories	99
CM703	Commodore Model 64 with Drives	119
CM704	Commodore Model 64 with Dataset	109
NS010	North Star Advantage	139
CC80	Matching Attache Case (5")	85
CC90	Matching Attache Case (3")	75
CC91	Matching Accessory Case	95
CC92	5.25" Diskette Case	49

computer case company

5650 Indian Mound Court
Columbus, Ohio 43213
(614) 868-9464

CALL TOLL FREE
800-848-7548

Circle No. 5

MICRO™

Letterbox

Dear Editor:

Mr. Tripp and Mr. Daley did a fine job of outlining two strongly contrasting viewpoints regarding BASIC in the February 1983 dual editorial. I would like to express my complete agreement with both viewpoints, with one reservation.

Mr. Tripp's critique of BASIC is not a critique of BASIC at all. It is a critique of a particular implementation of BASIC, most likely Microsoft BASIC. Every complaint he outlines can be fixed in a properly implemented BASIC, and some of the complaints *have* been fixed in some implementations.

For instance, I took delivery of my personal Wang 2200 in November 1972. Wang 2200 BASIC does indeed make effective use of disk facilities (at first only hard disks were supported because floppy disks had not yet been invented!). Since I also own a CBM 8032/8050 combination and several Apple IIs it pains me considerably when I contemplate the exceptionally poor DOSes these machines have.

The Wang also has superb string-handling facilities and has the ability to translate back and forth between decimal and hex. There is never any garbage collection with Wang strings, and binary object code can be loaded and manipulated as string arrays since both formatted and unformatted string arrays are allowed.

Mr. Tripp's point 4 (the large program problem with line numbers used as labels for GOSUBs and GOTOs) has, I understand, been licked in some extant BASIC interpreters. In any event, it is clear that that problem can be licked, as well as his points 1 and 2. Now that Mr. Tripp has brought the matter to my attention, it is clear that a good BASIC should support cursor commands such as CURSOR (R,C) to place the cursor, READ CURSOR TO (R,C), and even READ CURSOR LAST TO (R,C). The meaning of the first two statements is (I hope) evident, and the last means 'where was the cursor when the last CR/LF was input?'. All of these can be readily implemented.

This brings us to Mr. Tripp's sixth and final point — the slowness of the interpretation process. All of us know (presumably) about the use of BASIC compilers to fix this problem and also about the inconvenience of the compilation process. However, there are two ways to cure the problem, which do not involve a separate and distinct compilation. The obvious one is to use a newer, faster processor chip than the 6502 or 6809 with which MICRO readers are familiar.

The second method is to write a BASIC in which the stored program is essentially a compiled BASIC but where tables of variable names, tables of labels and such, are used along with an interpreter during the editing phase only, which makes the operator believe that the BASIC is fully interactive.

Even now, in the industry-standard Microsoft BASIC, a keyword such as GOSUB is not stored as ASCII "G, O, S, U, B". Instead, those ASCII characters are replaced with a single byte. Then, when listing the program, the computer makes the operator believe that the program does indeed store the GOSUB command as five ASCII characters. My suggestion is that we (considerably) extend this use of the computer to obscure from the operator the exact form of the stored program while persuading the operator during program input and editing that the stored form is conventional.

All of the complaints raised by Mr. Tripp are valid in some BASICs. Some of those complaints have already been fixed and all of them can be fixed. What we need is a properly written BASIC.

Hal W. Hardenbergh
Digital Acoustics, Inc.
1415 E. McFadden, Suite F
Santa Ana, CA 92705

Updates and Microbes

MICRO Calc Mistake

Listing 1 in MICRO Calc (58:47) by Loren Wright contained several errors. [CH] indicates the HOME key. The following lines contained errors or omissions. Note that line 115 is a new line.

```
20 CR$=CHR$(13):DL$=
   CHR$(20):RB$="[RVS]
   [OFF]":BL$="[20
   SPACES]":DI$="◆
   [OFF][CL]"
110 S$=S$(LL):IF RIGHT
   $(S$,1)="?"THEN
   PRINT"[RVS]"BL$CR$
   "[CU]";
115 PRINTS$DI$;
180 IFT$="←"THENS$(LL)
   =S$:GOSUB 5000:LL=
   1:GOTO110
2010 IFT$="@"ORT$=CR$
   ORT$="[CD]"ORT$=
   "[CU]"ORT$="←"ORT$
   =DL$ORT$="[CLR]"
   THENRETURN
8510 PRINT"[CLR]";:FOR
   I1=1TONL:S$=S$(II)
8520 PRINT"[YEL][CD]"
   S$"[RVS]"LEFT$(BL$
   ,20-LEN(S$)):NEXT:
   PRINT"[CH][CD]";:
   RETURN READY.
```

Corrections for Apple Micro Calc
61 THEN
65 "SS" instead of "BB"
1000 (Include) POKE 216,0

Model Rocket Line Change

In listing 1 of "Model Rocket Simulalation in BASIC" by David Eagle (56:31), substitute ↑ for the ± in line 120.

Missing Line

I caught an omission in the program BANK from "Discrete Event Simulation in Pascal" in the January issue (56:21). Add the following line: newtime := round(rnexp(1,uariv)) + time; just before the line "schedule ('d',newtime);" in 'service'.

Jeff LaBarre
Santa Monica, CA

Review Correction

The review of WP 6502 Version 1.3a from Dwo Quong Fok Lok Sow (57:99) needs a slight revision. Replace the last sentence in the documentation section with: "The disk is supplied with a number of text files already in place, complete with errors to be corrected with the editing features of WP 6502."

Earl Morris
Midland, MI

MICRO™

At Last! Two new ways to expand the usefulness of your Rockwell AIM or Cubit CPU computer are available for immediate, off-the-shelf delivery.

To discover how two new, state-of-the-art circuit cards from Design Dynamics can expand the use you get from your AIM or Cubit computer by providing bus compatible Analog to Digital Interface and Full Color Graphics, please read on.

Until now, if you needed a complete Analog to Digital Interface or Full Color Graphics display for your AIM or Cubit computer, you had to design and build it yourself.

But today, Design Dynamics fills each need on 4½" x 6½" cards. Just look at the features packed into each card:

A total control interface

The AIM/Cubit-compatible interface has been designed to provide you with a flexible, total control interface which includes A to D functions, D to A functions, a clock with user ports section and a User Prototyping Area. Circuit Card with all functions, $1,100.

Boards with selected functions

Analog to digital input is handled on 16 channels with 12 bit resolution. Maximum conversion time is 35 µsec. per channel. Card with only A to D function, $375.

Digital to analog output provides control of 0 to 10 Volts. 8 µsec. conversion is provided by double-buffered, 4 channel, 12 bit D/A converters. Card with only D to A function, $675.

A 24-hour time of day clock, with independent crystal timebase, includes an alarm mode for scheduling events. Two 16 bit timers, each with 16 bit prescaler and start/stop control can count multiple source pulses.

And, a User Prototyping Area provides power, ground bus and grid area for custom signal conditioning. Clock with Prototyping Area Card, $275.

Start-up software for each circuit card is provided.

Full Color Graphics

Now you can expand your system display from limited alphanumerics to a full color CRT display which includes two graphic modes, multicolor mode and text mode, viewed on your own color CRT.

Design Dynamics Full Color Graphics uses no system RAM, and includes it's own 16K dynamic RAM memory. It provides 35 planes of vertically stacked display, 32 sprites in front of graphic plane and internal anti-collision management.

Graphics I provides pattern graphics in 15 colors, 256 x 192 pixels; while Graphics II offers more complex colors and patterns. The Multicolor mode displays in positions of 64 x 48, with four colors per 8 x 8 pattern. The Text mode pattern plane is broken into 40 x 24 positions for text-only display. Full Color Graphics Card, $175.

Startup software included assists the user in becoming familiar with the extensive capacities of this board.

Full documentation included

A Data Pack which includes full documentation for each board makes it simple to put the A to D Interface and Full-Color Graphics to immediate use. Or, if you need to be sure these boards will fit your applications, you may order the Data Pack separately for only $15 per card.

Motherboard available

A fully buffered Motherboard for system expansion of eight cards allows convenient placement of boards for prototyping. Expansion Motherboard is $175.

Save time, money; call today

Get these AIM and CUBIT function-expanding cards *right now,* or call Jack Schnabel for complete information.

DESIGN DYNAMICS
1830 Soscol Avenue • Napa, California 94559 • (707) 257-6000

Rockwell AIM is a trademark of Rockwell, International • Cubit CPU is a trademark of Cubit, Incorporated

Circle No. 8

FOX 20:™
The magazine for *VIC-20 users
*1982

FOX 20 will provide you, each month, with 5 or more ready-to-run programs on cassette-exciting, imaginative, game, educational, and utility programs for most VIC memory configurations. In additon, you will receive Foxtales, an informative Video Newsletter, on Side 2. Novice and pro alike, **FOX 20** is the magazine for you. Be a sly little fox for only $53 a year - $6.50 single issue.

FOX 20 is a division of:
Foxfire Systems, Inc.
3811 Newton
Pasadena, Texas 77503

To order: Send check (U.S. Funds), add 5% Sales Tax for Texas residents, Canada and Overseas $63, orders pre-paid. Dealer and author inquiries invited.

FOX 20:™
The magazine for *VIC-20 users
P. O. BOX 507
Deer Park, Texas 77536
(713) 473-6723

*VIC-20 is a trademark of Commodore Business Machines

Circle No. 9

COMPU SENSE

QUICK BROWN FOX **$60.95**
The #1 word processor!
GENERAL LEDGER **$19.95**
(VIC-20)
CHECK MINDER
VIC-20 **$19.95** C-64 **$24.95**
HOME INVENTORY **$19.95**
(VIC-20)
CENTIPOD **$27.95**
Like Centiped, only better!
FROGEE **$27.95**
The exciting arcade game of Frogger.
MOTOR MOUSE **$29.95**
What a cheese'ee game!
CRIBBAGE
VIC-20 **$14.95** C-64 **$17.95**
This is the game of Cribbage.
STAR TREK
VIC-20 **$12.95** C-64 **$17.95**
Excellent adventure game!
MASTER MIND
VIC-20 **$12.95** C-64 **$19.95**
Makes you think.
ROACH MOTEL **$9.95**
Kill the bugs!
YAHTZEE 1.1 **$12.95**
YAHTZEE 2.1 **$14.95**

TO ORDER:
P. O. BOX 18765
WICHITA, KS 67218
(316) 263-1095

Personal checks accepted
(Allow 3 weeks) or
C.O.D. (Add $2.00)
Handling charges $2.00
VIC-20® is a registered trademark of Commodore

Circle No. 10

MICRO™ Editorial

The New Wave

When we decided to run a feature on the "new wave" of computers almost a year ago, we knew several new systems with different types of configurations would be entering the market in 1983. We believe these computers are an important trend and — because of the low prices — will help draw more people into the world of microcomuters. As discussed in last month's editorial our latest section, The Learning Center, will offer new users programming techniques and applications for these systems.

But we want readers who have been with us over the past few years to know that we are going to continue to support them. Apple and PET owners will still find high-level information and programs. And we'll also offer *advanced* material for VIC-20, TRS-80 Color Computer, Commodore 64 and Atari users, in addition to The Learning Center.

There is no doubt that the nature of the microcomputer world is changing — it's no longer only for the highly technical computerists using complicated and expensive systems. Technology has allowed manufacturers to produce sophisticated yet easy-to-use computers at an inexpensive price. No longer is there the huge gap between strictly game machines and intimidating computers. There are many varieties in the middle. Micro will keep you up to date on future "new wave" systems and how to use them.

MICRO Moves North!

We're still tripping over boxes and packing foam, but everyone agrees that our move to southern New Hampshire was a good idea. Our brand new offices among the pine trees will provide an inspiring setting to continue to mold MICRO into an exciting magazine.

Of course, as with any move, we (and our readers, advertisers, and authors) suffered thorugh quite a bit of confusion. For instance we were in the new building for three days before the phones actually worked. So if you tried to call our old number you were connected with the few souls left in Chelmsford who had no information except our new number — which wasn't working yet! We apologize for any inconveniences.

A big reason for the trek north is that over half the staff lives in southern New Hampshire. Commuting time has been cut considerably, which gives us more time to devote to the magazine. So, as soon as the phone lines are connected and the last box gets unpacked, we'll all be smiling and back to work as usual.

Our new address is: 10 Northern Blvd., Northwood Executive Park
P.O. Box 6502
Amherst, NH 03031
Our new phone number is: (603) 889-4330

Marjorie J. Morse

Marjorie Morse

MICRO™

Lyco Computer Marketing & Consultants

TO ORDER CALL US **TOLL FREE 800-233-8760**
In PA 1-717-398-4079

FREE DUST COVER with Purchase of

ATARI 800 48K $489.00
ATARI 400 64K $349.00
810 DISK DRIVE.....$419.00
ATARI 1200 64K RAM...$CALL $

ATARI®
A Warner Communications Company

ATARI HARDWARE

810 DISK DRIVE	$419.00
410 RECORDER	$75.00
1010 RECORDER	$75.00
850 INTERFACE	$164.00

PACKAGES

CX482 EDUCATOR	$119.00
CX 483 PROGRAMMER	$54.00
CX488 COMMUNICATOR	$219.00
CX419 BOOKEEPER	$189.00
KX7104 ENTERTAINER	$69.00

NEW RELEASES

400 KEYBOARD	$99.00
MINER 2049er	$32.75
FROGGER	$25.75
PREPPIE	$19.75
SEA DRAGON	$24.75
STRATOS	$24.75
DISKY	$39.95
MONKEY WRENCH 2	$52.75

MONITORS

NEC JB1260	$125.00
NEC JB1201	$155.00
NEC TC1201	$315.00
AMDEK 300G	$159.00
AMDEK COLOR I	$329.00

MODEMS

ANCHOR MARK I	$79.00
ANCHOR MARK II	$79.00
HAYES SMART	$239.00
HAYES MICRO II	$309.00
CAT	$144.00
J-CAT	$ CALL $

PERCOM DISK DRIVES

SINGLE DRIVE AT88	$389.00
ADD ON	$289.00
SINGLE DRIVE 40S1	$529.00
ADD ON	$329.00
DUAL DRIVE 40S2	$845.00
DUAL HEAD SINGLE DRIVE 44S1	$649.00
DUAL HEAD DUAL DRIVE 44S2	$789.00

SAVE on these in-stock PRINTERS

PROWRITER.....$375.00
NEC 8023A.....$439.00
SMITH CORONA TP1 ...$569.00

STARWRITER	$1475.00
PRINTMASTER	$1675.00
OKIDATA 82A	$419.00
OKIDATA 83A	$639.00
OKIDATA 84	$1029.00
OKIDATA TRACTOR	$63.00

PRINTER CABLES for Atari

CITOH	$35.00
EPSON	$35.00
NEC	$35.00
OKIDATA	$35.00
SMITH CORONA	$35.00

DISKETTES : In Stock

BASF	$19.00
ELEPHANT	$21.00
MAXELL MDI	$34.00
MAXELL MDII	$44.00

BUSINESS SOFTWARE

VISICALC	$159.75
LETTER PERFECT	$115.75
LETTER PERFECT ...ROM	$159.75
DATA PERFECT	$75.75
TEXT WIZZARD	$79.75
SPELL WIZZARD	$64.75
FILE MANAGER 800+	$69.75
ATARI WORD PRO	$109.75

POLICY DURING APRIL

In-Stock items shipped within 24 hours of order. Personal checks require four weeks clearance before shipping. No deposit for COD orders. PA residents add sales tax. All products subject to availability and price change. Advertised prices show 4% discount offered for cash. Add 4% for Mastercard and Visa.

Circle No. 11

TO ORDER CALL TOLL FREE
800-233-8760
In PA 1-717-398-4079
or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740

NAME BRAND **SUPER SALE**

Bulk Diskettes* with envelopes

*Now Get High Quality at a Low Price Manufactured by a Major Disc Company For MDS Without Their Name on Diskettes *Minimum order 20 diskettes with Tyvek envelope and storage shipping box *Quantity Discounts 100 deduct 3%, 1000 deduct 5%, 10 000 deduct 10% 100% Certified 1 Year Warranty

5¼" Soft Sectors

SINGLE SIDE SINGLE DENSITY W/HUB RING	$1.69*
SINGLE SIDE DOUBLE DENSITY W/HUB RING	$1.99*
DOUBLE SIDE DOUBLE DENSITY W/HUB RING	$2.79*

8" Soft or 32 Sectors

SINGLE SIDE SINGLE DENSITY	$1.79*
SINGLE SIDE DOUBLE DENSITY	$2.29*
DOUBLE SIDE DOUBLE DENSITY	$3.09*

PRINTERS

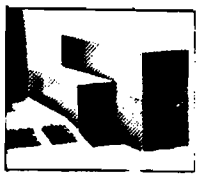
All EPSONS available	$call
GEMINI 10 by Star Micronics	$359.00
GEMINI 15 by Star Micronics	$549.00
Okidata Microline 80	$call
Okidata Microline 82	$469.00
Okidata Microline 83A	$call
Okidata Microline 84	$call

MICROBUFFER

MBP-16K Parallel	$149.00
MICROBUFFER IN-LINE For Most Printers	
32K Parallel	$289.00
64K Parallel	$339.00
32K Serial	$289.00
64K Serial	$339.00
64K Memory Expansion Modules	$169.00

TANDON DISK DRIVE ENCLOSURES

Complete with Chassis & Power Supply: Fully assembled silver or beige chassis with external card edge connector for easy cable installation for 5¼" drives. With MDS 120 days warranty $59.00

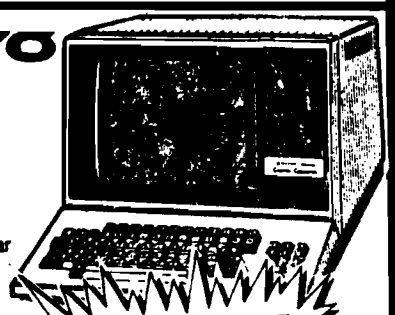


BARE DRIVES

TM100 1 40 Trk	$199.00
TM100 2 40/40 Trk	$269.00
TM100 4 80/80 Trk	$339.00
SIEMANS FDD100 8 SS/DD 8 in	$279.00
TM50 SS DD 40 Trk Thinline	$199.00
TM84B 1 SS/DD 8" 77 Trk Thinline	$369.00
TM84B 2 DD/DD 8" 77 Trk Thinline	$479.00

Add $59.00 For Complete 5¼" Drive System

SANYO



MCB 1000 High Performance Business System

Includes FREE MicroPro Software: • WordStar • SpellStar • MailMerge • CalcStar • And More • Z-80A C.P.U.
- 64KB RAM
- 328KB mini floppy disk drive
- CP/M operating system
- 12" non-glare green phosphor video display screen
- Centronics parallel printer port
- RS232C serial port
- Additional disk drives up to 2.3MB

$1595

SAVE $2000*

(* If purchased separately)

CASH ORDERS ONLY

Free shipping on all prepaid orders over $100.00 in U.S.A.

MDS MICRO DATA SUPPLIES

22295 EUCLID AVE.
EUCLID, OHIO 44117

Call (216) 481-1600

WE ACCEPT
- Visa
- MasterCard
- Checks
- Money Order
- C.O.D.

ALL PRICES ARE FOR MAIL ORDER ONLY. Prices, Specifications and Offerings subject to change without notice

ADD $3.00 FOR SHIPPING & HANDLING. $6.00 Extra for C.O.D. Orders. Ohio Residents add 6.5% Sales Tax

DEALER INQUIRIES WELCOME

Circle No. 12

MICRO™

PET Vet

Loren Wright

One convenience offered by Commodore computers is the load-and-run function of the shifted RUN/STOP key. With BASIC 4 PETs and CBMs, pressing the key loads the first program on drive 0 of the disk and immediately RUNs it. With the VIC, Commodore 64, and pre-BASIC 4 PETs, pressing the key causes the next program on tape to LOAD and immediately RUN. Doing this from within a program is a different story, though.

There are many applications for LOAD-and-RUN programming, and even more ways to accomplish it. You can load one program from another without destroying the variables defined in the first. This is a technique described by Jim Strasma in Part 1 of his "It's All Relative" series (MICRO 55:37, December '82) under the topic of chaining. Strasma's application was a mailing list program that used one start-up module to define a number of constants, with some based on the current equipment configuration. This way he could avoid redefining these in every new module. The technique also allowed him to pass parameters from one module to another. For more details see Strasma's article.

It is also possible to perform this kind of chaining without changing the values of pointers if you make sure the first program is always longer than the second. If this isn't the case, then add some extra REM statements at the end of the first program. This technique is essential if you have a program that is too long to fit into memory at once. Just find a convenient place to split it, make the first part longer than the first, and include an appropriate LOAD statement for the second part. Variables do not have to be redefined and are passed intact to the second program.

We now have techniques for chaining programs without destroying variables. However, there are situations when neither of these techniques will do the job. One is when you want to load a machine-language program at the top of memory, followed by a BASIC driver. Another is when you need to do a NEW before running the second program. In both cases we need the pointers set to their default values, but the programmed LOAD leaves them as they were for the original program. Also, since a NEW causes the program in memory to disappear, you can't very well LOAD a new program if the LOAD instruction no longer exists! *(Continued on page 14)*

PET Vet Listing

```
10 Q$=CHR$(34): NC=158: KB=623:
   DN$=",8": PO$="■■■"
20 PRINT"■": FORI=1TO3: READNA$(I
   ): PRINT NA$(I)"■": NEXT
30 INPUT"■INPUT PROGRAM #";N
40 PRINT"■NEW"
50 PRINT"■■LOAD"Q$NA$(N)Q$DN$
60 PRINTPO$"RUN"
70 POKE NC,4: POKE KB,19: FORI=1
   TO3: POKE KB+I,13: NEXT
1000 DATA PROGRAM 1,PROGRAM 2,
    PROGRAM 3
```

SJB DISTRIBUTORS. THE MOST COMPETITIVE PRICES ON COMMODORE.

commodore

NEW COMMODORE PRODUCTS

CBM 64	Call
CBM B500	$ 695
CBM B700	2990
CBM 1520 Plotter	259
CBM 1701 Color Monitor	279
B Series Software	Call

SOFTWARE FOR CBM 64

Word Processing (WordPro 3+)	$ 69
Word-Pac (tape)	60
The Assistant Series	
Writer's Assistant (easy and flexible)	99
File Assistant (database with merge)	99
Spreadsheet Assistant	99
Pers. Finance Assist. (great reports)	45
Busicalc (Spreadsheet)	55
Coco II (build your own games easily)	45
Home Accounting Package	39
General Ledger, A/R, A/P (with check writing)	ea. 139
CBM EasyFinance	50
CBM EasyScript	80
CBM EasyFile	80
Data Manager	70
Stock (investment analysis)	80
Pet Emulator (emulates 4.0 basic)	30
Sprite-Magic (use joystick to design sprites)	19
Assembler Package (cassette or disk, compiled, includes editor, loader, disassembler)	39
Spacebelt	20
Retroball	34

INTERFACES & ACCESSORIES

80 Column Expander	$159
VIC 1600 Modem	95
VIC 1650 (auto answer, auto dial)	150
VIC 1525 Graphic Printer	329
VIC 1530 Datasette Recorder	65
VIC 1541 Disk Drive	329
VIC Switch (connect 8 64's or Vics to printer, dd)	149
IEEE Interface (64)	85
PET-IEEE cable	33
IEEE-IEEE cable (2m)	43
Parallel Interface (Epson, Okidata, IDS, NEC)	80
RS-232 Printer Interface (Okidata, Diablo, etc.)	60
Programmers Reference Guide	18
Verbatim Diskettes (10 per box)	26
Victree (Programmers Utility)	75

VIC PRODUCTS & ACCESSORIES

8K RAM Memory Expansion Cartridge	$ 40
16K RAM	70
24K RAM	105
VIC IEEE Interface	75
VIC 3 Slot Expander	27
VIC 6 Slot Expander	70
RS-232 Printer Interface	65
Cassette Interface	27
Home Finance Package (6 tapes)	47
Gorf (64 also)	30
Omega Race	30
Arcade Joystick - Heavy duty w/2 firing buttons! Great for the VIC or 64	25

MONITORS - GREAT RESOLUTION (64 OR VIC)

Amdek Color I	$ 299
Amdek II or III	call
Panasonic CT160	279
Comrex 6500 - 13" Color	299
Transtar 20 (High Resolution Green Phosphor)	129
Video/Audio Cable	15

PRINTERS - LETTER QUALITY

CBM 8300, 40 cps	$1450
Diablo 620, 25 cps	949
ComRiter, 17 cps	819
Transtar 130, 16 cps (auto load, wp features!)	769
NEC 7700 series	2350
NEC 3500 series	1600

PRINTERS - DOT MATRIX

CBM 8023, 150 cps/graphics	589
Epson FX Printer, 160 cps	529
Epson MX-80 w/Graftrax	349
CBM Graphics for Epson	65
Okidata 82A, 120 cps (serial and parallel)	429
NEC 8023A (parallel)	439
Okidata 92	559
Star Gemini, 10	360
Star Gemini, 15	499

COMMODORE BUSINESS SERIES

SuperPet (5 languages, 2 processors)	$1409
CBM 8032 Computer, 80 Column	1029
CBM Memory Expansion, 64K	359
CBM 8050, 1 mg. Dual Drive	1259
CBM 8250, 2 mg. Dual Drive	1500
CBM D9060, 5 mg. Hard Disk	2240
CBM D9090, 7.5 mg. Hard Disk	2600
CBM 2031, 170K Single Drive (New)	489
DC Hayes Smart Modem	220

BUSINESS SOFTWARE

WordPro 4+ or 5+	$ 309
Administrator	489
VisiCalc (expanded)	199
The Manager (database)	199
BPI A/R, G/L, Job Cost, Inventory, Payroll	ea. 325

MasterCard, Visa, Money Order, Bank Check

COD (add $5) accepted.
Add 3% surcharge for credit cards.
In stock items shipped within 48 hours,
F.O.B, Dallas, Texas
All products shipped with manufacturer's warranty.

Prices are subject to change without notice.

TO ORDER CALL TOLL FREE
800-527-4893
800-442-1048
(Within Texas)

Business Hours
Mon.- Fri. 8 to 6, Sat. 10-2

Write for free catalog.

GAME OF THE MONTH

Adventu-Writer (make your own adventure game)	39

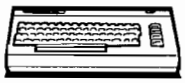
SJB DISTRIBUTORS INC.
10520 Plano Road, Suite 206
Dallas, Texas 75238
(214) 343-1328

Circle No. 13

COMMANDER

THE MONTHLY JOURNAL FOR COMMODORE COMPUTER USERS

SUPER PET



VIC-20

PET/CBM

GET YOUR MONEY'S WORTH

You've probably made a sizeable investment in your computer equipment. **COMMANDER** can help you make the most of it. Each issue brings you the *no-nonsense* advice you need to stay on the leading edge of this constantly changing field. **COMMANDER** will be your reference to the world of computers . . . with the best, most comprehensive coverage you can get!!

Subscribe now and take advantage of our limited offer of $4 off the one year subscription.

☐ 1 YR. U.S. $22☐ 2 YR. U.S. $40☐ 3 YR. U.S. $58

(Prices do not include $4 Discount)Prices in U.S. Funds

Washington residents please add 7.8% sales tax

—Subscription Orders Only—

Toll Free Number: 1-800-426-1830

(except WA, HI, AK)

COMMANDER

P.O. BOX 98827

TACOMA, WASHINGTON 98498

(206) 565-6816

Circle No. 14

MICROSPEC

Quit Playing Games . . .

Disk Based Software to Make Your Computer Get Down to Business

Disk Data Manager—Create and manage your own data base. Allows you to create, add, change, delete, search, sort, print, etc. Up to 1200 records on a single disk.
VIC 20. . . 59.95 CBM 64 . . . **89.95**

Payroll System—Full featured, complete payroll system. Even prints checks.
VIC 20. . . 89.95 CBM 64 . . . 99.95

Mailing List—Up to 1200 records on a single disk. Presorts by Zip Code. Prints on stock up to four labels wide.
VIC 20. . . 44.95 CBM 64 . . . 54.95

Inventory Package—Maintains quantity on hand, cost, sales price, reorder point, etc. Generates suggested reorder, sales report, and sales analysis.
VIC 20. . . **89.95** CBM 64 . . . 99.95

General Ledger—Up to 75 accounts! Generates Balance Sheet, Income Statement, Update Report, etc.
VIC 20. . . 89.95 CBM 64 . . . 99.95

Checkbook Manager—Up to 25 expense categories. Tracks all outstanding checks until they are paid.
VIC 20. . . 49.95 CBM 64 . . . 49.95

CONTACT US FOR ALL YOUR DISK BASED SOFTWARE NEEDS

Call for specifics on Hardware Configurations. Send Self-Addressed Stamped Envelope for Catalogue of Games and other Applications

DEALER INQUIRIES WELCOME

VISA

P.O. Box 863085
Plano, Texas 75086
(214) 867-1333

MasterCard

VISA and MASTERCARD Accepted

Circle No. 15

PET VET *(continued)*

Having just finished preparing MICRO's new book "Mastering Your VIC-20" for publication, I wanted to be able to demonstrate the programs at the West Coast Computer Faire. The book comes with a cassette, but obviously that is not the appropriate medium for a demonstration. I needed a menu program that would display a list of the programs on the screen and automatically LOAD and RUN any program selected. I tried making my menu program longer than the longest program by adding REMs at the end, but that didn't leave enough room to store the variables created by other, shorter programs. Also, several programs involved lowering the top-of-memory pointer to protect RAM-defined character sets. When a longer program is loaded after one of these, there isn't enough memory available. Obviously (since nothing is impossible), a different technique was required. I read about this technique in the early PET literature, but rather than digging through back issues, I decided to rediscover it myself.

Programming the Keyboard Buffer

This technique has a number of applications besides automatic loading, including having a program add lines to itself or write a whole new program! One commonly used variation is to convert machine language into BASIC DATA statements.

The addresses involved are different for the different Commodore machines, so consult the table for the Addresses for your machine. Below is a demonstration program for loading one of three programs. (It is up to you to make sure there actually are programs by those names on the disk!)

The Table

Function	Variable	BASIC 1	BASIC 2/4	VIC/C64
No. of chars. in keyboard buffer	NC	525	158	198
Start of keyboard buffer	KB	527	623	631

Program Description

What the program accomplishes is very simple. First a NEW statement is executed, then the LOAD, and finally a RUN — all as if they were typed in the direct mode. The keybaord buffer holds up to ten keys in sequence, and the PET processes them one at a time in the order they entered the queue. The program puts the commands on the screen in the proper positions. Line 70 puts a 4 in address NC, telling the PET that there are four unprocessed keys left in the buffer. Then four characters, a home and three returns, are put into the first through fourth positions in the buffer. The program is done, but the PET must process those four characters. The effect is as if you typed the commands yourself and then pressed the RETURN key.

Change the values of NC and KB according to your Commodore computer. If you are working with cassette, you are limited to the next program on the tape. To experiment with the automatic load you will also have to change DN$ to ",1" or " " and add three more cursor-down's to PO$.

MICRO™

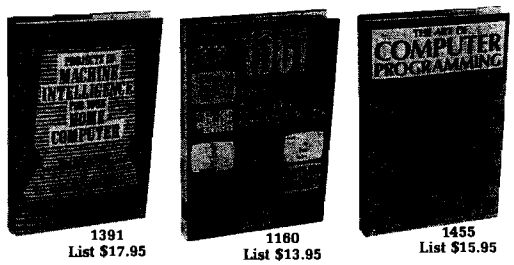
Micro The Computer Book Club®

"The ONLY Book Club for micro users"

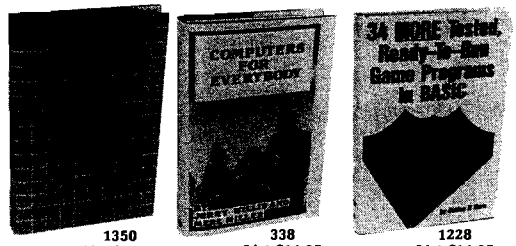
Play games . . . develop new programming skills . . . organize your finances . . . earn extra income . . . even build your own microcomputer . . .

HAVE IT ALL FOR ONLY $2.95

Select any 5 books, (values to $103.75) for only $2.95 when you join The Computer Book Club®!



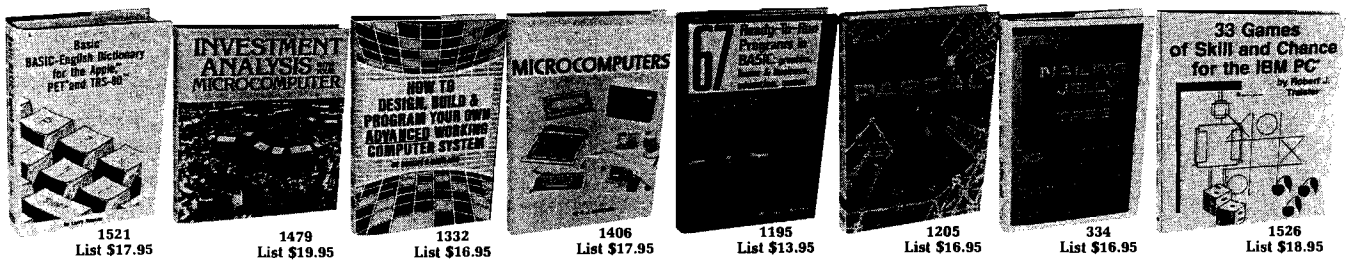
1391 List $17.95 | 1160 List $13.95 | 1455 List $15.95



1350 List $15.95 | 338 List $14.95 | 1228 List $14.95

1407 List $17.95 | 1513 List $19.95 | 337 List $19.95 | 1496 List $17.95 | 1276 List $15.95 | 1108 List $14.95 | 1396 List $13.95 | 1330 List $16.95

1485 List $21.95 | 1295 List $16.95 | 1277 List $19.95 | 335 List $14.95 | 1369 List $21.95 | 1062 List $13.95 | 1183 List $14.95 | 1422 List $14.95



1521 List $17.95 | 1479 List $19.95 | 1332 List $16.95 | 1406 List $17.95 | 1195 List $13.95 | 1205 List $16.95 | 334 List $16.95 | 1526 List $18.95

7 very good reasons to try Micro The Computer Book Club®

Blue Ridge Summit, PA 17214

- **Reduced Member Prices.** Save 20% to 75% on books sure to increase your computer know-how
- **Satisfaction Guaranteed.** All books returnable within 10 days without obligation
- **Club News Bulletins.** All about current selections—mains, alternates, extras—plus bonus offers. Comes 13 times a year with dozens of up-to-the-minute titles you can pick from
- **"Automatic Order."** Do nothing, and the Main selection will be shipped automatically! But . . . if you want an Alternate Selection—or no books at all—we'll follow the instructions you give on the reply form provided with every News Bulletin
- **Continuing Benefits.** Get a Dividend Certificate with every book purchased after fulfilling membership obligation, and qualify for discounts on many other volumes
- **Extra Bonuses.** Take advantage of added-value promotions, plus special discounts; on software, games, and more
- **Exceptional Quality.** All books are first-rate publisher's editions, filled with useful, up-to-the-minute information

Micro The Computer Book Club®

Blue Ridge Summit, PA 17214

Please accept my membership in The Computer Book Club® and send the 5 volumes circled below, billing me $2.95 plus shipping and handling charges. If not satisfied, I may return the books within ten days without obligation and have my membership cancelled. I agree to purchase 4 or more books at reduced Club prices (plus shipping/handling) during the next 12 months, and may resign any time thereafter.

334 335 337 338 1062 1108 1160 1183
1195 1205 1228 1276 1277 1295 1330
1332 1350 1369 1391 1396 1406 1407 1422
1455 1479 1485 1496 1513 1521 1526

Name _____ Phone _____

Address _____

City _____

State _____ Zip _____

Valid for new members only. (Orders outside U.S. or Canada must be prepaid in International Money Orders in U.S. dollars. Canada must remit in U.S. dollars.) This order subject to acceptance by The Computer Book Club®. MCRO-583

Circle No. 16

MICRO™

From Here To Atari

Paul S. Swanson

This month's column is based largely on a letter I received from Jose H. Nieto, a MICRO reader in Venezuela. His letter outlined a few questions that are commonly asked in reference to specific applications. His application included drawing geometric shapes.

Mr. Nieto's first question was about monitors. When drawing shapes in mode 8, you will get color artifacts on any compatible monitor or television because of the mechanics of the signal, although some monitors will reduce the artifacts a bit. However, using the computer for plotting single points to define geometric shapes does not really require color in mode 8. This mode actually is a one-color mode, allowing two luminances. I would prefer to do such work with a good black and white or green phosphor monitor. The image is much clearer than that obtained by either a television or a color monitor.

The Atari 800 computer can be connected to almost any raster scan monitor that uses the same sync signals a television uses. The Atari 400 cannot be connected to a monitor; its only video output is combined with its only audio output and modulated for television use.

The second question in the letter deals with connecting an Atari 800 computer to a stereo amplifier. First, the sound output from the Atari is monaural. You can still use a stereo amplifier, but you will not get stereo sound. The jack on the side of the Atari 800 has connections for the monitor and the sound channel. This jack has five pins on it. The ground pin is the lowest pin on the jack (pin 2). This should be connected to the ground wire of anything you connect to this jack. The "live" connection for sound is pin 3, which is the top pin on the left side as you look at the jack. If you connect both audio input lines to this pin (the "live" from each channel) your stereo amplifier should work just fine. The proper place in your amplifier to connect the Atari output is the auxiliary input or the phonograph input (if there is no auxiliary input).

For color monitors, pin 4 is the proper place to connect the live video wire. Black and white does not require the color information and I have had better results connecting those monitors to pin 1, which is composite luminance. Pin 4 is composite video, which will carry the color burst required for color pictures, but not required for black and white.

Another question in Mr. Nieto's letter refers to the use of a PRINT #6 command with a mode 0 screen. This will normally leave inverse video spaces at the end of each line. If you want to access a mode 0 screen with PRINT #6, don't use a GRAPHICS 0 statement. Instead, declare the mode with the following statement:

```
OPEN #6,0,0,"E:"
```

The screen is opened with a GRAPHICS statement as the screen handler (device S:). Using the screen editor, device E:, will eliminate the white squares. The white squares you get when you use GRAPHICS statements with PRINT#6 is actually the cursor, unreversed after it is moved. Use the device E: and you won't have to worry about it.

Since the white squares left behind are unreversed cursors, there is another simple solution to this. Turn off the cursor right after you declare GRAPHICS 0. This can be done by a POKE 752,1.

When you use GRAPHICS 0 and the POKE 752,1, you can use COLOR, PLOT, and DRAWTO with a mode 0 screen. Try, for example, the following short program:

```
10 GRAPHICS 0
20 POKE 752,1
30 PRINT
40 COLOR ASC("A")
50 PLOT 0,0
60 DRAWTO 10,10
```

Lines 10 and 20 open the screen and shut off the cursor. Line 30 eliminates the single white square that would be left if the PLOT were to be executed without it. The COLOR in mode zero is the character to print. The PLOT and DRAWTO will set a diagonal line comprised of A's on the screen. Eliminate line 30 to see the cursor that it eliminates. Change the COLOR statement for other characters. The control characters will print as characters in this method, so you can declare COLOR 28 in line 40 and get a line of up arrows. Even the ESC character works (COLOR 27). Only RETURN (COLOR 155) gives unusual results.

One last question concerns the graphics available on printers. There is no redefinable character set on the Atari 820 printer and I know of no graphics available on it either. The printer I use is an Epson MX-80 with Graftrax Plus. The character set on that printer is also not programmable, but you can address all of the dots on each line individually in its graphics modes. It is rather simple to write a screen dump program for the Epson, if you dump the screen sideways. If you have an Epson with any Graftrax option, read the graphics section of your Epson manual and you will see how this can be done.

Both of the 800 series Atari 40-column printers also have graphics capabilities. If you do not need the 80-column width, the suggested $299.00 retail price of these printers may be worth considering.

New Hardware

In other news, Atari has a few products to add to their computer line. In addition to the well-publicized 1200, Atari has a new Program Recorder, model number 1010, that is scheduled for first quarter (it should be available by the time you read this). The 1010 is like the 410 except it has a case design more similar to the new look of the Atari 1200. Also reported is that this new recorder has an improved mechanism. It will be in the same price range as the Atari 410.

In the second quarter, two new printers are scheduled. The Atari 1020 is listed as a 40-column color printer. It actually functions like a drum plotter with colored pens. It uses the 40-column sized paper and can be addressed as a printer. The characters printed to it are plotted in normal, compressed, or expanded mode, as controlled by the program. The suggested retail price for the 1020 is $299.00.

The Atari 1025 is a new 80-column printer that does not require the 850 interface. It reportedly does not have graphics capabilities. Suggested retail price for the 1025 is $549.00.

Write to Paul at 97 Jackson St., Cambridge, MA 02140.

MICRO™

MICRO™

CoCo Bits

by John Steiner

The winds of change blow surely through the computer industry, and the Color Computer is not to be left behind. A newer, more powerful DOS is rumored to be on the way, though as I write this there is still no official word from Tandy. The release cannot be too far off as there is mention of an Extended Disk BASIC 1.1 ROM in "The Sands Of Egypt," a newly released adventure game from Radio Shack. There is at least one new command added to the ROM. To execute the program from the 1.1 ROM, just enter 'DOS'. The 1.0 version must use the RUN file command. Tandy will let us know when they are ready.

I noted in a previous column that the color output of my TDP 100 was different from the standard Radio Shack CoCo. This is indeed the case; I have seen the differences in many programs, including the one just mentioned. Computerware says this is a problem on early model TDPs, and they are sending me modification data that will correct the color problem. The new board is now being installed in Radio Shack CoCos, and I am wondering if anyone who has one has noticed a difference from the earlier boards. If you have any information on this, please pass it along.

I am using my TDP and disk system, but with a borrowed disk controller card. The 6809 and controller card failed. The computer has been repaired, and the controller is awaiting a new ROM chip. It should be in any day now. I had been running a tape-based program, and had shut the computer off to reinstall the controller. When I powered up again, the system was down. There is no buffering between the processor and ROM slot, nor is there any between the controller input and ROM. A failure of either unit can easily damage both. As a precaution, be sure the power is off several seconds before reinstalling a card in the slot. Hardware hackers take note: a bi-directional buffer would be relatively easy to build and would allow protection for equipment on the line.

Modems and Things

I received a letter from Paul Van De Plas of Vancouver, BC. He has a Hayes Stack Smart modem, and a CoCo. He would like to use the auto-dial feature and the built-in Morse Code send/receive feature of the modem, but has been unable to access them using the CoCo. If you have any information on using the Hayes modem with the CoCo, pass your information along.

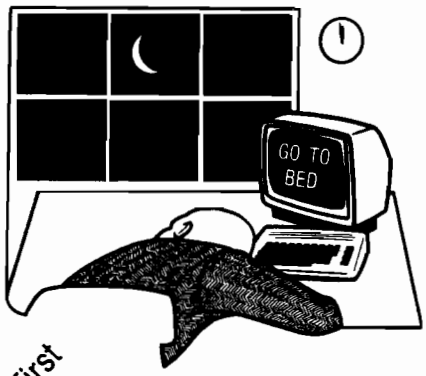
Though I have been using CoCo as a radioteletype terminal in my ham radio hobby, I have never had a telephone modem. When I needed one, I used an old, acoustic coupler, an originate-only unit that I borrowed from a friend. I never felt that the cost of a modem was worth what use I would get out of it. That changed when I happened across a Mura MM100 direct connect, originate/answer unit. The modem is under $100 and takes up only a little space on my crowded desk. I placed it on top of my

(Continued on page 18)

The MIDNITE SOFTWARE GAZETTE

The PAPER

A Bi-Monthly Journal of Notes, Reviews and Articles
Five Years of Service to the PET® Community



The First Independent U.S. Magazine for Users of Commodore Brand Computers

EDITORS: Jim and Ellen Strasma
$20 US / YEAR
Sample Issue free on request, from:
635 MAPLE, MT. ZION, IL 62549 USA
217/864-5320

Circle No. 18

maxell

the top of the line

at the lowest price!

Call our Modem Hotline (anytime) - 619-268-4488 for exclusive monthly specials. Our free catalog contains more than 600 fantastic values.

ABC Data Products
(formerly ABM)

8868 CLAIREMONT MESA BLVD.
SAN DIEGO, CALIFORNIA 92123

ORDERS ONLY 800-854-1555 | ITT TELEX 4992217 | INFORMATION 619-268-3537

Circle No. 19

ATTENTION PROGRAMMERS!!

DATASOFT is currently seeking programs and programmers to add to their rapidly growing and expanding operation. A leading marketer and developer of personal computer software, DATASOFT offers experienced assembly-language programmers the opportunity to join their staff to develop and translate arcade games such as ZAXXON™, as well as to author original material for their games, education and home management product lines. DATASOFT pays competitive salaries, plus bonuses based on product performance. Relocation assistance is available, if needed.

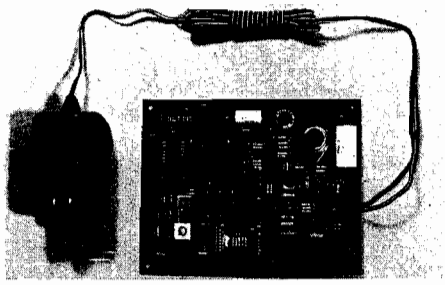
If you have working knowledge of Atari, Apple, TI, or Commodore operating systems, graphics, animation and sound, call or write Melinda Storch at:

Datasoft Inc.®
COMPUTER SOFTWARE
9421 Winnetka Ave.
Chatsworth, CA 91311
(213) 701-5161 / (800) 423-5916

ZAXXON and SEGA are registered trademarks of Sega Enterprises. DATASOFT is a registered trademark of Datasoft, Inc.

Circle No. 20

Boulder Logical Testing, Inc.
is now offering a new EPROM Programmer
FOR $195.00



- Microprocessor based
- Programs 2716, 2732, 2732A and 2532 EPROMS
- Zero Insertion Force (ZIF) sockets are standard
- 25 or 22 volt programming voltage option
- Copy from EPROM to EPROM automatically
- RS 232 interface with selectable baud rates
- Thorough user documentation includes example software drivers for popular computer systems, including Apple*, IBM PC*, and CP/M*
- Command set can be used from host computer or terminal
- Comes complete with power supply

To order, or for more information write:

Boulder Logical Testing, Inc.
PO Box 902
Boulder, CO 80306

Ordering information: send money order or check only, no CODs. Price includes shipping costs and documentation. Colorado residents include 3% sales tax.
*Apple is trademark of Apple, Inc. IBM PC is trademark of IBM, CP/M is trademark of Digital Research.

Circle No. 21

CoCo Bits *(continued)*

radioteletype modem, next to my ham radio, sandwiched between the disk drives. I have had a lot of fun with it, and gained some practical use by accessing the local bulletin board and the educational timesharing network available in North Dakota. I suppose a subscription to CompuServe will be next.

A new accessory available for CoCo is a video monitor output adapter. This device connects to CoCo's RF output and delivers a high-quality signal to a standard video monitor. There is no modification required to the computer, and it should provide an excellent video signal. Flex, in its 64 character-per-line mode, will be the test of this promising new accessory. Though color graphics are nice, the color TV is a poor monitor at best. Any improvement along this line would certainly be welcome.

I have a new drive unit on line and am happy with it so far. It is an MPI from Frank Hogg Labs. Though noisier than my Radio Shack drive and slightly longer, its color and style fit well with the system. The drive, a 40-track unit, allows me to read Flex files written on 40-track disks.

Flex allows you to specify different system and working drives. For example, the DOS will keep track of the fact that drive 0 might be 35-track double density, while drive 1 is double sided, double density, 80 track. That information is stored in the directory, which is always written in single-density format. By allowing different drive styles, a much greater compatibility is possible.

While on the subject of Flex, I must comment on the ease of getting printed information from files. Flex allows all commands to be prefaced with a P, which will send all file output to a printer. Flex allows you to change the baud rate from 110 to 9600. You may choose a carriage return plus line feed, or the standard Radio Shack carriage return where the printer must send its own line feed. You may set the number of stop bits as well.

Service Manuals

In the last few weeks, I have received several letters regarding the availability of service manuals for the CoCo disk drive units. Radio Shack Computer Center managers are informing people that manuals are not available. There are two manuals covering the disk units — one covers the drive and the other the drive controller card. The part numbers for the manuals are MS-260-3022 and MT-260-3022 respectively. The pair costs less than $10.00. According to a local store manager, these manuals are available from Radio Shack National Parts Division. If you provide your dealer with the above data, you should have no trouble obtaining them.

Next month I will have more comments on using Flex, Star-DOS, and other CoCo operating systems.

Contact Mr. Steiner at 508 Fourth Ave. NW,
Riverside, ND 58078

valFORTH™ for Atari* 400/800

Professional Software for the Hobbyist

VALPAR INTERNATIONAL

FORTH has been used for years by ATARI* and others in programming their arcade games. FORTH is *fast*, 15-20 times faster than BASIC, and can make use of every capability for your computer. And it's no longer the province of the professional programmer! With valFORTH and the additional packages described below, you can create programs in an afternoon that would previously have taken weeks of hard work!

WHAT? YOU DON'T ALREADY KNOW FORTH?

Then take advantage of our special offer on *Starting Forth* by Leo Brodie. Widely acclaimed as the best book available on the subject, this entertaining treatment of Forth will make you comfortable with this exciting language quickly and easily. When you order *Starting Forth*, you get a free copy of our "Notes for the valFORTH User," including references, by page, to *Starting Forth*.

These are the utilities developed by Valpar International's software specialists and used to create our commercial software products.

valDOS NEW! By Popular Demand!

valDOS. Now you can use your valFORTH system to read and write normal Atari DOS Files created with valDOS or other sources. Also includes valDOS File Editor for creating and reading FORTH Source Code without "screens." (Not a general file editor.)

(Over 25 pages of documentation. Requires valFORTH.)

valFORTH™

Package contains: fig-FORTH kernel with mathematical and stack operations machine-coded for higher speed than normal fig-FORTH; line editor AND screen editor, debugger, sound and graphics commands, floating point, advanced 6502 assembler, diskcopiers, and much more! (Over 110 pages of documentation)

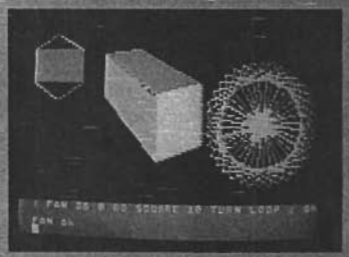
GENERAL UTILITIES AND VIDEO EDITOR

Utilities: 4 array types, 4 case types, text on graphics 8, extensive string manipulation and keyboard input, STICK & PADDLE, randoms, bit manipulation, and much, much more.

Editor: *fast, powerful, complete* valFORTH screen editor 1.1. A professional-quality tool that makes editing a pleasure.

(Over 60 pages of documentation. Requires valFORTH.)

PLAYER-MISSILE GRAPHICS, CHARACTER EDITOR, & SOUND EDITOR



Player-Missile: Create, move, color, change images of and bound players and missiles with high level commands. Full support of 5th player, multicolor players, etc. etc. All critical sections in machine code.

Character Editor: Compose character sets with joystick. Simultaneous display of created characters. Make images for players and missiles.

Sound Editor: Simple independent control of all four voices (one joystick per voice) and audio-control register. Create any single-setting sound with graphical and tabular readout.

(Over 35 pages of documentation. Requires valFORTH.)

Coming Attractions:
* TARGET COMPILER
* 3D WORLD

For more information *fast* and a 4TH ♥ IF HONK THEN bumpersticker, send 25¢ and a self-addressed, stamped envelope.

DISPLAY FORMATTER

Fast, simple creation of all types of display lists, with automatic 4K boundary jumping. Automatic or user controlled memory allocation, and formatting for horizontal and vertical scrolling, and display list interrupts.

(Over 30 pages of documentation. Requires valFORTH.)

TURTLE & valGRAPHICS, AND ADVANCED FLOATING POINT ROUTINES

All graphics modes supported — even GTIA and "7+". Draw and fill commands *faster* and *smarter* than Basic. "Turn-toward" for "chasing" and vanishing point effects; point labeling, etc. Also SIN, COS, ATN, ATN2, etc. added to floating point.

(Over 35 pages of documentation. Requires valFORTH.)

Text Compression and Auto Text Formatting

A unique, two-part utility!

- Text Compression allows the packing of text into much less space than normally required. Useful for wordy Adventure games, "artificial intelligence," etc.!
- Auto Text Formatting takes both normal and compressed text and routes it to the video screen "windows."

(Over 20 pages of documentation. Requires valFORTH.)

valFORTH alone requires 24K
valFORTH plus one or more packages requires 32K minimum
Memory requirements include 10-12K working space
All products are now on non-protected disks
Over 350 pages of detailed documentation!

valFORTH 1-1	$59.95
General Utilities and Video Editor	49.95
Player Missile Graphics, Character Editor, and Sound Editor	49.95
Display Formatter	39.95
Turtle and valGraphics and Floating Point Routines	54.95
Text Compression and Auto Text Formatting	39.95
valDOS and valDOS File Editor	59.95
Heavy-Duty Professional Binder (provided free when ordering 3 or more packages at one time)	13.00
Starting Forth	15.95

Plus Shipping and Handling
VISA and MASTERCARD accepted

Circle No. 17

VALPAR INTERNATIONAL
3801 E. 34TH STREET
TUCSON, ARIZONA 85713
Call Toll-Free 800-528-7070 In Arizona call (602) 790-7141

Principal Software Authors: Stephen Maguire and Evan Rosen

Atari is a trademark of Atari, Inc., a division of Warner Communications

THE ATARI 1200XL

by Paul S. Swanson

Differences between the Atari 1200XL and earlier 400 or 800 computers are significant. The most obvious difference is the physical appearance. The case design is sleek, particularly in comparison to the 400 and 800 systems. The keyboard is similar, but the two most troublesome keys have been moved off the main keyboard. These two keys are BREAK, which can accidentally stop a program running on a 400 or 800 if you miss BACK S by one key, and the Atari key, which inverts the video output if you miss the question mark by one key. These two keys have moved up to the metal strip above the keyboard.

There have been five keys added to the function keys available on the 400 and 800. One is the HELP key, which can be strobed by a program to sense user distress or any other defined symptom. There is also a built-in demo program in the 1200XL that uses the HELP key to control its functions.

The other four keys, F1 through F4, can be used in combination with SHIFT and CONTROL (renamed from the 400 and 800 CNTL) to provide a total of 12 keyboard-initiated functions. With no SHIFT or CONTROL, these keys act like the arrow keys do with CONTROL down, so they contribute single keystroke cursor control. With SHIFT down, the four keys add four additional cursor controls. SHIFT F1 will move the cursor to the upper left corner of the screen, SHIFT F2 to the lower right, SHIFT F3 to the beginning of the physical line, and SHIFT F4 to the end of the physical line.

CONTROL with these four function keys creates more involved functions. CONTROL with F1 locks out the keyboard, which can be restored by pressing the same combination again. CONTROL with F2 turns off the video display, which is restored by pressing any keyboard key. CONTROL with F3 toggles between keyboard click and no keyboard click. CONTROL with F4 changes the character set to the new built-in international character set.

Atari omitted the cartridge door from the 1200XL. This welcome change circumvents the physical weakness of the 400 and 800 case designs. The only part on my 400 or 800 that has ever failed is the cartridge door hinge, a problem I don't anticipate with my 1200XL. The cartridge has been moved to an open slot on the left side of the computer. This should provide many possibilities. With no cover containing the cartridge, larger cartridges (actually external to the computer) can be constructed that bypass the restrictive size constraints of 400- and 800-compatible cartridges. A slight alteration of dimensions on the cartridge slot may make some third-party cartridges incompatible with the 1200XL, but all of the Atari-produced cartridges will work easily.

The rest of the external accessories on the 1200XL are identical to the ones on the Atari 800 except that there are only two controller jacks on the 1200XL and four on the 800. There is a monitor jack as well as a modulator for connection to a television, a serial port jack, and the selector for channels 2 or 3.

Operating System Improvements

As indicated by the four function keys, there have been operating system changes. The international character set, implemented by CONTROL with F4, replaces the keyboard graphics characters with a set of 29 international characters with diacritical marks with the letters as well as the British pound and inverted exclamation point. As with the 400 and 800, you can also invent your own character set. L2, which is a LED indicator on the keyboard, lights when the international character set is selected.

The keyboard click is no longer produced by the keyboard. All of the functions are still possible, but they are channeled through the television speaker. When the keyboard click is suppressed, all of the other sound channels still operate normally.

Turning off the video display can speed up processor time by about 25%, which is done by eliminating DMA (Direct Memory Access). The real-time clock (which does exist on all Atari computers, contrary to Commodore's advertising claims) continues to run, so the vertical blank interrupt is still in operation with this function implemented. From timing loops that I ran on both the 1200XL and on the 800, it appears that the vertical blank interrupt has also been rewritten and runs faster on the 1200XL.

The additional memory that was added to bring the 1200XL to 64K is bank selected into the 6502's address space. This implementation excludes direct access to the additional memory from BASIC; however, it can be utilized by machine-language programs and subroutines, including those called from BASIC.

Although I have no documentation on it, I believe the video output circuitry has been improved on the 1200XL. The colors are brighter and the contrast is sharper. In switching from normal television viewing to using the 1200XL in the normal text mode, I have found it necessary to turn down the contrast significantly or set color registers 1 and 2 closer together in luminance.

The locations used to read and write controller ports 3 and 4 on the 400 and 800 are redefined on the 1200XL. These locations (Port B) are used to implement, under program control, the new features available from the four new function keys. The other new LED, L1, is lighted when the keyboard is disabled. Writing to the Port B registers controls L1 and L2 the way the function keys would. Port B also controls the bank selection options.

Another operating system enhancement is the addition of four graphics modes. All four of these new modes are actually available on the 400 and 800, but can be declared by a GRAPHICS statement from BASIC on the 1200XL because they are operating-system supported. To implement them on the 400 or 800, you must build your own custom display list.

The four modes include two character graphics modes and two map modes. You must build and locate in

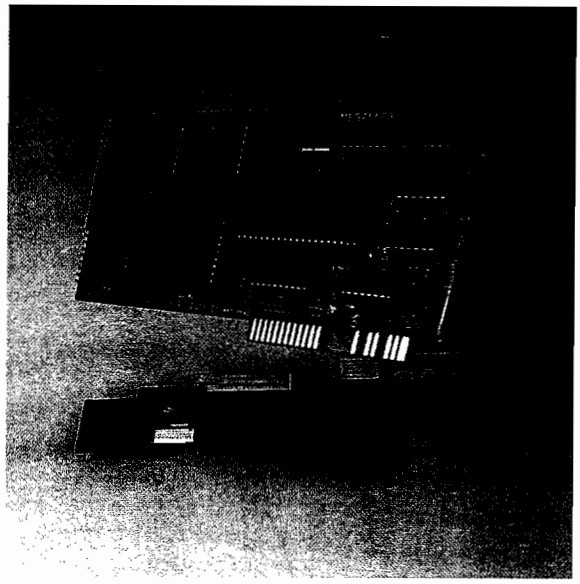
MEGAFLEX® ABILITY

You Pick The Disk System, MegaFlex Controls It!

WITH SOFTDRIVERS FOR A FLEXIBLE FUTURE!

MEGAFLEX—a universal floppy disk controller and modern alternative to the Apple drive system offering increased storage, improved reliability and . . . FLEXIBILITY.

Enjoy megabytes of online storage with your choice of micro, mini, or maxi drives—or even 6Mb with the Amlyn cartridge pack! Ideal for high-capacity storage now, winchester-disk backup later.



The MEGAFLEX secret is to autoboot soft-drivers that match the needs of your drive system. All hardware functions are software-controlled. MEGAFLEX can match new drive capabilities without hardware changes. Drive-dependent ROMs have been eliminated.

APPLE *III*? OF COURSE!!

MEGAFLEX is compatible with BASIC, CP/M, Pascal, VISICALC, SOS and DOS-emulation on the Apple III, Apple II, Franklin Ace and Basis. All language features and operating system commands (LOAD, BRUN, etc.) are standard. If you can operate Apple drives you can operate MEGAFLEX! Your Apple software will run without modification too.

BRIDGE THE APPLE FORMAT BARRIER!

The MEGAFLEX diskette does what Apple's cannot—read and write diskettes from other computers! Software-controlled industry-standard IBM 3740 or System 34 type formats allow the MEGAFLEX library of reformatting software to read and write Altos, Radio Shack, Osborne, and IBM PC diskettes. (Call for the latest software details.)

MORE STORAGE, MORE UNIVERSAL FEATURES, LOWEST COST

MEGAFLEX with 8″ maxi or high density 5.25″ minis gives you 1.2 Megabyte of formatted data per diskette for 8 times the file and data size!

MEGAFLEX offers flexible software choices:

- data rate (250/500 Kbits per second),
- single and double density recording, and
- single/double sided drive operation (max 4 drives).

MEGAFLEX has the lowest chip count of any controller today! This means less power, a cooler Apple and better reliability.

Lowest price, highest performance, that's MEGAFLEX!™

MEGABYTER™
A Division of SVA

11722 SORRENTO VALLEY ROAD
SAN DIEGO, CA
(619) 452-0101
TWX 910-335-2047 APPLE TWO SDG

TRADEMARKS CP/M—Digital Research

Circle No. 22

memory a character set when using the character graphics modes. The character set information is interpreted the same way bytes on a four-color map mode display are interpreted except that, when the character is printed in inverse video, bit combination 11 uses register 3 instead of register 2. Although much of the literature from Atari that deals with these two modes calls them four-color modes, they are actually five-color modes, restricting each character to four of the colors. The colors for each character are background (register 4), register 0, register 1, and either register 2 or register 3.

The new map modes are 160 × 192 dot displays. One is a two-color mode and the other is a four-color mode. The addition of these modes means that all of the internal IR modes except one are now operating-system supported. The one that isn't is a variation of mode 0 using ten scan lines per character line instead of the eight used in mode 0. This enhances the appearance of the characters by allowing two extra scan lines for the lower-case descenders. Judging by the number of programs that I have seen using this mode, the demand may dictate that Atari may never bother to support it in their operating systems. It can produce nice output, but mode 0 is usually considered good enough.

Concerning the Competition

This new offering by Atari is not without competition; for example, the Commodore 64. The Commodore system is a sophisticated assembly of electronics, implementing several of the same features implemented by the Atari computers. The Commodore has more and larger sprites than the Atari's player/missiles. The Commodore has three sound channels as opposed to the four in the Atari computers, but the Commodore has more controls for attack and decay as well as finer selection of frequency. However, the sound channels on the Atari computers are more capable of producing sound effects.

There are 16 colors available on both the Atari line of computers and the Commodore 64, but the Atari computers also give a selection of luminances. Correctly written graphics routines on the Atari have much more depth than routines written on computers such as the Commodore 64 and Apple because of these luminances.

The Atari display output is also more in tune with the television — the normal video output in the home where these computers are intended to be. There are only 160 color clocks across the visible area of a standard color television. This makes it physically impossible to have a selection of two different colors for the text mode if 40-character lines are used. If you have a Commodore 64 and you try to select certain combinations of color for the text mode, evey other letter will be illegible because the television cannot display the number of color clocks required. The Atari text mode uses one color with two luminances for the characters in the text mode with at least two dots for the width of each dot or line in the characters. This makes the Atari characters appear shorter and fatter than the characters used on other computers, but also makes them much more legible because the method used on Ataris is more consistent with the way the television processes the signal.

New Peripherals

No drastic physical change in appearance would be complete without color-coordinated peripherals. Already available is the Atari 1010, which is an upgrade of the Atari 410 Program Recorder (cassette drive). The case is consistent with the design of the 1200XL and the internal circuitry has undergone a few improvements. Some existing cassette-based programs may give you problems on the 1010 and 1200XL combination, but Atari includes a notice in the literature supplied with the 1200XL that promises to solve any problem you may have with APX cassette programs, free of charge, of course.

Available in the second quarter, possibly by the time you read this, are two new printers. The Atari 1020 is advertised as a 40-column color printer. It uses multicolored pens to plot letters and/or graphics. The Atari 1025 is an 80-column printer that does not require the 850 interface. I have been told that it does not have graphics nor some of the other fancier options afforded by third parties (I'll be keeping my Epson MX-80 with Graphtrax Plus), but not requiring the 850 makes its suggested retail price of $549 very attractive for 80 columns of printed output. Other 80-column printers may have similar prices, but require an additional $200 for the 850. Other new peripherals are expected in the near future.

Looking Ahead

Atari will be announcing new computers in June. Although not much has been released officially, two new computers, called the 600XL and 800XL, are expected. The new computers will have BASIC as a built-in feature according to the sources I have tapped for information.

Rumors concerning the new computers should not be trusted, but are interesting. I have heard mumblings of built-in speech synthesizers and modems. If these prove to be true, I would say that Atari is making real progress in advancing the quality of home computers. The internal graphics circuitry used in the Atari seems to be, by far, the most advanced available in any computer in the same price range.

Third-party suppliers should become very active in producing new peripheral devices for the 1200XL. Although this computer has been criticized for being a "black box" with no open circuitry that the true hacker can start cutting up and rearranging (probably one of the stronger selling points for the less sophisticated Apple computer line), the 1200XL actually offers something not easily available on the older Atari computers. The open cartridge slot has 13 of the 16 address lines with a select line governing the other three, all eight data bus lines, and several other control lines available. Little imagination is required to think of what can be done with this slot.

You may contact Paul Swanson at 97 Jackson Street, Cambridge, MA 02140.

MICRO™

New Microprocessors for Your Apple II

by Phil Daley and John Hedderman

Do you feel left out? Does everyone seem to have a new computer but you? You can have a new computer without selling your old one.

There are several add-on boards for the Apple II that increase its capabilities, provide new features, teach you new languages, and allow you to run software written for processors other than the 6502. We discuss several of these boards: the Excel-9 6809 by Norell Data Systems, the Mill 6809 by Stellation II and the Dtack Grounded 68000. We have also been working with an Apple III and will present a short summary of its features.

Let us start off by saying that we like the Apple II and the 6502 microprocessor and have not found serious deficiencies in either. However, after using a 6809 microprocessor, you cannot help but be impressed with the increased capabilities of the 'cadillac' of 8-bit microprocessors. Just having 16-bit registers makes a large difference in programming ease, and being able to long branch to any portion of memory makes writing location-independent code a snap. When you add on the power of a structured operating system like OS-9, the results are dramatic. Certainly no one will argue the advantages of a 68000 microprocessor. With 32-bit registers and a 12.5 MHz clock, the speed differential on identical code is enormous. When you add the programming ease and addressing capability of the 68000, you have amazing potential.

TSC's Flex

The Excel-9 board comes with the complete FLEX manual from Technical Systems Consultants and is well written, easy to understand and complete. Unfortunately the Apple documentation from the Norell people was written in Japanese and translated into English, or at least that's the way it reads. The documentation is somewhat sketchy but gives a satisfactory explanation of the setup and operation of the Excel-9 board.

The first step in using a new disk is to always make a backup before something disastrous happens. To format a Flex disk, a program called NEWDISK is executed. We typed in 'NEWDISK 2' to refer to formatting a disk in Apple drive 2. Unfortunately that did not work, as a 'DRIVES NOT READY' error message was displayed. Apple normally calls the drives in each disk controller card '1' and '2'. The slot number of the controller card and the drive number of the disk are necessary to completely specify the desired drive. Flex numbers the drives, starting with '0', and continuing in sequence. The Apple drive #1 is considered as Flex drive #0 and Apple drive #2 is Flex drive #1. This is probably less confusing than the CP/M system of Apple drive '1' being CP/M drive 'A'.

Typing 'NEWDISK 1' sucessfully formatted a blank disk. All commands in Flex are executed by typing the program name. The documentation says to make a backup the proper command is 'BACKUP'. When we typed 'BACKUP', a 'NOT FOUND' error was received. A quick CATalog of the disk revealed a COPY command, and by typing 'COPY 0,1' produced a back up of the Flex master disk. Next we 'LINKed FLEX' to produce a bootable disk. The bootstrap program has to know physically where the system DOS resides on the disk, and the program LINK stores the starting track and sector of FLEX.SYS in track 0, sector 1 so that upon booting a disk, the boot program can load Flex into memory.

All of the normal Flex utilities are included on the system disk, and they all work as specified in the Flex manual. Using the LIST, ASN, CAT, etc., commands produced the standard Flex results. To try the 'P' command, you attach a regular printer cable to your interface and printer, we used an Epson MX-80 with the Apple serial card. Typing 'P,CAT 0' produced a hardcopy catalog list.

To delve a little deeper into the innards of Apple Flex, a sector read utility was used to examine the backup copy of the Flex system disk. The track and sector linking that Flex uses to access the sectors in an interleaved fashion seemed to be rather haphazard and not sequentially as on other 6809 computers. It appears that the Apple NEWDISK writes the sectors in a physically sequential process and then links them ahead four or five sectors to optimize access speed, whereas Flex on a regular 6809 computer interleaves the

sectors as it writes them, and then links them in a logically sequential process. This difference prompted us to write a test program (see listing 1), a Flex utility to change the name and number of the Flex formatted disk that is stored in the system information record, using the Flex Advanced Programmers Manual. This program was written and tested on a standard 6809 computer first and then retyped, with no changes, into the TSC Text Editor supplied on the Apple Flex disk.

The program reads the System Information Record (one sector) by calling one of the Flex user routines and, after making the appropriate changes to the information, writes the sector back, referring to its track and sector (Track 0, Sector 3). By employing two different methods of accessing the disk, we gave the system a double test. Each method worked even though the track-sector reference to the disk sector was not the Apple disk sector referenced. Apparently Flex on the Apple translates this difference internally.

The Excel-9 board also has a basic system monitor, which we used to examine some memory. The monitor can do things like Dump, Disassemble, Jump, Compare, etc. One note: when within a monitor command, the documentation doesn't mention the accepted method of return to the command mode. (Reset works, but it's messy.) A {space}{return} seems to work most of the time.

The Excel-9 board works very well and fully implements the Flex operating system on the Apple. It worked without flaw on our Apple II. We couldn't use it on our Revision 7 Apple II without removing the Apple serial card from slot 1. Norell is aware of the problem and is working on a cure. If you want to learn about Flex, the Excel-9 is the board for you.

Stellation II's Mill

Standard software for the Mill, a 6809 board, includes a macro-assembler and debugger for writing machine-language programs on the 6809. Included is a demonstration package for multi-processing — running the 6502 at 1/5th speed and the 6809 at full speed at the same time. Also available is a Pascal speed-up kit to increase the operating speed of Apple Pascal. The multi-tasking operating system, OS-9, is available from Microware and includes a structured basic, BASIC09, a sort of hybrid with Pascal.

The documentation that is packaged with all parts of the Mill system is complete, understandable, and well written. But learning an entirely new operating system requires effort on your part. OS-9 is a 'Unix-like' heirarchical tree-structured system and, as such, takes someone who is used to a simple system like Apple DOS a while to get the hang of it. More on Apple operating systems in next month's magazine.

McMill is a powerful Macro-Assembler that comes with the Mill. It would be very useful for developing software for other 6809-based computers with more limited development systems; eg., the TRS80-C.

To test the assembler for the 6809 Mill, we tried a bubble sort program (see listing 2) from a programming manual. Any normal text editor that makes standard DOS text files can be used to write the assembly file. We used Apple Writer as it is one of the few that is DOS compatible and doesn't word wrap when you don't want it to. After removing the errors due to erroneous labelling, the file assembled correctly. The question arose as to the proper ORiGination, not covered in the manual. The Mill uses addressing which is $8000 relative to the Apple addressing due to conflicting areas of memory. We tried $A000 (assuming that would translate to $2000), which placed it in memory at $A000, the middle of DOS: this crashed the system, so we tried $2000.

This address inversion is the trickiest part of using The Mill. Since having two processors addressing the same RAM memory can cause conflicts, The Mill inverts the most significant memory address bit (A15) prior to being placed on the Apple memory address bus. This means that there are both 'logical' and 'physical' names for each memory location. The physical name is the actual RAM address in the Apple, the logical address is the inverted address that the 6809 thinks its using. In this manner, the conflict with high memory interrupt vectors is overcome. Since the Apple vectors are in ROM, there would be no way to change them for the 6809. Inverting the addresses places the 6809 vectors at the top of the $7000 (physical) addresses.

The address inversion also creates another problem. The Assembler creates the code at the ORiGinated address. The output of the Mill Assembler is a text-type file of hexadecimal code. A program called 'LOAD09' is a utiliy on the assembler disk that loads the created file into RAM memory. Unfortunately, it will also try to load the program into ROM or DOS, anywhere the program was ORG'ed. Since this leads to system crashes, it is necessary to ORG the program somewhere in Apple RAM that is available. Then the program must be BLOADed and called at a different address, such as the inverted one. Those of you who have been following closely will see that this places definite constraints on the addressing modes of the original program. The bubble sort program was then rewritten to use only Program Counter Relative addressing to result in a position independent program that can run anywhere in memory. The 6809 assembler makes its very easy to write position independent code, calculating all the relative offsets to the PCR.

McMill is a powerful Macro-Assembler that comes with the Mill. It would be very useful for developing software for other 6809-based computers with more limited development systems...

A call to the Mill people prompted the suggestion we try $7FFE as the origination point with the first two bytes of zero to overwrite the interrupt vector. That would place the program at $8000 (relative $0000). This is probably the most satisfactory solution.

'MUG' is a program to interactively run and debug 6809 programs. It includes memory dump and disassembly listings, breakpoint insertion, memory search and modify, register change and dump and a calculator mode. It is easily modifiable so that the Mill can reside in any slot. (This is starting to become a problem with all these add-on cards cluttering up our Apple.) The program and documentation are easy to use and, once you overcome the address inverting, easy to understand.

The Mill also has demonstration

programs of the two processors running concurrently. While it is most impressive, it would take some time to learn how to make the best use of this feature.

Another set of programs is the Pascal speed-up kit. This group of files substitutes for the SYSTEM.APPLE and SYSTEM.MISCINFO files on APPLE1: and requires the Mill to be present in the Apple when operating. The directions are clear and established the files on our newly made APPLE1: (now called MILL09: so that we don't get it mixed up). The speed difference is quickly apparent. You can even specify a screen print of which processor is running at the current time, although it didn't show until we turned off the 80-column card.

This is the smoothest running board we tested with the least bugs and best software. It is also the only unit tested that includes software for Pascal. The complete outfit including OS-9, while expensive, is probably the most useful adaptation for the Apple.

Dtack Grounded's 68000

If you are adventuresome and want to experiment with a fast microprocessor, Dtack Grounded makes a 68000 board that is compatible with the Apple. The connection instructions are simple (the board plugs into slot 2) and the demonstration programs are impressive and work flawlessly. A cross-assembler is available from Phase-0, and it appears to work correctly.

Again, it is necessary to use a text editor to make the file (see listing 3) before assembling. The editor must supply carriage returns at the end of each line. ASSEM68K has several nonstandard requirements such as: it uses an 'ASC' pseudo-op instead of 'DC' for ASCII characters, it requires a ';' for comments, and it does not allow an 'END' statement. The Phase-0 documentation does mention the first two, it does not mention the third. The operation of the actual assembly is easy; it only requires a source file name (optional drive number), object file name (default 'source name.OBJ0') and printer slot (default 0 for monitor). The listing format is relatively neat except that comments over 18 characters long wrap around to the next line (on an 80 column printer) and make the listing harder to read.

The assembler disk comes with an upload and monitor programs so that you can learn and try out 68000 code running on the Dtack board. Unfortunately the documentation on the upload and monitor portions is incomplete and hazy. Producing the object code proved to be much easier than uploading it to the 68000 board. We first tried the 'UPLOAD' program supplied on the Phase-0 disk. The program does not handle extra DOS parameters, such as drive number. We had to LOAD, CATALOG, and RUN to read from a second drive.

...for those of you who like the 6502 but want a system with more features, there is the Apple III, the most sophisticated 6502 we've seen.

Next we used the MONITOR program to look at memory. At first we couldn't find the uploaded program and made a quick call to Dtack Grounded. Mr. Hardenburg told us that the upload program loads the program into the Dtack board wherever it is organized. He also gave us a quick memory map, not included in the documentation to help in deciding where to write programs. $0 to $1000 is the bootstrap program, from $1000 to $10FF are the RAM vectors, and from $1100 to $1EFF is the system stack, building down. Therefore $2000 is a safe place for programs with scratch space in the $1100 area.

Since we had ORiGined the program at $1000, it had been overwritten by the RAM vectors. Setting the ORG to $2000, uploading the code and running the monitor program found the program intact and residing at its proper location. The next problem was how to gracefully exit from the program. We tried JuMPing to $122, the monitor input idle. Upon execution, the program returned to the monitor and we dumped the registers. Unfortunately, they showed no results; in fact, except for the Stack Pointer (A7), none had changed at all.

After several program changes/runs and no results, we tried modifying the registers directly with one line programs. When this also gave no changes, we gave up on the Phase-0 monitor program. There seems to be a problem with either the register dump or the execution command.

Using the Monitor that is in the Bootstrap ROM of the Dtack board, while a more complicated procedure, produced better results. We examined the sample programs included with the Dtack board and determined the proper procedure for executing a program on the board. It includes loading the FP BASIC from Dtack on the language card, uploading the file, executing the file and ending with a register dump to view the results (see listing 4). The REM statements are ignored by the Applesoft interpreter and used by the Monitor program as data codes. In this manner, data can be transferred from the Dtack board to the Apple, and vice-versa. There are other CALLs to perform other functions as well. We also discovered the opcode 'TRAP #15', which dumps the 68000's registers and exits the program. TRAPping to vectors 0-14 can provide other exits and routines by changing the vector pointers. In any case, using the Monitor program from BASIC works quite well.

The Dtack board has tremendous potential but requires knowledge of the user. It has the most primitive software at this point, but also the most potential for future development.

Apple's III

For those of you who like the 6502 but want a system with more features, there is the Apple III, the most sophisticated 6502 computer that we've seen. It contains the best elements of the Apple II with improvements of many of its shortcomings, and has additional features not found on many systems in its price field. In as much as we have heard the bad reviws that followed its release, and that we like the Apple II even with its faults, we awaited with trepedation the arrival of the III, and have somewhat procrastinated the testing and experimentation necessary to write this review.

The folks at Apple need not fear, however, as we liked the computer more than we thought we would and, indeed, if not for its price tag, would be tempted to buy one. The question remains whether a 'personal' computer

would need all its features and memory, and it is probably better suited in a business atmosphere.

The Apple III has a built-in disk drive that is much quieter than the old Apple drives. We have recently seen some new drives for the II that are much quieter and more professional sounding. They all seem to work quite reliably however. It also has an RS-232 serial port, two ports that support two analog and two digital devices, B & W and color video ports and an external speaker plug. The III runs on SOS (Sophisticated Operating System), one of the two biggest improvements over the II. It is a system reminiscent of the II's Pascal environment (which it is written in), including its exasperating file names, but it is much better than DOS and CP/M.

There is a standardized interface for all block and character devices, and each device is named and configurable. There is a Utility Configuration program to set up initial parameters and to save your personal configuration to disk. Then, assuming you don't buy any additional hardware, you transfer that file to all your disks that you want to boot and you are all set. We added a Vista Timecard III to our system in about ten minutes, faster than it took us to learn how to set the right time after it was connected. The Timecard is accurate to several seconds per week.

The file system uses pathnames and supports multiple heirarchies in the directory system, a branching tree file structure. Fifty-one files can be entered in a volume file directory, but there can be 1663 sub-directory entries on that volume. File names can be 15 characters long, something we never thought much about on the II, but having worked more on a Flex system allowing only eight characters per name, we really appreciate additional file name lengths. Pathnames can be up to 128 characters.

The other major improvement is in the console, CRT display. Not only does the III standardly display 80 characters per line, upper and lower case (the biggest defect in the II), but also the character set is in RAM, leading to user-definable character sets. In fact, Apple supplies several fonts on the Apple Writer III and Business BASIC disks. We found it fascinating to write an example in Gothic font. While we can't think of any real use for a Gothic font, we could definately see some uses for math symbols and Greek letters. The keyboard also supports type-ahead (no use to someone who types as slow as we do) 128 character codes directly generated from the keyboard (256 if you count the 'open apple' which sets the high bit of all keys on the keyboard), a numeric keypad, and all keys repeatable if held for more than 1/2 second. We especially like the two-speed cursor control keys, which go faster if you push down harder.

The Monitor III that came with the system is the best green monitor that we've seen — crisp and clear. However, the idea that 'normal' is black letters on a green screen and 'inverse' is green letters on a black screen must have been thought up by someone not connected with reality. Actually, in deference to us computer types, the default condition is 'inverse'.

The Apple III appears to us to be a smooth working machine, but too expensive to be a home computer. It is more appropriate than the Apple II for business use.

So what about the IIe? The reviews and our personal observation point out that the structural changes, while major in fact, are minor in operation.

All the products reviewed lived up to our expectations; some can make your Apple do things you wouldn't have dreamed possible. If you want increased performance or to learn about new microprocessors, you don't have to buy a whole new computer.

Manufacturers' Addresses

Excell-9
Norell Data Systems
3400 Wilshire Blvd.
P.O. Box 70127
Los Angeles, CA 90010

The Mill
Stellation Two
P.O. Box 2342
Santa Barbara, CA 93120

Dtack Grounded
Digital Acoustics
1451 E. McFadden, Suite F
Santa Ana, CA 92705

ASSEM68K
Phase Zero, Ltd.
2509 N. Campbell Ave. Suite 130
Tucson, AZ 85719

Apple III
Apple Computer Co., Inc.
Cupertino, CA 95014

The Apple IIe

So what about the IIe? We can hear the questions now. Also, we get telephone calls several times a week asking what are the differences between the II+ and the IIe. Unfortunately, we have not received a product for review. The most important things to determine, such as the changes to the F8 ROM, will have to wait until we can do some experimentation or until some of our readers who own a IIe write in to let us know the differences.

The reviews and our personal observation point out that the structural changes, while major in fact, are minor in operation. To someone who uses a II+ with all the trimmings (a language card, an 80-column board, etc.), the changes will seem minor indeed. The keyboard layout is similar to the III and we never did get comfortable with that. Not having the double quote over the '2' really slowed down my already snail's pace typing rate. We did like the cursor keys, however.

Most software seems to run OK; the original figure of 85% appears to have been a major understatement designed to shock the software community. Unless a routine uses strange entries into the Monitor ROM, the programs should run OK. Eight-column boards are strange creatures, being completely incompatible with each other, so don't expect all programs for an 80-column board to work on the IIe. They wouldn't run on a II+ without the specifically designed board, either.

We promise a more technical in-depth report on the exact changes when we are able to get that information. If anyone has a IIe, write us about the changes. We know that the binary-language loader is certainly faster than the old system master 'HELLO'.

(Continued on next page)

Listing 1: Flex Example (Assembly)

```
FLEX ASSEMBLY LANGUAGE PROGRAM : NAMEDISK FOR FLEX ON THE APPLE

                 * DISK RENAME/RENUMBER
                 *
                 * COPYRIGHT (C) 1982
                 * THE COMPUTERIST, INC.
                 * 34 CHELMSFORD STREET
                 * CHELMSFORD, MA 01824
                 * ALL RIGHTS RESERVED
                 * USER-CALLABLE ROUTINES

            CC0C WKDRV   EQU    $CC0C     WORK DRIVE LOCATION
            CC14 LINBUF  EQU    $CC14     LINE BUFFER POINTER
            CD03 WARMS   EQU    $CD03     DOS WARMSTART ENTRY
            CD15 GETCHR  EQU    $CD15     GET CHARACTER
            CD18 PUTCHR  EQU    $CD18     PUT CHARACTER
            CD1B INBUF   EQU    $CD1B     INPUT BUFFER LINE
            CD1E PSTRNG  EQU    $CD1E     PRINT STRING
            CD27 NXTCH   EQU    $CD27     GET NEXT BUFFER CHARACTER
            CD39 OUTDEC  EQU    $CD39     OUTPUT DECIMAL NUMBER
            CD3F RPTERR  EQU    $CD3F     REPORT DISK ERROR
            CD48 INDEC   EQU    $CD48     INPUT DECIMAL NUMBER

                 * FMS EQUATES

            D403 FMSCLS  EQU    $D403     FMS CLOSE
            D406 FMS     EQU    $D406     FMS CALL

                 * SYSTEM EQUATES

            C840 FCB     EQU    $C840     SYSTEM FCB
            C841 ERLOC   EQU    FCB+1     ERROR NUMBER
            C843 FCBDRV  EQU    FCB+3     DRIVE NUMBER

                 * MAIN PROGRAM

                         ORG    $C100

C100 20   01        CALL   BRA    CALL2
C102 02             VN     FCB    2         VERSION NUMBER
C103 10BE CC14      CALL2  LDY    LINBUF    POINT TO NEXT BUFFER CHARACTER
C107 A6   A4                LDA    ,Y        GET FIRST CHAR
C109 81   0D                CMPA   #$D       IS IT A CR?
C10B 26   08                BNE    SPEC1
C10D B6   CC0C              LDA    WKDRV     GET WORKING DRIVE NUMBER
C110 B7   C843              STA    FCBDRV    STORE IN FCB
C113 20   15                BRA    CALL3
C115 81   30        SPEC1  CMPA   #'0       IS FIRST CHAR A '0'?
C117 27   0C                BEQ    SPEC2
C119 81   31                CMPA   #'1       IS CHAR A '1'?
C11B 27   08                BEQ    SPEC2
C11D 86   15                LDA    #21       FORCE ILLEGAL FILE SPECS ERROR
C11F B7   C841              STA    ERLOC     STORE IN FCB
C122 16   00C8              LBRA   ERROR
C125 80   30        SPEC2  SUBA   #$30      CONVERT ASCII TO BINARY
C127 B7   C843              STA    FCBDRV    STORE DRIVE NUMBER IN FCB
C12A 8E   C840      CALL3  LDX    #FCB      POINT TO FCB
C12D 86   10                LDA    #16       OPEN SYSTEM INFO RECORD
C12F A7   84                STA    ,X        STORE IN FCB
C131 BD   D406              JSR    FMS       CALL FMS
C134 1026 00B5              LBNE   ERROR     CHECK FOR ERRORS
C138 8E   0007              LDX    #7        GET RECORD CODE
C13B A7   84                STA    ,X        STORE IN FCB
C13D BD   D406              JSR    FMS       CALL FMS
C140 1026 00A9              LBNE   ERROR     CHECK FOR ERRORS

C144 8E   C23E              LDX    #STRNAM   POINT TO STORAGE AREA
C147 C6   80                LDB    #128
C149 6F   80        CALL4  CLR    ,X+       CLEAR STORAGE
C14B 5A                     DECB
C14C 26   FB                BNE    CALL4

C14E 8E   C1F6              LDX    #CURNMS   CURRENT NAME STRING
C151 BD   CD1E              JSR    PSTRNG    PRINT IT
C154 8E   C890              LDX    #FCB+80   POINT TO VOLUME NAME
C157 C6   0B                LDB    #11       PREPARE TO OUTPUT 11 CHAR
C159 A6   80        LOOP1  LDA    ,X+       LOAD DATA, BUMP ADDRESS
C15B BD   CD18              JSR    PUTCHR    OUTPUT NEXT CHARACTER
C15E 5A                     DECB
C15F 26   F8                BNE    LOOP1     PRINT UNTIL DONE
C161 8E   C208              LDX    #NEWNMS   NEW NAME STRING
C164 BD   CD1E              JSR    PSTRNG    PRINT IT
C167 BD   CD1B              JSR    INBUF     INITIATE BUFFER INPUT
C16A 8E   C23E              LDX    #STRNAM   POINT TO STORAGE AREA
C16D BD   CD27      LOOP2  JSR    NXTCH     GET NEXT BUFFER CHARACTER
C170 25   04                BCS    FILL1     EXIT ON NON-ALPHA CHAR

C172 A7   80                STA    ,X+       STORE IN WORK AREA
C174 20   F7                BRA    LOOP2     TRANSFER UNTIL DONE
C176 8C   C23E      FILL1  CMPX   #STRNAM   WAS FIRST CHAR NON-ALPHA
                                             (CR)?
C179 27   20                BEQ    NUMBER    IF SO, SKIP TO DISK NUMBER
C17B 86   FF                LDA    #$FF
C17D A7   84                STA    ,X        FLAG END OF NAME
C17F 8E   C23E              LDX    #STRNAM   POINT TO INPUT NAME
C182 108E C890              LDY    #FCB+80   POINT TO NAME IN FCB
C186 C6   0B                LDB    #11
C188 A6   80        FILL2  LDA    ,X+       GET CHAR FROM INPUT NAME
C18A 81   FF                CMPA   #$FF      END OF NAME?
C18C 27   07                BEQ    FILL3     IF SO, PAD WITH NULLS
C18E A7   A0                STA    ,Y+       ELSE, TRANSFER TO FCB
C190 5A                     DECB
C191 26   F5                BNE    FILL2     TRANSFER UNTIL FLAG
                                             REACHED
C193 20   06                BRA    NUMBER    NO PADDING NEEDED
C195 4F             FILL3  CLRA
C196 A7   A0                STA    ,Y+       STORE NULLS IN FCB
C198 5A                     DECB
C199 26   FA                BNE    FILL3     PAD UNTIL DONE

C19B 8E   C21A      NUMBER LDX    #CURNMB   CURRENT NUMBER STRING
C19E BD   CD1E              JSR    PSTRNG    PRINT IT
C1A1 8E   C89B              LDX    #FCB+91   POINT TO VOLUME NUMBER
C1A4 5F                     CLRB             SUPPRESS PADDING
C1A5 BD   CD39              JSR    OUTDEC    OUTPUT DECIMAL NUMBER
C1A8 8E   C22C              LDX    #NEWNMB   NEW NUMBER STRING
C1AB BD   CD1E              JSR    PSTRNG    PRINT IT
C1AE BD   CD1B              JSR    INBUF     GET LINE FROM INPUT BUFFER
C1B1 10BE CC14              LDY    LINBUF    POINT TO FIRST CHARACTER
C1B5 BD   CD48              JSR    INDEC     INPUT DECIMAL NUMBER
C1B8 25   1C                BCS    WRITE     EXIT ON NON-NUMERIC
                                             CHARACTER
C1BA 5D                     TSTB
C1BB 27   19                BEQ    WRITE     EXIT ON INVALID NUMBER
C1BD 5F                     CLRB
C1BE A6   A0        NUMB2  LDA    ,Y+       GET NEXT DIGIT
C1C0 81   0D                CMPA   #$D       IS IT A CR?
C1C2 27   03                BEQ    NUMB3     EXIT IF SO
C1C4 5C                     INCB
C1C5 20   F7                BRA    NUMB2     GET DIGITS UNTIL CR REACHED
C1C7 C1   04        NUMB3  CMPB   #4
C1C9 22   0B                BHI    WRITE     ACCEPT IT IF LESS
                                             THAN 10000
C1CB 108E C89B              LDY    #FCB+91   POINT TO VOLUME NUMBER
C1CF 4F                     CLRA
C1D0 A7   A4                STA    ,Y        CLEAR OUT NUMBER BYTES
C1D2 A7   21                STA    1,Y
C1D4 AF   A4                STX    ,Y        STORE VOLUME NUMBER IN FCB
C1D6 8E   C840      WRITE  LDX    #FCB      POINT TO FCB
C1D9 6F   88 1E             CLR    30,X      TRACK NUMBER IN FCB
C1DC 86   03                LDA    #3        SECTOR NUMBER
C1DE A7   88 1F             STA    31,X      STORE IN FCB
C1E1 86   0A                LDA    #10       WRITE ONE SECTOR
C1E3 A7   84                STA    ,X        STORE IN FCB

C1E5 BD   D406              JSR    FMS       CALL FMS
C1E8 26   03                BNE    ERROR     CHECK FOR ERRORS
C1EA 7E   CD03              JMP    WARMS     RETURN TO FLEX

C1ED BD   CD3F      ERROR  JSR    RPTERR    REPORT ERROR
C1F0 BD   D403              JSR    FMSCLS    CLOSE FMS
C1F3 7E   CD03              JMP    WARMS     RETURN TO FLEX

                 * STRING EQUATES

C1F6 43 55 52 52    CURNMS FCC    'CURRENT NAME =   ',$04
C208 4E 45 57 20    NEWNMS FCC    'NEW NAME?        ',$04
C21A 43 55 52 52    CURNMB FCC    'CURRENT NUMBER = ',$04
C22C 4E 45 57 20    NEWNMB FCC    'NEW NUMBER?      ',$04
                 * NAME STORAGE AREA

C23E                STRNAM RMB    128

                           END    CALL
```

Listing 2: Mill Example (Assembly)

```
DEMO PROGRAM FOR THE MILL

                    *   McMill Demo
                    *   Feb 25, 1983
                    *   From Programming the 6809
                    *   Modified for relative addressing

                        ORG    $7FFE
7FFE 0000               FCB    $00,$00
8000 308D0028 BUBBLE    LEAX   BASE,PCR  GET TABLE
8004 C62C               LDB    #LENGTH
8006 5A                 DECB
8007 3085               LEAX   B,X       Point to end
8009 6F8D001E           CLR    EXCHG,PCR Clear exchange flag
800D A684     NEXT      LDA    ,X        A=Current entry
800F A182               CMPA   ,-X       Compare with next
8011 2C0E               BGE    NOSWIT    Go to Noswit if curr.&=next
8013 3404               PSHS   B         Save B
8015 E684               LDB    ,X        Get next
8017 E701               STB    1,X       Store in current
8019 A784               STA    ,X        Store current in next
801B 3504               PULS   B         Restore B
801D 6C8D000A           INC    EXCHG,PCR Set exchange flag
8021 5A       NOSWIT    DECB             Decrement B
8022 26E9               BNE    NEXT      Continue until 0
8024 6D8D0003           TST    EXCHG,PCR Exchanged=0?
8028 26D6               BNE    BUBBLE    Restart if not 0
802A 39                 RTS
802B          EXCHG     RMB    1
              BASE      EQU    *
802C 1A       LENGTH    FCB    26
802D 4744525548         FCC    "GDRUHBSEQIGNVC"
8033 4253455149474E56
803B 43
803B 4B4654554F         FCC    "KFTUOPLFDSQC"
8041 504C4644535143
```

Listing 3: 68000 Example (Assembly)

```
68000 ASSEMBLY LANGUAGE PROGRAM : STRING SEARCH

                        ORG     $2000
002000: 207C00002040    MOVEA.L #MAIN,A0       ;LOW RANGE LIMIT
002006: 227C00002053    MOVEA.L #TEST,A1       ;HIGH RANGE LIMIT
00200C: 7003            MOVE.L  #3,D0          ;TEST STRING LENGTH
00200E: 48A7000A        MOVEM   D1/D3,-(SP)    ;SAVE REGISTERS
002012: 48E70900        MOVEM.L A0/A3,-(SP)    ;ON STACK
                        *
                        * SEARCH FOR FIRST CHAR OF TEST STRING
                        *
002016: 1611            MOVE.B  (A1),D3        ;FIRST TEST CHAR
002018: 95CA            FIRST   SUBA.L  A2,A2  ;CLEAR A2 TO START
00201A: 0C10002A        CHKEND  CMPI.B  #'*',(A0)  ;END OF MAIN?
00201E: 671E            BEQ.S   RETRN          ;YES. GO EXIT
002020: B618            CMP.B   (A0)+,D3       ;FIRST CHAR MATCH?
002022: 66F6            BNE.S   CHKEND         ;NO. KEEP SEARCHING
                        *
                        * FIRST TEST CHAR FOUND, COMPARE REST
                        *
002024: 3200            MOVE    D0,D1          ;STORE TEST LENGTH
002026: 5541            SUBQ    #2,D1          ;D1=COUNT-2
002028: 2649            MOVEA.L A1,A3          ;STORE TEST ADDR.
00202A: 528B            ADDQ.L  #1,A3          ;LOOK AT CHAR TWO
00202C: 2448            MOVEA.L A0,A2          ;STORE MAIN ADDR.
00202E: 538A            SUBQ.L  #1,A2
002030: 0C10002A        LOOP    CMPI.B  #'*',(A0)  ;END OF MAIN?
002034: 6708            BEQ.S   RETRN          ;IF SO, RETURN
002036: B10B            CMPM.B  (A3)+,(A0)+    ;COMPARE CHARS
002038: 66DE            BNE.S   FIRST          ;MATCH FAILED
00203A: 51C9FFF4        DBF     D1,LOOP        ;KEEP COMPARING
00203E: 4E4F            RETRN   TRAP    #15    ;DUMP REGS, EXIT
002040: 57415348494E    MAIN    ASC     'WASHINGTON BULLETS*'
002053: 544F4E          TEST    ASC     'TON'
```

Listing 4: 68000 Example (BASIC)

```
APPLESOFT PROGRAM : UPLOADS MACHINE LANGUAGE PROGRAM
ONTO 68000 BOARD AND EXECUTES IT.

10 A = PEEK (49280): CALL  - 936
20  PRINT CHR$ (4);"BLOAD UTIL2,A$8600,D1"
    :CALL 38383
30  PRINT CHR$ (4);"BLOAD SEARCH.OBJ0,D2,A$6000"
40 P = 38383:Q = P - 102: CALL 38380
50  CALL Q: REM 07 002000 6008 0056
60  CALL Q: REM 02 002000
70  CALL P - 27
80  PRINT
    :PRINT "IF THE TEST STRING WAS FOUND, REGISTER"
90  PRINT "A2 WILL BE  00 00 20 47."
100  PRINT
     :PRINT "IF THE TEST STRING WAS NOT FOUND,"
110  PRINT "REGISTER A2 WILL BE  00 00 00 00120  END
```

MICRO™

Transition Period

Commodore's product line has evolved quickly over the 4½ years since the first PETs were delivered. The heart of the PET is the 6502 microprocessor, designed by Commodore's (then) newly acquired subsidiary, MOS Technology. At that time the company's only computer product was the small-keyboard PET, long since phased out in favor of models with full-sized business and graphics keyboards. From these evolved an 80-column model, which later gave rise to the 96K CBM 8096 and the SuperPET. With the exception of the SuperPET (or the 6809 half of it), all are essentially the same computer.

Then Commodore introduced the VIC-20, billed as the "first full-featured computer for under $300." It was named after its technological centerpiece, an inexpensive CRT controller called VIC (Video Interface Chip). Many things about the VIC-20 are radically different. Some are obvious compromises to obtain a low price: the keyboard has fewer keys and no numeric keypad; the screen consists of only 23 rows of 22 characters; only 5K of RAM is included; a separate TV or monitor is required; the IEEE-488 bus is gone; and no cassette player is included. There are many improvements aimed particularly at the home market (as opposed to the hobbyist market of the PET): color, sound, a controller port, programmable function keys, and program cartridge capability. The VIC, even with all its differences, is easily identifiable as a descendant of the PET. The operating system is essentially the same as the PET and CBM; PET/CBM BASIC programs run on a VIC with relatively few adjustments; and the processor is still a 6502.

Commodore New Wave

by Loren Wright

— Commodore 64 — First in New Wave

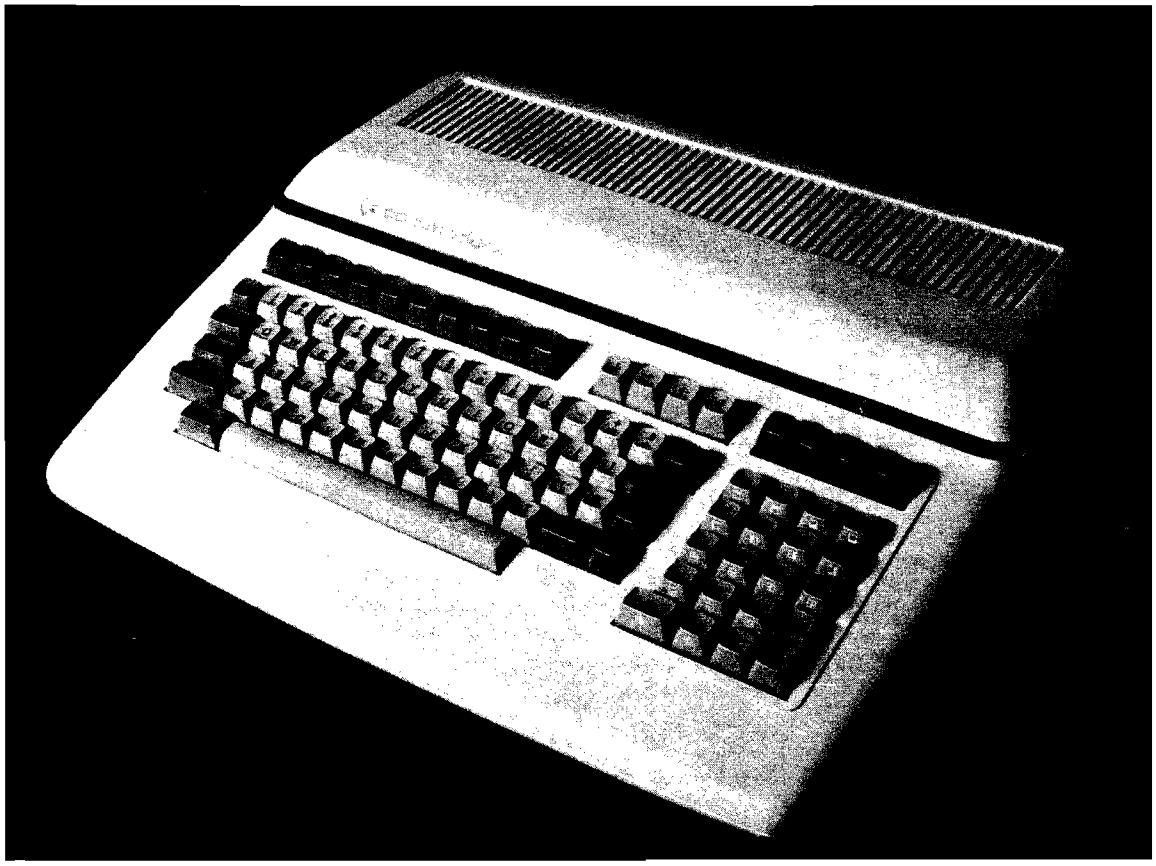
At the same time that Commodore was gearing up for a massive VIC-20 advertising campaign last fall, it introduced the Commodore 64. The Commodore 64 looks very much like the VIC-20 (it even appears that the same mold was used for the case), but the inside is a lot different. Featured are four new chips: the 6510 microprocessor, the 6526 CIA interface chip, the 6581 SID sound chip, and 6567 VIC II video chip.

The 6510 is a 6502 with an added I/O port, which is used to switch RAM, ROM, and I/O in and out of the 64K address space. There is actually 64K of RAM in there that could be used all at once, if you wanted to design your own operating system. The practical advantage is that things like the character generator and I/O chips can share the same address space and get switched in only when needed. Under the BASIC operating system provided, this means that about 8K more RAM is available for program storage. As in the VIC, the most fundamental tasks, such as getting a character from the keyboard and putting a character on the screen, are concentrated in a single 4K ROM called the KERNAL. This design feature, combined with the built-in switching, makes the C64 ideal for such languages as Pascal and FORTH, which have much of their own operating systems built in. When the BASIC ROMs permanently occupy address space as they do in the PET, these languages use additional space, leaving less for programs, or operate from the disk, making operation slow. With the C64, the BASIC ROMs can be switched out and RAM (or cartridge ROM) can be switched into the same space.

The VIC II is a considerable improvement over the VIC. It allows 25 rows of 40 characters, fine scrolling, high-resolution graphics, and eight hardware-maintained movable object blocks (called sprites). With the relative abundance of RAM, high-resolution graphics and user-defined character sets are more practical than they are with the limited memory of the VIC-20. For detailed information on the capabilities of the VIC II chip, see my December 1982 PET Vet column (MICRO 55:54).

The SID is an extremely powerful sound chip with four different waveforms, envelope generators, filters, ring modulation, and synchronization for each of three voices. See my February 1983 PET Vet column (MICRO 57:71) for more on the synthesizer-like capabilities of the SID chip.

The CIA adds a time-of-day clock to the familiar 6522 VIA. Two of them are



Commodore's C 128/80

used in the C64 — one for the keyboard and one for the parallel user port.

The C64 apparently will give rise to a whole line of portable computers designed to compete favorably with the Osborne portable computers. The unit is smaller than the Osborne, and it will be available in three different models: 1. a 5-inch black and white monitor and single 5¼-inch floppy drive, 2. a 5-inch color monitor and single floppy drive, and 3. a color monitor and dual floppy drive. The third will sell for less than $1700; the others less.

This line was previewed in prototype versions at the Consumer Electronics Show in Las Vegas in January. The machines are apparently a C64 complete with 6510, SID, VIC II, and CIA's, but in a different configuration. Further details were unavailable from Commodore, and there is no telling exactly when these machines will appear in stores.

6509 Addresses 1 Megabyte

Three new models, all based on another new processor (the 6509) were announced last year. They have received various names (500 and 700 series; PET II and CMB II; B, P, and BX series), but now it seems that only one, the C 128/80, will appear soon.

Like the C64 and VIC-20, the keyboard and computer are combined in a single low-profile unit. When this unit is connected to a black and white monitor, it produces an 80-column display of 9x14 characters. Both business and PET-graphics character sets are included. The keyboard is expanded considerably. There is a separate numeric keyboard with double-zero, 'clear-entry', and double-size 'enter' keys. There is a row of 10 programmable function keys across the top of the main keyboard. The cursor controls are on four separate keys, rather than only two. Also, unlike the VIC and the C64, the unit includes a switching power supply and 'whisper' fan.

As its name implies, the C 128/80 includes 128K of RAM, and that is expandable internally to 256K and externally to 896K. This is accomplished with the new 6509 microprocessor, which performs its own bank switching to address this much memory. The instruction set is identical to that of the 6502.

The BASIC supplied in 24K of ROM is a further advancement beyond BASIC 4.0 (5.0?). Apparently programs written in BASIC 4.0 will run under the new BASIC.

The C 128/80 has a number of different interfaces. The IEEE-488 interface makes it possible to use any Commodore peripheral, including floppy and hard disk drives and several printers. The IEEE-488 bus is also used by Hewlett-Packard, Tektronix, and Fluke in many of their scientific instruments. The RS-232C interface, accomplished with a 6551 ACIA chip, makes a great number of peripherals available including auto-answer/auto-dial modems and inexpensive printers. Also available is the parallel user port supported by the 6526 CIA chip.

The 6581 sound chip, with all the sound capabilities, is also included.

Breakers on the Horizon

Commodore's Max machine (or Ultimax?) has not appeared yet, perhaps because of the current success of the VIC-20. Also, I presume there will be machines in the same line as the C 128/80, only with different features (a color monitor, included disk drives?).

Commodore showed a prototype of a hand-held computer, but sources indicate that product won't see the light of day.

You may contact Loren Wright at P.O. Box 6502, Amherst, NH 03031.

MICRO™

IS 16-BITS THE SOLUTION ?

by Keith Roberson

Sixteen-bit microprocessors have made a significant impact on the imagination of microcomputer users, for a good reason. But it's just possible that for the novice small business user the distinction is moot, if not actually irrelevant. This article discusses why 8-bit technology, while far from the leading edge, can be expected to have a long and active life in the small computer marketplace.

The 8-bit microprocessor has come a long way since its inception; from a hobbyist's toy to the foundation of the burgeoning business computer marketplace. The inroduction of a new microprocessor generation — the high peformance 16-bit systems — marks another step toward maturity. This event has moved some industry observers to see in it an abrupt end to the product life of the eight-bit business computer system.

The new computers do, in fact, have certain advantages over the older technology. Faster and more powerful than their 8-bit predecessors, they are already making 8-bit technology obsolete.

Or are they?

Ask a potential small business computer buyer for an opinion on the subject. Chances are he won't know what you're talking about — or care.

And why should he?

Most microcomputer business systems buyers are technologically uninformed. Many prefer to keep it that way. Trend-setters in their own right, they accept the microcomputer as a potentially useful tool for solving their business and information management problems. Accountants, clerical workers, executives, or professionals, few have used anything more complex than an adding machine or a hand-held calculator before. They want economical, efficient, and productive solutions to their business problems. The details of a technology that achieves this could hardly interest them less.

They must deal with many factors and are confronted by an incredible number of factors in making a choice among business computers for their own applications.

Asking them to choose between 8-bit and 16-bit microprocessor technology is to add to the complexity of an already enormously confusing mass of variables. Since any such decision in the last analysis must be based on the harsh realities of business, the objective of the selection process is to choose the computer that does the best job in the specific installation, at the lowest cost.

The issue of 8-bit *versus* 16-bit business computers must be secondary to this concern. Nevertheless, the 16-bit computer system is drawing vast amounts of attention to itself. To an uninformed observer, the claims carry a measure of validity. Some discussion of these claims, and a comparison of eight-bit and 16-bit systems may help clarify the issues for prospective business computer buyers.

Bits, Bytes, and MIPS

Sixteen-bit systems are often described as high performance machines. That there are major differences between 8- and 16-bit microprocessors cannot be disputed. However, that does not mean 16-bit technology is the fix-all-problems solution to business microcomputer problems. Many factors can affect overall system performance, and the most ad-

COMPU SENSE

VIC-20®

VIC-20®	**Personal Computer**	$169.95
VIC-1515	**Printer**	334.95
VIC-1530	**Datasette**	67.50
VIC-1541	**Disk Drive**	375.00
VIC-1010	**Expansion Module**	139.95
VIC-1311	**Joystick**	9.95
VIC-1312	**Game Paddles**	19.95
VIC-1600	**Telephone Modem**	99.95
VIC-1210	**VIC 3K Memory Expander Cartridge**	34.95

Plugs directly into the VIC's expansion port. Expands to 8K RAM total.

VIC-1110	**VIC 8K Memory Expander Cartridge**	52.50

8K RAM expansion cartridge plugs directly into the VIC.

CM101	**VIC 16K Memory Expander Cartridge**	99.95
CM102	**24K Memory Expander Cartridge**	119.95
VIC-1011A	**RS232C Terminal Interface**	39.95

Provides interface between the VIC-20 and RS232 telecommunications modems. Connects to VIC's user port.

PETSPEED - Basic Compiler for Commodore 130.00
Compile any Pet Basic program. The only optimizing compiler. Programs compiled with Petspeed run up to 40 times faster. Petspeed code is unlistable and compiled programs cannot be tampered with. No security device required for compiled programs. Available NOW for the Commodore 64.

Star Gemini 10 Printer	Call for price
Star Gemini 15 Printer	Call for price
SMD Monitor	Call for price

CARDBOARD 6 $87.95
An expansion interface for the VIC-20. Allows expansion to 40K or accepts up to six games. May be daisy chained for more versatility.

CARDBOARD 3 $39.95
Economy expansion interface for the VIC-20.

CARD "?" CARD/PRINT $79.95
Universal Centronics Parallel Printer Interface for the VIC-20 or CBM-64. Use an Epson MX-80 or OKIDATA or TANDY or just about any other.

CARDETTE $39.95
Use any standard cassette player/recorder with your VIC-20 or CBM-64.

LIGHT PEN $29.95
A light pen with six good programs to use with your VIC-20 or CBM-64.

HOME & BUSINESS PROGRAMS For VIC-20 & C-64

CW-107A	**Home Calculation Program Pack**	$48.95
CPV-31	**Data Files** - your storage is unlimited	14.95
CPV-96	**Household Finance Package** - to keep records of all your household expenses	30.95
CPV-208	**Bar-Chart** - display your numerical data	8.95
CH	**Turtle Graphics** - learn programming	34.95
CH	**VIC Forth** - is a powerful language for BASIC programming	49.95
CH	**HES MON** - is a 6502 machine language monitor with a mini-assembler	34.95
CH	**HES Writer** - time-saving word processing tool	34.95
CH	**Encoder** - keep your personal records away from prying eyes	34.95
CT-21	**Statistics Sadistics** - statistical analysis	14.95
CT-121	**Total Time Manager 2.0** - creates personal or business schedules	15.95
CT-124	**Totl Label** - a mailing list and label program	13.95
CT-125	**Totl Text BASIC**	15.95
CT-126	**Research Assistant** - keep track of reference data	17.50
CT-140	**Totl Text Enhanced**	29.95
CM-152	**Grafix Designer** - design graphic characters	12.95
CQ-5	**Minimon** - allows you to program, load, save, or execute machine language programs	13.95
CT-3	**Order Tracker**	15.95
CT-4	**Business Inventory** - to maintain record of inventory	15.95
CS	**Home Inventory** - lists your home belongings	17.95
CS	**Check Minder** - (V-20 & 64) keep your checkbook the right way	14.95
CS	**General Ledger** - a complete general ledger	19.95
CHC-504	**HES Writer** - word processor	39.95
CHC-503	**Turtle Graphics II** - utilizes the full graphics of your 64	49.95
CHC-502	**HESMON** - machine language monitor w/mini-assembler	34.95
CHP-102	**6502 Professional Development System**	29.95
CFC	**Data Files** - a management program	27.95
CPV-327	**HESCOM** - transfers data and programs bidirectionally between VICs at three times the speed of a disk drive	40.95
CPV-328	**HESCOUNT** - monitors program execution	19.95
CHV	**HESPLOT** - Hi-res graphics subroutines	12.95
CPV-367	**Conversions** - figures, volume, length, weight, area, and velocity to all possible configurations	7.95
CC	**The Mail** - your complete mail program Cassette / Disk	24.95 / 29.95
CPV-220	**Client Tickler**	16.95
CPV-221	**Club Lister**	13.95
CPV-224	**Depreciator**	9.95
CPV-236	**Investment Analyst** - keep track of investments and investment opportunitites	12.95
CPV-251	**Present Value**	10.95
CPV-269	**Super Broker**	12.95
CPV-270	**Syndicator** - calculates whether to buy or sell	13.95
CPV-274	**Ticker Tape** - maintains investments profile	14.95
CPV-276	**Un-Word Processor** - screen editor	16.95
CPV-286	**Phone Directory** - never lose a phone number again	9.95
CS-111	**Checkbook** - home "utility" program	14.95
CPV-294	**Calendar My Appointments** - print a calendar for every month in any year.	14.95
CPV-296	**The Budgeter** - place your personal finances in order	12.95

CS1 **QUICK BROWN FOX** $60.50
The Word Processor of this decade!

COMMODORE SOFTWARE

VIC-1211A **VIC-20 Super Expander** $57.99
Everything Commodore could pack into one cartridge - 3K RAM memory expansion, high resolution graphics plotting, color, paint and sound commands. Graphic, text, multicolor and music modes. 1024x1024 dot screen plotting. All commands may be typed as new BASIC commands or accessed by hitting one of the VIC's special function keys. Includes tutorial instruction book. Excellent for all programming levels.

VIC-1212 **Programmer's Aid Cartridge** $45.99
More than 20 new BASIC commands help new and experienced programmers renumber, trace and edit BASIC programs. Trace any program line-by-line as it executes, pause to edit. Special KEY command lets programmers redefine function keys as BASIC commands, subroutines or new commands.

VIC-1213 **VICMON Machine Language Monitor** $48.99
Helps machine code programmers write fast, efficient 6502 assembly language programs. Includes one line assembler/disassembler.

NEW GAMES FOR YOUR VIC-20®

CC58	**Astroblitz** - This game is challenging, even to a VIC-MASTER! Navigate your ship carefully to avoid being hit by enemy fire.	$39.95
CC60	**Terraguard** - Speed and careful skill will enable you to once again destroy the aliens. Too slow? You're destroyed by their beam.	39.95
CC98	**Serpentine** - This game will test your patience & skill. Object - to survive long enough to lay eggs and raise your young	39.95
CC500	**Intruder-Scrambler** - In your bomber, invade the defending scramble system, dodging rockets, to blow up enemy posts, etc.	19.95
CC101	**Choplifter** - Rescue the American hostages & return them safely to the U.S. You will encounter tanks, jets and killer satellites.	39.95
CC102	**Black Hole** - Your mission is, simply, to survive! Your ship must not be hit by space objects or sucked into the Black Hole!	39.95
CC104	**Apple Panic** - Speed is required! Destroy the apple monsters by digging holes in the brick floors for them to fall into.	39.95
CC65	**Video Mania** - Introducing your enemies: EVIL EYE, WALWOKER, KILLERBOX. Your only defense - throw your alien zapper!	39.95
CS1	**Flags of Nations** - A game that challenges players to identify flags of various widely-known nations of the world.	10.95
CS2	**Flags of Nations** - Second Edition - A field of 34 flags of lesser known nations of the world	10.95
CS3	**Cities and States** - A game that draws a map of a state or states and asks players to name key cities in those states.	10.95
CS4	**Cities of the World** - Deals with important cities of nations throughout the world.	10.95
CS5	**Mountains and Rivers** - Draws large geographical area maps. You identify major mountain ranges, rivers & bodies of water	10.95

NEW GAMES FOR YOUR C-64

Tank Arcade (Also for VIC-20) - Pre-determine how many hits it will take to wipe out your opponent. Then, on with the battle! Battlefield changes. $13.95

Roadracer - Choose the type of track & a time or lap race. Use steady control at speeds of 50 to 200 miles per hour. Hit the wall & lose valuable time. 13.95

Shootout at the OK Galaxy (Also for VIC-20) - 30 alien warships have entered your war zone. Shields up? Energy level OK? Defend yourself. 19.95

Galaxy - Have you ever wanted to conquer the universe? Send your galactic fleets out to explore, solar system by solar system. From 1 to 20 players. 19.95

Bomber Attack - Ground to air warfare. You're in command of a supersonic bomber over enemy terrain. Drop all 25 bombs on key locations. 14.95

Midway Campaign - Your computer controls a huge force of Japanese ships trying to conquer Midway Island. Your only advantage is surprise. 19.95

Dnieper River Line - A fictionalized engagement between Russian & German forces in 1943. Soviet forces, controlled by the computer, seek to overrun your line and capture sufficient objectives to attain victory. Four levels of difficulty. 25.00

Tanktics - Armored combat on the Eastern front of WWII. You start outnumbered 2 to 1 but you choose your tank types before the battle. 24.50

Guns of Fort Defiance - You are the commander of a 19th artillery piece in a besieged fort. Choose type of ammo. Set the cannon's elevation, deflection. 20.00

Computer Baseball Strategy - You, the manager of the home team, test you skill against a wily and unpredictable opponent, your computer. 15.95

Lords of Karma - Like an intriguing puzzle! Decipher secrets while exploring a mythical, magical city & countryside. Avoid the lurking monsters! 20.00

North Atlantic Convoy Raider - It's the Bismarck convoy raid of 1941! The computer controls the British ships. Will you change history? 19.95

Planet Miners - Compete against others and the computer to stake valuable mining claims throughout the solar system in the year 2050. 19.95

Conflict 2500 - In 2500 AD, earth is threatened by attacking aliens with an infinite # of attack strategies with which to tease the defending player 19.95

Nukewar - Nuclear confrontation between two hypothetical countries. Defend your country with espionage, bombers, missiles, submarines, etc. 19.95

Computer Acquire - New Second Edition! The object is to become the wealthiest person in this "business" game - hotel acquisitions & mergers. 20.00

Andromeda Conquest - Vast scale space strategy game of galactic colonizing and conquest. Strange life forms & alien technologies - exciting! 19.95

Telengard - Microcomputer Dungeon Adventure game. Time fantasy and role-playing. 50 levels of ever-more complex mazes to explore & survive! 25.00

MORE — MORE — MORE

Prices subject to change.

TO ORDER:
P. O. Box 18765
Wichita, KS 67218
(316) 263-1095

WRITE FOR FREE CATALOG

Personal checks accepted (Allow 3 weeks)
or C.O.D. (Add $2) *Handling charges $2.00*

VIC-20® is a registered trademark of Commodore

Circle No. 23

vanced technology is not always the most appropriate.

Comparisons of the two technologies are usually made on the basis of performance. But software availability, expansion capability, and price are factors a buyer must definitely consider in the process of selecting a high-ticket capital item like a computer that will be used for several years. And performance itself must be viewed within the parameters of realistic settings and the constraints these impose on a system.

Performance

Perhaps the most widely used — and most misleading — measure of a computer's performance is the number of binary digits (bits) the Central Processing Unit (CPU) can access or manipulate at one time. Bits constitute the atoms of a computer. They are processed in clusters (words) of 8, 16, 32, 64, or more at one time. The larger the word, the more information a computer can process in a given time. Hence the concern over word size.

A 16-bit processor such as the Intel 8086 has twice the word size of Zilog's 8-bit Z80, but that does not mean it has twice the performance. Sixteen-bit microprocessor chips come in many forms and configurations. The Intel 8088, for example, processes a 16-bit word internally, but transfers it *via* an 8-bit external data path (bus), in order to allow designers of 16-bit systems to incorporate it into existing 8-bit designs. On the other hand, Motorola's 68000 has a 32-bit internal architecture and a 16-bit bus. yet both are considered 16-bit microprocessors.

The performance of these wider bandwidth computers is nearly always limited by their application. The application itself can impose conditions that nullify any performance advantage provided by a state of the art technology. For example, word processing — which accounts for the bulk of current small business microcomputer applications — involves the extensive manipulation of alphanumeric data. But alphanumeric data is represented in an 8-bit format. In such an application, the capabilities of a 16-bit microprocessor are grossly underutilized. The buyer ends up paying for excessive unused capability. However, extensive scientific calculations — such as Fourier transforms, for example — could show dramatic increases in through-put by utilizing the wider internal bandwidth of 16-bit processors.

MIPs and Clock Speed

A better performance benchmark than word size, or the number of bits in a system, is the number of instructions or operations a processor can execute in a given amount of time.

The unit of measure generally used is MIPS (million instructions per second). This value is largely dependent on the CPU clock speed. Typically, the new 16-bit processors have clock speeds about 50% faster than 8-bit chips, and have bus widths that in many instances allow them to handle twice as much data at a time.

Faster clock speeds account for most of the perceived performance increases of 16-bit microprocessors over their 8-bit predecessors. The Intel 8086, for example, runs at 6 MHz. Taking into account its word size, and a slightly more powerful instruction set, it could be said to run twice as fast as an 8-bit Z80A running at 4 MHz. However, in many applications the 8086 is comparable in performance to a Z80B running at 6 MHz, and slower than a Z80H running at 8 MHz.

Internal and External Memory

One of the indisputable advantages of a 16-bit system is its inherent ability to address or access more random access memory (RAM or main memory), typically a megabyte or more, compared to the 64-kilobyte limit of the 8-bit chips. However, advances made through a technique called bank switching, allow 8-bit micros to access many times their nominal 64K limit.

A large internal memory makes it possible to design a comuter that can use a sophisticated operating system (OS) — a program always resident in RAM that oversees the loading and execution of the application programs. A large RAM also reduces the need for frequent access to the much slower disk memory by keeping larger portions of program and data in the much faster RAM.

But RAM is one of the most expensive parts of a computer system. And a megabyte of internal memory is far in excess of what today's most popular applications programs are designed to use.

Most business application programs available today are disk intensive. That is, they rely heavily on disk media for external memory or data storage. In these popular configurations, the critical path to high performance is not RAM size but speed of the disk storage unit. This is not at all affected by the CPU, whether it be 8-bit or 16-bit.

There are two main types of disks and disk drives: hard and floppy. A hard disk costs about five times as much as a floppy, is orders of magnitude faster, and offers on-line storage accessibility measured in megabytes, as opposed to the kilobytes of storage offered by floppies.

Two other features of fast disks that are not remarkably affected by the use of 16-bit microprocessors are direct memory access (DMA) — which optimizes the transfer time from disk to memory, and buffering, the use of a temporary memory as a holding area to reduce frequent disk access.

Software

Most often considered a separate issue, software is intimately associated with performace. The finest hardware is useless without strong applications and systems software. What sometimes appears to be a troublesome performance problem is often resolved through the use of redesigned software.

Eight-bit based systems enjoy a distinct advantage here, especially when coupled with Digital Research's CP/M operating system (or a compatible variation of it), the industry's de-

In these popular configurations, the critical path to high performance is not RAM size but speed of the disk storage unit. This is not at all affected by the CPU, whether it be 8-bit or 16-bit.

facto operating system. Thousands of CP/M applications exist for 8-bit systems, written and tested over the many years since microcomputers emerged as a valid tool for use in business and other settings. Compared to the few hundred applications presently available for 16-bit machines, the advantage of implementing an 8-bit system is clear, where the objective is to be up and running with the largest selection of application software from which to select suitable programs.

Another factor to consider in evaluating the appropriateness of 16-bit over 8-bit installations, is the fact that the bulk of 16-bit programs are derived from direct analogs of the 8-bit versions and do not really take advantage of the 16-bit's potential for increased performance. While this may not be an important concern for users who plan to develop their own applications programs, it can be a significant consideration for those whose objective is to use the new machine as soon as possible to solve pressing business or professional problems.

Interestingly enough, many of the most talked about recent advances in applications software (such as multi-user and shared data base programs) were developed on 8-bit machines and then ported over to the 16's. This is not quite so strange as it sounds; most 16-bit machine manufacturers still rely on 8-bit computer sales for their bread and butter.

Expandability and Multi-user Capability

The ability to expand — whether it be the size of internal or external memory, or the number of users it will support — is a frequently over-looked but important measure of performance.

The decision on which system to obtain must be based as much on current needs as on the needs of the predictable future.

What if more users need to access the system? What will happen to system performance if more users are added? Is the current storage capacity adequate? If not, can the storage capacity be expanded? Will it be possible to modify the system readily, or must it be scrapped and another system selected to replace it?

In short, is it a system with a reasonable productive life in the setting where it will be used?

For most serious business computer users, multi-user capability is a must. As a business grows, more ways to use a computer are soon discovered. Within a surprisingly short time, it becomes evident that a single-user system — be it a 16-bit business computer or an 8-bit personal computer enhanced to its limits to fit into such a setting — is inadequate.

Along with the increased number of applications, the need for easy access to common files and the most recent information also grows.

If anticipated computer usage for the future justifies selecting a system that can easily accommodate additional users and memory expansion, at least two major options are currently available: additional single-user workstations or multi-user systems.

Multi-user systems are available in two architectures: *time sharing* and *distributed processing*.

Time-Sharing

Time-sharing allows several users to share a single computer apparently simultaneously. The system behaves as if the computer serves only that one user, when, in fact, the user controls only one time slice of many. Time-sharing has its origins in traditional mainframe and minicomputer operations, and is based on the most economical utilization of expensive resources.

A user program takes control of a CPU, does some processing for a while, then gives up the CPU to another user.

The immediate cost advantage of time-sharing is the ability to add more users without significant hardware expenditures. However, performance rapidly deteriorates as more users are added. Even 16-bit microprocessors show significant degradation, sometimes slowing down to unacceptable response rates with the addition of only one or two users.

Additional overhead is also paid in performance and memory space for the operating system to handle such necessary functions as memory management, scheduling and device polling.

Distributed processing

In distributed processing, several microcomputers are linked together to perform like one big computer. users may be added without significant degradation of system performance.

The costs associated with extensive cabling, stand-alone computer installations and their maintenance, must be evaluated against the advantages of this approach. Cost effectiveness is always an important consideration in any decision involving a business.

An interesting compromise to both types of systems has been made by a number of systems that utilize a dual processor architecture. users share 8-and 16-bit processors transparently; indeed, users may run 8-bit and 16-bit on the same system concurrently. This provides the opportunity to use 8-bit software available today, and offers a viable migration path to the 16-bit as more sophisticated software for it becomes available. However, true parallel operation is not generally possible on these systems. Most of them use tightly-coupled processors, and, in most cases, only one processor is running at a time. This reduces the cost/benefit factor to the user by providing only very slight performance advantage over a single CPU system.

An easy-to-use 16-bit operating system may be hampered by a lack of software.... Even 16-bit microprocessors show significant degradation sometimes slowing down to unacceptable response rates....

All things considered, buyers of business computer systems may need to pay more attention to future expansion capability than processing speed when evaluating the performance benefits of 8-bit *versus* 16-bit computers.

Price Considerations

If price were the only consideration, 8-bit machines would win hands down, since they make up virtually the entire low-priced end of the market. However, price must be coupled with performance; an Apple may be bought

for under $2000, but could cost twice as much in additional hardware to perform the most simple business functions at reasonable levels. Nor will it ever be able to do the things a true business computer can do. By the same token, an IBM mainframe could do it all, but the cost would not be justified.

Picking a system with the best price performance ratio is a matter of deciding which features are needed and finding the lowest-priced system with those features.

Other Considerations

Possibly the worst obstacle to more rapid integration of microcomputers into the society can be found in the industry's lack of standardization. Incompatible and untried operating systems abound. Users often are unwittingly obliged to serve as a test site.

Among multiuser systems, some operating systems may be promising at this point of their development. It remains for the individual user to evaluate the benefits of any operating system, whether it be MS/DOS, CP/M86, UNIX, or a UNIX look-alike, running in a 16-bit environment, over a CP/M, or CP/M compatible, operating system running in an 8-bit environment. Each approach has it benefits, along with its disadvantages. An easy-to-use 16-bit operating system may be hampered by a lack of software. A multiuser MP/M operating system may be forcing the 8-bit microprocessor to function at the extremes of its capability, leaving it vulnerable to malfunctions.

A recently developed approach has been adopted by several manufacturers, who have designed multi-user networked operating systems that are compatible with CP/M. These include MUSE, CP/NET, MMOST, NSTAR and several others.

Given the need for caution in making capital investments, the probability remains that eight-bit operating systems with CP/M capability, will continue to remain the preferred choice of business computer users for some time to come.

The Solution

Today's small-business computer users are looking for ways to handle their work more efficiently. The 16-bit systems are appropriate in some applications, but in many more the best choice may depend more on established 8-bit technology.

There is no question that the bigger processors will eventually replace the 8-bit processors. The current lack of 16-bit software, however, and the predictable need to expand capacity, combine to make high performance distributed processing systems a viable alternative while waiting for the microcomputer to mature more fully. Indeed, some industry analysts think that the new 32-bit processors will take over before the 16-bits have a chance to take hold, and most are now in agreement that the 8-bit will still be around for many years to come.

Z80, Z80A, Z80B, Z80H are registered trademarks of Zilog. **8086, 8088** are registered trademarks of Intel Corporation. **CP/M** and **MP/M** are registered trademarks of Digital Research Corporation.® **UNIX** is a trademark of Bell Laboratories. **Zeus4** is a registered trademark of OSM Corporation.

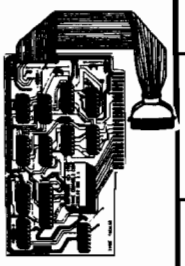
You may contact Keith Roberson at OSM Computer Corporation, Mountain View, CA.

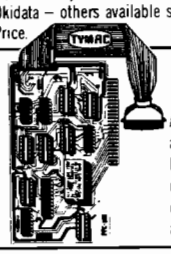
MICRO™

APPLE HARDWARE

THE TACKLER ™ – DUAL • MODE PARALLEL INTERFACE FOR THE APPLE® 2 BOARDS IN ONE FOR NO MORE COMPATIBILITY PROBLEMS!

An intelligent board to provide easy control of your printer's full potential. Plus a standard parallel board at the flip of a switch – your assurance of compatibility with essentially all software for the APPLE®. Hires printing with simple keyboard commands that replace hard to use software routines. No disks to load. Special features include inverse, doubled, and rotated graphics and many text control features, available through easy keyboard or software commands. Uses Industry standard graphics commands. This is the first truly universal intelligent parallel interface! Change printers – no need to buy another board. Just plug in one of our ROM'S and you're all set. ROM'S available for Epson, C. Itoh, NEC, and Okidata – others available soon. Specify printer when ordering. Call for Price.



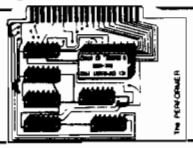


THE UPGRADEABLE PPC-100 PARALLEL PRINTER CARD

A Universal Centronics type parallel printer board complete with cable and connector. This unique board allows you to turn on and off the high bit so that you can access additional features in many printers. Easily upgradeable to a fully intelligent printer board with graphics and text dumps. Use with EPSON, C. ITOH, ANADEX, STAR-WRITER, NEC, OKI and others with standard Centronics configuration. **$139.00**

IF YOU WANT GRAPHICS AND FORMATTING THEN CHOOSE THE PERFORMER

for Epson, OKI, NEC 8023, C. ITOH 8510 provides resident HIRES screen dump and print formatting in firmware. Plugs into Apple slot and easy access to all printer fonts through menu with PR# command. Use with standard printer cards to add intelligence. **$49.00** specify printer.



THE MIRROR FIRMWARE FOR NOVATION APPLE CAT II®

The Data Communication Handler ROM Emulates syntax of an other popular Apple Modem product with improvements. Plugs directly on Apple CAT II Board. Supports Videx and Smarterm 80 column cards, touch tone and rotary dial, remote terminal, voice toggle, easy printer access and much more. List $39.00 Introductory Price **$29.00**

MINI ROM BOARDS

Place your 2K program on our Mini Rom Board. Room for one 2716 EPROM. Use in any slot but zero. Only **$34.95**

Circle No. 24

DOUBLE DOS Plus

A piggy-back board that plugs into the disk-controller card so that you can switch select between DOS 3.2 and DOS 3.3 DOUBLE DOS Plus requires APPLE DOS ROMS. **$39.00**

SOFTWARE APPLE

Super Pix

Hires screendump software for the Epson, OKI, C. Itoh and Nec 8023. Use with Tymac PPC-100. Special **$19.95** (Specify Printer)

***Mr. Lister* – Customer Contact Profiler & Mailer**

A Super Mail List Plus more — up to 1000 Entries on single 3.3 Disk (only 1 Drive required) — 2 second access time to any name — full sort capabilities — Dual Index Modes — supports new 9 digit Zip. Easy to follow manual — Not Copy Protected — 4 user defined tables with 26 sort selections per table — Beta tested for 6 months — user defined label generation. Introductory Price $135. **$99.00** Dealer &: Dist. Inquires Invited.

APPLE LINK

A communications system for the Apple® (Requires Hayes Micro Modem). Transmit and receive any type of file between APPLES®, Automatic multi-file transfer, real time clock indicating file transfer time. Complete error check. Plus conversation mode. Only one package needed for full transfers. Compatable with all DOS file types. (requires Hayes Micro Modem) **$59.00**

THE APPLE CARD/ATARI CARD

Two sided 100% plastic reference card Loaded with information of interest to all Apple and Atari owners. **$3.98**

NIBBLES AWAY II

AGAIN! Ahead of all others.

- **AUTO-LOAD PARAMETERS** . . . Free's the user from having to Manually Key in Param values used with the more popular software packages available for the Apple II.
- **EXPANDED USER MANUAL** . . . incorporates new Tutorials for all levels of expertice; Beginners Flowchart for 'where do I begin' to 'Advanced Disk Analysis' is included.
- **TRACK/SECTOR EDITOR** . . . An all new Track/Sector Editor, including the following features: Read, Write, Insert, Delete Search, and impressive Print capabilities!
- **DISK DIAGNOSTICS** . . Checks such things as: Drive Speed, Diskette Media Reliability, and Erasing Diskettes.
- **HIGHEST RATED** . . . Best back up Program in Softalk Poll (Rated 8.25 out of 10).
- **CONTINUAL UPDATES** . . . Available from Computer Applications and new listings on the source. **$69.95**

Dealer and Distributor Inquiries Invited.

VISA MasterCard

MICRO-WARE DIST. INC.
P.O. BOX 113 POMPTON PLAINS, N.J. 07444
201-838-9027

Alspa Computer, Inc.

The price-performance leader. Includes Z80A, 1 or 2 full 8" drives (double density, double sided), 3 serial and 1 parallel port, and winchester port. Prices start at less than $2000. DEALER and OEM inquiries invited.

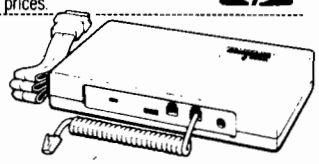
SPECIALS on INTREGATED CIRCUITS

6502	7.45	10/ 6.95	50/ 6.55	100/ 6.15
6502A/6512A	8.40	10/ 7.95	50/ 7.35	100/ 6.90
6520 PIA	5.15	10/ 4.90	50/ 4.45	100/ 4.15
6522 VIA	6.45	10/ 6.10	50/ 5.75	100/ 5.45
6532	7.90	10/ 7.40	50/ 7.00	100/ 6.60
2114-L200		2.45	25/ 2.30	100/ 2.15
2716 EPROM		4.90	5/ 4.50	10/ 4.00
2532 EPROM		7.90	5/ 7.45	10/ 6.90
6116 2K×8 CMOS RAM		7.90	5/ 7.45	10/ 6.90
4116 RAM		8 for 14		
Zero Insertion Force 24 pin Socket (Scanbe)				2.00

Hewlett Packard

Write or call for prices.

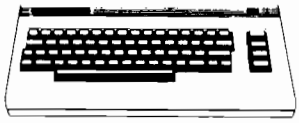
Anchor Automation Signalman Modems



FREE SOURCE MEMBERSHIP WITH SIGNALMAN

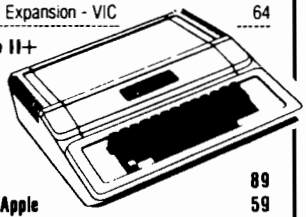
All Signalman Modems are Direct Connect, and include cables to connect to your computer and to the telephone. Signalman Modems provide the best price-performance values, and start at less than $100. **Dealer and OEM inquiries invited**

Mark I RS232	(99)	79
Mark II for Atari 850	(99)	79
Mark IV for CBM/PET with software	(169)	119
Mark V for Osborne (software available)	(129)	93
Mark VI for IBM Personal Computer	(279)	195
Mark VII Auto Dial/Auto Answer	(179)	119
Mark VIII Bell 212 Auto Dial/Answer	(399)	319
DC HAYES Smartmodem		**219**
DC Hayes Smartmodem 1200		545



Apple Emulator for Commodore 64	**89**
Screenmaker 80 COLUMN CARD for C64	**155**
Solid Oak 2 Level Stand for C64 or VIC	**29**
C64/VIC Switch (networking)	135
BACKUP V1.0 tape copier for C64 or VIC	**20**
CARDBOARD/6 Motherboard - VIC	79
CARDAPTER/1 Atari VCS Adapter - VIC	69
CARDPRINT Printer Interface - C64/VIC	64
CARDBOARD/3s Motherboard - VIC	32
CARDETTE/1 Cassette Interface - C64/VIC	32
CARDRITER Lightpen - C64/VIC	32
CARDRAM/16 RAM Expansion - VIC	64

We carry Apple II+ from Bell & Howell



Apple Emulator for Commodore 64	**89**
16K RAM Card for Apple	**59**
Solid Oak 2 Level Stand for Apple	**29**
Super Serial Card	149
Thunderclock Plus	119
Z80 Softcard and CP/M (Microsoft)	235
Parallel Printer Interface/Cable	79
Apple Dumpling (Microtek) Printer Interface	**115**
Apple Dumpling with 16K Buffer	**160**
Grappler + Interface	140
TG Products for Apple in stock	
DC Hayes Micromodem II	299
Videx 80 Column Card	239
Hayden Software for Apple 20% OFF	
PIE Writer Word Processor	**120**

commodore

See us for Personal, Business, and Educational requirements. Educational Discounts available.

PETSCAN $245 base price

Allows you to connect up to 30 CBM/PET Computers to shared disk drives and printers. Completely transparent to the user. Perfect for schools or multiple word processing configurations. Base configuration supports 2 computers. Additional computer hookups $100 each.

Commodore COMMUNICATES!

COMPACK $115

Intelligent Terminal Package includes: ACIA hardware based interface; DB25 Cable and STCP Software with remote telemetry, transfer to/from disk, printer output, XON-XOFF control, user program control, status line.

VE-2 IEEE to Parallel Interface 110

Includes case, power supply, full 8-bit transmission, and switch selectable character conversion to ASCII.

VIC 20 Products		VIC Sargon II Chess	32
BACKUP V1.0	**20**	**VIC GORF**	**32**
VIC RAM Cards in stock		**Meteor Run (UMI)**	**39**
VIC SuperExpander	**53**	**VIC Radar Ratrace**	**24**
VIC 16K RAM	**69**	**Amok (UMI)**	**20**
Thorn EMI Software		Snakman	15
HES Software		Rubik's Cube	13
VIC Omega Race	32	Programmers Reference	15
Spiders of Mars (UMI)	39	Renaissance (UMI)	39
Programmers Aid	45	VIC Adventure Series	

VICTORY Software for VIC and C64

Street Sweepers	12	Maze in 3-D	12
Night Rider	11	Cosmic Debris	12
Treasures of Bat Cave	12	Grave Robbers Advent.	11
Games Pack I	12	Games Pack II	12
Victory Casino	8	Adventure Pack I	12
Adventure Pack II	12	Trek	11

Commodore 64 Programmers Reference Guide	17
Compute!'s First Book of PET/CBM	11
POWER ROM Utilities for PET/CBM	78
WordPro 3+ - 32K CBM, disk, printer	195
WordPro 3+/64	**69**
WordPro 4+ - 8032, disk, printer	300
SPELLMASTER spelling checker for WordPro	**170**
VISICALC for PET, ATARI, or Apple	190
PETRAX PET to Epson Graphics Software	**40**
SM-KIT enhanced PET/CBM ROM Utilities	**35**
Programmers Toolkit - PET ROM Utilities	35
Branding Iron EPROM Programmer for CBM	**75**
PET Spacemaker II ROM Switch	36
2 Meter PET to IEEE or IEEE to IEEE Cable	40
Dust Cover for PET, CBM, 4040, or 8050	8
VIC or C64 Parallel Printer Interface	64
CmC IEEE-RS232 Printer Interface — PET	120
SADI Intelligent IEEE-RS232 or parallel	235
ZRAM - CBM 64K RAM, Z80, CP/M	550
Programming the PET/CBM (Compute!) — R. West	**20**
Compute! First Book of VIC	11
Whole PET Catalog (Midnight Gazette)	**8**
Color Chart Video Board for PET	125
PET Fun and Games (Cursor)	11
Disk ICU—Disk Recovery System for PET/CBM	40
Copy-Writer Word Processor for C64, PET/CBM	70

FlexFile for PET CBM, C64 $110

Database, Report Writer with Calculations, Mailing Lists

FORTH for PET full FIG model — Cargill/Riley	**$50**
Metacompilor for FORTH for independent object code	**30**
KMMM PASCAL for PET/CBM/C64	**85**
EARL for PET/CBM Disk-based ASSEMBLER	**65**
Super Graphics — BASIC Language Exercises	**45**
Fast machine language graphics routines for PET/CBM	
RAM/ROM for PET/CBM	4K $75 8K $90

DISK SPECIALS

Scotch (3M) 5" ss/dd	10/ 2.25	50/ 2.10	100/ 2.05
Scotch (3M) 5" ds/dd	10/ 3.15	50/ 2.90	100/ 2.85
Scotch (3M) 8" ss/sd	10/ 2.40	50/ 2.20	100/ 2.15
Scotch (3M) 8" ss/dd	10/ 2.95	50/ 2.70	100/ 2.65

We stock VERBATIM DISKS

Write for Dealer and OEM prices.

BASF 5" or 8"	10/ 2.00	20/ 1.95	100/ 1.85
NEW BASF Qualimetric Disks also in stock			
Wabash 5"ss/sd	10/ 1.80	50/ 1.75	100/ 1.70
Wabash 5" ss/dd	10/ 2.00	50/ 1.95	100/ 1.90
Wabash 8" ss/sd	10/ 2.00	50/ 1.95	100/ 1.90

We stock MAXELL DISKS

Write for dealer and OEM prices.

Disk Storage Pages	10 for $5	Hub Rings 50 for $6
Disk Library Cases	8"—3.00	5"—2.25
Head Cleaning Kits	11	

CASSETTES—AGFA PE-611 PREMIUM

High output, low noise, 5 screw housings.

C-10	10/ .61	50/ .58	100/ .50
C-30	10/ .85	50/ .82	100/ .70

SPECIALS

Zenith ZVM-121 Green Phosphor Monitor	100
BMC 12A 12" Green Monitor	80
VOTRAX Personal Speech System	280
VOTRAX Type-N-Talk	160
VOICE BOX Speech Synthesizer (Apple or Atari)	
Prowriter 8510 parallel	389
Okidata 92	520
Daisywriter 2000	1050
Many printers available (Star-Gemini, Brother, OKI, etc.)	
We Stock AMDEK Monitors	
Amdek DXY-100 Plotter	600
A P Products	15% OFF
Watanabe Intelligent Plotter 990	6-pen 1290
ISOBAR 4 Outlet Surge Suppressor/Noise Filter	**49**
We stock Electrohome Monitors	
dBASE II (8" format)	300
Panasonic TR-120M1P 12" Monitor (20 MHz)	149
Panasonic CT-160 Dual Mode Color Monitor	285

USI Video Monitors—Green or AMBER 20 MHz hi-res. Dealer and OEM inquiries invited

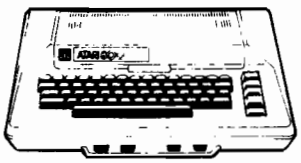
ALL BOOK and SOFTWARE PRICES DISCOUNTED

Synertek SYM-1 Microcomputer SALE 189
KTM-2/80 Synertek Video and Keyboard 349

ZENITH data systems

Z29 Terminal (new detached keyboard)	**680**
ZT-1 Intelligent Communications Terminal with Monitor	**479**
Z100 16-bit/8-bit System	**CALL**

We stock entire Zenith line.



ATARI SPECIALS

WE STOCK ENTIRE LINE—write for prices.

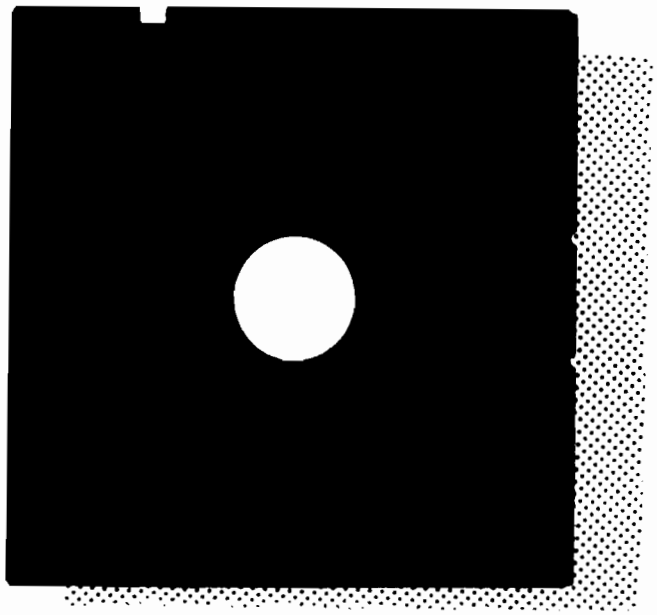
Atari 1200	**749**	**QIX**	**34**
Voice Box	100	**Anchor Modem—Atari**	**79**
Inside Atari DOS	18	**Atari Graphics** (Compute!)	11
Thorn EMI Software		**First Book of Atari**	11
EduFun Software		**APX Software**	

252 Bethlehem Pike
Colmar, PA 18915 215-822-7727 **A B Computers**

WRITE FOR CATALOG

Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders. Prices listed are on cash discount basis. Regular prices slightly higher. Prices subject to change.

GET MORE DATA ON YOUR CBM DISK



by Werner Kolbe

The Commodore operating systems write numerical data to disk as ASCII strings. This article shows you how to write up to four times more numerical data on the CBM disk using the computer's internal binary format instead of ASCII strings.

The data sheet for your floppy disk system tells you how many bytes you can store on a disk, but normally it does not tell you how effectively the system uses the space to store your data. In the case of the PET/CBM disks up to four times more data can be stored when a compressed format is used for storing numbers.

I use my Commodore PET in conjunction with a 4040 floppy disk system to store huge amounts of statistical data upon dogs' tooth failures. These data are mostly integers and only a few are large enough to require floating-point format. I noticed that these data, after they were written on the disk, needed more bytes than they had used in my computer. The reason is simple: the data files on disk or tape are represented in ASCII format, whereas in the computer they are stored more compactly. Storing the data in a compressed form is not less reliable than storing it in the ASCII code. In both cases you will get a wrong number (or read error) if the wrong byte is read from disk.

The operating system of the PET treats all external devices in the same manner, and so it sends data to the disk in the same format as it sends data to the printer, which is the ASCII format. Thus, an integer that can be represented by two bytes may need as many as eight bytes. You could save three-quarters of the space by writing the data in the same compressed format on the disk as it is stored in the computer.

The simplest and fastest way to store compressed data is to assign the value of the number that will be stored on disk to a special variable and then PEEK the storage area of that variable. The BASIC interpreter allocates storage area to the simple variables (not arrays) in the sequence of their occurrence behind the BASIC program, and so it is a good idea to use the special variable as the first variable of your program. This variable will then be located directly behind the BASIC program and can be PEEKed easily, using the contents of RAM locations 42, 43 that point to the beginning of the variable list.

The same principle is used to read the data back. Using GET, you read the appropriate number of bytes from the disk, then POKE the number into the storage area of the special variable. The latter will return the value of the number.

Listing 1, Compress Example 1, shows you how this method works if you want to store integers on the disk. The special variable is called XX%, but it will work with other names as well. The % identifier must be there to ensure that its content is represented by only two bytes. The variable YY points to the first of these two bytes, which you must send to the disk to write a number. To read it back into the computer POKE the ASCII values of the characters received from the disk into the storage area of the dummy variable XX%. The example writes two files on the disk to demonstrate the difference in the space needed for the same numbers. Then the compressed numbers are read back into the computer to complete the process.

Listing 2, Compress Example 2, shows how the compression can be done for floating-point variables. The only difference is that now the special variable must be of the floating-point type and that five bytes are used to represent a number. This time the space used on the disk is about three times less.

Finally, listing 3 shows how you can write both compressed integers and floating-point numbers within the same program. This time the variable XX is second in the list and you get the location of its storage area by adding 7 to the pointer YY%.

The example programs will run on all ROMS of the PET/CBM, VIC, and Commodore 64 computers. I do not know how other personal computers store numbers on disk, but if it is done in ASCII then a similar procedure could be used to increase the available space on disk.

You may contact Werner Kolbe at Vd Kamlaan 65, 2625 Kn Delft, Netherlands.

LISTING 1

```
100 REM EXAMPLE HOW TO COMPRESS INTEGERS
110 REM ------------------------------
120 REM
130 REM TWO SAMPLE FILES ARE WRITTEN.
140 REM ONE IN STANDARD ASCII FORMAT
150 REM AND ANOTHER IN COMPRESSED FORMAT.
160 REM THE COMPRESSED VERSION NEEDS ABOUT
170 REM FOUR TIMES LESS SPACE ON THE DISK
180 REM
190 REM XX% IS THE DUMMY VARIABLE
200 REM WHICH MUST BE THE FIRST VARIABLE
210 REM IN THE PROGRAM.
220 REM
230 REM YY POINTS TO THE PLACE WHERE
240 REM THE VALUE OF XX% IS STORED IN
250 REM INTERNAL COMPRESSED FORMAT.
260 REM
270 XX%=0
280 YY=PEEK(42)+256*PEEK(43)+2
290 REM
300 REM
310 OPEN1,8,3,"@1:STANDARD,S,W"
320 OPEN2,8,4,"@1:COMPRESSED,S,W"
330 REM
340 A=15E3
350 REM
360 REM WRITE 1000 NUMBERS
370 REM
380 FORI=ATOA+1E3
390 REM
400 REM 1. USING THE STANDARD METHOD
410 REM
420 PRINT#1,I;CHR$(13);
430 REM
440 REM 2. IN COMPRESSED FORMAT
450 REM
460 REM ASIGN IT TO THE DUMMY VARIABLE
470 XX%=I
480 REM
490 REM AND WRITE PETS INTERNAL REPRESENTATION
500 REM AS CHARACTERS ONTO THE DISK
510 REM
520 FORJ=0TO1:PRINT#2,CHR$(PEEK(YY+J));:NEXTJ
530 NEXT I
540 CLOSE1:CLOSE2
550 REM
560 REM
570 REM HERE I SHOW HOW TO READ BACK
580 REM THE COMPRESSED DATA.
590 REM
600 OPEN2,8,4,"1:COMPRESSED,S,R"
610 REM
620 REM WE GET A PAIR OF TWO CHARACTERS
630 REM AND POKE IT INTO THE STORAGE
640 REM AREA OF XX%.
650 REM
660 FORJ=0TO1
670 GET#2,A$:IFA$=""THENA$=CHR$(0)
680 POKE(YY+J),ASC(A$)
690 NEXTJ
700 REM
710 REM XX% NOW CONTAINS THE NUMBER
720 REM WHICH WE CAN USE.
730 REM
740 PRINTXX%:IFST=0THEN660
750 CLOSE2
READY.
```

COMPRESS
requires:
PET, CBM, VIC, or
Commodore 64 and a
Commodore disk drive

LISTING 2

```
100 REM EXAMPLE HOW TO COMPRESS FLOATING
110 REM POINT NUMBERS.
120 REM
130 REM
140 REM XX IS THE DUMMY VARIABLE
150 REM
160 REM AGAIN WE WRITE 1000 NUMBERS
170 REM IN STANDARD AND IN COMPRESSED
180 REM FORMAT ONTO THE DISK
190 REM
200 REM
210 XX=0:YY=PEEK(42)+256*PEEK(43)+2
220 OPEN1,8,3,"@1:STANDARD,S,W"
230 OPEN2,8,4,"@1:COMPRESSED,S,W"
240 FORI=1TO1000
250 A=RND(1)/100
260 PRINT#1,A;CHR$(13);
270 XX=A
280 FORJ=0TO4:PRINT#2,CHR$(PEEK(YY+J));:NEXTJ
290 NEXT I
300 CLOSE1:CLOSE2
310 REM
320 REM AND HOW TO READ IT BACK
330 REM
340 OPEN2,8,4,"1:COMPRESSED,S,R"
350 FORJ=0TO4
360 GET#2,A$:IFA$=""THENA$=CHR$(0)
370 POKE(YY+J),ASC(A$)
380 NEXTJ
390 K=XX:PRINTK:IFST=0THEN350
400 CLOSE2
READY.
```

LISTING 3

```
100 REM EXAMPLE HOW TO COMPRESS BOTH
110 REM INTEGERS AND FLOATING POINT
120 REM NUMBERS.
130 REM
140 REM XX% AND XX ARE DUMMY VARIABELS
150 REM
160 REM YY% AND YY ARE THE POINTERS TO
170 REM THEIR STORAGE AREA.
180 REM
190 XX%=0:XX=0
200 YY%=PEEK(42)+256*PEEK(43)+2
210 YY=YY%+7
220 REM XX IS THE SECOND VARIABLE
230 REM
240 OPEN2,8,4,"@1:MIXTURE,S,W"
250 FORI=1TO100
260 A=I+RND(1)
270 REM WRITE IN PAIRS I,A
280 REM FIRST I AS COMPRESSED INTEGER
290 REM
300 XX%=I:FORJ=0TO1:A$=CHR$(PEEK(YY%+J)):PRINT#2,A$;:NEXT
310 REM
320 REM AND THEN A
330 XX=A:FORJ=0TO4:PRINT#2,CHR$(PEEK(YY+J));:NEXTJ
340 NEXT I
350 CLOSE1:CLOSE2
360 REM
370 REM AND HOW TO READ IT BACK
380 REM
390 OPEN2,8,4,"1:MIXTURE,S,R"
391 REM
392 REM FIRST THE INTEGER
393 REM
400 FORJ=0TO1
410 GET#2,A$:IFA$=""THENA$=CHR$(0)
420 POKE(YY%+J),ASC(A$)
430 NEXTJ
440 I=XX%
441 REM
442 REM AND THEN THE FLOATING POINT
443 REM
450 FORJ=0TO4
460 GET#2,A$:IFA$=""THENA$=CHR$(0)
470 POKE(YY+J),ASC(A$)
480 NEXTJ
490 A=XX:PRINTI,A:IFST=0THEN400
500 CLOSE2
READY.
```

MICRO™

Parameter Passing in Assembly Language

Part I

A discussion of several methods of passing data to and from assembly-language subroutines. 6502, 6809, 68000, and 16032 microprocessors are used for programming examples.

by Randall Hyde

Programming in assembly language has long been thought a black art accomplished only by skilled magicians and sorcerers. An assembly-language program requires many more statements to accomplish the same goal than does the equivalent Pascal or BASIC program. Because of the extra tedium involved, most people shy away from assembly language because there is too much to learn. In reality, anyone with enough patience can master assembly language easily. One of the major impediments to learning assembly language is learning to pass data to and from assembly-language subroutines. This article discusses various methods of passing data on the 6502, 6809, and the newer 6800 and 16032 microprocessors.

Passing Data in the Microprocessor Registers

The simplest method used to pass data to and from an assembly-language subroutine is *via* the processor registers. Any register capable of storing retrievable data is usable for passing data to and from an assembly-language subroutine. On the 6502 data can be passed in the accumulator, X register, Y register, and the processor status register. On the 6809 data can be passed in the A or B accumulator, the X or Y index register, or the U stack pointer register. On the 16032 data can be passed in any of the eight general purpose registers, the static base register, or the frame pointer. Of the four microprocessors described here, the 68000 provides the richest complement of processor registers. Eight general purpose and seven address registers are available to the programmer in addition to the 68000 stack pointer.

The 6502 A, X, and Y registers are eight bits wide, and so you are limited to transferring a single byte within one of the registers. Single-bit boolean/binary values can be passed in the carry, zero, minus, and overflow flags. Such variables can be tested easily using the 6502 branch instructions.

The 6809 A and B accumulators are both eight bits wide and can be used for passing 8-bit data to and from a 6809 subroutine. If you need to transfer 16 bits of data, the A and B accumulators can be concatenated to form a single 16-bit accumulator — the 6809 D accumulator. Sixteen-bit data also can be passed in the 6809 X, Y, and U registers. Single-bit data can be passed in the 6809 zero, carry, overflow, and negative flags. Thirty-two-bit data can be passed by passing the low order 16 bits in one register and the high order 16 bits in a second register.

The 16032 microprocessor from National Semiconductor is far more advanced than its 8-bit brothers, the 6502 and 6809. Almost all of the 16032 registers are 32 bits wide, allowing you to pass a considerable amount of data in a single register. Furthermore, there are eight general purpose registers, three special purpose registers, and the pro-

cessor status register available for use. Since a considerable amount of data can be passed using the 16032 registers, passing parameters in the registers is much more practical on the 16032 than on the 6502 or 6809.

When it comes to passing data in the processor registers, the 68000 wins hands down. With 15 32-bit undedicated registers and a 32-bit stack pointer, the 68000 provides more registers than any popular microprocessor currently on the market.

Deciding when data is to be passed in one of the processor registers is the only problem left to solve. Since registers are a scarce commodity, judicious use of the register resources is recommended. Suppose you need to pass three bytes of data to a 6502 subroutine. Since you have three registers available, it would seem natural to pass the parameters in the A, X, and Y registers. In most cases such a choice would be very poor because these registers may be required for calculations going on outside the subroutine being called. For example, assume you are passing three bytes of data to a 6502 subroutine in the A, X, and Y registers. Furthermore, suppose you wish to call this subroutine repeatedly (i.e., from within a loop). The 6502 X and Y index registers are perfect for controlling loops and variables external to some subroutine. Unfortunately the X and Y registers will not be available for such use if they must be used for passing parameters to a subroutine, especially since you cannot push the X or Y register onto the stack without disturbing the contents of the accumulator. For this reason the use of the 6502 X and Y registers for passing parameters should be avoided as much as possible.

The 6809's register set is much more capable of handling parameters passed in the register than is the 6502. First, the 6809 has more registers than the 6502; second, the registers are twice as wide as the corresponding 6502 registers. Nevertheless, you should avoid passing parameters in the X, Y, and U registers for the same reasons outlined in the preceding paragraph.

The NS16032, with its eight general purpose registers, removes many of the restrictions on passing parameters in registers. Still, the frame pointer and Static Base register shouldn't be used for passing data to and from subroutines. That leaves the eight general purpose 16032 registers. These registers (called R0...R7) are each 32 bits wide, and so the 16032 is capable of passing over ten times the data in the CPU registers that the 6502 can! There's only one problem with using the 16032's general purpose registers to pass parameters: they *are* general purpose. This means that any of the eight registers can be used as accumulators, pointers, counters, or index registers. On the 6502 and 6809 the X and Y registers are special purpose index, pointer, or counter registers. Therefore, as you are writing a program, you know what the intended purpose of these registers is. Caution must be exercised with the 16032 because any of the registers can be used as an index register.

The 68000's register set provides so many registers for the programmer to use that many of the comments pertaining to register usage in the previous paragraphs simply don't apply. The 68000 data registers are most advantageously utilized for passing pure data (since most of the arithmetic and logical operations work best with the data registers) and the 68000 address registers should be used for passing pointer data since these registers are best suited for address manipulation.

Once you understand the disadvantages of passing data in the processor registers (i.e., there are only so many of them and they may not always be available for parameter usage) the next step is to decide when and how data is to be passed in the processor registers. The most obvious candidate is a routine that requires a single byte to be passed to it or returns a single byte upon completion. A good example is the READ CHARACTER and WRITE CHARACTER routines found in most monitors and operating systems. The WRITE CHARACTER subroutine (PUTC) usually requires a single character written to the system console. The READ CHARACTER subroutine (GETC) usually returns a single character read from the system terminal.

On the 6502 the accumulator is the best choice when data needs to be transferred to or from a subroutine. On the 6809 the A or B accumulator can be used to transfer the data. On the 16032 any of the eight general purpose registers can be used, although R0 (for consistency's sake) should be used.

When additional byte parameters must be passed to a subroutine, additional registers must be consumed. On the 6502 the Y register should be used first (because the X index register is used for counting more frequently than the Y register), and the X register last. The 6809 has two 8-bit accumulators that can be used for passing 8-bit parameters. Once these are exhausted you should use the Y register, followed by the X register, followed by the U stack pointer. Remember, the X, Y, and U registers are all 16 bits wide so two bytes can be passed in these registers by packing and unpacking the data. Packing two bytes into one of these registers is accomplished easily by loading the A and B accumulators with 8-bit data and then transferring the 16-bit D accumulator to one of the index registers. Of course there may be a more direct way, such as loading the 16-bit register with an immediate value or loading from some 16-bit memory location.

When additonal byte parameters must be passed to a subroutine additional registers must be consumed. On the 6502 the Y register should be used first...the X register last.

The 68000 and 16032 provide so many general purpose registers that there is little chance of running out of them. On these processors you can simply use additional registers to pass the extra parameter bytes. Although the 68000 and 16032 sport a complete set of 32-bit registers, packing and unpacking the data to preserve space is inefficient; there are better methods for passing parameters.

When you need to pass 16-bit values

to or from a subroutine, the 6502's limitations quickly become apparent. The 6502's registers are all eight bits wide, so you must use two registers to pass a 16-bit value. Most contemporary programs pass the low order eight bits in the Y register and the high order eight bits in the accumulator. Sometimes it would be more convenient to pass the low order byte in the accumulator (because arithmetic and logical operations take place in the accumulator and the low order byte is always operated on first), but my experience has proven that consistency (even with other people's code) is more important than convenience. If you need to pass more than one 16-bit value on the 6502, you will have to resort to another technique for parameter passing as the 6502 has only three bytes of register storage.

Passing 16-bit data on the 6809 is much simpler than on the 6502. The 6809 features a full set of 16-bit data and index registers. To pass a single 16-bit datum, your best bet is to use the 6809 D accumulator (which is a concatenation of the A and B 8-bit accumulators). If you need to pass two 16-bit values, the D accumulator and Y index registers are your best bet. The Y register should be used in place of the X register because most 6809 programmers use the X register more often than the Y register (this is a carry-over from the 68000 days). If more than three words of data need to be transferred you should choose a method other than using registers to transfer the parameters.

The 68000 and 16032 provide quite a few 32-bit registers, so passing 16- or even 32-bit data to a subroutine is no problem. In general, however, no more than four pieces of data should be transferred in the processor registers because these processors support more advanced methods of parameter transfer.

If you are using the 6502 you should try to limit yourself to two bytes of data passed in the 6502 registers (be it two 8-bit values or a single 16-bit value). On the 6809 you should limit yourself to three 8-bit values or two 16-bit values (using A, B, and Y registers). On the 68000 or the 16032 you should use registers to pass parameters only if you are passing less than five parameters. If you must exceed these limtis, then you should use one of the other methods described in this article.

Passing Parameters in a Fixed Location

Once your register list is exhausted, one of the easiest ways to pass data is by placing it in some fixed memory location before calling the appropriate subroutine. This method is quite popular with several monitor programs such as the Apple monitor and the Atari operating system (PEEKs and POKEs to BASIC hackers). By using this method of data transfer a virtually unlimited number of parameters can be passed. All you need to do is load up the appropriate memory locations and call the subroutine.

There are two principle disadvantages to this method of parameter transfer. First, loading data into a register and then storing it into some memory location requires more code and takes longer to execute than simply loading a CPU register with the parameter. Secondly, if you reserve a memory location for use by a subroutine (parameter or local variable) it cannot be used for anything else unless you are absolutely certain that no conflict will arise. Good programming practices dictate that you do not use such memory locations for anything else, otherwise the code will not be as maintainable as it could be.

One advantage to this method is that all three processors mentioned in this article support some sort of abbreviated addressing (zero page on the 6502, direct page on the 6809, absolute with a short displacement on the 68000 and 16032) that lets you access certain memory locations using a shortened instruction. Furthermore, accessing such absolute memory locations is usually faster than any other method of parameter transfer other than register usage.

One of the major problems with passing parameters in a fixed location is that you must move the data into the fixed location before calling the subroutine.

Certain global values (such as the horizontal and vertical positions for the cursor in a video screen driver) are perfect for this type of parameter transfer since all code in the system can have immediate and easy access to the data. This method of parameter passing is also perfect for quick and dirty programs where the extra time and thought required to pass data in some other manner is too excessive to be justified (this is especially true on the 6502, which is rather poor with respect to parameter handling.

Passing the Address of a Parameter Block

One of the major problems with passing parameters in a fixed location is that you must move the data into the fixed location before calling the subroutine. In many applications (like calling a disk operating system OPEN command, or a similar routine) a considerable amount of data needs to be passed to the subroutine, but the parameter values remain static over the lifetime of a program. In these situations it is more efficient to pass the address of a parameter block instead of moving the parameters to some fixed location. One way to accomplish this is to pass the address of the parameter block in one or more of the CPU registers. This is how, for example, Apple's DOS expects the address of the IOB (input output block) to be passed to the RWTS subroutine. On the 6502, most people pass the low order byte of the address of the parameter block in the Y register and the high order byte in the accumulator. On the 6809 one of the index registers (probably the X index register) is the best choice for the job. On the 16032 any of the eight general purpose registers are fine for this type of parameter transfer.

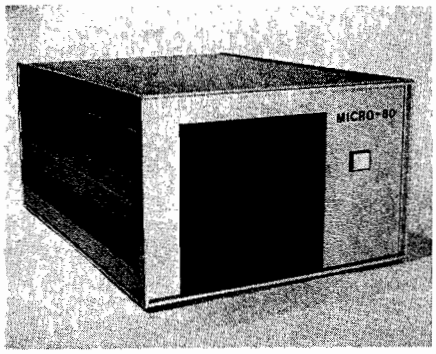
Depending upon the circumstances, the above form of parameter addressing is quite useful if you have a set of stock parameter blocks you wish to pass to a subroutine. For instance, the RWTS example mentioned earlier could benefit from this type of parameter transfer. In your program you define several IOBs and switch between them simply by loading the address of the new IOB.

Example:

```
LDY #IOB1 ;GET IOB-LO
LDA /IOB1 ;GET IOB-HI
```

(Continued on next page)

NEW FROM D & N MICRO PRODUCTS, INC.



MICRO-80 COMPUTER

Z80A CPU with 4MHz clock and CP/M 2.2 operating system. 64K of low power static RAM. Calendar real time clock. Centronics type parallel printer interface. Serial interface for terminal communications, dip switch baud rates of 150 to 9600. 4" cooling fan with air intake on back of computer and discharge through ventilation in the bottom. No holes on computer top or side for entry of foreign object. Two 8" single or double sided floppy disk drives. IBM single density 3740 format for 243K of storage on each drive. Using double density with 1K sectors 608K of storage is available on a single sided drive or 1.2 meg on a double sided drive. Satin finish extruded aluminum with vinyl woodgrain decorative finish. 8 slot backplane for expansion. 48 pin buss is compatible with most OSI boards. Uses all standard IBM format CP/M software.

Model 80-1200	$2995
2 8" single sided drives, 1.2 meg of storage	
Model 80-2400	$3495
2 8" double sided drives, 2.4 meg of storage	
Option 001	$ 95
Serial printer port, dip switch baud rate settings	

Software available in IBM single density 8" format.

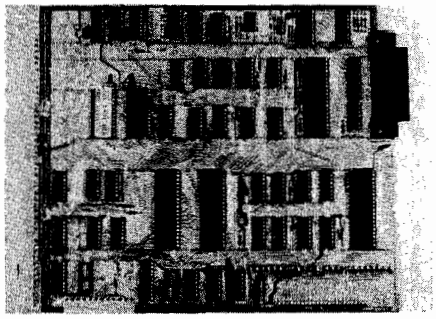
Microsoft		Digital Research		Micropro	
Basic-80	$289	PL/1-80	$459	Wordstar	$299
Basic Compiler	$329	Mac	$ 85	Mail-Merge	$109
Fortran-80	$410	Sid	$ 78	Spellstar	$175
Cobol-80	$574	Z-Sid	$ 95	Super Sort I	$195
Macro-80	$175	C Basic-2	$110	**Pascal**	
Edit-80	$105	Tex	$ 90	Pascal/MT +	$429
Mu Simp/Mu Math	$224	DeSpool	$ 50	Pascal Z	$349
Mu Lisp-80	$174	**Ashton-Tate**		Pascal M	$355
		dBase II	$595		

Convert almost any static memory OSI machine to CP/M® with the D & N-80 CPU Board.

Z80A CPU with 4MHz clock. 2716 EPROM with monitor and bootstrap loader. RS-232 serial interface for terminal communications or use as a serial printer interface in a VIDEO system. Disk controller is an Intel 8272 chip to provide single or double density disk format. 243K single density or 608K double density of disk storage on a single sided 8" drive. A double sided drive provides 1.2 meg of storage. DMA used with disk controller to unload CPU during block transfers from the disk drives. Optional Centronics type parallel printer port complete with 10 ft. cable. Optional Real Time Calendar Clock may be set or read using 'CALL' function in high level languages. Power requirements are only 5 volts at 1.4 amps. Available with WORDSTAR for serial terminal systems.

INCLUDES CPM 2.2

D & N-80 serial	$695
D & N-80 serial w/Wordstar	$870
D & N-80 video	$695
Option001	$ 80
parallel printer and real time calendar clock	



D & N-80 CPU BOARD

OTHER OSI COMPATIBLE HARDWARE

IO-CA10X Serial Printer Port **$125**
Compatible with OS-65U and OS-65D software

IO-CA9 Parallel Printer Port **$175**
Centronics standard parallel printer interface with 10 ft. flat cable

BP-580 8 Slot Backplane **$ 47**
Assembled 8 slot backplane for OSI 48 pin buss

24MEM-CM9	**$380**	**24MEM-CM9F**	**$530**
16MEM-CM9	**$300**	**16MEM-CM9F**	**$450**
8MEM-CM9	**$210**	**8MEM-CM9F**	**$360**
BMEM-CM9F	**$ 50**	**FL470**	**$180**

24K memory/floppy controller card supports up to 24K of 2114 memory chips and an OSI type floppy disk controller. Available fully assembled and tested with 8, 16, or 24K of memory, with floppy controller (F). Controller supports 2 drives. Needs separated clock and data inputs. Available Bare (BMEM-CM9F) or controller only (FL-470). Ideal way to upgrade cassette based system

C1P-EXP Expansion Interface **$ 65**
Expansion for C1P 600 or 610 board to the OSI 48 pin buss. Requires one slot in backplane. Use with BP-580 backplane

BIO-1600 Bare IO card **$ 50**
Supports 8K of memory, 2 16 bit parallel ports may be used as printer interfaces. 5 RS-232 serial ports, with manual and Molex connectors

DSK-SW Disk Switch **$ 29**
Extends life of drive and media. Shuts off minifloppy spindle motor when system is not accessing the drive. Complete KIT and manual

Disk Drives and Cables

8" Shugart SA801 single sided **$395**

8" Shugart SA851 double sided **$585**

FLC-6 6 ft. cable from D & N or OSI controller to 8" disk drive **$ 69**

5 1/4" MPI B51 with cable, power supply and cabinet **$450**

FLC-5 1/4 8 ft. cable for connection to 5 1/4 drive and D & N or OSI controller, with data separator and disk switch **$ 75**

Okidata Microline Printers

ML 82A Dot Matrix Printer **$534**
120 CPS, 80/120 columns, 9.5" paper width, friction or pin feed

ML 83A Same as 82A except **$895**
16" paper width, 132/232 columns with tractor feed

ML 84 Same as 82A except 200 CPS, **$1152**
16" paper width, 132/232 columns, 2K buffer, dot addressable graphics, with tractor feed

D & N Micro Products, Inc.
3684 N. Wells St.
Fort Wayne, Ind. 46808
(219) 485-6414

VISA MasterCard

TERMS $2.50 shipping, Foreign orders add 15%.
Indiana residents add 4% sales tax.

Circle No. 26

```
        JSR  RWTS


        LDY  #IOB2
        LDA  /IOB2
        JSR  RWTS


        LDY  #IOB3
        LDA  /IOB3
        JSR  RWTS

        ETC


IOB1    BYT     IOB DATA HERE
IOB2    BYT     IOB DATA HERE
IOB3    BYT     IOB DATA HERE

        ETC
```

If the data in IOB1 differs greatly from the data in IOB2, and both of them differ from the data in IOB3, then this form of parameter transfer is much more efficient than moving the data into an IOB at a fixed memory location. Another advantage to passing this data is the fact that it does not tie up the use of some specific address in the CPU's memory space. All you need to pass is the address of the beginning of the parameter block and, therefore, it can be located anywhere in memory; in fact, it can be relocated at will. Note that any amount of data can be transferred using this technique; the only limitation is that you have enough memory to hold all the parameters. On the 6502 it is best to keep the size of the parameter block below 256 bytes due to the 8-bit limitation of the 6502 index registers. Rarely will a subroutine require more than 30-40 bytes of parameters, so this shouldn't be a problem.

Now that you can pass the address of the parameter to a subroutine, the only problem left is retrieving the actual parameters. While the 16032, 68000, and 6809 microprocessors let you access a data table indirectly through a register, the 6502 does not. To access the parameter data on the 6502 you must store the address into a pair of consecutive zero page memory locations and access the data table using the (ZP),Y or (ZP),X addressing modes. For example, assume that you have a parameter block of the form:

```
PBLOCK BYT PARM1
       BYT PARM2
       BYT PARM3
       ADR PARM4
```

and the subroutine has the calling sequence:

```
LDY  #PBLOCK
LDA  /PBLOCK
JSR  SUBROUTN
```

Inside "SUBROUTN" you would store the accumulator and Y registers into a pair of zero page variables (say PARMPTR); then you could access the parameters using the (ZP),Y addressing mode as follows:

```
LDY  #0
LDA  (PARMPTR),Y  ;Fetches PARM1


LDY  #1
LDA  (PARMPTR),Y  ;Fetches PARM2


LDY  #2
LDA  (PARMPTR),Y  ;Fetches PARM3


LDY  #3
LDA  (PARMPTR),Y  GET PARM4-LO
INY
LDA  (PARMPTR),Y  GET PARM4-HI
```

On the 6809 accessing the parameter block is much easier. Assuming the address of the parameter block is passed in the X or Y register (it doesn't matter which one) the calling sequence would be:

```
LDX  #PBLOCK     Get address of
                 parameter block
JSR  SUBROUTN    Call subroutine
```

To access the data in the parameter block you would use the code:

```
LDA  ,X          Access PARM1


LDA  1,X         Access PARM2


LDA  2,X         Access PARM3


LDD  3,X         Access PARM4
                 (16 bits)
```

The 68000 calling sequence and parameter access method is almost identical to that for the 6809. The calling sequence is:

```
MOVE.L  #PBLOCK,A0
JSR     SUBROUTN
```

Within the subroutine you would access data in the parameter block using statements of the form:

```
MOVE.B 0(A0),D0  ;Copy PARM1 into D0


MOVE.B 1(A0),D1  ;Copy PARM2 into D1


MOVE.B 2(A0),D2  ;Copy PARM3 into D2


MOVE.W 3(A0),D3  ;Copy PARM4 into D3
```

The 16032 calling sequence and parameter access mechanism is identical to that of the 68000. The calling sequence is:

```
MOVD    #PBLOCK,R7  ;Get address
                    ; of parameters
JSR     SUBROUTN
```

Within the subroutine you would access data in the parameter block using statements of the form:

```
MOVB 0[R7],R0  ;Copy PARM1 into R0


MOVB 1[R7],R0  ;Copy PARM2 into R0


MOVB 2[R7],R0  ;Copy PARM3 into R0


MOVW 3[R7],R0  ;Copy PARM4 into R0
```

To return data to the calling subroutine simply move the data into the parameter block. Since the calling program presumably knows the address of the parameter block, it can extract data easily from the parameter block after the subroutine returns.

Although the address of the parameter block was passed in the registers in these examples, this address is nothing more than a 16-bit (or 32-bit) parameter, which can be passed to the subroutine in any fashion possible. It could be passed in a register (as we have done here), in a dedicated memory location, or even from within some other parameter block! In fact, any method used for passing 16 or 32 bits of data to a subroutine can be used to pass the address of a parameter block to the subroutine.

(Continued on next page)

HDE inc.

BOX 120
ALLAMUCHY, N.J. 07820
201-362-6574

HUDSON DIGITAL ELECTRONICS INC.

THE TASK* MASTERS

HDE supports the *TIM, AIM, SYM and KIM (TASK) with a growing line of computer programs and peripheral components. All HDE component boards are state-of-the-art 4½" x 6½", with on board regulation of all required voltages, fully compatible with the KIM-4 bus.

OMNIDISK 65/8 and 65/5
Single and dual drive 8" and 5¼" disk systems. Complete, ready to plug in, bootstrap and run. Include HDE's proprietary operating system, FODS (File Oriented Disk System).

DM816-M8A
An 8K static RAM board tested for a minimum of 100 hours and warranted for a full 6 months.

DM816-UB1
A prototyping card with on-board 5V regulator and address selection. You add the application.

DM816-P8
A 4/8K EPROM card for 2708 or 2716 circuits. On board regulation of all required voltages. Supplied without EPROMS.

DM816-CC15
A 15 position motherboard mounted in a 19" RETMA standard card cage, with power supply. KIM, AIM and SYM versions.

DISK PROGRAM LIBRARY
Offers exchange of user contributed routines and programs for HDE Disk Systems. Contact Progressive Computer Software, Inc. for details.

HDE DISK BASIC
A full range disk BASIC for KIM based systems. Includes PRINT USING, IF . . . THEN . . . ELSE. Sequential and random file access and much more. $175.00

HDE ADVANCED INTERACTIVE DISASSEMBLER (AID)
Two pass disassembler assigns labels and constructs source files for any object program. Saves multiple files to disk. TIM, AIM, SYM, KIM versions. $95.00

HDE ASSEMBLER
Advanced, two pass assembler with standard mnemonics. KIM, TIM, SYM and KIM cassette versions. $75.00 ($80.00 cassette)

HDE TEXT OUTPUT PROCESSING SYSTEM (TOPS)
A comprehensive text processor with over 30 commands to format and output letters, documents, manuscripts. KIM, TIM and KIM cassette versions. $135.00 ($142.50 cassette)

HDE DYNAMIC DEBUGGING TOOL (DDT)
Built in assembler/disassembler with program controlled single step and dynamic breakpoint entry/deletion. TIM, AIM, SYM, KIM AND KIM cassette versions. $65.00 ($68.50 cassette)

HDE COMPREHENSIVE MEMORY TEST (CMT)
Eight separate diagnostic routines for both static and dynamic memory. TIM, AIM, SYM, KIM and KIM cassette versions. $65.00 ($68.50 cassette)

AVAILABLE DIRECT OR FROM THESE FINE DEALERS:

Progressive Computer Software
405 Corbin Road
York, PA 17403
(717) 845-4954

Johnson computers
Box 523
Medina, Ohio 44256
(216) 725-4560

Falk-Baker Associates
382 Franklin Avenue
Nutley, NJ 07110
(201) 661-2430

Perry Peripherals
P.O. Box 924
Miller Place, NY 11764
(516) 744-6462

Lux Associates
20 Sunland Drive
Chico, CA 95926
(916) 343-5033

Laboratory Microcomputer Consultants
P.O. Box 84
East Amherst, NY 14051
(716) 689-7344

Circle No. 27

Passing Parameters on the Stack

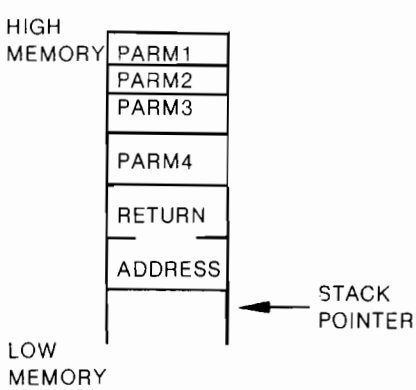
The favorite method for passing parameters used by many high-level languages such as Pascal is passing the parameters on a stack. The 6809, 68000, and 16032 are extremely well suited for passing data to and from procedures on a stack. The 6502 is, unfortunately, poorly suited for this application.

The 6502's problem with passing parameters on the stack is that the stack is of a very limited size (256 bytes). Furthermore, it's hard to get at data stored on the stack before a subroutine call. Nevertheless, there are times when pushing data onto the stack is the easiest way of passing data to a subroutine, especially if it is recursive. Although difficult, such parameter passing techniques are not impossible to perform on the 6502.

To pass data to a 6502 subroutine on the stack simply push the data onto the stack and call the routine as follows:

```
LDA  PARM1
PHA
LDA  PARM2
PHA
LDA  PARM3
PHA
LDA  PARM4
PHA
LDA  PARM4+1
PHA
JSR  SUBROUTN
```

There are two important points to keep in mind when passing data to a 6502 subroutine in this fashion. First, don't forget that upon entering the subroutine the subroutine return address is on the top of the stack. Second, the data appears on the stack in the reversed order (i.e., PARM4 + 1 is on the top, PARM4 is below that, PARM3 is below PARM4, etc.). Upon entering SUBROUTN the 6502 stack would look like this:



Note that the 6502 stack grows down and always points to the next available location on the stack. There are several ways to get the data off of the stack. One way is to POP it off (saving the return address in a temporary location and re-pushing it later) as follows:

```
SUBROUTN PLA
         STA  RTNADR
         PLA
         STA  RTNADR+1
         PLA
         STA  SPARM4+1
         PLA
         STA  SPARM4
         PLA
         STA  SPARM3
         PLA
         STA  SPARM2
         PLA
         STA  SPARM1
```

Push return address back onto the stack.

```
         LDA  RTNADR+1
         PHA
         LDA  RTNADR
         PHA

         RTS
```

This method of getting at parameters on the 6502 stack requires the use of some permanent memory locations. If you can't afford to use fixed memory locations to hold data, you can access it in place on the stack by copying the stack pointer into the X register and then using the index addressing mode:

```
TSX
LDA $107,X ;Get PARM1

LDA $106,X ;Get PARM2

LDA $105,X ;Get PARM3

LDA $104,X ;Get PARM4-LO

LDA $103,X ;Get PARM4-HI
```

Remove parameters from stack and return.

```
PLA          ;Get LO return address
STA $106,X   ;Save where rtn adrs
             ;must go
PLA          ;Get HI return address
STA $107,X   ;Save at correct spot
PLA          ;POP parameters
             ;off stack
PLA
PLA
PLA
PLA
RTS
```

The routine shown here modifies the contents of the 6502 accumulator and X register. Most well-written subroutines will need to preserve the contents of the 6502 registers, which usually means they will need to be pushed onto the stack. If you push all three registers onto the stack, keep in mind that the offsets required to access PARM1, PARM2, PARM3, and PARM4 are increased by three. A better version of the former program that preserves all the 6502 registers is:

```
SUBRTN PHA              ;Save Acc
       TXA
       PHA              ;Save X
                        ;register
       TYA
       PHA              ;Save Y
                        ;register
       TSX              ;Get copy of
                        ;SP register

       LDA  $10A,X      ;Read
                        ;PARM1

       STA  $109,X      ;Store data
                        ;into
                        ;PARM2

       LDA  $108,X      ;Read
                        ;PARM3

       LDA  $107,X      ;Get LO
                        ;PARM4
                        ;value
       STA  PARM4SAV
       LDA  $106,X      ;Get HI
                        ;PARM4
                        ;value
       STA  PARM4SAV+1
```

Prepare for subroutine return

```
       LDA  $105,X      ;Get HI rtn
                        ;adrs byte
       STA  $10A,X      ;and
                        ;reposition
                        ;on the stack
       LDA  $104,X      ;Get LO rtn
```

```
                        ;adrs byte
STA  $109,X             ;and store
                        ;back onto
                        ;the stack
LDA  $103,X             ;Get Acc
                        ;value
STA  $108,X
LDA  $102,X             ;Get X
                        ;register
                        ;value
TAX
LDA  $101,X             ;Get Y
                        ;register
                        ;value
TAY
PLA                     ;POP extra
PLA                     ;junk off of
PLA
PLA
PLA                     ;the stack
PLA
PLA
PLA                     ;Get Acc
RTS                     ;value
```

As you can probably tell, manipulating parameters on the 6502 stack is not a pleasant task.

The 6809 provides a much better method to access data on the stack than does the 6502. To begin, the 6809 supports *two* stacks: a hardware stack (where return addresses are kept) and a user stack (which is reserved for user-defined data). Both 6809 stack pointers can be used as index registers, so there's no need to waste one of the valuable index registers in order to access data on the stack. Finally, both stack pointers are 16 bits long so parameter data of any length (even when recursive subroutines are being used) can be accommodated.

While parameter data can be passed on either the hardware or the user stack, it is probably easiest to pass parameters on the user stack and use the hardware stack for storing return addresses and processor registers. Assuming that you have the same parameters defined for the 6502 example above, the calling sequence for a 6809 subroutine would be:

```
LDB   PARM1
LDA   PARM2
PSHU  A,B
LDA   PARM3
PSHU  A
LDX   PARM4
PSHU  X
JSR   SUBROUTN
```

After this code is executed the 6809 stack looks something like this:

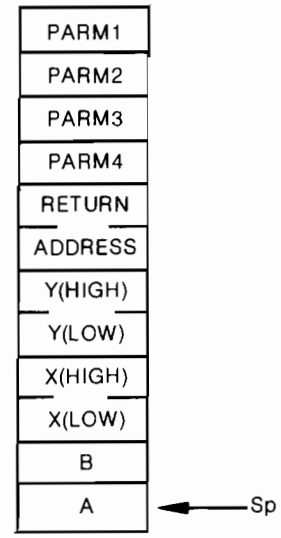
Here is the 6809 routine:

```
SUBRTN PSHS A,B,X,Y

       LDA  4,U       ;Get PARM1

       LDA  3,U       ;Get PARM2

       LDA  2,U       ;Get PARM3

       LDD  ,U        ;Get PARM4

       LEAU 5,U       ;Remove PARMs
                      ;from user
                      ;stack
       PULS A,B,X,Y   ;Restore 6809
                      ;registers
       RTS            ;All done
```

Notice that passing parameters on the 6809 stack is accomplished easily, especially when using the two stacks available to the user. Although it is a little more difficult to pass the parameters on the hardware stack (because the return address gets sandwiched in there) it is still much simpler to perform this task than it is to pass the data on the 6502 hardware stack. The 6809 code necessary to accomplish this task is:

```
SUBRTN PSHS A,B,X,Y ;Save 6809
                    ;registers

       LDA  12,S    ;Get PARM1

       LDA  11,S    ;Get PARM2

       LDA  10,S    ;Get PARM3

       LDD 8,S      ;Get PARM4

Get ready to leave

       LDD  6,S
       STD  11,S
       PULS A,B,X,Y
       LEAS 5,S     ;Remove
                    ;parameter
                    ;storage
       RTS
```

The stack, after an invocation of this subroutine, looks like this:



The Motorola 68000 chip has eight address registers. A7 is the 68000 hardware stack pointer; the remaining seven registers can be treated like the user stack pointer on the 6809. The only difference between using the hardware stack pointer and any of the other address registers is that you must remember to respect the four-byte return address that is pushed onto the hardware stack during a subroutine call. The calling sequence for a 68000 subroutine is

```
MOVE.B PARM1,D0
MOVE.B D0,(SP)+     ;Note: SP is a
                    ;synonym for A7
MOVE.B PARM2,D0
MOVE.B D0,(SP)+
MOVE.B PARM3,D0
MOVE.B D0,(SP)+
MOVE.B PARM4,D0
MOVE.B D0,(SP)+
JSR    SUBRTN
```

Since the hardware stack is being used in this particular case, you must remember to index past the return address when fetching data from the stack. The code to accomplish this is

```
SUBRTN MOVEM A0-A6/   ;Save registers
             D0-D7,
             -(SP)

       MOVE.B 68(SP),  ;Get PARM1
             D0
```

```
MOVE.B 67(SP),   ;Get PARM2
       D1

MOVE.B 66(SP),   ;Get PARM3
       D1

MOVE.W 64(SP),   ;Get PARM4
       D1

MOVE.L (SP),D0   ;Move return
                 ;address down
MOVE.L D0,4(SP) ;four bytes
MOVEM  (SP)+,
       A0-A6/
       D0-D7
ADDA   #4,SP     ;Adjust stack
RTS              ;pointer
```

The National Semiconductor 16032 chip was designed with a stack architecture in mind. For this reason it handles parameters passed on the stack quite well. Assuming you have the same parameters as used in the previous three examples, the calling sequence for a 16032 subroutine would be:

```
MOVEB  PARM1,TOS
MOVB   PARM2,TOS
MOVB   PARM3,TOS
MOVW   PARM4,TOS
JSR    SUBRTN
```

You can see that it is much easier to push the data onto the stack with the 16032 than it was with the 6502, 6809, and 68000 microprocessors. The 16032 fully supports memory-to-memory data transfers, alleviating the problem of having to load the data into a register and then pushing it onto the stack.

Although the 16032 has only one hardware stack, the instruction set recognizes that the return address will be placed on top of the parameters and allows you to make adjustments for this. To access the parameters on the stack you would use the code

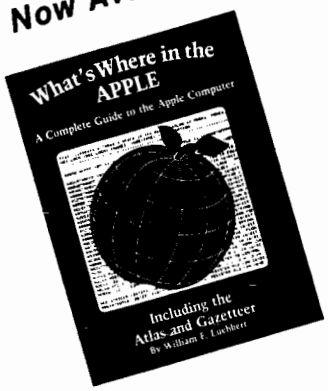
```
Save the NS16032 general purpose
registers and copy the stack pointer
into the 16032 frame pointer. This
also allocates zero bytes of local
storage for this particular subroutine.

SUBRTN ENTER [R0,R1,R2,R3,R4,
  R5,R6,R7],0

MOVB    6[FP],R0 ;Get PARM1

MOVB    5[FP],R0 ;Get PARM2

MOVB    4[FP],R0 ;Get PARM3

MOVW    2[FP],R0 ;Get PARM4

Now close up shop and quit

EXIT    [R0,R1,  ;Restore
        R2,R3,   ;registers
        R4,R5,   ;Return and pop
        R6,R7]   ;parameters
RTS     5
```

"Passing Parameters" will conclude next month.

You may contact the author at Lazer MicroSystems, Inc., 1791 Capital Unit G, Corona, CA 91720.

MICRO

Revised Edition Now Available!



All New Text Added To The Original Atlas And Gazetter...

"What's Where in the Apple"
A Complete Guide to the Apple Computer.

This revised edition of the famous Apple Atlas will:

- provide you with more information on the Apple's memory than is available elsewhere
- guide you with a numerical Atlas and an alphabetical Gazette — to over 2000 memory locations of PEEKs, POKEs and CALLs
- give you names and locations of various Monitor, DOS, Integer BASIC and Applesoft routines and tell you what they're used for
- enable you to move easily between BASIC and machine language
- guide you through the inner workings and hidden mechanisms of the Apple

All Apple users will find this book helpful in understanding their machine and essential for mastering it. The Atlas and the All New Guide are now available in one 256-page Wire-O-Bound book for only **$24.95.**

If you own the original **What's Where in the Apple** purchase **The Guide** to complete your edition. This 128-page Wire-O-Bound version contains all the new material to be used with the memory map and atlas. Don't miss this opportunity to complete your edition of **What's Where in the Apple.** Order your copy of The Guide now...only **$9.95**

Please Rush ☐ **What's Where in the Apple** @ $24.95 *(plus $2.00 s/h)*
☐ **The Guide** @ $9.95 *(plus $2.00 s/h)*

I'm paying by ☐ Check ☐ MO ☐ VISA ☐ MasterCard

Name _____

Credit Card # _____

Address _____

Card Expires _____ Total Enclosed $ _____

City _____ State _____ Zip _____

Mail to: **MICRO,** 34 Chelmsford Street, P.O. Box 6502, Chelmsford, MA 01824

Or Call Toll-Free: **1-800-354-8118** (In PA, 1-800-662-2444)

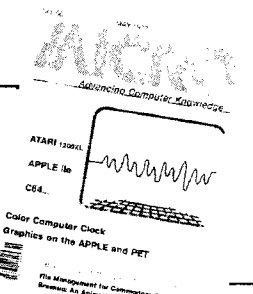
Circle No. 28

SAVE 20%!!!!

Subscribe to MICRO™

Use This Postage Paid Card to Order the Next 12 Issues of MICRO and SAVE $6.00 Off Newsstand Price!

MICRO™ SAVE 20%



☐ Yes! Enter my subscription to MICRO, and send me the next 12 issues for just **$24.00.** I save **$6.00 off** the newsstand price!

Name _____

Address _____

City _____

State _____ Zip _____

I'm paying by ☐ Check ☐ MO ☐ VISA ☐ MasterCard

Card # _____ Exp.Date _____

Signature _____

I OWN A:
☐ Commodore 64 ☐ VIC-20
☐ Apple ☐ PET
☐ Atari 400 ☐ Atari 800
☐ Other: _____

For Faster Service Call:
1-800-345-8112
(In PA: 1-800-662-2444
VISA or MasterCard Only

A Feast Of Computing Ideas...

Order These Books From MICRO™

MICRO™ Books

NEW for VIC-20 Users!
Mastering Your VIC-20
With 8 BASIC Projects

A book that makes learning to program your VIC-20 fun! Contains 8 projects and programs. Games, utilities — even a VIC-20 version of "VisiCalc." All 8 programs on cassette to help you learn faster.

☐ Mastering the VIC-20 @ $23.95

NEW for OSI Users!
MICRO on the OSI

Includes Machine-Language enhancements and BASIC Aids, hardware modifications for enhanced/reversed video, programs for control code and upper/lower case entry. A valuable programming tool.

☐ Micro on the OSI @ $19.95

Best Sellers for APPLE Users!
MICRO on the APPLE

Programming aids, utilities, games, enhancements. Together Volumes 1, 2, and 3 contain over 100 programs on diskette. Fully documented and explained.

☐ 3-Volume Gift-Boxed @ $59.95
☐ Vol.1 ☐ Vol.2 ☐ Vol.3 $24.95 ea.

Please rush the MICRO Books I've checked above to:

Name _____

Address _____

City _____ State _____ Zip _____

(Allow 6-8 weeks for delivery)

I'm paying by: ☐ Check ☐ MO ☐ VISA ☐ MC

Total Enclosed: $ _____
(Add $2.00 s/h per book. MA res. add 5% tax)

Visa/MC # _____

Exp. Date: _____

MICRO™ Reader Service...

For More Information On The Advertisers In This Issue!

MICRO™ Reader Service Card

The numbers below correspond to the numbers imprinted on the advertisements in this issue of MICRO. This card valid for 90 days only.

Mail information to:

Name _____

Address _____

City _____ State _____ Zip _____

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76	77	78
79	80	81	82	83	84	85	86	87	88	89	90	91
92	93	94	95	96	97	98	99	100	101	102	103	104

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 60, CHELMSFORD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

MICRO™

34 Chelmsford Street
P.O. Box 6502
Chelmsford, MA 01824

SAVE 20%!!!!

Subscribe to MICRO

Use This Postage Paid Card to Orde the Next 12 Issue of MICRO and SAV $6.00 Off Newsstand Price!

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 60, CHELMSFORD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

MICRO™

34 Chelmsford Street
P.O. Box 6502
Chelmsford, MA 01824

A Feast O Computin Ideas...

Order These Books From MICRO

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 60, CHELMSFORD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

MICRO™

34 Chelmsford Street
P.O. Box 6502
Chelmsford, MA 01824

MICRO Reader Service..

For More Information On The Advertisers In This Issue

FOR YOUR APPLE II

Industry standard products at super saver discount prices

SOFTWARE

ARTSCI	List	SGC
Magicalc	$149.00	$ 99.00
Magic Window II	149.00	99.00
DBase (Apple)	695.00	475.00
BRODERBUND		
Payroll	$395.00	$295.00
Choplifter	34.95	25.00
Arcade Machine	44.95	29.95
Serpentine	34.95	25.00
Home Accountant	74.95	55.00
Home Accountant Plus	150.00	109.00
DATAMOST		
Snackattack	$ 29.95	$ 22.50
Thief	29.95	22.50
Swashbuckler	34.95	24.95
Zork I, II, or III	39.95	27.95
Starcross	39.95	27.95
Format II	250.00	175.00
System Saver & Fan	59.95	49.00
Multiplan	275.00	199.00
ON LINE		
Mystery House	$ 24.95	$ 19.95
Cranston Mannor	34.95	24.95
Frogger	34.95	24.95
Screen Writer II	129.95	99.95
Memory Management II	49.95	39.95
PEACHTREE		
GL, AR, AP, Inventory, Payroll	ea. $400.00	ea. $295.00
Micro Buffer II	299.00	249.00
SENSIBLE SOFTWARE		
Super Disk Copy III	$ 29.95	$ 22.95
DOS Plus	24.95	17.95
SERIUS SOFTWARE		
Bandits	$ 34.95	$ 26.95
Epoch	34.95	26.95
Fly Wars	29.95	22.95
Gorgon	39.95	29.95
Sneakers	29.95	22.95
Joy Port	74.95	59.95
Wizardry	49.95	34.95
Night of Diamonds	34.95	26.95
Star Maze	34.95	26.95
PFS	125.00	89.95
PFS Report	95.00	69.95
PFS Graph	125.00	89.95
Data Capture 4.0	64.95	49.95
Merlin/Assembly Lines	119.95	89.95
Merlin	64.95	49.95
ASCII Express Pro	129.95	99.95
Transend II	149.00	119.00
Transend I	89.00	65.00
DB Master	229.00	165.00
DB Utility Pack	99.00	79.00
STRATIGIC SIMULATION		
All Software	$ 59.95	$ 39.95
All Software	39.95	27.95
SYNERGISTIC SOFTWARE		
Wilderness & Dungeon	$ 32.50	$ 24.95
GPLE	64.95	49.95
TG Joystick	59.95	45.00
Select-A-Port	59.95	45.00
Wordstar		325.00
Spellstar		175.00
SuperCalc		175.00
VisiCalc		179.00

SPECIAL AND NEW

FRANKLIN ACE 1000 COMPUTER

Hardware and Software compatible with Apple II **$950**

FRANKLIN ACE 1000 COMPUTER plus DISK DRIVE, CONTROLLER, and MAGICALC **$1,250**

EXPAND-A-RAM™ PLUS MAGICALC™

Everything that Visicalc™ can do and much more —plus additional memory. Fully compatible with Visicalc. Includes DOS, CP/M, Pascal Disk Emulator. No preboot or Apple modification required.

64K EXPAND-A-RAM plus MAGICALC **$375**

128K EXPAND-A-RAM plus MAGICALC **$449**

APPLEsurance II™

Diagnostic Disk Controller and System Assurance Package. Standard disk controller plus automatic check of system hardware **$99**

5¼" DISK DRIVE

Use with either standard Apple II disk drive or APPLEsurance II **$249**

GRAPHITTI CARD

Prints HIRES page 1 or 2 from onboard firmware. Features: True 1:1 aspect ratio, prints emphasized mode, reverse mode, rotates 90 degrees . . . plus more. Compare all this with the Grappler. We think you'll agree that this is the best graphics card on the market. Specify for use with EPSON, NEC-8023, C-ITOH Prowriter, or Okidata.

(List: $125) **$89**

PARALLEL PRINTERS

NEC 8023 or C-ITOH 8510

(Virtually identical) Specifications: • 100 CPS dot matrix printer • 80 column print—136 characters per line • Tractor/friction feed • 7 different print fonts included • 2K printer buffer • Proportional spacing • Bit image graphics and graphic symbols.

NEC 8023 or C-ITOH **$475**

NEC 8023 or C-ITOH 8510 with Parallel Interface and Cable **$550**

EPSON 100 with Parallel Interface and Cable **$775**

BROTHER Daisywheel Printer **$895**

VERSAcard FROM PROMETHEUS

Four cards on one! With true simultaneous operation. Includes: (1) Serial Input/Output Interface, (2) Parallel Output Interface, (3) Precision Clock/Calendar, and (4) BSR Control. All on one card. Fully compatible with CP/M* and Apple Pascal*.

(List: $249) **$169**

WORD PROCESSING SPECIAL WITH WORDSTAR AND SUPERCALC!

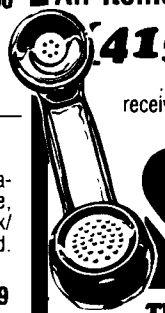
Do professional word processing on your APPLE. All necessary hardware and software included. Complete 80 column video display, enhanced character set, 16K memory board, Z-Card with CP/M* software, Wordstar and word processing software and SuperCALC.

(List: $1,228) . . . **Special at $795**

Z-80 CARDS	List	SGC
Microsoft Softcard Z-80	$399.00	$289.00
ALS Synergizer	749.00	595.00
U-Z-80 Processor Board		125.00
Microsoft+Premium Syst.		595.00
80-COLUMN CARDS		
Smarterm 80-Col Display	$345.00	$225.00
Smarterm Expanded Character Set		40.00
Combination Smarterm & Exp. Char. Set		260.00
Videx Videoterm		275.00
Videx Enhanacer II	149.00	125.00
Videx VisiCalc Preboot	49.95	45.00
MODEMS FOR YOUR APPLE II		
Hayes Smartmodem 300		$229.00
Hayes Smartmodem 1200	699.00	550.00
Micromodem II		279.00
Hayes 100 Baud		Call
Apple Cat II	389.00	299.00
D Cat Modem	199.00	175.00
MONITORS		
Amdek 300G Green		$159.00
Color—Taxam RGB with Interface		395.00
PARALLEL INTERFACE		
Centronics Compat. PRT-1		$ 69.00
JOYSTICK Replaces two Apple Paddle Controllers	$ 59.00	$ 39.00
FUNCTION STRIP	$ 79.00	$ 65.00
MEMORY EXPANSION		
Prometheus 16K RAM Module complete	$169.00	$ 65.00
5¼" FLOPPY DISKS		
Box of 10 with hub rings		
With other purchase		$ 19.95
Without other purchase		23.00

All equipment shipped factory fresh. Manufacturers' warranties included. California customers add 6½% tax. Include payment by personal check, money order, or cashier's check with order and SGC will pay shipping charge. Call for amount of shipping charge when paying by credit card.

All items are normally in stock



(415) 490-3420

. . . And we'll be here to help after you receive your order. Feel free to call the SGC Technical Staff for assistance.

SGC

The mail order specialists

342 Quartz Circle, Livermore, CA 94550

Circle No. 29

Real Time Clock for Color Computer

by John Steiner

Use this subroutine to add a real-time clock and date function to your programs.

TIME$/SUB
requires:
TRS-80 Color Computer

After owning my TRS-80C for one year, I have few complaints and a lot of praise. The graphics are impressive for such a low-priced system, and the number of external hook-ups is more than adequate for the average user. Plus, you get the built-in RS-232 and joystick ports, which are extra cost on many other systems.

I finally purchased a disk drive and soon became accustomed to the invisible disk operating sytem. The DOS is an integral part of extended BASIC in the color computer. But I was jealous of my friends who owned the Model III with the built-in TIME$ and DATE$ commands.

One of the applications I have for my machine is a mailing list and file system for the local ham radio and computer clubs. It would be nice for a program to tell me the last time a file was created, or updated. I suffered with this shortcoming for many weeks until I saw the light.

The color computer has a built-in elapsed timer, which on power up counts from 0 to 65535. The timer is available only in Extended BASIC, and I have heard that it is interrupt driven, updating approximately every 1/60th of a second.

For the cost of a few extra reserved variables, I wrote a subroutine that can be inserted into any program. Listing 1 contains the routine, which begins at 60000.

You need to make only a couple modifications to any individual program to use the routine. After clearing memory and dimensioning strings, call the subroutine that begins at 60000, and immediately thereafter call the routine at 60100 as in the example.

```
10 CLEAR200:DIM N$(100)
20 GOSUB 60000: GOSUB 60100
```

The routine at 60000 enters the correct date and time into DATE$ and TIME$. Lines 60100 to 60160 update the correct time using the timer function.

Now, throughout the program, sprinkle GOSUB 60100 sparingly. Suitable places for the GOSUBs are at the beginning of menu screens, just before and after long sorts, and *always* just before printing TIME$ to the printer, screen, disk, or cassette.

The timer will keep reasonably correct time, unlike most software clocks, because it is interrupt driven.

When the routine is called, line 60100 determines whether or not one minute has expired. If it has not, no time is wasted, and the program reassigns or, in the case of initialization, assigns the current time to TIME$. If it has been a while since the last GOSUB, one minute is subtracted from the value of the timer and added to the variable MIN. If the hour must be updated, it is also done in line 60120. Another check is made to see if the timer is greater than 3550. This indicates that more than one minute has passed since the last update. If this is the case, the program loops back to 60110 and again updates the timer, hour, and minute variables.

You may find that the clock will gain or lose time. By changing the value of TIMER in lines 60100, 60110, and 60130, you can speed up or slow down the clock. You can make it as accurate as you want.

After the routine determines that the time is now correct, lines 60140 and 60150 format and assign the correct time to TIME$. Your program can get by without updating the time for 18 minutes, before the TIMER function in the color computer resets itself, so be sure that it is accessed more often than that. Again, it's a good idea to put the access routine in the menus, which are usually accessed often.

Now your computer has TIME$ and DATE$ capacity too!

***Listing 1:* TIME$/Subroutine**

```
50000 '*************************
50010 '*    TIME$/SUB          *
50020 '*  JOHN STEINER         *
50030 '* APRIL 11, 1982        *
50040 '*************************
50050 'TO CHANGE TIMER SPEED ADJUST
      LINES 60100,60110 AND 60130
60000 CLS:PRINT:PRINT"ENTER TODAYS DATE"
60010 LINEINPUT"MM/DD/YY: ";DATE$
60020 LINEINPUT"ENTER THE CORRECT HOUR: ";HOUR$
60030 IFVAL(HOUR$)<1ORVAL(HOUR$)>12THEN
      PRINT"INCORRECT TIME":GOTO60020
60040 HOUR=VAL(HOUR$)
60050 LINEINPUT"ENTER THE CORRECT MINUTE: ";MIN$
60060 IFVAL(MIN$)<0ORVAL(MIN$)>59THEN
      PRINT"INCORRECT MINUTES":GOTO60050
60070 MIN=VAL(MIN$)
60080 TIMER=0
60090 RETURN
60100 IFTIMER<3550THEN60140
60110 TIMER=TIMER-3550
60120 MIN=MIN+1:IFMIN>59THENMIN=MIN-60:
      HOUR=HOUR+1:IFHOUR>12THENHOUR=HOUR-12
60130 IFTIMER>3550THEN60110
60140 MIN$=STR$(MIN):IFLEN(MIN$)=2THEN
      MIN$="0"+RIGHT$(MIN$,1)ELSE
      MIN$=RIGHT$(MIN$,2)
60150 TIME$=STR$(HOUR)+":"+MIN$
60160 RETURN
```

John Steiner is an electronics instructor in the Fargo, ND school system. His hobbies include programming, amateur radio, and writing. He has written articles for several publications and is at present completing a book on Electronics. Also in the works is a book on BASIC. John's computer system includes a TPD-100 with disk system, and an Epson MX-80 printer.

VIDEO TERMINAL BOARD 82-018

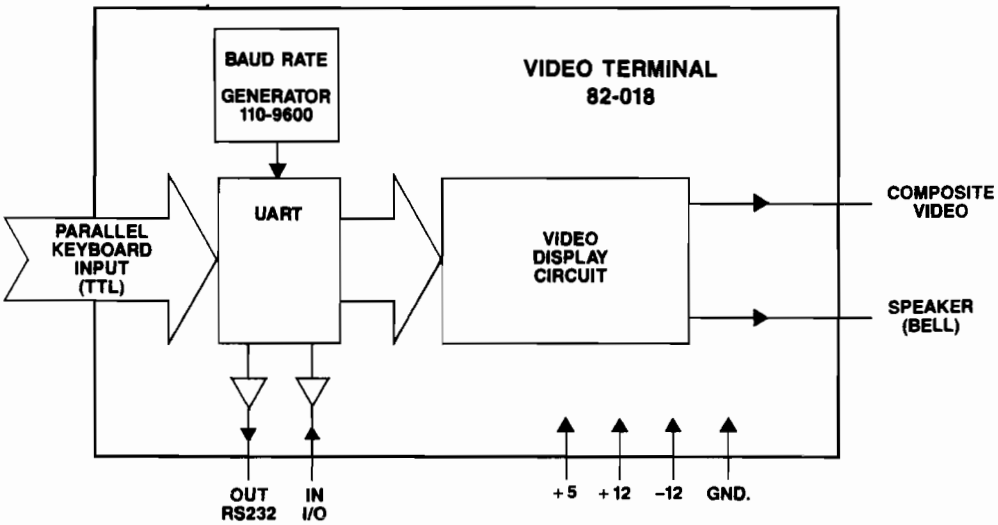
This is a complete stand alone Video Terminal board. All that is needed besides this board is a parallel ASCII keyboard, standard NTSC monitor, and a power supply. It displays 80 columns by 25 lines of UPPER and lower case characters. Data is transferred by RS232 at rates of 110 baud to 9600 baud — switch selectable. The UART is controlled (parity etc.) by a 5 pos. dip switch.

Complete source listing is included in the documentation. Both the character generator and the CRT program are in 2716 EPROMS to allow easy modification to your needs.

This board uses a 6502 Microprocessor and a 6545-1 CRT controller. The 6502 runs during the horz. and vert. blanking (45% of the time). The serial input port is interrupt driven. A 1500 character silo is used to store data until the 6502 can display it.

Features

- **6502 Microprocessor**
- **6545-1 CRT controller**
- **2716 EPROM char. gen.**
- **2716 EPROM program**
- **4K RAM (6116)**
- **2K EPROM 2716**
- **RS232 I/O for direct connection to computer or modem.**
- **80 columns x 25 line display**
- **Size 6.2″ x 7.2″**
- **Output for speaker (bell)**
- **Power +5 700Ma.**
  +12 50Ma.
  –12 50Ma.



This board is available assembled and tested, or bare board with the two EPROMS and crystal.

Assembled and tested	#82-018A	$199.95
Bare board with EPROMS and crystal	#82-018B	$ 89.95

Both versions come with complete documentation.

JOHN BELL ENGINEERING, INC.

ALL PRODUCTS ARE AVAILABLE FROM **JOHN BELL ENGINEERING, INC.** • 1014 CENTER ST., SAN CARLOS, CA 94070
ADD SALES TAX IN CALIFORNIA • ADD 5% SHIPPING & HANDLING 3% FOR ORDERS OVER $100

MasterCard

SEND $1.00 FOR CATALOG

(415) 592-8411
WILL CALL HOURS: 9am - 4pm

10% OUTSIDE U.S.A.
ADD $1.50 FOR C.O.D.

VISA

#249

&GET

by David Dice

A machine-language program to allow customized input routines.

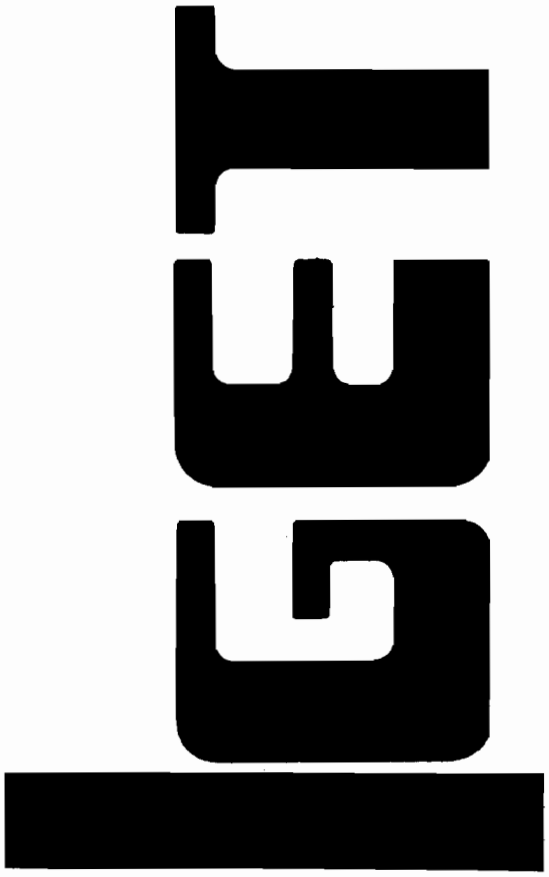
AMPERGET
requires:
Apple II with Applesoft BASIC

Many functions are not available with Applesoft's INPUT routine. For instance, have you ever had to input strings containing commas or colons? Would you like to use the 'ESC' key as an exit parameter? Or how about using the backspace key to back up to a previous line of data? One possibility is using the GET routine, but this method has disadvantages. You can GET the character and then concantenate strings with a statement like GET X$: TE$ = TE$ + X$. However, you can no longer use the retype (— >) or backspace (< —) keys to copy or delete material from the string. Another problem is that it accelerates the need for garbage collection — that annoying, time-consuming process of freeing up memory that happens when you do a lot of string swapping in Applesoft. I tempered the problem by developing &GET.

Initially my approach to using the retype and backspace keys involved the MID$ command in a short Applesoft subroutine that found the correct part of the string. This worked at least as fast as I could type but there was some noticeable screen flicker. And because I was doing even more string manipulations, it compounded the problem of memory space!

When presented by a problem that BASIC doesn't solve efficiently you have two alternatives. Quit, or try another language! In this case I wanted the program to run quickly, so the logical approach was to write the input routine in machine language. Fortunately, the procedure could be greatly simplified by using a number of monitor and Applesoft routines and pointers. The result was &GET.

&GET takes about half a page of memory. It could be assembled in an area not affected by the Applesoft program that calls it, and page 3 of memory (starting at $300) is just such a convenient location. The &GET routine is accessed by a command using the syntax '&G,A$' where A$ can be any string variable. The & vector in byte $3F5 contains a JMP code to the beginning of the routine. The first few lines of the program check to ensure that the correct syntax was used and find the location of the string variable specified. Provided that the syntax is right the



routine begins to input the string. If there is an error, then you will hear the Apple's beep and be presented with the message 'SYNTAX ERROR'.

You put the characters you pick up in the input buffer, page 2 of memory. This is where the Apple's normal input routine, GETLIN, stores characters; since you're not using GETLIN there won't be any problems.

First the buffer is cleared by filling it with spaces. You are now ready to get the data. The monitor input routine, RDKEY, is used to input a character. If the character is one of 'ESC', ' ← ', ' → ', or 'RETURN' then special action is taken:

1. If 'ESC', then $1B (decimal 27), the ASCII code for ESCAPE, is placed in the variable and the routine exits.
2. If ' ← ', then the cursor is moved backwards and the length of the string is decremented. If an attempt

Amperget Listing

```
                    **********************************
                    *
                    * &GET
                    *
                    * COPYRIGHT (C) 1983
                    *
                    * MICRO INK
                    * 34 CHELMSFORD ST.
                    * CHELMSFORD, MA 01824
                    *
                    * WRITTEN BY D. DICE
                    *
                    **********************************
                    *
                    * APPLESOFT ROUTINES AND POINTERS
                    *
                    BASL     EPZ $28
                    CHKCOM   EQU $DEBE
                    CHKSTR   EQU $DD6C
                    CHRGET   EPZ $B1
                    CROUT1   EQU $FD8B
                    COUT     EQU $FDED
                    ERROR    EQU $D412
                    FRETOP   EPZ $6F
                    GETSPA   EQU $E452
                    IN       EQU $200
                    PTRGET   EQU $DFE3
                    RDKEY    EQU $FD0C
                    RNGBEL   EQU $FF3A
                    SYNTAX   EPZ $10
                    VARPT    EPZ $83
                    NULL     EQU $00
                    SPACE    EQU $20
                    FORWARD  EQU $95
                    BACKWARD EQU $88
                    RETURN   EQU $8D
                    ESCAPE   EQU $9B
                    AMPER    EQU $3F6
                    *
                             ORG $300
                    *
                    * SET UP & JUMP
                    *
0300 A9 0B          LDA #BEGIN
0302 8D F6 03       STA AMPER
0305 A9 03          LDA /BEGIN
0307 8D F7 03       STA AMPER+$1
030A 60             RTS
                    *
                    * ENTRY POINT FROM AMPERSAND
                    *
030B C9 47          BEGIN    CMP 'G          ;CHECK FOR 'G'
030D F0 05                   BEQ START
030F A2 10                   LDX #SYNTAX     ;SYNTAX ERROR
0311 4C 12 D4                JMP ERROR
0314 20 B1 00       START    JSR CHRGET
0317 20 BE DE                JSR CHKCOM
031A 20 E3 DF                JSR PTRGET
031D 20 6C DD                JSR CHKSTR
0320 A2 00                   LDX #NULL       ; FILL THE BUFFER WITH SPACES
0322 A9 20          CLRBUF   LDA #SPACE
0324 9D 00 02                STA IN,X
0327 CA                      DEX
0328 D0 F8                   BNE CLRBUF
032A 20 0C FD       GETCHAR  JSR RDKEY       ; READ THE KEYBOARD
032D C9 95                   CMP #FORWARD    ; CURSOR AHEAD?
032F F0 1A                   BEQ RDSCRN
0331 C9 88                   CMP #BACKWARD   ; BACKSPACE?
0333 F0 1B                   BEQ BACKWDS
0335 C9 8D                   CMP #RETURN     ; RETURN?
0337 F0 35                   BEQ RETURN1
0339 C9 9B                   CMP #ESCAPE     ; ESCAPE?
033B F0 29                   BEQ ESCAPE1
033D 20 ED FD       PRINT    JSR COUT
0340 9D 00 02                STA IN,X        ; STORE VALUE IN BUFFER
0343 E8                      INX             ; INCREMENT COUNT OF ENTRIES
0344 E0 F8                   CPX #$F8        ; IF GREATER THAN 248
0346 B0 50                   BCS BELL        ; THEN RING BELL
0348 4C 2A 03                JMP GETCHAR     ; ELSE GET NEXT CHARACTER
034B B1 28          RDSCRN   LDA (BASL),Y    ; READ THE SCREEN
034D 4C 3D 03                JMP PRINT
0350 CA             BACKWDS  DEX
0351 E0 FF                   CPX #$FF        ; IF < 0 THEN
```

is made to back up past the end of the line, then the variable contains the code for BACKSPACE, $8 and the routine exits.

3. If ' → ', then RDSCRN picks up the current character displayed on the monitor.
4. If 'RETURN', then the routine exits.

The Applesoft routine GETSPA is used to find space for the string variable, the high bit is stripped off, and then TSFR moves the input buffer to the variable specified in the &G,A$ statement. If the data is not one of these special characters, then it is added to the end of the input buffer. A check is made on the length of the input string. At 249 characters the bell begins to ring. At 255 characters the buffer has been filled, a backslash is printed, and the routine starts over.

The Applesoft program AMPERGET.DEMO (listing 2) shows how the routine can be used. The subroutine, GET THE DATA, uses the &GET. You may type in data, use the backspace and retype keys normally, and end a line by typing 'RETURN'. If you type a 'RETURN' without any entry, then the current value of the variable L1$ or L2$ is left unchanged. If you backspace past the beginning of INPUT LINE 2, then AMPERGET.TEST will take you back to INPUT LINE 1. Line 270 loops back to the beginning of the data input routine so that it is impossible to back up past the beginning of GET THE DATA. Typing 'ESC' at any time clears all the entries and repeats the GET THE DATA subroutine.

You lose the cursor editing functions ESC I,J,K,M (or A,B,C,D) by using this routine. This is a small price to pay for getting the ESC key back for an 'escape' function, and the ability to enter any character into an Applesoft string.

David Dice holds a Ph.D. in chemistry and a Bachelor's degree in education. He has taught chemistry, science, math, and computer science from the Junior High to the University levels. Mr. Dice is owner of Digipac Computer Consulting, a company specialzing in the production and distribution of quality educational and business software, and a fulltime high school teacher. You may contact him at Digipac, 907 River St. E., Prince Albert, Sask. 56V 0B3.

```
0353 F0 06        95            BEQ OUT        ; EXIT WITH BACKSPACE
0355 20 ED FD     96            JSR COUT       ; ELSE PRINT THE VALUE
0358 4C 2A 03     97            JMP GETCHAR
035B A2 00        98  OUT       LDX #NULL      ; EXIT WITH BACKSPACE
035D A9 08        99            LDA #BACKWARD-$80
035F 9D 00 02    100            STA IN,X
0362 E8          101            INX
0363 4C 6E 03    102            JMP RETURN1
0366 A2 00       103  ESCAPE1    LDX #NULL      ; EXIT WITH ESCAPE CODE
0368 A9 1B       104            LDA #ESCAPE-$80
036A 9D 00 02    105            STA IN,X
036D E8          106            INX
036E 8A          107  RETURN1    TXA            ; TRNSFR LEN TO ACCUM.
036F A0 00       108            LDY #NULL      ; AND STORE IT
0371 91 83       109            STA (VARPT),Y
0373 20 52 E4    110            JSR GETSPA     ; GET SPACE FOR STRING
0376 A0 01       111            LDY #$01
0378 A5 6F       112            LDA FRETOP
037A 91 83       113            STA (VARPT),Y  ; STORE ADDRESS OF STRING
037C C8          114            INY
037D A5 70       115            LDA FRETOP+1
037F 91 83       116            STA (VARPT),Y
0381 A0 00       117            LDY #NULL
0383 B1 83       118            LDA (VARPT),Y
0385 AA          119            TAX
0386 A8          120            TAY
0387 CA          121  TSFR      DEX
0388 88          122            DEY
0389 BD 00 02    123            LDA IN,X       ; GET CHARACTER
038C 29 7F       124            AND #$7F       ; CLEAR $80 BIT
038E 91 6F       125            STA (FRETOP),Y ; STORE CHARACTER
0390 E0 00       126            CPX #NULL
0392 D0 F3       127            BNE TSFR
0394 20 8B FD    128            JSR CROUT1     ; EXIT AND CLEAR LINE
0397 60          129            RTS
0398 E0 FF       130  BELL      CPX #$FF       ; IF STRING LEN EQUALS 255
039A F0 06       131            BEQ END        ; THEN DO AGAIN
039C 20 3A FF    132            JSR RNGBEL     ; ELSE  RING THE BELL
039F 4C 2A 03    133            JMP GETCHAR
03A2 A9 DC       134  END       LDA #$DC       ; PRINT SLASH AND START OVER
03A4 20 ED FD    135            JSR COUT
03A7 20 8B FD    136            JSR CROUT1
03AA 4C 22 03    137            JMP CLRBUF
03AD             138            END
```

```
DICE AMPERGET.DEMO

10  REM ************************
20  REM *        AMPERGET.DEMO
30  REM *        BY DAVID DICE
40  REM *      COPYRIGHT (C) 1983
50  REM *          MICRO INK
60  REM ************************
70 ES$ =  CHR$ (27):CB$ =  CHR$ (
     8):D$ =  CHR$ (4)
80  PRINT D$;"BRUN AMPERGET.OBJ"
90  HOME
100  PRINT "'RETURN'=COPY 'ESC'=V
     OID '<-'=BACKSPACE"
110  VTAB 3: PRINT "INPUT LINE 1:
120  VTAB 13: PRINT "INPUT LINE 2:
130  GOSUB 170
140  IF ES THEN L1$ = "":L2$ = ""
     : GOTO 90
150  GOSUB 310
160  END
170 ES = 0
180  VTAB 3: HTAB 15: PRINT L1$
190  VTAB 3: HTAB 15: & G,A$
200  IF A$ = "" THEN A$ = L1$: VTAB
     3: HTAB 15: PRINT L1$
210  IF A$ = CB$ THEN 180
220  IF A$ = ES$ THEN ES = 1: RETURN
230 L1$ = A$
240  VTAB 13: HTAB 15: PRINT L2$
250  VTAB 13: HTAB 15: & G,A$
260  IF A$ = "" THEN A$ = L2$: VTAB
     13: HTAB 15: PRINT L2$
270  IF A$ = CB$ THEN 180
280  IF A$ = ES$ THEN ES = 1: RETURN
290 L2$ = A$
300  RETURN
310  HOME
320  PRINT "THE VALUES ENTERED WERE:
330  PRINT "LINE 1: ";L1$
340  PRINT : PRINT "LINE 2: ";L2$
350  RETURN
```

MICRO™

OSI — OSI — OSI — OSI

192K RAM BOARD

ONLY **$895**

The CCS - 192K is divided into 4 - 48K bank partitions. Each board can be dip switch selected for 1 of 4 192K banks.

When combined with our CPU the 192K RAM Board implements a four user computer system at a logic cost of $1595 (special package price).

Combine this logic package with our Tilios Operating System and you have a multi-user computer with throughput that has to be seen to be believed.

7250 COMMERCE CIRCLE EAST • FRIDLEY, MINNESOTA 55432 • PHONE (612) 574-9493
(800) 328-2419

Circle No. 31

Modifying and Using MAE on the PET

by F. Arthur Cochrane

Modifications to the MAE assembler described here allow output to an ASCII printer and listing output to a disk file.

MAE is the Macro Assembler Text Editor for the PET written in fast machine-language code by Carl Moser of Eastern House Software. There are versions for 3.0 ROMs, 4.0 ROMs, and 8032 PETs. MAE has many powerful features, including macros and conditional assembly.

Patches to MAE for ASCII Printers

The following patches were provided by Carl Moser for ASCII printers:

1. To prevent sending a cursor down character before each line replace the following code with five NOPs:

```
74C4 A9 11        LDA #$11
74C6 20 53 72     JSR $7253
```

It will look like this:

```
74C4 EA EA
74C6 EA EA EA
```

2. To eliminate line feeds make the following changes:

	From	To
6467	20	60
5553	0A 0A 0A	0D 0D 0D
556D	0A 0A	0D 0D
5571	0A	0D
5572	12	07
558D	0A	0D
5584	0A	0D
55A1	0A 0D 0A	0D 0D 0D
68F5	C9 0A	C9 0D

3. To output true ASCII instead of PET ASCII change 7210 20 B4 74 JSR $74B4 to 20 53 72 JSR $7253.
4. If you are using paper wider than 80 columns then the output can be widened slightly by changing the following to six NOPs (EA):

```
6027 C0 39        CPY #$39
6029 90 02        BCC $602D
602B A0 39        LDY #$39
```

5. If your printer is not device 4 then change 725A from 4 to your printer's device number.
6. To fix the STP (word processor) for no linefeeds to the printer make the following changes:

	From	To
0931	20 36 09	EA EA EA
0934	A9 0A	EA EA
0963	C9 0A	C9 0D
099F	C9 0A	C9 0D

7. To avoid losing an occasional character with any printer, enter these patches:

```
7256 85 2B 78
726D A5 2B 58
```

Send Listing Output to Disk

If you wish to send the listing to a disk file instead of a printer, then this patch will allow it. The following code tells the printer to listen.

```
7259 A9 04      LDA #$04
725B 85 D4      STA $D4
725D 20 D5 F0 JSR $F0D5 ($F0BA,
Upgrade BASIC)
7260 20 48 F1 JSR $F148 ($F12D,
Upgrade BASIC)
```

Change this code to the following to tell the disk to listen:

```
7259 EA EA
725B EA EA
725D EA EA EA
7260 20 F0 77   JSR $77F0
77F0 A9 08      LDA #8
77F2 85 D4      STA $D4
77F4 20 D5 F0   JSR  $F0D5  ($F0BA,
                BASIC 2.0)
77F7 A9 68      LDA #$68
77F9 4C 43 F1   JMP  $F143  ($F128,
                BASIC 2.0)
```

To use this disk output routine you must open the file in BASIC as OPEN8,8,8,"dr:name,s,w" then SYS to the MAE warm start ($5003). Now when the TO IEEE command is given the output goes to the open file. When you are through with the file give the BA command to return to BASIC and CLOSE8 to close the file.

Using the Relocating Loader

To start, all zero page locations that you want to be relocated should be assigned starting at location zero with the .DI pseudo-op. On the PET there are few locations in zero page to use. They are fixed and will be set with the .DE pseudo-op, but the .DI feature would be useful to develop software for another computer. The absolute (non-zero-page) locations used should be assigned with the .DI command starting at location $0400, but be sure to keep the same relative relationships among the labels that you desire in the final code. Now start assembly at location $2000. Note that MAE defaults the start of assembly at location $0400, not at location $2000, so you must explicitly use the .BA pseudo-op.

Now assemble the file with the AS command. After the two passes, give the OU command with a file name to put the relocatable file onto disk. Once the output command has done its pass the relocatable loader is ready to be used.

When the loader asks for the file name, give it the name of the file created with the OU command. For the zero-page offset and absolute offset enter numbers to add to the zero page and absolute locations (as defined with the .DI command). This will assemble them at their correct addresses. On the PET these will usually be zero. Next, for the program execution offset, give the location (in hex) where you want the program to run. For the program store offset, give the location (in hex) where you wish the program to be stored in memory. This allows programs to be stored in a different location than they will execute.

The relocating loader allows you to give someone only the relocatable file while retaining the source code. Here are examples of the output generated using the relocating loader.

```
FILE NAME? "0:REL.REL.04AUG"
Z-PG OFFSET? 0
ABS OFFSET? 0
PGM EXE OFFSET? 7800
PGM STORE OFFSET? 7800

LOAD MAP:
Z = 0000-0000
A = 0000-06BB
E = 7800-7800
S = 7800-7800
```

For information on the MAE user group send a SASE to ASM/TED Users Group (ATUG), c/o Brent Anderson, 200 S. Century, Rantoul, IL 61866, (217) 893-4577.

Mr. Cochrane has a Master of Science in electrical and computer engineering. Most of his programming on the PET is in assembler using the MAE assembler. You may contact him at 1402 Sand Bar Ferry Rd., Beech Island, SC 29841.

Richvale Telecommunications

10610 BAYVIEW (Bayview Plaza)
RICHMOND HILL, ONTARIO, CANADA L4C 3N8
(416) 884-4165

$185.00 Canadian
$149.00 U.S.
PLUS CUSTOMS BROKERAGE, HANDLING AND MAILING CHARGE.

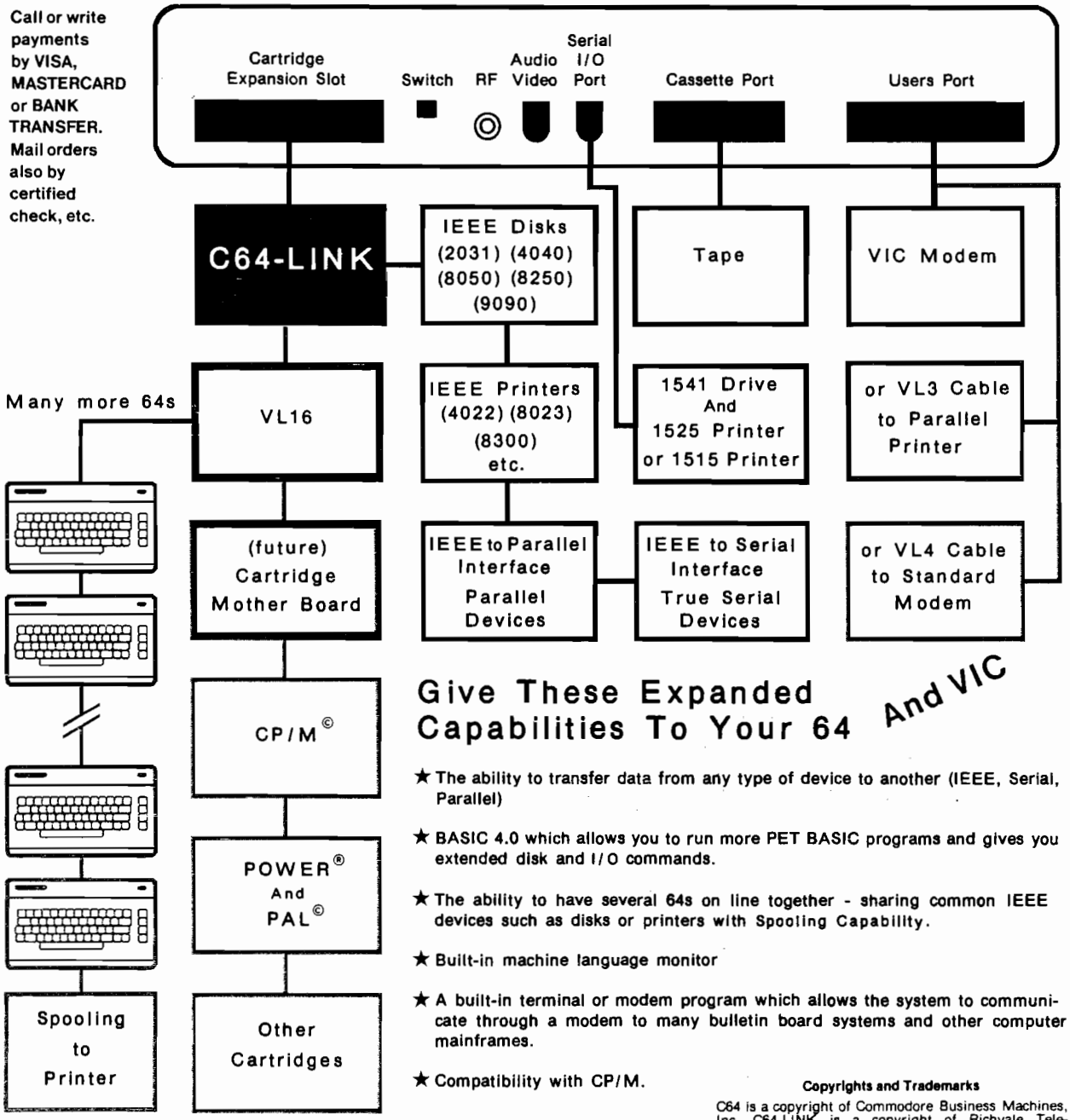
RTC

C64-LINK©
The Smart 64

Also available for VIC 20

RTC

Call or write payments by VISA, MASTERCARD or BANK TRANSFER. Mail orders also by certified check, etc.



Give These Expanded Capabilities To Your 64 And VIC

- ★ The ability to transfer data from any type of device to another (IEEE, Serial, Parallel)
- ★ BASIC 4.0 which allows you to run more PET BASIC programs and gives you extended disk and I/O commands.
- ★ The ability to have several 64s on line together - sharing common IEEE devices such as disks or printers with Spooling Capability.
- ★ Built-in machine language monitor
- ★ A built-in terminal or modem program which allows the system to communicate through a modem to many bulletin board systems and other computer mainframes.
- ★ Compatibility with CP/M.

Contact your local Commodore dealer or RTC.

Copyrights and Trademarks

C64 is a copyright of Commodore Business Machines, Inc. C64-LINK is a copyright of Richvale Telecommunications. CP/M is a registered trademark of Digital Research. POWER is a trademark of Professional Software. PAL is a copyright of Brad Templeton.

Circle No. 32

68000 INSTRUCTIONS

by Joe Hootman

Perhaps some of the most fascinating instructions implemented by the 68000 are those calssified as miscellaneous (see table 1).

Two trap instructions are available to the 68000 user. TRAP is a form of program control instruction; it loads the program counter with a particular vector and pushes the program counter and the status register onto the system stack. The TRAP instruction can select one of 16 trap vectors to be executed; any time the TRAP instruction is executed the machine will be vectored to the specified vector. The TRAPV instruction will perform in a similar manner to TRAP except that the V bit is checked; if it is set, the trap vector is fetched and executed. The TRAPV instruction is used extensively in two's complement arithmetic. When either of the TRAP instructions is executed, the processor goes into the exception processing state.

The Check Register Against Bounds (CHK) is used to compare the value in a register against a bound. The upper bound is expressed as a two's complement word-length integer located at the designated EA. The lower bound is zero. If the bound is exceeded, the CHK is selected and the processor goes into exception processing. This instruction allows the checking of array bounds by verifying that a data register contains a valid subscript. It is used extensively in higher-level languages such as Pascal.

Two special instructions are used to call subroutines. These instructions are the link (LINK) and Unlink (UNLK) instructions. The link statement is used to reserve space on the stack to be used by a called subroutine. The LINK instruction pushes the designated register onto the stack. The updated stack pointer is then loaded into the designated register. The specified 16-bit sign-extended displacement is then added to the stack pointer. This allows the subroutine to access the data area on the stack by indexing off the designated register.

UNLK is used to return the reserved data area. LINK and UNLK can be used to maintain a linked list of local data and parameters on the stack for nested subroutines. Generally the LINK state-

Figure 1

High Byte	Low Byte	
		Even Address
Operation word (opword)		word number 1
		Even Address +2
Immediate data/operand (if any)		word number 2
		Even Address +4
Source Effective Address extension (if any)		word number 3
		Even Address +6
Destination Effective Address extension (if any)		word number 4
16 bits wide		

***Figure 2:* Opword Field Using a Single EA**

Opword Field Using a Single EA

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X	X	X	Effective Address Mode			Register		

Figure 3

Opword Format Using Two EAs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	Size		Effective Address Register			Mode			Effective Address Mode			Register		

ment is used before the subroutine call and the UNLK is used just before the RTS at the end of the routine.

The NOP instruction just passes the control of the program to the next instruction.

Addressing Modes

Six general addressing mode categories are available to the 68000 user. Variations of the general categories extend the total number of addressing modes to 14. The 14 address modes combined with the 56 basic instructions give a potential of 784 separate instructions. However, some of the instructions do not make use of all the addressing modes. The addressing modes that are implemented in the 68000 are listed below.

1. Direct Addressing Using Register
   a. Data Register Direct
   b. Address Register Direct
2. Direct Memory Addressing
   a. Absolute Short
   b. Absolute Long
3. Indirect Memory Addressing
   a. Register Indirect
   b. Post-increment Register Indirect
   c. Pre-decrement Register Indirect
   d. Register Indirect with Displacement
   e. Register Indirect with Index and Displacement
4. Implied Register Addressing
5. Program Counter Relative Addressing
   a. PC-relative with Displacement
   b. PC-relative with Index and Displacement
6. Immediate Addressing
   a. Immediate
   b. Quick Immediate

The 68000 expects to find instructions in a prescribed sequence with the first 16-bit word being the operation word (opword). Every instruction must have an operation word; this word tells the processor what addressing mode is being used and how many extension words are associated with the instruction. Not every instruction needs to make use of the extension words. For example, the instructions that make use of implied addressing do not make use of the extension word.

The general format for the total instruction is given in figure 1.

A major difference between the 68000 and existing 8-bit processors is the way the 68000 accesses memory. The 68000 generally expects to fetch instructions from the memory on even addresses. If the address bus is forced to an odd address the processor will go

***Table 1:* Miscellaneous Instructions**

Mnemonic	Data Size/CCR	Name	Comments
NOP	— CCR X N Z V C - - - - -	No Operation	No operation occurs and execution continues with the next instruction. Opword Format 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 0 1 0 0 1 1 1 0 0 1 1 1 0 0 0 1
TRAP	— CCR X N Z V C - - - - -	Trap	This instruction forces the processor to one of 16 trap vectors. The processor will be put into the exception processing state. Opword Format 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 0 1 0 0 1 1 1 0 0 1 0 0 | Vector The vector field specifies which trap vector contains the new program counter to be loaded.
TRAPV	— CCR X N Z V C - - - - -	Trap on Overflow	This instruction initiates exception processing if the overflow condition is true. The new program counter is loaded from the TRAPV exception vector. Opword Format 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 0 1 0 0 1 1 1 0 0 1 1 1 0 1 1 0
CHK	16 CCR X N Z V C - • U U U	Check Register Against end Bounds	This instruction compares a specified data register to an upper bound. The upper bound, a 16-bit two's complement integer, is exceeded then the processor picks up the CHK exception vector and enters exception processing. Opword Format 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 0 1 0 0 | Register | 1 1 0 | Effective Address Mode | Register The field specifies the data register whose contents are to be checked. The effective address specifies the upper bound of the word to be checked against Dn. The following effective address modes cannot be used: 2, 13, 14.*
LINK	— CCR X N Z V C - - - - -	Link and Allocate	The following steps are executed: 1. The contents of the specified address register are pushed onto the stack. 2. The address register is then loaded with the updated SP. 3. A 16-bit sign-extended offset is added to the SP. Opword Format 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 0 1 0 0 1 1 1 0 0 1 0 1 0 | Register Displacement The register field specifies the address register through which the link is to be constructed; the displacement field specifies the two's complement integer that is to be added to the stack pointer.
UNLK	— CCR X N Z V C - - - - -	Unlink	The stack pointer is loaded with the specified address register and the address register is loaded with the long word data from the top of the stack. Opword Format 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 0 1 0 0 1 1 1 0 0 1 0 1 1 | Register The register field specifies the address through which the unlinking is to be done.

* See next month's table.

Products for Commodore, Atari, Apple, and others!

NEW THE MONKEY WRENCH II
A PROGRAMMERS AID FOR ATARI 800
NEW AND IMPROVED — 18 COMMANDS
PLUGS INTO RIGHT CARTRIDGE SLOT

If you are a person who likes to monkey around with the ATARI 800, then THE MONKEY WRENCH II is for you!! Make your programming tasks easier, less time-consuming and more fun. Why spend extra hours working on a BASIC program when the MONKEY WRENCH can do it for you in seconds. It can also make backup copies of boot type cassette programs. Plugs into the right slot and works with ATARI BASIC cartridge.

$59.95

The MONKEY WRENCH provides 18 direct mode commands. They are: AUTO LINE NUMBERING — Provides new line numbers when entering BASIC program lines. RENUMBER — Renumbers BASIC's line numbers including internal references. DELETE LINE NUMBERS — Removes a range BASIC line numbers. VARIABLES — Display all BASIC variables and their current value. Scrolling — Use the START & SELECT keys to display BASIC lines automatically. Scroll up or down BASIC program. FIND STRING — Find every occurrence of a string, XCHANGE STRING — Find every occurrence of a string and replace it with another string. MOVE LINES — Move lines from one part of program to another part of program. COPY LINES — Copy lines from one part of program to another part of program. FORMATTED LIST — Print BASIC program in special line format and automatic page numbering. DISK DIRECTORY — Display Disk Directory. CHANGE MARGINS — Provides the capability to easily change the screen margins. MEMORY TEST — Provides the capability to test RAM memory. CURSOR EXCHANGE — Allows usage of the cursor keys without holding down the CTRL key. UPPER CASE LOCK — Keeps the computer in the upper case character set. HEX CONVERSION — Converts a hexadecimal number to a decimal number. DECIMAL CONVERSION — Converts a decimal number to a hexadecimal number. MONITOR — Enter the machine language monitor.

In addition to the BASIC commands, the Monkey Wrench also contains a machine language monitor with 16 commands used to interact with the powerful features of the 6502 microprocessor.

NEW VIC RABBIT CARTRIDGE AND CBM 64 RABBIT CARTRIDGE

"High-Speed Cassette Load and Save!"

$39.95
(includes Cartridge and Manual)

Expansion Connector on the VIC Cartridge

"Don't waste your Life away waiting to LOAD and SAVE programs on Cassete Deck."

Load or Save 8K in approximately 30 seconds! Try it — your Un-Rabbitized VIC takes almost 3 minutes. It's not only Fast but VERY RELIABLE.

Almost as fast as VIC Disk Drive! Don't be foolish — Why buy the disk when you can get the VIC Rabbit for much, much less!

Easy to install — it just plugs in.
Expansion Connector on rear.
Works with or without Expansion Memory.
Works with VIC Cassette Deck.
12 Commands provide other neat features.

Also Available for 2001, 4001, and 8032

Now for the "64" STCP — 300/1200 Baud
Standard Terminal Communications Package

'PFO' IOD OOA CP<D1>D2 BELL = 12:30:00 10:14:36

Don't settle for non-standard Communications Protocol! Access Micro Net, Source, Bulletin Boards, Local Mainframe, etc.

- Complete Package — Includes RS232 Interface Board and software (does not include modem)
- Communicates in Industry Standard ASCII
- Upload/Download to/from Disk
- Automatic File Translation
- Can be controlled from keyboard or user supplied basic or machine language program

Specify 3.0 or 4.0 ROMS or 8032 Commodore Computer 4040 or 8050 or PEDISK II Disk or CBM64 on 1541.

Price: $129.95

PET BASIC SCROLL PROGRAM
Scroll thru Basic Programs using cursor up/down keys. Specify computer. $6.00 on cassette, $9.00 on diskette.

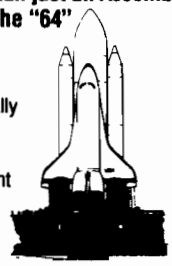
65C02 MAE
Same as our MAE but enhanced for the new 65C02 Opcodes. Turns your computer into a development system for the new ROCKWELL 65C02 Microprocessor. $200.00 — Specify Computer.

6800 CROSS ASSEMBLER
A Cross Assembler based on the MAE that runs on the PET, Apple, or Atari but assembles opcodes for the Motorola 6800 microprocessor. Turns your computer into a development system for the Motorola 6800 Microprocessor. $200.00 — Specify Computer.

ATARI and VIC Cartridges
EHS can supply large quantities of ATARI and VIC Cartridges for software developers. If you need cartridges, call for pricing.

More than just an Assembler/Editor! Now for the "64"
MAE
for PET APPLE ATARI

It's a Professionally Designed Software Development System



~~$169.95~~ New Price $99.95

Blast off with the software used on the space shuttle project!

- Designed to improve Programmer Productivity.
- Similar syntax and commands — No need to relearn peculiar syntaxes and commands when you go from PET to APPLE to ATARI.
- Coresident Assembler/Editor — No need to load the Editor then the Assembler then the Editor, etc.
- Also includes Word Processor, Relocating Loader, and much more.
- Options: EPROM Programmer, unimplemented opcode circuitry.
- STILL NOT CONVINCED: Send for free spec sheet!

ATARI AND PET EPROM PROGRAMMER
Programs 2716 and 2532 EPROMs. Includes hardware and software. PET = $75.00 – ATARI (includes sophisticated machine language monitor) = $119.95

TRAP 65
TRAP 65 is a hardware device that plugs into your 6502's socket. Prevents execution of unimplemented opcodes and provides capability to extend the machines' instruction set.
For PET/APPLE/SYM.
Reduced from $149.95 to $69.95

5¼ INCH SOFT SECTORED DISKETTES
Highest quality. We use them on our PETs, APPLEs, ATARIs, and other computers. $22.50/10 or $44.50/20

Prowriter Printer - Excellent dot matrix print Parallel = $489 00
Serial = $600 00 IEEE = $589 00

DC Hayes Smart Modem = $235 00
DC Hayes Micro Modem II = $289 00

Rana Disk Drive - 375
4 Drive Controller - 114

EPROMS 2716 = $4.50 2532 = $7.50
Over 40 Commodore Programs by Baker (on 4040) = $25.00

Eastern House
3239 Linda Dr.
Winston-Salem, N.C. 27106
(919) 924-2889 (919) 748-8446
Send for free catalog!

VISA · MasterCard

Circle No. 41

Design & Illustration: Paul Stephenson

ADVENTURE.
THE KEY IS YOUR COMPUTER

KONGO KONG

Climb ladders, avoid the barrels the crazy ape is rolling at you, and rescue the damsel. Fast machine code action.

Available for VIC-20 and COMMODORE 64

PLAYED WITH JOYSTICK OR KEYBOARD

$19.95

GRAVE ROBBERS

Introducing the first GRAPHIC ADVENTURE ever available on the VIC-20! With realistic audio-visual effects, you explore an old deserted graveyard and actually see the perils that lie beyond.

Available for VIC-20 and COMMODORE 64

PLAYED WITH KEYBOARD

$14.95

ANNIHILATOR

Defend your planet against the hostile aliens. All machine code makes this "Defender-like" program one of our best arcade games.

Available for VIC-20 and COMMODORE 64

PLAYED WITH JOYSTICK

$19.95

ADVENTURE PACK I

(3 programs)

Jack and the Beanstalk Defeat the Giant in your quest for his Golden Hen.

Computer Adventure Re-live the "excitement" of getting your computer. An adventure with a very different flavor.

Moon Base Alpha You must find a way to destroy the meteor that is racing towards your base, or else all moon colonies will be demolished!

Available for VIC-20 and COMMODORE 64

PLAYED WITH KEYBOARD

$19.95

ADVENTURE PACK II

(3 programs)

African Escape As the sole survivor of a plane crash, you must find your way out of the dark continent.

Hospital Adventure You are a spy whose mission is to complete the bungled assassination attempt on the evil dictator, who is recuperating in the hospital under heavy guard.

Bomb Threat Get back to town to warn the authorities of the bomb planted by the terrorists who left you prisoner at their hideaut.

Available for VIC 20 and COMMODORE 64

PLAYED WITH KEYBOARD

$19.95

CHOMPER MAN

High speed machine action. Don't let the bullies catch you in a game packed full of machine code.

Available for COMMODORE 64

PLAYED WITH JOYSTICK OR KEYBOARD

$19.95

Programs for the **VIC-20** and the **COMMODORE**

VICTORY SOFTWARE®

VICTORY SOFTWARE INC.
7 Valley Brook Road, Paoli, PA 19301
(215) 296-3787

All programs fit standard memo
All programs a on cassette tap

Check your local dealer or order directly.
ORDERING We accept personal cfecks, money orders, VISA, and MASTER-CHARGE. Charge orders please include number and expiration date. Overseas orders please use charge or have check payable through a U.S. bank. Add $1.50 postage and handling per order. PA residents please add 6% sales tax.

Circle No. 39

The MICRO™ LEARNING CENTER

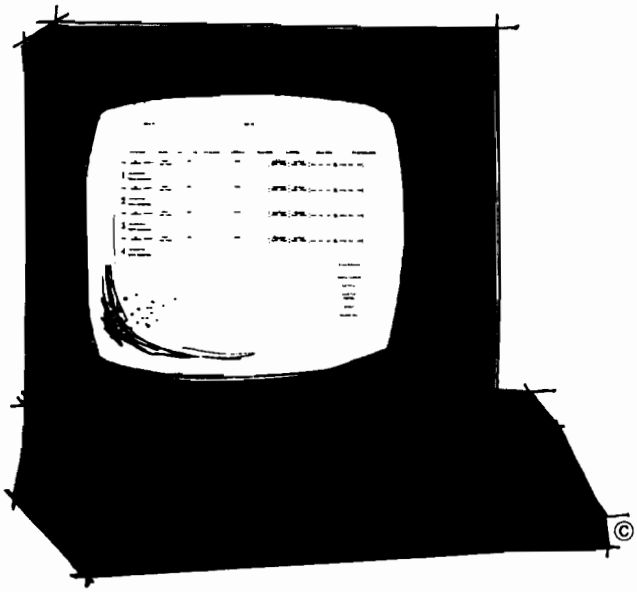
[redacted]

Atari 800 **Texas Instrument**

Features:

- **Breakup for the VIC-20**
  by Loren Wright and Phil Daley
  A simple animation demonstration of PEEKing and POKEing screen memory.

- **Addressfile**
  by Brian Zupke
  A file management system for the VIC-20 and other Commodore computers



HAVE YOU FLOWN YOUR ATARI TODAY?

Imagine yourself at the controls of a small, single-engine plane, 10,000 feet in the air, on your approach to the runway and safety. You're running low on fuel, but your instruments show that you're on the glide path, and lined up with the runway. It's a beautiful, sunny day, and you can see the airport in the distance, across the grassy fields. But the crosswind is tricky, and it will take all your skill to land safely. You're coming down now, and the runway is getting closer. A bit left, OK, now lower the power, fine, now put down the flaps. Pull the nose up a bit more, you're a little low. Watch the power! Don't stall. OK. Here comes the runway. You hear the squeal of tires on pavement, your pulse quickens, you're down, but watch it, you're pulling right! Brakes, brakes! Left more! You've stopped safely! Good job. The first real-time flight simulator for ATARI is now available from MMG Micro Software. Written entirely in machine language, there are four levels of difficulty, landings in clear or foggy weather, landings with or without instruments, and with or without the real-time view from the cockpit. **Final Flight!** requires Atari 400/800, 24K, 1 joy stick, and is offered on tape or disk for the same suggested retail price of $29.95.

MMG

Circle No. 38

Fin
Fligh
is available
your local dealer or dire
from **MMG Micro Software.** Ju
send check or money order to P.O. B
131, Marlboro, N.J. 07746 or for Mastercar
Visa, and C.O.D. deliveries call **(201)431-3472.** Plea
add $3.00 for postage and handling. New Jersey residents ac
6% sales tax.

Atari is a registered trademark of Atari,

BREAKUP

by Loren Wright and Phil Daley

BREAKUP
requires:
VIC-20
Easily modifiable for other PETs

A simple animation demonstration of PEEKing and POKEing screen memory.

Get out your paddles or joystick and get ready to hit the ball with your bumper. By knocking down a wall of bricks, you can now show all your friends how dexterous you are.

Breakup is a simple animation display game that presents the principles of PEEKing and POKEing the screen memory to plot and erase characters on the screen, and to test for collisions. It includes a "ball" that moves around the screen, rebounds from struck objects, and knocks out blocks in the wall of bricks. It also includes a player-controlled "bumper" to keep the ball from going out-of-bounds and being lost, a defined playing field with three walls from which to bounce the ball, and two rows of blocks to be knocked off by your bouncing ball.

The game keeps score; 5 for each block and 1 for each bounce off the bumper. The previous highest score during each session is kept and displayed at the end of the game. More than 170 points qualifies you for a bonus of 50 points, new blocks, and an extra ball. You can choose to play with a joystick, paddles, or keyboard. If you choose the keyboard, the Function keys F1 and F7 move the bumper up and down. If you hold down the shift key at the same time, it increases the bumper movement speed.

Operating Instructions

1. Type in the BREAKUP program and type 'RUN'.
2. You are first asked whether you have a joystick, paddles, or will play from the keyboard. Choose the correct response — J, P, or K.
3. If you specify hardware that you don't have, you will have to press the RUN/STOP key to recover.
4. The program will display the playing field, the brick wall, and your bumper. When you are ready to start play, press the button on the joystick or paddle X, or the space bar if you are using the keyboard.

The Program

The ball starts from a random position on the left side of the screen and travels to the right, hitting a block. This causes the block to disappear, adds five to your score, and the ball rebounds toward the left. Here is the challenge: You must position the bumper to keep the ball from travelling out-of-bounds and off the screen, thereby losing the ball. If you are successful, the ball will rebound back toward the blocks. If you miss the ball, the game freezes so that you can see how close (or how far) you were from the ball when you missed it. You are allowed three balls (four if you get the bonus), so try not to miss. You will notice that the angle and speed of the ball increase the closer you hit the ball to the ends of the bumper. Hitting the ball near the center of the

bumper helps to restore the ball to the original direction.

The program stores the high score from game to game until the computer is turned off. Challenge your friends to a match and see who can get the highest score.

To create animation on a computer screen, we need to know how the computer displays information on the screen and how we can change that information. The television is constantly updating the information being displayed, about 30 times per second. A computer that uses memory-mapped video stores the screen information in RAM (Random Access Memory) so that the video display routine can have this information available at all times. The location in memory where this information is stored on the VIC depends on the amount of memory available on your particular VIC. The formula for determining this location is:

```
1. S = 4*(PEEK(36866)AND 128) +
   64*(PEEK(36869)AND112)
```

S returns with the start of video display memory location.

To plot a character in a given location on the screen, we must use the following equation:

```
2. A = S + X + 22 * Y
```

where A is the location in memory to either PEEK or POKE the character, depending upon the program usage. X and Y are the Cartesian coordinates of the location, starting at 0,0 in the upper left corner of the screen.

Appendix D of the *VIC Programmer's Reference Guide* contains a table of the codes to be POKEd in screen memory and the resulting characters. As an example, type equation 1 above into your VIC to assign the proper value to S, and then type 'POKE 36879,8 : POKE S, 81'. A small white ball should appear on your screen in the upper left corner. We used equation 2 above with X = 0 and Y = 0. By changing the X and Y values in that equation and POKEing the resulting locations with 81, you can print little white balls anywhere on the screen. X values can range from 0 to 22 and Y values from 0 to 20.

How do you erase them? Look up 'space' (which is 32 in the screen code chart and POKE that number into the same location as you originally plotted the ball. This effectively erases whatever is on the screen position. Location 36879 is the memory position for the background and border colors on the screen. Appendix B of the *VIC Programmer's Reference Guide* lists all the possible backgrounds and borders and the correct number to be POKEd into this location. The reason for turning the background black and plotting the ball white is that white is the default color when POKEing the screen memory locations.

POKEing an 81 to the S location with the screen set to white results in an invisible ball being plotted. To see the ball on a white screen we have to change the color of the ball. Each screen memory location also has associated with it a screen color memory location. For example, if you POKE 38520,2, this will change the white ball to red. Since two locations must be POKEd for any color other than white, we use white as the ball color to improve the speed of the program and lessen the complexity of the plotting routines.

Paddles and Joysticks

Using paddles or a joystick helps to make the game more interesting and easier to play, but programming them is a little obscure, especially for the novice BASIC programmer. If you don't understand the following information on first reading, look at the BREAKUP program and see how it is done, then reread this section.

There are five switches on a joystick. Four of them are controlled by the joystick and one is for the fire button. You must PEEK the locations of the VIA to determine which joystick switches are closed. The values are normally 0 and change to 1 (or −1) when closed. We also have to POKE the Data Direction Register to input mode. For VIA #1 this is accomplished easily by a POKE 37139,0. Changing VIA #2 to input disables keyboard input, so we change only the DDR when we are polling the joystick and return it to normal during the rest of the program. Therefore, we POKE 37154,127, read the joystick, and POKE 37154,255 immediately afterward.

Reading switches 0, 1, 2, and 4 are done as follows:

```
POKE 37139,0
S0 =((PEEK(37137)AND4)=0)
S1 = -((PEEK(37137)AND8)=0)
S2 =((PEEK(37137)AND16)=0)
S4 = -((PEEK(37137)AND32)=0)
```

Reading switch 3 is a bit trickier:

```
POKE 37154,127
S3 = -((PEEK(37152)AND128)=0)
POKE 37154,255
```

Because BREAKUP uses only up and down readings of the joystick, the switch 3 program lines are not needed. Switch 0 is North, switch 1 is South, switch 2 is West, switch 3 is East, and switch 4 is the fire button.

The VIC paddles are read in the same manner except that an actual value may be PEEKed for the paddle settings. The switches are the same. Here are the lines necessary to implement paddle readings:

```
POKE 37139,0                Set DDR to Input
PX = PEEK(36872)            Paddle X value
SX = -((PEEK(37137)
  AND16 = 0)                Paddle X switch
PY = PEEK(36873)            Paddle Y value

POKE 37154,127
SY = -((PEEK(37152)
  AND128) = 0)
POKE 37154,255              Paddle Y switch
```

Program Description

Lines 1000-1020 (the subroutine to plot the bumper) calculate the screen position of the bumper and then POKE the locations either with a bar (B = 245) or a space (B = 32), depending on whether the subroutine is called to plot or erase the bumper. Lines 3070 and 3090 erase and draw the ball, respectively. Line 3080 looks at the screen location to see if a block has been hit. If so, the subroutine at line 5000 is called to update the score and put "english" on the ball if it was called because of bumper contact.

Lines 4050-4060 plot the block walls. If you remove lines 4050 and 4060, the game turns into "PONG" with the player scoring one point for each rebound from the bumper. Line 4140 picks a random spot along the left-hand side of the screen for the new ball to appear.

Line 3050 tests for out-of-bounds in the Y range and line 3100 tests for out-of-bounds in the X range. If the ball is out-of-bounds it reverses the ball direction. Line 3120 tests for hitting the bumper. If it has hit the bumper then the 5000 subroutine is called to update the score and put english on the ball, as previously mentioned.

As a whole, the routine moves rather slowly due to the limitations of BASIC language. We employed several techniques to keep the program moving as fast a possible. Variables used most often (those concerning the ball and bumper) were initialized at the beginning of the program so that they would be placed in the beginning of the variable storage area. The screen update subroutines were placed near the beginning of the program so that the interpreter would spend less time looking for the line numbers of the subroutines. In line 2010 (the bumper moving routine) a return is placed in the first line so that when the bumper is not to be moved, the program doesn't spend time replotting an unmoved bumper. That is why the ball moves much faster when you are not changing the bumper location on the screen.

The screen POKEs can be any character (see the screen code chart in the programming manual). Breakup uses 32 (the space) to erase the screen. The ball is 81, the blocks are 102, the bumper is 245. These could be changed easily to different shapes. The scoring subroutine at line 5000 can be changed to make the score advance either faster or slower. Also, the scoring could be changed to increase the bonuses or provide snappy comments at the conclusion of the game. The block-plotting routine at line 4050 could be modified to increase or decrease the number of columns of blocks. If you plot them nearer to the bumper, then the 15 in line 3080 will need to be decreased commensurately.

Changes for Other PETs

The POKEs in 3050, 3080, 3110, 3130, 3150, 4020, 4170, 4250, 6000, and 6090 may be removed. They concern screen color and bounce sounds.

```
4000 COL = 40
4010 S0 = 32768
```

After typing the program as is, other PET owners should make these changes:

```
2090 S = 0:IFA$ = " = "THENS = 1
2100 IFA$ = CHR$(20)THENS = -1
2110 IFA$ = " ∧ "THENS = 3
2120 IFA$ = CHR$(148)THENS = -3
```

Line 4010 is also different for the C-64:

```
4010 S0 = 1024
```

You may contact the authors at MICRO Magazine, Box 6502, Amherst, New Hampshire 03031.

Breakup Listing

```
10 X=0:Y=X:X1=X:Y1=X:S0=X:COL=X:S1=X:S2=S:S3=X:DY=X:DX=X:GOTO4000
1000 S1=S0+COL*A:S2=S0+COL*(A+1):S3=S0+COL*(A+2)
1010 POKES1,B:POKES2,B:POKES3,B
1020 RETURN
2000 TS=INT(PEEK(36872)/17)+2
2010 IFTS<A+2ANDTS>A-2THENRETURN
2020 B=SP:GOSUB1000
2030 A=TS:GOTO2150
2040 00=-((PEEK(PA)AND4)=0):01=((PEEK(PA)AND8)=0)
2050 IFQ0<>0THENS=-3:GOTO2130
2060 IFQ1<>0THENS=3:GOTO2130
2070 RETURN
2080 GETA$:IFA$=""THENRETURN
2090 S=0:IFA$=" "THENS=1
2100 IFA$=" "THENS=-1
2110 IFA$=" "THENS=3
2120 IFA$=" "THENS=-3
2130 B=SP:GOSUB1000
2140 A=A+S
2150 IFA<2THENA=2
2160 IFA>17THENA=17
2170 B=D:GOSUB1000
2180 RETURN
3000 OK=1
3010 ONPDLGOSUB2000,2040,2080
3020 XI=X:Y1=Y
3030 Y=Y+DY:IFY>19THENY=19
3040 IFY<2THENY=2
3050 IFY=2ORY=19THENDY=-DY:POKEV3,230
3060 X=X+DX:IFX>19THENX=19
```

(Continued on page 78)

Addressfile

by Brian Zupke

The program Addressfile stores names, addresses, phone numbers, and memos in cassette files. Entries can be entered, deleted, or changed, and files can be updated at any time. By using the search routine, you can locate any entry having a particular name, address, phone number, or memo.

The program was written on a 13K VIC-20. However, if the REM statements are deleted, it will run with 8K. Since the program is somewhat large, I am offering to make a copy of it for anyone who sends me either a blank tape and a self-addressed, stamped envelope, or $3.00.

To allow for fast and easy data entry and editing, INPUT statements were used. This means that no colons or commas are allowed. The program has eight major functions: ADD ENTRIES, CHANGE ENTRIES, VIEW ENTRIES, DELETE ENTRY, SEARCH FILE, SAVE FILE, LOAD FILE, and END PROGRAM. Below are detailed descriptions of each function.

ADD ENTRIES — Add new entries to the file.

Added entries are put at the end of the current file, or the file presently in memory. After typing an entry, the computer asks if you want to make any changes. The default is 'N'. If you type 'Y', then the cursor moves to the top of the entry and it is re-entered. If you don't want to change a certain line in the entry, just hit RETURN; otherwise make the change and then hit RETURN. Be sure to hit RETURN on each line of the entry since the computer is INPUTing all of them again. If you do not want to make any changes then the computer will ask if you are finished adding entries to the file. The default is 'N'. You must type a 'Y' to return to the menu.

CHANGE ENTRY — Change any part or all of an entry.

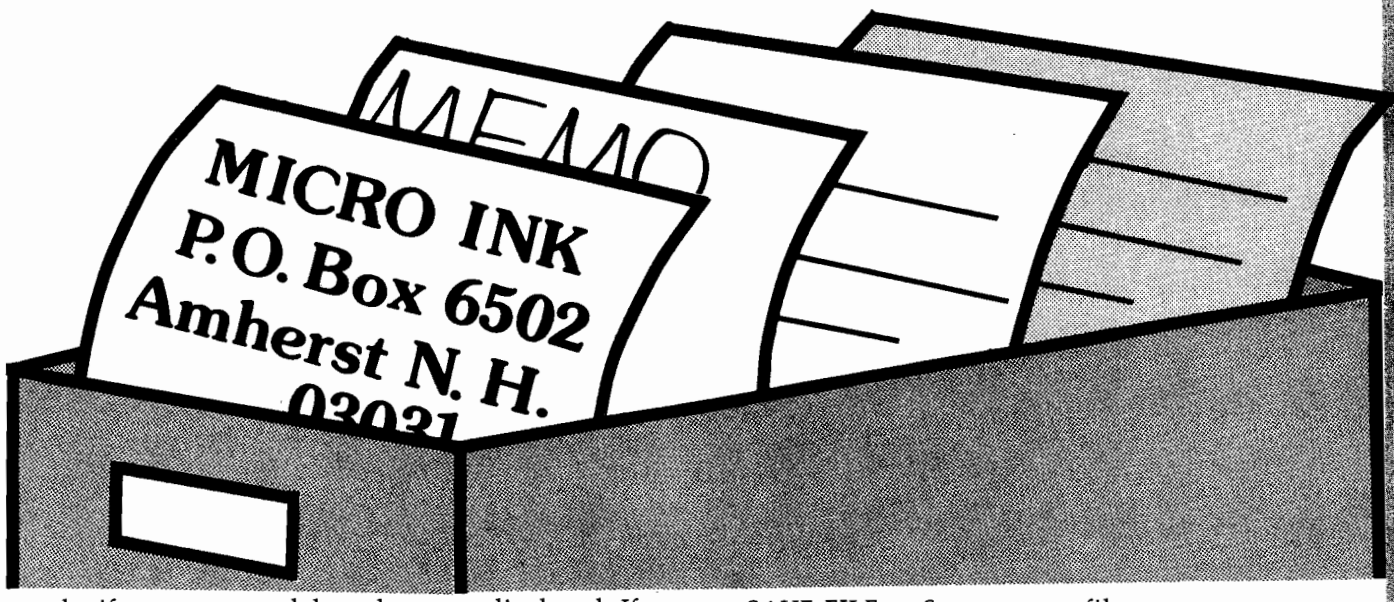
The computer first asks for the number of the entry to be changed. If you don't want to change an entry respond with a number less than one, which returns control to the menu. When a valid entry number is typed in, that entry is listed. Question marks are printed out before each line so that the entry can be re-entered. Change the entry as you would when adding entries. After the changes are made, the computer asks if you want to change another entry. The default is 'Y'. Control is returned to the menu if you type 'N'.

VIEW ENTRIES — View any or all entries in the file.

The computer first asks for a starting entry to be viewed. When a number is typed in, the corresponding entry is displayed. The computer will then ask if you want to see another entry. The default is 'Y'. If the response is 'N', then control is returned to the menu; otherwise, the entry immediately following the one on the screen is displayed. For example, if the sixth entry is on the screen, the seventh entry will be displayed next. After the last entry is displayed and you request to see another entry, the computer will display an end-of-file message briefly and control is returned to the menu.

DELETE ENTRY — Delete an entry in the file.

The computer asks for the number of the entry to be deleted. Once a number is typed in, the corresponding entry is displayed. The computer



asks if you want to delete the entry displayed. If you type 'Y', then the entry is deleted and all the entries above that one are moved down one. For example, if entry six is deleted, then entry seven becomes entry six, eight becomes seven, etc. If you type 'N' the entry is not deleted. In either case, the computer will ask for another entry number. If you type a number less than one, control is returned to the menu.

SEARCH FILE — Search the current file for a specific name, address, etc.

There are four things you can search for in the file: the name, address, phone number, or memo. After choosing one, the computer will ask if you want a full printout of each entry in the file that matches. The default is 'Y'. If you type 'N' then only the entry number is printed. Then the computer will ask if you want a pause after each match is printed. Again the default is 'Y;. With the pause, a key will have to be pressed after each match is listed. Then the computer will ask for the specific item for which to search. If you are searching for a name and two names are typed in (first and last), the computer will check both names individually. This is true for names in an entry. If either name typed in matches either name in an entry, then a match is found. When searching for an address, a match is found if the street is the same, the city is the same, or if the state and zip code are the same as the entry's. With the phone number and memo, a match is found when they are the same as the entry's. When all entries have been checked an end of search message is displayed. A key has to be pressed to return to the search menu. If you are through searching, press '5' to return to the menu.

SAVE FILE — Save current file on cassette tape.

The computer asks for a name under which to save the file. After you type in the name, the computer will display the press play and record message. Once you start the tape, the computer will open and save the file, and once the file is saved, it is CLOSED and control is returned to the menu. When you save a file you must be sure of two things: don't overwrite another file, and make sure that each line in every entry has at least one character or number in it. If you fail to do this the entries will contain the wrong information when the file is loaded back in. This happens because the computer will pass over a blank line and INPUT# the first non-blank line in the same variable, which causes the entries to 'bunch up.'

LOAD FILE — Load a file from tape into the computer.

The computer asks for the name of the file you wish to load. When a name is typed in, the corresponding file is loaded into the computer. Control is returned to the menu after the file is loaded.

END PROGRAM — Terminates program.

Caution should be taken with this function. Be sure that you save any wanted files on tape before quitting.

(Addressfile Listing on next page)

Addressfile Listing

```
10 REM     ****************
11 REM     *              *
12 REM     * ADDRESSFILE *
13 REM     *              *
16 REM     *     BY       *
17 REM     *              *
18 REM     * BRIAN ZUPKE *
19 REM     *              *
20 REM     ****************
99 REM
100 DIMN$(100),S$(100),C$(100),T$(100),Z$(100),M$(100),P$(100):E=1:Q$="? "
130 REM * PRINT MAIN MENU *
150 PRINT"■■■■    ADDRESSFILE"
160 PRINT"■■";SPC(8)"MENU":PRINT"■■■■  1) ADD ENTRIES":PRINT"  2) CHANGE ENTRIES
"
170 PRINT"  3) VIEW ENTRIES":PRINT"  4) DELETE ENTRY":PRINT"  5) SEARCH FILE"
175 PRINT"  6) SAVE FILE":PRINT"  7) LOAD FILE":PRINT"  8) END PROGRAM"
177 REM INPUT USERS CHOICE
180 PRINT"■■       -CHOICE-■";INPUT"      ";C
182 IFC=8THENEND
185 REM JUMP TO CORRECT ROUTINE
190 ONCGOTO200,300,400,700,800,500,600
191 GOTO150
195 REM * ADD ENTRY TO FILE *
200 PRINT"■■    ADD ENTRIES■"
210 PRINT"ENTRY #"E"■"
220 PRINT"NAME:":INPUTN$(E)
230 PRINT"STREET:":INPUTS$(E)
240 PRINT"CITY:":INPUTC$(E)
250 PRINT"STATE:":INPUTT$(E)
260 PRINT"ZIP-CODE:":INPUTZ$(E)
265 PRINT"PHONE:":INPUTP$(E)
270 PRINT"MEMO:":INPUTM$(E)
275 REM CHECK IF NEW ENTRY NEEDS TO BE CHANGED
280 INPUT"CHANGES■■■■■■■■■";Y$:IFLEFT$(Y$,1)<>"Y"THEN294
285 REM IF CHANGE THEN CURSOR UP & INPUT ENTRY AGAIN
290 PRINT"■■■■■■■";GOTO220
293 REM CHECK IF DONE ADDING ENTRIES
294 E=E+1:INPUT"DONE■■■■■■■■■";Y$:IFLEFT$(Y$,1)="Y"THEN150
295 GOTO 200
297 REM * CHANGE ENTRY IN FILE *
299 REM INPUT ENTRY # TO BE CHANGED
300 PRINT"■■ CHANGE ENTRY■":INPUT"ENTRY #";X:PRINT"■";G$;N$(X):PRINTG$;S$(X)
305 REM DISPLAY ENTRY
310 PRINTG$;C$(X):PRINTG$;T$(X);PRINTG$;Z$(X):PRINTG$;P$(X):PRINTG$;M$(X)
315 REM CURSOR UP & ENTER ENTRY
320 PRINT"■■■■■■■";INPUTN$(X):INPUTS$(X):INPUTC$(X)
321 INPUTT$(X):INPUTZ$(X):INPUTP$(X):INPUTM$(X)
325 REM CHECK FOR MORE ENTRY CHANGES
330 INPUT"ANY MORE■■■■■■■■■";Y$:IFLEFT$(Y$,1)="Y"THEN300
340 GOTO150
360 REM * VIEW ENTRIES IN FILE *
370 REM INPUT STARTING ENTRY
400 PRINT"■■  VIEW ENTRIES■":INPUT"STARTING ENTRY";Y:PRINT"■■■■■"
410 FORX=YTOE-1:PRINTN$(X);PRINTS$(X);PRINTC$(X)
415 PRINTT$(X),Z$(X);PRINTP$(X);PRINTM$(X)"■■■■■"
420 PRINT"ENTRY #";X:INPUT"CONTINUE    Y■■■";Y$:IFLEFT$(Y$,1)<>"Y"THEN150
425 REM CONTINUE VIEW
427 REM PRINT END OF FILE MESSAGE WHEN DONE
430 PRINT"■■■■■":NEXTX:PRINT"END OF FILE":FORX=1TO2000:NEXT:GOTO150
480 REM * SAVE FILE ON TAPE *
500 PRINT"■■  SAVE FILE■■■■■"
505 REM INPUT NAME FOR FILE
510 PRINT"FILENAME:■":INPUTA$
512 IFA$="STOP"THEN150
515 REM SAVE ALL ENTRIES IN NAMED FILE
520 OPEN1,1,1,A$
530 PRINT#1,E;FORX=1TOE-1
540 PRINT#1,N$(X):PRINT#1,S$(X):PRINT#1,C$(X)
543 PRINT#1,T$(X):PRINT#1,Z$(X):PRINT#1,P$(X):PRINT#1,M$(X)
545 REM END SAVE
550 NEXT:CLOSE1:GOTO150
580 REM * LOAD FILE FROM TAPE *
600 PRINT"■■■■LOAD FILE■■■■■"
605 REM INPUT NAME OF FILE TO LOAD
610 PRINT"FILENAME:■":INPUTA$
612 IFA$="STOP"THEN150
615 REM LOAD NAMED FILE FROM TAPE
620 OPEN1,1,0,A$
630 INPUT#1,E;FORX=1TOE-1
640 INPUT#1,N$(X):INPUT#1,S$(X):INPUT#1,C$(X)
643 INPUT#1,T$(X):INPUT#1,Z$(X):INPUT#1,P$(X):INPUT#1,M$(X)
645 REM END OF LOAD
650 NEXTX:CLOSE1:GOTO150
680 REM * DELETE ENTRY FROM FILE *
700 PRINT"■■  DELET ENTRY■■"
701 REM INPUT ENTRY # TO BE DELETED
705 PRINT"ENTRY #:":INPUTN
710 IFN<1THEN150
715 REM IF # IS OUT OF BOUNDS, STOP
720 IFNOE-1THENPRINT"ENTRY DOES NOT EXIST":GOTO705
723 REM PRINT ENTRY
725 PRINT"■";N$(N):PRINTS$(N):PRINTC$(N):PRINTT$(N),Z$(N):PRINTP$(N):PRINTM$(N)
;"■■"
727 REM LAST CHANCE NOT TO DELETE ENTRY
730 INPUT"DELETE":Y$:IFLEFT$(Y$,1)<>"Y"THEN700
735 REM MOVE ALL ENTRIES ABOVE # DOWN ONE
740 FORX=NTOE-1:N$(X)=N$(X+1):S$(X)=S$(X+1):C$(X)=C$(X+1)
743 T$(X)=T$(X+1):P$(X)=P$(X+1):M$(X)=M$(X+1)
745 Z$(X)=Z$(X+1)
747 REM REDUCE TOTAL # OF ENTRIES
750 NEXT:E=E-1:GOTO700
780 REM * SEARCH ENTRIES IN FILE *
800 PRINT"■■  SEARCH FILE■"
810 PRINT"  CHOICES:■■":PRINT"1) SEARCH FOR NAME"
820 PRINT"2) SEARCH FOR ADDRESS":PRINT"3) SEARCH FOR PHONE #"
825 PRINT"4) SEARCH FOR MEMO":PRINT"5) RETURN TO MENU"
830 INPUT"CHOICE";C:IFC<1ORC>5THEN800
840 IFC=5THEN150
842 REM  CLEAR FLAGS
845 P=0:W=0
850 PRINT"DO YOU WANT A FULL    PRINTOUT OF EACH      POSSIBLE MATCH?"
856 REM FOR A PRINTOUT OF EACH MATCH SET PRINTOUT FLAG
860 INPUT"■■■Y/N■■■";Y$:IFLEFT$(Y$,1)="Y"THENP=1
870 PRINT"DO YOU WANT A PAUSE   AFTER EACH LISTING?"
876 REM FOR PAUSE AFTER EACH MATCH SET PAUSE FLAG
880 INPUT"■■■Y/N■■■";Y$:IFLEFT$(Y$,1)="Y"THENW=1
885 REM JUMP TO CORRECT SEARCH INPUT
890 ONCGOTO900,940,980,970
895 REM * INPUT NAME TO SEARCH *
900 PRINT"NAME:":INPUTN$:IFN$="STOP"THEN800
910 FORX=1TOLEN(N$):IFMID$(N$,X,1)=" "THENS30
920 NEXT:L$=N$:R$="":GOTO1000
930 L$=MID$(N$,1,X-1):R$=MID$(N$,X+1):GOTO1000
936 REM * INPUT ADDRESS TO SEARCH *
940 PRINT"USE STANDARD          ABREVIATIONS."
950 PRINT"STREET/P.O.BOX:":INPUTS$:IFS$="STOP"THEN800
960 PRINT"CITY:":INPUTC$:PRINT"STATE:":INPUTT$:PRINT"ZIP CODE:":INPUTZ$:GOTO100
0
966 REM * INPUT MEMO TO SEARCH *
970 PRINT"MEMO:":INPUTM$:IFM$="STOP"THEN800
972 GOTO1000
976 REM * INPUT PHONE # TO SEARCH *
980 PRINT"PHONE #:":INPUTP$:IFP$="STOP"THEN800
995 REM * START OF SEARCH ROUTINE *
1000 PRINT"SEARCHING FOR ENTRY"
1006 REM JUMP TO CORRECT MATCH CHECK
1010 FORX=1TOE-1:ONCGOTO1100,1150,1210,1190
1015 REM CONTINUE SEARCH
1020 NEXTX:PRINT"END OF SEARCH":PRINT"HIT KEY TO CONTINUE"
1030 GETA$:IFA$=""THEN1030
1035 GOTO800
1037 REM MATCH FOUND-PRINT ENTRY *
1040 PRINT"ENTRY #";X:IFP=0THEN1060
1046 REM PRINT NAME,ADDRESS & MEMO OF MATCH
1050 PRINTN$(X):PRINTS$(X):PRINTC$(X)
1055 PRINTT$(X),Z$(X):PRINTP$(X):PRINTM$(X)
1060 IFW=0THEN1020
1070 PRINT"HIT KEY TO CONTINUE"
1076 REM PAUSE BEFORE CONTINUING SEARCH
1080 GETA$:IFA$=""THEN1080
1090 GOTO1020
1094 REM * CHECK NAME OF ENTRY FOR MATCH *
1096 REM FIND FIRST & LAST NAME
1100 FORY=1TOLEN(N$(X)):IFMID$(N$(X),Y,1)=" "THEN1120
1110 NEXTY:F$=N$(X):S$="-":GOTO1135
1120 F$=MID$(N$(X),1,Y-1):S$=MID$(N$(X),Y+1)
1130 REM MATCH IF SAME FIRST OR LAST NAME
1135 IFL$=F$ORR$=S$ORL$=S$ORR$=F$THEN1040
1137 REM NO MATCH FOUND, CONTINUE SEARCH
1140 GOTO1020
1142 REM * CHECK ADDRESS OF ENTRY FOR MATCH *
1148 REM MATCH IF SAME STREET
1150 IFS$=S$(X)THEN1040
1155 REM MATCH IF SAME CITY
1160 IFC$=C$(X)THEN1040
1165 REM MATCH IF SAME STATE & ZIP CODE
1170 IFT$=T$(X)ANDZ$=Z$(X)THEN1040
1175 REM NO MATCH FOUND, CONTINUE SEARCH
1180 GOTO1020
1182 REM * CHECK MEMO OF ENTRY FOR MATCH *
1188 MATCH IF SAME MEMO
1190 IFM$=M$(X)THEN1040
1195 REM NO MATCH FOUND, CONTINUE SEARCH
1200 GOTO1020
1207 REM * CHECK PHONE # OF ENTRY FOR MATCH *
1209 REM MATCH IF SAME PHONE #
1210 IFP$=P$(X)THEN1040
1215 REM NO MATCH FOUND, CONTINUE SEARCH
1220 GOTO1020
READY.
```

Breakup Listing *(continued)*

```
3070 POKES0+X1+COL*V1,SP
3080 IFX>15ANDPEEK(S0+X+COL*Y)=102THENPOKEV1,255:O=5:GOSUB5000
3090 POKES0+X+COL*V,BL
3100 IFX>1ANDX<19THEN3150
3110 IFX=19THENDX=-DX:POKEV3,230:GOTO3150
3120 IFX=1ANDY>A0RY=A-1ANDY>0ANDY<A+20RY=A+3ANDY<0THENFLAG=1
3130 IFFLAG=1THENFLAG=0:DX=-DX:POKEV3,220:GOSUB5000:GOTO3150
3140 BALL=BALL-1:OK=0:POKES0+X1+COL*V1,SP:RETURN
3150 POKEV1,0:POKEV3,0:RETURN
4000 V1=36874:V3=36876:CS=36879:PA=37137:POKE37139,0:COL=22
4010 S0=4*(PEEK(36366)AND128)+64*PEEK(36869)AND112)
4020 POKECS,30:GOSUB6000:POKE650,128
4030 A=2:B=0:GOSUB1000
4040 BALL=3:O=1
4050 FORX=16TO18STEP2:FORY=2TO19
4060 POKES0+X+COL*V,102:NEXT:NEXT
4070 IFBNUSTHEN4090
4080 SCRE=-1:GOSUB5000
4090 ONPDLGOTO4100,4100,4130
4100 BUT=-(PEEK(PA)AND(PDL*16)>=0)
4110 IFBUT<>1THEN4100
4120 GOTO4140
4130 GETA$:IFA$<>" "THEN4130
4140 X=2:Y=INT(RND(1)*14)+2:DX=1:DY=1
4150 GOSUB3000
4160 IFOKTHEN4150
4170 POKEV3,0
4180 ONPDLGOTO4190,4190,4220
4190 BUT=-(PEEK(PA)AND(PDL*16)>=0)
4200 IFBUT<>1THEN4190
4210 GOTO4230
4220 GETA$:IFA$<>" "THEN4220
4230 POKES0+X+COL*V,SP
4240 IFBALLTHENSCRE=SCRE-1:GOSUB5000:GOTO4140
4250 POKECS,27:PRINT"■■■■■■■GAME OVER!
4260 PRINT"■YOU SCORED■"SCRE"■"
4270 PRINT"POINTS!
4280 PRINT"■PREVIOUS HIGH WAS■"HI"■"
4290 PRINT"■PRESS SPACE TO CONT.":IFSCRE>HITHENHI=SCRE
4300 GETA$:IFA$=""THEN4300
4310 GOSUB6090
4320 GOTO4030
5000 SCRE=SCRE+0:PRINTV$"■BALL=■"BALL"■  SCORE=■"SCRE;
5010 IFNOTBNUSANDSCRE>170THENSCRE=SCRE+50:BALL=BALL+1:BNUS=-1:POKEV1,0
     :GOTO4050
5020 IFO=5THENDX=-DX:GOTO5080
5030 IFY<=RANDDY>0THENDY=-2*DY:GOTO5080
5040 IFY>=A+1ANDY<0THENDY=-2*DY:GOTO5080
5050 DY=DY/2
5060 IFDY>0ANDDY<1THENDY=1
5070 IFDY<0ANDDY>-1THENDY=-1
5080 O=1:RETURN
6000 POKE36878,15:PRINT"■"
6010 PRINT"■■■■■■VIC BREAKUP"
6020 PRINT"■■■■■■■■K■EYBOARD, ■P■ADDLE":PRINT"■■■J■OYSTICK OR ■Q■UIT?"
6030 GETA$:IFA$=""THEN6030
6040 IFA$="P"THENPDL=1:GOTO6090
6050 IFA$="J"THENPDL=2:GOTO6090
6060 IFA$="K"THENPDL=3:GOTO6090
6070 IFA$="Q"THENPRINT"■":END
6080 GOTO6030
6090 POKECS,8
6100 V$="■■■■■■■■■
```