COMMODORE WORLD

Issue 12
Volume 2, Number 7
$4.95 U.S.

THE NEWS MAGAZINE FOR COMMODORE 64 & 128 USERS

Learn to Program in HTML

It's easier than you think!

A Sneak Preview of the SuperCPU

The Mensch Connection

Meet the man behind the technology

Reviews

The CMD FD-2000

Internet and Unix guides

Plus...

Running your own BBS

The 65816 Super Processor

SECOND CLASS

SOFTWARE SUPPORT INTERNATIONAL

We Carry America's Largest Selection of C-64/C-128 Software!

ENTERTAINMENT

10 Game Grab Bag	$9.95	Pacman	$9.97
Arkanoid 2	$7.97	Plundered Hearts	$12.97
Beyond Dark Castle	$7.97	Pool of Radiance	14.97
Beyond Zork 128	$12.97	Portal	$14.97
Defender of the Crown	$9.97	Questron 2	$9.97
Double Dragon 2	$9.97	Rampage	$7.97
Heavy Metal	$12.97	Realms of Darkness	$12.97
Heros of the Lance	$14.97	Roadwars	$7.97
Keys to Maramon	$9.97	Sidearms	$7.97
Mean Streets	$9.97	Steel Thunder	$9.97
Monopoly	$12.97	Strip Poker	$14.97
Ms Pacman	$9.97	Trump Castle Casino	$12.97
Ogre	$9.97	Wheel Fortune 1-2-3	$9.97

SPORTS & FLIGHT

4th & Inches	$7.97	Tony LaRussa Baseball	$7.97
Champshp Baseball	$7.97	Tournament Tennis	$7.97
Champn Basketball	$7.97	WWF Wrestling	$7.97
Dave Winfield Batter Up	$9.97	Acrojet	$9.97
Fast Break	$7.97	Apache Strike	$7.97
Hardball	$7.97	Blue Angels	$7.97
Jack Nicklaus Golf	$9.97	F-14 Tomcat	$9.97
Jordan vs Bird	$7.97	F-19 Stealth Fighter	$12.97
Leaderboard Golf	$9.97	Flight Sim Games	$4.97
Pro Football Facts	$9.97	High Roller	$7.97
Pro Soccer	$7.97	Jet Combat Sim	$7.97
Pure Stat Baseball	$9.97	Skyfox	$7.97
Sporting News Baseball	$9.97	Super Huey I	$7.97
Star Rank Boxing	$7.97	Top Gunner	$9.97

ACCESSORIES

C-128 RGB Cable	$17.95	Floppy Disk Notcher	$6.95
Computer Hand 2	$6.95	Dust Covers - specify	$7.95
Convert A Com	$24.95	Ergostick Joystick	$16.95
Disk Bank 10/3.5"	$2.95	CBM 1200 Baud	$9.95
Disk Bank 10/5.25"	$2.95	Aprotek 2400 Baud	$49.95
Disk Bank 100/3.5"	$12.95	1351 Smart Mouse	$44.95
Disk Bank 100/5.25"	$12.95	MW 350 Interface	$44.95
Disk Bank 70/5.25"	$7.95	64 Power Supplies	$34.95
Disk Mailers	$0.39	128 Power Supplies	$49.95
Drive Cleaners	$6.95	Printer Ribbons	CALL
Serial Cable 6ft	$8.95	Userport Expander	$24.95
Serial Cable 10 ft	$9.95	Video Ram Upgrade	$49.95
Drive Power Cable	$7.95	3.5" 10 cnt.Floppy	$7.95
User Port Cable	$15.95	5.25" 20 cnt.Floppy	$4.95
Com Modem Adapter	$15.95		

PRODUCTIVITY

1750 Super Clone	$99.95	Geos 128 v2	$44.97
Basic Compiler 64	$12.97	Graphic Label Wizard	$14.97
Big Blue Reader	$29.97	Graphics Basic	$9.97
B/W Prog Tools	$14.97	Home Designer 128	$24.97
B/W Power C	$9.97	Manager, The	$12.97
B/W Turbo Cartridge	$17.97	Maverick v5	$24.95
C128 Graphics Bundle	$29.97	Model Diet	$9.97
Christmas Model Kit	$9.97	Newsroom	$14.97
CSM Protection Man 1	$14.97	On Line Help	$9.97
CSM Protection Man 2	$19.97	Outrageous Pages	$19.97
Data Manager 2	$14.97	Postcards	$14.97
Designer's Pencil	$9.97	Printmaster Plus	$19.97
Drive Align 1541/71	$12.97	Superbase 64	$19.97
Easy Working Tri Pak	$9.97	Superscript 64/128	$14.97
Geos 64 v2	$39.97	Swiftcalc w/Sideways	$14.97

EDUCATIONAL

Early Learning Friends	$9.97	Spellicopter	$9.97
European Nations & Loc	$9.97	Stickybear Math	$9.97
Keys to Typing	$9.97	Stickybear Numbers	$9.97
Memory Manor Cart	$9.97	Typing Tutor 4	$9.97
Magic Spells	$9.97	Word Attack	$9.97
Snoopy Sky Scramble	$9.97	Word Spinner	$9.97

REFURBISHED HARDWARE

C-64 Keyboard w/ PS	$79.95	1541 Clone Drive	$39.95
C-128 Keyboard w/PS	$139.95	40 Col Monitor	$99.95
C-128D Computer	$174.95	80 Col Monitor	$139.95
1541 Disk Drive	$64.95	Printers/Call for Make	$39.95
1571 Disk Drive	$129.95	Misc.	CALL

Items Listed Above Do Not Include Shipping. U.S 48 States - Add $5.50 per order. Alaska, Hawaii & Canada - add $5.50 for the first piece and $1.00 per each additional piece per shipment. Second Day Air shipping is available. Call for shipping charges. Call Or Write For Your Free c-64/128 Catalog Listing HUNDREDS Of Commodore Products And Special Offers For Your Computer. Our Order Takers Are On Duty 6:00 a.m. - 5:00 p.m. M - F and 7:00 a.m. - 3:00 p.m. Sat. - Pacific Time.

SOFTWARE SUPPORT

Software Support Int.
2700 N.E. Andresen Rd.
Suite D-4
Vancouver, Wa 98661
(360) 695-1393

CALL TOLL FREE TODAY! **1-800-356-1179**

Major Credit Cards Accepted.

CONTENTS

ISSUE 12 JANUARY 1996

COMMODORE WORLD
THE NEWS MAGAZINE FOR COMMODORE 64 & 128 USERS

GENERAL MANAGER
Charles R. Christianson

EDITOR
Doug Cotton

ASSISTANT EDITOR
Jenifer Esile

ADVERTISING SALES
Charles A. Christianson
(413) 525-0023

PHOTOGRAPHY
Wayne Wrubel

GRAPHIC ARTS
Doug Cotton
Jenifer Esile

ELECTRONIC PRE-PRESS & PRINTING
Mansir/Holden, Inc.

Cover Design by Jenifer Esile

Commodore™ and the respective Commodore product names are trademarks or registered trademarks of Escom GmbH. Commodore World is in no way affiliated with Escom GmbH., owner of the Commodore logo and technology. Commodore World is published 8 times annually by Creative Micro Designs, Inc., 15 Benton Drive, East Longmeadow MA 01028-0646. Second-Class Postage Paid at East Longmeadow MA. Annual subscription rate is US$29.95 for U.S. addresses, US$35.95 for Canada or Mexico, US$45.95 for all EC Countries, and US$57.95 to all other addresses worldwide. All subscription payments must be provided in U.S. Dollars. Mail subscriptions to CW Subscriptions, c/o Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow MA 01028-0646.

POSTMASTER: Send address changes to: CW Address Changes, c/o Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow MA 01028-0646.

FEATURES

REVIEWS

COLUMNS

DEPARTMENTS

See Our CHK-LIST Utility On Page 54

From The Editor

Happy New Year!

Yes, it's time to change all the calendars, reflect briefly on the year gone by, then turn to our minds to ringing in the new year with a bang. And, of course, that means it's time for me to extend our best wishes for a happy new year to all our readers, and to thank you all for making 1995 the success it was for us here at Commodore World and Creative Micro Designs, Inc. And now that the amenities are out of the way, let's cover the 'old business'.

At the close of last year (seems so long ago now, doesn't it?) I found myself on a small Canada Air flight from Hartford to Toronto. The mission: to attend the first World of Amiga show to be held since the demise of Commodore. This year's sponsor was none other than Wonder Computers, a Canadian firm with a very strong commitment to the Amiga and the Amiga market.

What struck me most about this show was not the kind of thing that would have piqued my interests at similar shows years before: things were still pretty much the same. Oh, there were new faces, and a lot of the old faces were missing; but the atmosphere and general aura created were—similar. This bodes well for Amiga Technologies, the new wing of ESCOM that is working to put Amiga computers back into peoples minds and homes. The Amiga community—in Canada, at least—was receptive to the show, and ready to buy whatever dealers had to offer. And this bodes well for vendor support as well; seeing a willing market is bound to convince a few vendors to stay on-board.

Now, not everything at the show had the big "A" on the front panel. I noticed some of our old friends at the TPUG booth were playing around with a C-65, and I also ran into a couple of the CW advertisers at the show. Jason Compton and Katie Nelson were there, too; some of you have seen Jason's previous work here or in his own *Amiga Report*. Jason and Katie are starting a series on HTML in this very issue!

Another CW author joined me at the show to make the rounds: Jim Butterfield. It's amazing how many people Jim can draw to himself. This can be a true advantage if you're with Jim at a show, since the booths are often crowded—getting close to a booth that had something I wanted to see and saying, "Hey, there's Jim Butterfield!" was generally enough to clear some space.

I also had a rather interesting chat with Matthew Desmond, who apparently had hoped to run into some 8-bit folks at the show. Matt was rather pleased to see the results of the CPS tests results we published in CW Issue #4, and could probably be heard for quite some distance as he chanted, "I won! I won!" While Matt probably didn't think of it at the time, whenever he wins, so do we all.

In all, it was an interesting show, and I was pleased to see a good turnout there.

Now, lets move ahead to this year. Naturally, there has been a lot of excitement over the upcoming release of the SuperCPU. In the issue you now hold, you'll find an update on this project (in *On The Horizon*), and a 'sneak preview' of the product itself (pg. 23). You can also find out more about the microprocessor at the heart of this product in *The Super Processor* (pg. 20), and get a glimpse at how it relates to the 6510 in *The Mensch Connection* (pg. 17).

My guess is that, after reading this issue, many of you will be anticipating what's to come in this new year as much as we are here at CW and CMD!

Doug Cotton
Editor

LOADSTAR SOFTWARE VALUES!

The Compleat* Series

The Compleat PRINT SHOP I: (The Print Shop by Broderbund or Printmaster required). Over 1300 artistic and never before published PRINT SHOP images. The smart, fast software package included allows you to quickly scan through the many PRINT SHOP images sequentially, by name, or by group number. Press a key and save the graphic you want in 2-block, 3-block and even PRINTMASTER graphic files! All that plus a printed guide to your new sea of graphic files. And if you like the first volume, you'll probably want to get your hands on **The Compleat PRINT SHOP II**, which also contains over 1300 PRINT SHOP images, previously published on LOADSTAR issues over the past ten years. Included is the same smart, fast software package included in The Compleat PRINT SHOP I. Each volume is $20.00.

The Compleat Programmer: Become the Commodore programmer you've always wanted to be! Megabytes of knowledge crammed and stuffed onto eight 5.25-inch disks or two 1581 disks! Plus we include all the tools, extensions, languages, assemblers, tutorials and utilities you'll need to create the same type of software you see on LOADSTAR! This massive collection is way over two megabytes of instructional text and valuable tools. $20.00. For $5 more, get C= Hacking MAG, not available separately, to complete your programming set.

The Compleat Maurice: A compilation of 26 solitaire card games written by Maurice Jones, the acknowledged master of card game simulations for the C-64/128. There's even a brand new, never before published game called Boomerang. Two 5.25 inch disks or one 3.5 inch disk. $20.00 postage paid!

The Compleat Dave: Two 1581 disks or three 1541 disks crammed with SID music. Over 250 classic melodies from yesteryear, arranged and transcribed by the Master of Music, Dave Marquis, and now they're available in one gigantic 8-hour collection. If you enjoy SID music, you owe it to yourself to get THE COMPLEAT DAVE. Two 1581 disks or three 1541 disks $20.00 postage paid!

The Compleat Walt: Ten years of Walt Harned's slideshows and multimedia events gathered into one huge collection: Seven 5.25-inch disks or three 3.5 inch disks! Over 250 pictures, including some that have never been published. The greatest one-man collection of art on any computer platform! $20.00 postage paid!

The Compleat Roger: 25 educational quiz programs, each carefully crafted by Roger Norton, an educator who uses C-64s. These programs come crammed on two 5.25-inch 1541 disks or one 1581 disk. $20.00 postage paid!

*** No, we didn't misspell "complete." Compleat is the ten dollar spelling of complete.**

Other Products

Geopower Tools - 19 Geos utilities: Calendar Printer, Fast Format, Geo Fetch (grab any portion of a screen as a Photo Scrap), Phoenix (resurrect a trashcanned file), Programmer's Calculator are just a few of the handy tools. Side Two is filled with Clip Art (in Photo Album format) and fonts. $9.95 (C-64/128) Item #080525

Songsmith - LOADSTAR's own music-making program. With this deluxe music editor/player you can easily transcribe music from sheet music or make up your own tunes. Songsmith comes with a slick 30-page manual and a jukebox player with eight tunes. $9.95

Game Star #1 Eight games from LOADSTAR #70-#100). The Tenement, Stack 'Em, The Sherwood Open, Gems, Stealth Bomber, Eagle Eyes, Moonraker & Circuitry. $9.95

Just For Fun - Eight original games. There are arcade games, educational games, puzzle games and just games that are just plain fun on this disk. $9.95

Fun Four - Four original games. A huge maze game, trivia game, solitaire and a space shoot 'em up -- all runnable from a menu. $9.95

Sport -In the early days of science, "sport" meant "mutant." This is a full novel on C-64 disk by author, Jeff Jones, about a murderous neo-human. Over 500 terrifying pages. Automatic presentation software included. Printing capability. Bookmarks. Warning! This is a *real* novel with strong content, frank language, violence and adult situations. If the hard-hitting action of *real* sci-fi/horror novels turns your stomach, please don't buy this book. Must be 18. 1 CMD HD disk , 2 3.5-inch disks or 3 5.25-inch disks. $5.95 $1.00 Shipping.

Brainpower/Brainstorm - 80-column word processor and idea processor for the C-128. These two programs are together on one disk. It comes with a detailed 32-page manual. $9.95 (C-128 only) Item #069421

Best Of Loadstar Compilations: LOADSTAR is over 11 years young! We have 5 anthology disks that take you through the evolution of LOADSTAR, from humble beginnings to the well-oiled machine we've become. If you want to hop on a time machine and go back, start with **The Best Of LOADSTAR #5** and move your way back to **The Best Of LOADSTAR #1.** Each of the 136 back issues are available for purchase with discounts on large orders for you collectors out there. Best Of LOADSTAR disks are available on 5.25-inch disks only. C-64 disk. $9.95.

Master Base - Database of users' groups for the 80-column C-128. This is a fast, powerful database program for handling addresses and mailing labels (includes barcode printing). The disk also has a file of over 600 users' groups addresses. $4.95 (C-128 only)

LOADSTAR is a monthly "magazine on disk" for the Commodore 64/128. Subscribers receive two 1541 disks (or one 1581 disk) in their mailbox every month filled with news, articles and programs. These non-PD, high-quality programs are written by the best home-based programmers in the field and edited by the crack LOADSTAR team of Fender Tucker and Jeff Jones. Subscription prices are at an all-time low of $69.95 for a 12-month subscription, or $19.95 for a three-month subscription. You may also elect to subscribe "by the month," where we charge your credit card $7.95 for each issue after it's shipped.

Loadstar Products Order form 1-800-594-3370

Quantity	Description	Price	Item#	Total
	3-month LOADSTAR subscription	$19.95	na	
	1-year subscription to LOADSTAR	$69.95	na	
	Month to month @$7.95	$7.95/mo	na	
	Compleat Print Shop I on 5.25-inch	$20.00	0009D5	
	Compleat Print Shop I on 3.5-inch	$20.00	0001D3	
	Compleat Print Shop II on 5.25-inch	$20.00	0010D5	
	Compleat Print Shop II on 3.5-inch	$20.00	0002D3	
	Compleat Programmer on 5.25-inch	$20.00	0005D5	
	Compleat Programmer on 3.5-inch	$20.00	0005D3	
	Hacking mag 3.5-inch only	$5.00	0006D3	
	Compleat Maurice 5.25-inch	$20.00	0007D5	
	Compleat Maurice 3.5-inch	$20.00	0007D3	
	Compleat Dave 5.25-inch	$20.00	070525	
	Compleat Dave 3.5-inch	$20.00	070523	
	Compleat Walt 5.25-inch	$20.00	070425	
	Compleat Walt 3.5-inch	$20.00	070423	
	Compleat Roger 5.25-inch	$20.00	0004D5	
	Compleat Roger 3.5-inch	$20.00	0004D3	
	Geopower Tools	$9.95	080525	
	Game Star #1	$9.95	080825	
	Just For Fun	$9.95	073525	
	Fun Four	$9.95	073525	
	Sport 5.25-inch	$5.95	070325	
	Sport 3.5-inch	$5.95	070323	
	Sport CMD High Density	$5.95	070327	
TOTAL FROM THIS COLUMN				

Qty	Description	Price	Item#	Total
TOTAL FROM LEFT COLUMN				
	Songsmith	$9.95	069525	
	Best Of LOADSTAR #5	$9.95	049525	
	Best Of LOADSTAR #4	$9.95	049425	
	Best Of LOADSTAR #3	$9.95	049325	
	Best Of LOADSTAR #2	$9.95	049225	
	Best Of LOADSTAR #1	$9.95	049125	
	Masterbase	$4.95	081025	
SUBTOTAL				
Louisiana Residents ad 4% sales tax				
Shipping $5.50 UPS Second day air. Pay no shipping if LOADSTAR subscription or any product from the Compleat series is ordered.				
Total Enclosed:				

Payment method:

❑ Check/money order made payable to "Softdisk Publishing" in US Funds

❑ MasterCard ❑ Visa ❑ American Express ❑ Discover

Card # ______________________

Expiration Date ______ / ______ / ______

Authorized signature: ______________________

Subscribers: I prefer: ❑ 5.25-inch diskettes ❑ 3.5-inch diskettes

Mail or Fax to:
Softdisk Publishing
P.O. Box 30008, Shreveport, LA 71130-0008
Questions: ○ 1-318-221-8718 ○ Orders 1-800-594-3370

BACKTALK

LETTERS AND QUESTIONS FROM OUR READERS

Dear CW,

This is addressed to T. J. Moyles in regard to his question in the last issue.

The protection scheme Cosmi uses is "cute", not devious. How they accomplish it is not that important. Here is how to make a bootable copy:

1. Take a 5 1/4" blank disk, and notch it so that you can write on both sides of the disk.
2. Format each side as a 1541 disk (i.e. 664 blocks free on each side.)
3. Fire up Maverick 5.0, select single or dual drives (which ever you have) and when you get to the main menu, select GCR NYBBLER.
4. When loaded, put side A of Swiftsheet 128 in the drive and copy to side A of backup disk.
5. When completed, flip to side B of Swiftsheet 128, place in drive, and copy to side B of backup disk.
6. Run a directory on both sides of the backup to insure they are identical to Swiftsheet 128.

VIOLA! You now have a bootable copy! Just autoboot it like you do the original. Enjoy.

About other 80 column spreadsheets, I like SwiftCalc 128, PocketPlanner 128, and if you are into GEOS, GeoCalc 128 is no slouch either. If you like to graph and chart data, GeoChart is a must to go with it. All of these programs are still available through Software International Inc. the last time I checked.

Best regards,
Robert J. Carberry

I'd like to voice our appreciation to Mr. Carberry and a number of other readers who responded to Mr. Moyles' questions. Since Mr. Carberry's response was the most complete of those we received, it was the one we decided to publish. However, we will pass along copies of all of the responses to Mr. Moyles for his perusal.

Dear CW,

I have a CMD 500 MB HD that I wish to use as an external drive on my Compac 286. Scott Eggleston, Editor of *the UNDERGROUND*, was kind enough to take a shot at it, but he had not actually done it, and therefore was not able to spell out a successful step by step instruction which I really need, because I'm not a computer wiz. Surely one of your acknowledged wizards (Dale Sidebottom, Doug Cotton, Jim Brain, Maurice Randall, or Steve Vander Ark, etc.) can come to my rescue? Help, help, help! I have an 8-bit SCSI card and the cable to mate the HD to the card, but I'm nervous about plugging in and firing up without divine guidance. I sure don't want to screw up the HD. I already have the CMD 85 MB HD running on my C128. I have had it for some time now with no trouble operating on an almost daily basis. Great machine!

I hope you can help me with my problem. Looking forward to hearing from you.

Best regards,
Robert J. Carberry

Frankly, hooking it up is simple... getting it to work, however, is the iffy part. SCSI, in fact, is about the only thing I can think of that's more finicky than a cat.

Here's what you should be aware of: Your HD contains a SCSI controller, which is similar to that 8-bit card for your 'other' computer. So similar, in fact, that they probably both have the same SCSI address (practically every SCSI controller on the planet uses SCSI device 7). This can be a source of conflict.

Also, consider that what the CMD HD controller does when you start your system is to begin a search of your drive for HD-DOS. Normally this is found right away, and booting is quick. If the drive gets formatted for another purpose (which I assume is the case here), this search can take a minuete or so on the larger drives.

Now, there's a fairly simple way around these two problems, but I think most users will find it cumbersome. The trick is to turn on the drive while holding down both the SWAP buttons on the front panel. This causes the drive to come up in INSTALL mode, where it will generally ignore other controllers and skip the search for a DOS.

The best solution, however, is to get yourself a new cable for inside the drive. Currently you have two cables inside your drive—one goes from the drive mechanism to the controller, and the other goes from the controller to the DB-25 connector on the rear panel. If you can get yourself a new cable that goes directly from the drive (50-pin SCSI header connector) to the rear panel (DB-25 female)—replacing the two cables and rear panel connector—you'll have an external drive that works like any other, without the quirks and hassles. Such cables aren't common fare, though, so you may have to try a custom cable house such as Cables To Go in Dayton, Ohio (800-225-8646).

Dear CW,

I would like to comment on Maurice Randall's disk drive article in Issue 11. I own two 1581 drives: one manufactured in Taiwan and the other in Hong Kong. The Taiwanese model freezes up with HD disks while the other model doesn't. I have saved Basic, CP/M and GEOS files to HD disks and they all work like a charm.

I don't know if this matters, but the P/N numbers on the bottom of each slightly differ. The 'non-HD' drive reads 255068-03. The 'HD' drive reads 255068-04, and German wording appears on this drive as well. Also, the gray face plates differ. Commodore is spelled in lower case letters on the 'non-HD' drive and the Commodore symbol appears on the 'HD' version with the first letter in Commodore capitalized.

Is it a 'bootleg' drive? Could I have stumbled onto Commodore's missing link? My opinion is that computers and peripheral devices will not perform certain functions unless they are programmed to do so in the first place.

Thanks for a great magazine and for keeping the faith!

Sincerely,
Kevin A. Komonyi

It wasn't uncommon for Commodore to manufacture products in more than one place, and I don't suspect you have a 'clone' 1581. The reason that HD disks work on one and not the other is more due to differences in the two mechanisms—HD disks require a stronger magnetic signal, and many DD (Double Density) drives simply won't have the ability to write to HD disks.

Commodore World reserves the right to edit letters published in this column for length, spelling, and clarity. If you have questions or comments, write to: *CW Backtalk, c/o Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow MA 01028.*

COMMODORE TRIVIA

by Jim Brain

Welcome to another edition of Commodore Trivia. As many of you may know, these trivia questions and answers have been donated by me to the Commodore community at large. Unlike other articles in *Commodore World*, these trivia questions have been placed in the public domain. I ask only that the trivia questions remain intact and unchanged, and that my name and address appear somewhere so users can contact me. The trivia is also used for a contest I run on the Internet; contact me at the included address for more information. Because curiosity has the best of me, I always welcome a note or postcard detailing where the trivia goes. I also welcome new questions—provided they come with the answers. Enjoy.

Jim Brain
Brain Innovations, Inc.
602 North Lemen
Fenton, MI 48430
brain@mail.msen.com

COMMODORE TRIVIA #11 QUESTIONS

$0A0 In the mid 1980's, Commodore introduced RAM Expansion Units for the Commodore 64, 64C, 128, and 128D. There were three of them. Give their model numbers, and what was different among them.

$0A1 Some of the CIA integrated circuits used on the C64 and C128 computers have a hardware defect. What is the result of this defect, and when does it occur? (May be more than one, but I need only one.)

$0A2 Name the Commodore machine(s) on which a Intel 8088 was an OPTIONAL coprocessor. (Hint, not the IBM clones.)

$0A3 On Commodore computers beside the Plus/4 series, there are three frequencies used to record the data on the tape. Name the frequencies used.

$0A4 Commodore Plus/4 series computers can not read any cassettes recorded on other Commodore computers. Why? (Hint: It has nothing to do with the nonstandard connector on the Plus/4.)

$0A5 During power-up, the Commodore 64 checks to see if it running in PAL-land or NTSC-land. How does it determine its location?

$0A6 What is the 65XX ML opcode for BRK?

$0A7 On the 65XX CPU, what gets pushed onto the stack when an interrupt occurs?

$0A8 Speaking of the stack, where is the stack located in the 65XX address map?

$0A9 On the 65XX CPU line, it is possible to set and clear a number of processor status flags. Examples include SEC and CLC to set and clear the carry flag. What flag has a clear opcode, but no set opcode?

$0AA When saving a text file to tape, the computer records 192 bytes of data, an inter-record gap, and then the same 192 bytes of data again. How wide is this inter-record gap, and why is it there?

$0AB On an unexpanded VIC-20, where does the screen memory start?

$0AC In Commodore BASIC, what is the abbreviated form of the "Load" command?

$0AD In Commodore BASIC, what is the abbreviated form of the "List" command?

$0AE On the Commodore 64, there is section of 4 kilobytes of RAM that cannot be used for BASIC programs. It is the favorite hiding places for many ML programs, however. What is its address in memory?

$0AF What is stored at locations $A004-$A00B, and why is it strange?

COMMODORE TRIVIA #10 ANSWERS

$090 The 65XX series of processors was modeled after the Motorola 6800. Motorola hampered the design groups' efforts to pursue product developments using the 6800. A core group of 8 designers left Motorola and went to MOS Technologies, which was the largest producer of calculator chips at the time. MOS decided it was time to go into the CPU business.

$091 The older brother to the 6502 was the 6501. The 6501 was pin-compatible with the 6800, which prompted a suit by Motorola. Eventually, MOS reached an agreement where they scrapped the 6501 marketing, but were free to market the 6502.

$092 151 opcodes are documented in the NMOS 6502 data book. The remaining 105 opcodes were not implemented, and exist as "don't care" states in the opcode matrix. That means that some seemingly invalid opcodes will actually perform pieces of two or more valid opcodes. Newer CPU systems trap all non-implemented opcode usages, but not the 6502.

$093 The architecture assumes that each opcode has two bytes in it and one byte can be fetched per cycle. For instructions that use only 1 byte, the extra fetched byte (actually the next opcode) is thrown away.

$094 Although this is a subjective answer, one could nominate NOP on the basis that NOP is generally believed to waste one execution cycle on a particular processor, namely one cycle on the 65XX line. However, one can argue that NOP simply means no operation, and has no ties to length of execution. You be the judge.

All other instructions must take at least two cycles: one for opcode fetch, one for operation.

$095 As stated above, it was calculator chips.

$096 Rockwell, MOS Technologies, and Synertek.

$097 1.022727143 MHz. It is derived by taking the main clock frequency (14.31818MHz) and diving it by 14.

$098 985.248449 kHz. It is derived by taking the main clock frequency (17.734472MHz) and dividing it by 18. Thus the PAL 64 actually runs slower than the NTSC one.

$099 Data is latched in to the 65XX on the falling edge of Phi0 (Phi1). The timing diagram in some books (64 PRG is one) is incorrect.

$09A The 75XX line used in the 264 series (Plus/4 and C16), and the 85XX series used in the C64C and C128 series.

$09B The answer is location $ff+10 mod 256 = $09. The answer involves explaining a (mis)features of the NMOS 65XX CPU line. The above code instructs the

65XX CPU to use zero-page addressing mode to load the accumulator. In zero-page addressing, the address need only be one byte wide ($ff in this case), because the high byte is considered to be $00. Now, as humans, we would expect the CPU would add 10 to 255 ($ff), giving 265 ($109) as the address to load the accumulator from. However, the CPU designers decided that zero-page addressing means that the high byte will be $00 all the time, no exceptions. If a situation like the above occurs, the low byte of the addition will be used as the low byte of the address (9 in this case), but the high-byte will be ZERO. All zero page addressing modes work this way. Note that the CMOS versions of the 6502 do perform the high byte "fix-up", so this behavior is only seen on the NMOS parts.

$09C This was a trick. The code is trying to use INDIRECT INDEXED indexing mode using the x register, but that addressing mode can only be used with the y register. If the code is changed to the following, legal code:

```
ldx #10
lda ($ff),y
```

Then, the above discussion for zero-page addressing holds true here as well. The effective address would have been (hi:lo) $100:$0ff, but is instead (hi:lo) $000:$0ff. The simple rule is: zero page means exactly that. There is no way to address outside of zero-page with zero-page addressing.

$09E $fffc. The address formed by reading $fffd and $fffc is stuffed into the IP, and the code is read starting there. $fffc is read first, since the 65XX line stores addresses in low byte, high byte format.

$09F Bill Mensch. He hand-designed these newer parts in the 65XX line in the same manner he and Chuck Peddle and others hand-designed the 6501 and 6502.

User Group Connection

Looking for a Commodore user group in your area? User groups can help you solve problems, keep you informed of new products and events that might be of interest, and give you a chance to share your computing experiences with others who enjoy Commodore computing. The listings below include: Name, Address, Meeting dates, Number of members, Contact person, and Phone.

ARIZONA

Busy-Bee User Group 3
Paseo Cuacha de Toro
Arivaca, AZ 85601

CALIFORNIA

Club 64
PO Box 514
Patton, CA 92369

Commodore Technical User's Group
2231 E. Trenton Ave.
Orange, CA 92667-4451

FLORIDA

Naples Area C.U.G.
10701 Rogers Lane
Bonita Springs, FL 33923

Stuart Area Commodore
P.O. Box 1446
Port Salerno, FL 33492

ILLINOIS

East Side Computer Club
P.O. Box 1347
Alton, IL 62002

S.W.R.A.P. User's Group
PO Box 528232
Chicago, IL 60652-8232

NEVADA

Clark County Commodore Computer Club
P.O. Box 28136
Las Vegas, NV 89102

OHIO

Ashtabula County Commodore User's Group
c/o 1527 Elmwood Dr.
Ashtabula, OH 44004

OREGON

Amiga Commodore Users Group
623 29th St.
Astoria, OR 97103

WASHINGTON

Spokane Commodore User's Group
PO Box 474
Mead, WA 99021

CANADA

Hamilton Commodore Users Group
35-1400 Limeridge Rd.,
East Hamilton, Ontario Canada L8W-1L7

Attention
User Groups!

Please send us your group's name, contact person and phone number, when you meet, number of users, and an address where users can contact you to:

CW UG Connection
c/o Creative Micro Designs
P.O. Box 646
East Longmeadow, MA 01028-0646

ON THE HORIZON

Commodore and Computer Industry News

EZ Loader Menu Program Updated to Version 3.10

EZ Loader, by David Schmoll, is a menu system designed to help you access your most used programs by a single menu from any disk drive, as long as the program is on the drive requested. Although the program concept was written to help CMD Drive users, EZ Loader works just fine on Commodore disk drives and, once set up, can be used as a menu for each disk. EZ Loader operates in all modes: 64 and 128, 40 or 80 column, as well as slow or fast mode. EZ loader can call a 64 program from the 128 side menu and drop into 64 mode to execute it. EZ Loader supports the TOD clock, and allows you to preset program execution times, so programs can run automatically from the menu.

This new version is mainly bug fixes and tweaks, but the known bugs were serious enough to justify the release. The only real new features are:

- the full 128 keyboard is now active in 64 mode
- 128 users can drop to 64 mode without leaving EZ Loader by pressing the ALT key

Registration notes: Due to requests users that some indication and advantage be given to those users who register this program, certain features in the evaluation copy have been disabled. Disabled features include the mouse, clock and date display (assuming an RTC (Real-Time Clock) is present), loading of saved data files, multiple screen access, screen blanking, entry exchange option, and timed event support, and auto booting of the first file. These options can be enabled once the registration has been completed and you receive and enter the proper codes that I will need to send you.

EZ Loader is available online from a number of Commodore Support boards. It can be freq'ed or downloaded from: The Silver Hammer BBS (303) 766-8035 (Fido 1:104/518). It is also available for FTP from: ccnga.uwaterloo.ca. Archive name is ezldr3-1.sfx.

Internet Book for Commodore Users Coming Soon

The Internet is a big place and it's not always easy to find out all the in's and out's about taking advantage of its features when checking it out for the first time. There's a lot of Commodore specific information available on the Internet, but you have to know how to get to the information, and that can be a catch-22.

How do I get on the Internet? What term programs can I use? What the heck is FTP? Can I browse the World Wide Web with my C-64?

Gaelyne Moranec, the former editor of CEE-64 Alive!, has been writing about Commodore computers and modeming for three years with columns in Commodore World and BBS Magazine. After answering countless letters helping others learn the ropes of using the Internet, Gaelyne has announced that she's working on a book for Commodore users who want to take advantage of the Internet. *The Internet For Commodore Users* has a tentative release date of March 31, 1996 and will be packaged with Nick Rossi's Novaterm 9.5 (shareware) term program.

The first half of the book discusses Commodore software and hardware needed, setting up a term program, hooking a modem up to the computer, and finding an Internet provider. The second half of the book covers the basics of using the features of the Internet, such as E-mail, FTP'ing, browsing the World Wide Web, dealing with Unix commands and other subjects, all from the point of view of a diehard Commodore user.

The Internet For Commodore Users is expected to retail for US$24.95. Distributors have not yet been announced.

Fax Software Update Announced

Maurice Randall has recently announced that geoFAX V2.0 is due to be released in about 6 weeks. Versions 1.5 and 1.6 are not considered to be completed versions, and V2.0 will be sent free-of-charge to all registered users of these versions when it becomes available. For those that are considering buying geoFAX prior to the release of V2.0, you will also get V2.0 for free when it begins shipping.

The V2.0 disk will include the remaining printer drivers (several of those announced were not ready for the current release), and a few of the existing ones will have improvements. Both Class 2 and Class 2.0 fax modems will be supported, and modem control during the handshaking period will also be improved over previous versions. The new version also sports an improved interface, and remembers changes such as the baud rate (such changes become your default setting the next time you run the program).

Other additions include an icon for choosing between tone and pulse dialing, a telephone directory, and a log file that describes your fax activity (this can be viewed in geoWrite).

The most drastic changes are what you see on the screen. The simulated fax machine display has been improved, and the dialogue boxes that pop-up have taken on a more graphical look.

Final Press Release from dieHard

On September 22, 1995 LynnCarthy Industries, Incorporated, publishers of *dieHard, the Flyer for commodore 8bitters* and *The Sewing Room* closed its doors.

The remainder of the *Flyer* subscriptions were to be fulfilled by *Electronics Now*, however it took four months to get everything worked out and by then *Electronics Now* felt the list was too old. The remainder of the *Spinner* subscriptions will be filled by *LOADSTAR*. The fulfillment needs for the *Flyer* are so great, that only an ad driven magazine could take it over. Commodore magazines are subscription driven, meaning the subscribers pay for the publication rather than the advertisers. And because of this, neither magazine can offer refunds. LCII does not exist any more, and therefore cannot offer refunds. *LOADSTAR* is a top notch magazine and is a blast to read. I hope you stay with it for many years to come.

LCII had many problems over the last year, the postal increase, and the cost of paper skyrocketing, to name a few. Time was taken off for additional complications during pregnancy, as well as paternity and maternity leave. But over all, LCII just grew too fast. *The Sewing Room* got so many subscribers so fast, LCII needed to do more marketing to bring in more capitol to cover the increase in costs. Which, in turn, brought in more subscriptions to fill. Neither magazine could stay on schedule. LCII staff found they were doing everything but publishing. And everyone was stressed out. It was not a healthy atmosphere. Certainly nothing like the company that it once was. So we closed the doors.

This is in no way a reflection on the commodore market. User Groups all over the nation are booming as ever. CMD and Parsec now both have store fronts. *LOADSTAR* has become a separate entity from Softdisk. (Congratulations all around!)

I want to thank everyone for staying with us. We really thought *dieHard* would be on a regular schedule and that it would continue. I really have enjoyed doing *dieHard* and will miss it dearly. It makes me sad to say good-bye.

Fare thee well--long live the lowercase "c" machines!--Brian L Crosthwaite

READY.

Address Change for Kasara

We've recently been informed of an address change for Kasara Microsystems, who still supplies a large number of chips and other parts for Commodore users. The new address is: Kasara Microsystems, 705 Office Park Drive, Hilton Head, SC 29928. Phone: (803) 842-5056, Fax: (803) 842-5057.

SuperCPU Update from CMD

During the past few weeks, there have been many changes with respect to our SuperCPU project. In our quest to keep users as up-to-date as possible on the status of this project, we've put together this informative update.

SuperCPU 64/10 Cancelled

Advanced orders for the SuperCPU have been instrumental in helping us track the demand for this product. Due to the extremely low percentage of orders we have received for the 10 MHz version of this product, however, we cannot justify the extra costs involved in stocking parts for this version. We have therefore decided to eliminate the 64/10, and from this point forward we will only accept orders for the 64/20 model.

SuperCPU 128

The most significant announcement is that we have determined that a C-128 version of the accelerator will be released. This decision has been based on the overwhelming number of requests we have received from C-128 users. Due to this response, we are confident that sufficient demand for a 128 version exists, and that we can justify the expense associated with its development.

The specifications of the C-128 version will be virtually identical to its C-64 counterpart, and will support both C-64 and C-128 modes. Naturally, the SuperCPU 128 will have additional RAM, ROM and custom logic to support these capabilities.

Co-Development

In order to accommodate the development of the C-128 version, our engineers have had to alter their initial development schedule to allow additional time to co-develop portions of the C-128 version. Due to this we estimate that an additional six to eight weeks will be required, which will push initial deliveries for the C-64 version into April (the C-128 version should be available approximately 90 days later).

The main thrust of this co-development is to attempt to utilize the same PC board for both the C-64 and C-128 versions. If we are successful with this approach, we can reduce the overall development cost of the C-128 version by amortizing many of the fixed costs associated with the design and production over both versions.

While we may be able to use the same circuit board for both units, the chip sets for the two versions will be substantially different, and this may prohibit our ability to upgrade a 64 version to a 128 version. In addition, the extra circuitry required to produce the C-128 version may effect the retail price of this unit by as much as $100.00. Our goal is to keep the cost of the C-128 version below $300.00, but a firm estimate will not be available until later in the development cycle. Until then, we will not be taking advanced orders on the C-128 version.

A Better Product Overall

With the additional development time we believe that both the C-64 and C-128 versions will benefit. This time allows us to explore the possibility of additional features and capabilities which had previously been scrapped due to the tight development schedule. Additional information will be released as we are able to confirm the feasibility of these additional features.

Although we're disappointed that our initial production schedule for the C-64 version cannot be met, we're excited about the potential this decision brings to the Commodore market. By supporting the C-128 platform, we believe that we will see more third party development as a whole, which will benefit both C-64 and C-128 users by further strengthening our market.

In closing, we would like to request that any questions be channelled to CMD by mail or E-mail at the addresses listed below. We will not be fielding questions about these products by phone until such time as the designs have been finalized.

Mail Inquiries:

SuperCPU Inquiries
c/o Creative Micro Designs, Inc.
P.O. Box 646
East Longmeadow, MA 01028-0646

E-Mail (Internet):

cmd.support@the-spa.com (Technical Info)
cmd.sales@the-spa.com (General Info)

ERROR CORRECTIONS

Okay, we blew the correction to Al Anger's E-mail address in Issue 11's Error Corrections. The address should have read:

d014066c@cdfreenet.seflin.lib.fl.us

Worse yet (if that's possible), some idiot here (not mentioning any names, Doug) apparently forgot to create a legend for the Disk Drive Speed Comparison chart presented on page 16 of Issue 11. We've included a similar (corrected) chart in the FD-2000 review in this issue.

Finally, thanks to Jim Brain for quickly reacting to an error we made concerning his mail-server info. Jim quickly noticed the problem and added a function to the server to resolve the issue.

Just For Starters

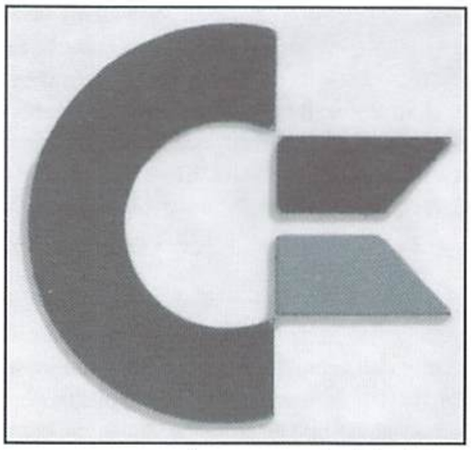
by Steve Vander Ark

TIPS ON DEVICE NUMBERS & PRINTING COMMODORE GRAPHICS

So far in this column, I've talked about a number of pieces of equipment which you add onto your Commodore computer. You might remember that way back in one of the early columns, I used the term peripheral to refer to any piece of hardware you add onto your computer. The subject of the first half of this column, then, is peripherals. Specifically, I'm going to talk about adding a second disk drive.

I've talked about disk drives at some length before. I listed a few handy commands for things like formatting a disk or getting a directory of what's on a disk. At that time, I gave you certain parameters for these commands. Those are symbols or words which follow the command itself to modify or specify what that command means. The parameters of a disk drive command usually include a number to tell the command which drive it is intended for. There is no particular reason, after all, why you have to have only one drive. In fact, most Commodore users eventually add at least one drive of some kind onto their systems. In a disk drive command, you have to give a device number in order to direct your command to the proper drive.

The first drive you hook up is given the number 8. The reason it isn't number 1 is that other devices, such as a printer or even the screen of the monitor or television, are also assigned numbers automatically, so numbers lower than 8 are already taken. Device numbers from 8 on up are set aside for disk drives. If you add on a second disk drive, you will want that device to be given the number 9. Trouble is, when you first get that drive, you will most likely find that it is already pre-set to number 8. On top of that, you will quickly discover that there isn't even a port available on the back of your computer for a second disk drive! Obviously, you will need to know a few tricks to getting things set up correctly.



First of all, let's get the drives hooked up to the computer. As I mentioned, there won't be a second port open on the back of your Commodore, so you can't just plug it in there. You will find, however, that there are two ports on the back of each disk drive itself. You can use those double ports to "daisy chain" more than one unit together. You do that by plugging the second drive into the back of the first one. Your Commodore computer knows enough to send messages along the cables through the linked drives until it gets to the one you specify. Of course, then you need to have the right numbers set so the message you send knows where to stop.

Now, let's get the numbers set the way we want them. It isn't hard at all to change the number on a drive. If you have a 1571 drive, it's particularly easy. Look on the back of the 1571 and you will see that there are two tiny switches set in a rectangular hole near the power cord. These are used to assign a number to that particular drive. There are four possible settings using these switches. (You can flip them either up or down using a pen.) Here's how you set them for various numbers:

left switch	right switch	device number
UP	UP	8
DOWN	UP	9
UP	DOWN	10
DOWN	DOWN	11

Unfortunately, the 1541 drive doesn't have those handy switches, so you'll have to change the number by sending a command to the drive itself each time you start up your system. Of course, this means that you can't turn on a second drive until the first one has been set to a number different than 8. Here's the procedure and the command you need to send:

1) turn on the drive you want to be number 9
2) type in the following program lines, pressing return after each one, of course:

```
10 open 15,8,15
20 print# 15, "m-w" chr$(119
) chr$(0) chr$(2) chr$(9+32)
chr$(9+64)
30 close 15
```

Once you have this program entered, type:

```
run
```

This will change the device number of the drive you just turned on, making it device 9.

3) turn on the other drive, which is set to 8

If you are always going to have to go through this procedure as you start up, you might want to consider changing the device number of one of your 1541 drives permanently to number 9. You can do this by means of a simple procedure using only a knife, a Phillips-head screwdriver to open the case, and a small bowl.

I know what you're thinking:Whoa! Open the case?! Yes! It's really not hard at all. And there's nothing better for giving other people the impression that you are a computer expert than opening the case of some piece of hardware. (This is definitely an operation to be performed at the kitchen table, for example, so the rest of your family can gape at you in amazement with a new-found respect.) Make sure that when you get the thing opened up, you look over the highly complicated stuff inside while saying "Now, which component shall I repair this time?" Of course, since no one knows what a lot of that stuff in there really does (at least, no one I know), it's best if you don't actually touch anything.

Okay, here's the procedure. Unplug the drive from both the power and the computer, turn the drive upside down, and unscrew those four screws that hold it together. Once the screws are removed, turn the drive back over and remove the top. There's a metal plate kind-of-thing which covers the really interesting bits, so you'll have to remove that as well. Stpe one—you're in.

This is the point where you will very likely lose a screw or two, so as you work, drop the screws into the bowl. Also, make sure you are paying very close attention to how everything comes apart. Observation is everything in a job like this; in a few minutes you'll be needing those screws and also needing to know where to put them back in.

As you look over the "innards" of your drive, make sure you're facing the front panel and then look along the left edge about in the middle. You're looking for two "jumpers," which are little metal connections on the circuit board. These jumpers will be labeled either "1" and "2" or "A" and "B," depending on which model of 1541 drive you have. You're going to use your knife to cut those jumpers in one of the following patterns:

DEVICE #	JUMPER A/1	JUMPER B/2
8	don't cut	don't cut
9	cut	don't cut
10	don't cut	cut
11	cut	cut

Once you've done this, you can put the whole thing back together again. Make some off-hand remark like "There, that ought to do it!" (for the benefit of those you are trying to impress), and then mark that 1541 drive with the new number so you don't forget.

I know that may seem a bit daunting, but it's worth it. Two disk drives are so much better than one, especially when you use programs such as GEOS which actively support two drives. Even without GEOS, though, that extra drive will definitely come in handy. Copying disks is much simpler with two drives. Creating and maintaining a library of disks (and we all do this; as the disks accumulate and it becomes harder and harder to remember what you put where) becomes a lot simpler with two drives, since you can copy files from various disks onto archive disks by categories. Many programs can be loaded from either disk drive, too, and some let you write data to a second drive. All in all, a second disk drive can be a very handy addition to your Commodore computer.

Back to BASIC

We've been looping a lot lately. We've been talking about IF...THEN commands and GOTOs. Those are very powerful concepts; I dare say that not a program exists that doesn't use some form of looping. But this time around, I thought we might wander in another direction altogether. Let's create a screen display. We can use such a display for the beginning of a program, for example, or to dress up a simple menu. Commodore's built-in BASIC language makes this kind of work very easy and a lot of fun, too. Essentially, we'll just use the PRINT command to place various lines, spaces, characters, and shapes on the screen. With a little patience, we can design quite a nice display and even draw pictures.

The various shapes we'll be using are the ones you see printed on the keyboard. We'll place them in our PRINT statements and when we run the program, they'll be zapped up on the screen, one line at a time. Before we start, though, let's clear the whole screen so we can work. Press and hold the shift key while you press the button on top row marked CLR/HOME. Your screen will clear and the little blinking cursor will move to the upper left hand corner. You can make that screen clear part of a program by simply placing that SHIFT-CLR/HOME combination in a PRINT statement. Try it. Type this line (substitute the SHIFT-CLR/HOME key combo where you see {CLEAR/HOME}) and RUN it:

```
10 print "{CLEAR/HOME}"
```

You'll see a heart shape on the screen where you typed that key combination. That symbol is how your program shows SHIFT-CLR/HOME in a command. When your program runs, that PRINT statement will not place anything on the screen; in fact, it will remove everything from the screen. You can PRINT other handy keypresses in this way, including the cursor keys and keys to change colors of the text that will be printed.

Okay, so now you have the first line of your fancy display all set. Your screen is clear and you are ready to create a masterpiece. There is one small hitch, however. PRINT commands place things on the screen in thin horizontal lines. It's hard to see just what your whole picture will look like without a lot of trial and error. If you're only adding a few frills to a menu, that won't be too hard to do. To place a box around some text, for example, you just do this:

```
10 print "{CLEAR/HOME}" : re
m clear screen
20 print "{CRSR DN}{CRSR DN}
" : rem move cursor down 2 l
ines
30 print "{SPACE}{SHFT o}{3
CMDR y}{SHFT p}"
```

Notice how the shift key gives you the character on the right while the Commodore key gives you the character on the left.

```
40 print "{SPACE}{CMDR h}HI!
{CMDR n}"
50 print "{SPACE}{SHFT l}{3
CMDR p}{SHFT @}"
```

Now RUN this program. See how nifty this looks? And see how easy it is to make cool displays?

I have created entire screens full of this kind of thing before. I've used those graphic characters to write large title screens for games that I've written and dressed up many other programs with a little extra pizzazz. Whenever I did a whole screen, though, I always planned it first on graph paper. That way I could copy the right things into my PRINT statements to make it all come out the way I wanted. There are forty characters across the screen and 24 lines of characters in all. Just be careful: you can't place a character in the very last spot on the screen, the lower right hand corner, or the whole thing will automatically shift up on the screen one line and your top row will disappear. Try it once and you'll see what I mean.

Commodore's built-in BASIC is a very capable programming language. Neat features like these character graphics make it possible for even the novice programmer to create a very nice-looking screen display.

Foreign Exchange

By Gaelyne R. Moranec

COMPUTING DOWN UNDER

I'm not what you would consider the "jetset" type. I'm a hermitess/homebody who spends most of my time using my computer. For the last six months, I've been tapping at the keys in Reynella, South Australia, about 11,000 miles from my US origins. Reynella is a suburb of Adelaide, the capital city of South Australia. The seasons are the opposite in the southern hemisphere, so when you're huddled up to your power supply for warmth this winter, I'll be coping with the summer heat here.

More than a year ago I decided to visit Rod Gasson, the author of QWKRR128. We met through the FIDONet Commodore echo, and have spent many hours typing mail to each other over the years. We thought it would be nice (and less time consuming) to be able to talk with each other in person. There were many times I thought I'd never make it; that the ticket prices would be too high, that I'd chicken out (I HATE flying!), or Rod would get "cold feet" and rescind his invitation. None of my worst fears were founded, though.

It takes some planning to move ones Commodore equipment from one hemisphere to the other! I had one suitcase filled with some of my Commodore equipment,while Rod supplied the major items: the C128, 1901 color monitor, and the power supplies I needed. My own C128 and monitor were left behind because, in North America, the computer and monitor are NTSC, while here in "Oz" they are PAL. The power to run the equipment is also different. In the U.S. it's 110v, while here in Australia they use 240v.

I didn't leave any of my other equipment behind. I brought an FD-4000 disk drive, my 5 meg RAMLink along with its battery backup (I didn't even try to see if the battery would hold for the trip!), my RAM Expansion Unit, 14.4K modem, SmartTrack trackball and my CMD hard drive, loaded with every file I thought I'd want. I also packed the user manuals for each device and many of my most used reference books.

While packing my suitcase, I kept having a vision of some movie-like Gorilla in the back of the airport mauling the luggage. I used reference books to create a padding layer in the suitcase, then put each device into anti-static plastic bags. I used sweaters and other clothes to wrap each item. The idea was to pack the suitcases so I could fit as much as was safe, and at the same time protect the electronic items from the luggage mauling gorilla.

I left Saginaw, Michigan around 1:30 on a Wednesday morning, and arrived in Sydney, Australia at 6:00 Friday morning. Because I crossed the International dateline, Thursday never existed for me. I was prepared to have my luggage ripped apart in customs, so I had receipts for each Commodore item to prove they were for my own use. I found that they weren't very interested in my nifty peripherals. But I didn't get away that easy; I still had to deal with the gorilla. With 400 other travelers, I waited and watched as the luggage came up a conveyer belt and was *shot* 3 feet, where it hit a wall and landed on the revolving conveyer. I watched other people's suitcases crash, and upon impact, I sometimes got to see what they had brought with them! As I watched, I kept thinking of my drives and RAMLink, wondering if I'd packed them well enough to handle the impact. After what felt like an eternity, my suitcase finally hit the wall (yes, I cringed). Did they make it? Remarkably, they did. I didn't test things immediately (I had other things to do), but within a couple days I was reasonably sure most items had made it unscathed.

I was met at Sydney's Kingsford Smith Airport by Mark Dowsett, who authored the documentation for QWKRR128, and Darren "DaZZa" Gibbs, who is a Commodore user and the person responsible for bringing FIDONet CBM conference into Australia. Mark is an Air Traffic Controller and uses his computers and modem to relax in his free time.

One of my first objectives upon arriving was to get access to a computer and modem so I could send Rod a message to let him know I made it through the flight. My first lesson in using the phone system here in Oz was that the dial tone sounds like a distant phone ringing, waiting for someone to answer, only the "ring" doesn't stop.

Also, BBSing here isn't cheap, as each local call costs 25 cents. While it may not seem like much, it does add up.

Once I'd caught up with my mail, Mark gave me a tour of Sydney. In three days of walking around the city (including a ferry ride to the Taronga Park Zoo, a walk around the Rocks area, watching a play, a tour of the Royal Botanical Gardens, and of course a visit to the famous oddly shaped Opera House) I was exhausted. The highlight of my visit in Sydney was visiting the new air traffic control tower with Mark, where I met another Commodore user. It was quite a moment to find myself chatting about my Commodore equipment while getting a spectacular view of Sydney, with the very latest in navigational technology all around us.

After my weekend in Sydney, I traveled on to Adelaide to meet Rod. He repairs VCRs and Camcorders from his home, and when not repairing video equipment he can be found in front of his computer. If he's not reading mail with QWKRR128 or Browser, he's busy adding programming touches to these programs. His workshop is, politely put, "cluttered"—half of it devoted to his business, the other half filled with computer related "stuff". His system includes his C128, a 210 MB hard drive, 9 MB RAMLink, 1581, 1571, SwiftLink, and a laser printer.

We visited inventive Commodore user Paul Gardner-Stephens, the author of 64Net, to pick up the 128 I would be using. Paul's 64Net connects up to 4 Commodores to a PC in a networking system, and is distributed in the U.S. by Performance Peripherals. Paul is a nineteen year old whiz kid studying Computer Science at Flinders University. His computers were in a shed behind his house, with four different computer systems set up in pairs facing each other with cables running between them. On the floor, a computer mouse was trying to escape it's connection and managed to evade being stepped on. After watching Paul's hyperactive fingers type, showing us things on each computer, I wouldn't doubt that he uses his foot to control that mouse on the floor.

It took me about a week to get the power supplies, computer and accessories all together and set up. The size of the power strip/surge protectors used gave me a shock due to the large and unusual shape of the plugs, but it was good to sit down to "my" trusty Commodore and return to happily being a hermitess and tapping away at my keyboard.

Have I noticed much difference in using my Commodore "down under"? Not as much as I thought I would. I understand that PAL systems have an extra screen line in 80 column mode, but I haven't really "noticed" it. My first impression was that there was more border on all the screens, in both 64 and 128 mode. In some of the programs I use regularly I've noticed that the cursor flashes more rapidly, and Rod tells me that most programs written in the U.S. which track time don't track it correctly in Australia—but it's not something I've found to be much of a problem.

I've visited two user groups so far and have found them to be very similar to those I attended when I lived and computed in the States; we come from all walks of life and ranges of computer experience, from the beginners to pros. It's interesting to note that the Guru's aren't necessarily those who run the meetings. I also found it fascinating (and comforting) that Commodore users, no matter where they are in the world, can start talking about our computers and lose all track of time! Sound familiar?

HTML and the Web: An Introduction

By Katherine Nelson and Jason Compton of Amiga Report Magazine

In today's society, a new type of social status is arising. The basis of the status? Connectivity. Millions of people are connected through traditional phones, cellular phones, pagers, fax machines, and of course the newest craze: the Internet. Various forms of communication through the Internet are available, and more are being created. One of the most popular is the World Wide Web, also known as the WWW and simply "the Web."

The WWW is a way to access information on more topics than you can imagine in an attractive and useful manner. This information is accessed by a program on your machine if you have a direct internet connection, or by a program on your "host" machine (your internet access provider). This program, or "browser", relays the information to you in text form if you use a text-based browser such as Lynx. Alternately, Mosaic, Netscape, and other graphical browsers allow for viewing in-line graphics, listening to sounds, playing movies, and other such "multimedia" experiences.

The advantages of using such graphical browsers are clear in that they allow for more detailed presentations. However, as Commodore 8-bits do not currently support the level of connectivity required for a graphical browser, using Lynx is your best bet. For maximum compatibility, using an 80-column VT-100 compatible terminal is necessary. This will allow Lynx's formatting to display properly.

Most Web sites use buttons, or "links" to allow you to travel from place to place ("page" to "page"), viewing more specific information on a topic, or examining a different topic completely. These links are selected in the graphic browsers through the use of a mouse, and by the keyboard in the text-based browsers.

Every page has an address called a URL (Uniform Resource Locator). This is how a page is accessed for the first time in a browser or through the links. These addresses generally start with http:// followed by a directory and possibly a file name. Your internet provider will be able to tell you the address to use for any documents you eventually create.

What kind of file could incorporate all of these graphics, sounds, and links into a presentable package? A simple text file written in the HTML (HyperText Markup Language) format. The graphic and sound files are stored separately, and accessed by the browser when it sees the commands, or "tags" (surrounded by '<' and '>' characters) imbedded in the text document.

Beginners HTML

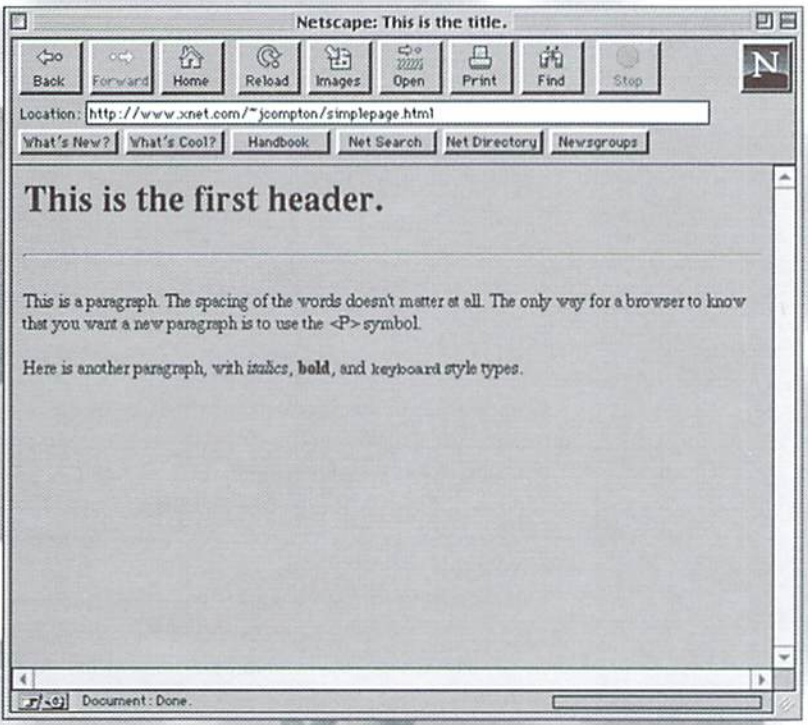
Don't be intimidated by the sound of "HyperText Markup Language". The basics are easy to learn and simple to use; no programming background is necessary. Many computer beginners have created their own HTML documents, or "home pages." What does a simple HTML document look like?

Note: The "begin" and "end" lines surrounded by dashes are not part of the documents. They are merely there to show where the document begins and ends, and give its filename.

```
<HTML>¶
<HEAD>¶
<TITLE> This is the title. </TITLE>¶
</HEAD>¶
<BODY>¶
<H1> This is the first header. </H1>¶
¶
<HR>¶
¶
<P> This is a paragraph. The spacing of the w
ords doesn't matter at all.¶
¶
The only way for a browser to know that you w
ant a new paragraph is to use the &lt;P> symb
ol. <P> Here is another paragraph, with <EM>i
talics</EM>, <STRONG>bold</STRONG>, and <KBD>
keyboard</KBD> style types.¶
¶
</BODY>¶
</HTML>¶
```

What does this look like in a browser? Well, the title would be shown either as a window title, or somewhere else near the top of the screen. Within the window or screen, the graphic (above right) would be shown.

Starting from the beginning, the first tag is <HTML>. This simply lets the browser know that the document is indeed an HTML document. The



Location: http://www.xnet.com/~jcompton/simplepage.html

This is the first header.

This is a paragraph. The spacing of the words doesn't matter at all. The only way for a browser to know that you want a new paragraph is to use the <P> symbol.

Here is another paragraph, with *italics*, **bold**, and `keyboard style` types.

</HTML> likewise shows that the end of the document has been reached.

Most tags follow the <TAG_NAME> ...text... </TAG_NAME> rule, with beginning and ending markers delineating the area of effect. Tags may be nested, but each inner tag must be COMPLETELY enclosed by the outer tags. A tag can not have its ending portion sooner in the document than the ending portions of the tags it contains.

The next tag, <HEAD>, with its companion </HEAD>, and its contents <TITLE> and </TITLE> demonstrate the nesting rules admirably. The <HEAD> section is completely between the <HTML> tags, and the <TITLE> section is completely between the <HEAD> tags. The <HEAD> tags simply enclose information about the document that doesn't necessarily appear within the document itself, such as the title. The <TITLE> tags enclose the text that is to be the title of that particular page.

The <BODY> and </BODY> tags surround the text and tags that should be shown in the browser window. All links, pictures, and text should be between these two tags. (And remember that these tags are also contained by the <HTML> and </HTML> tags).

The <H1> and </H1> tags surround what are known as "headers". Headers are specially sized text that can be used to separate topics in different areas of the page. There are six different size headers, with <H1> being the largest, <H3> being the medium size, and <H6> being the smallest. Lynx and other text-only browsers usually do not show different sizes of text, but still set them apart from the rest of the document to make them obvious.

The next two tags, <HR> and <P>, demonstrate two of the more common exceptions to the <TAG_NAME> ...text... </TAG_NAME> rule. These are stand-alone tags. <HR> instructs the browser to place a simple horizontal line at that location. This is good for separating areas of text on a page, and is considered to be good form.

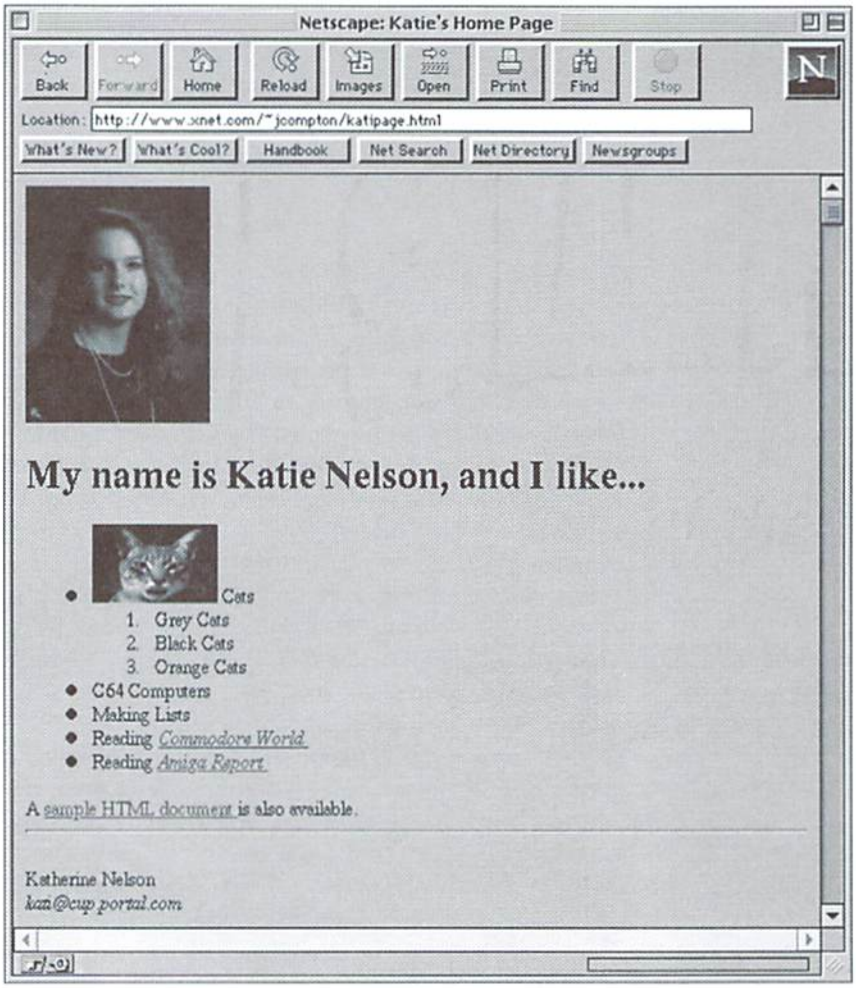
Also note that to have the browser actually display <P>, a special character "&lt;" had to be used in place of the "<". Alternately, we could have used "&gt;" in place of the ">" symbol. This is to prevent the browser from following the instruction.

<P> is one of the most often used tags in HTML. Any time a new paragraph is to begin, a <P> is necessary. The example above illustrates how browsers ignore spacing, and this is true also of return characters. Another way to break a line (not shown) without causing a new paragraph is the stand-alone tag <BR>. When a browser encounters <BR>, it will immediately go to the beginning of the next line before continuing with the current paragraph.

You shouldn't be surprised if some browsers do not actually use bold or italics where they are placed above. The tags are actually <EM> for EMPHASIS, and <STRONG> for STRONG emphasis. The convention is, however, that <EM> is italic, and <STRONG> is bold. Lynx's ability to display these niceties is limited-but don't worry, the text will still appear, in plain form if nothing else.

These tags are enough for a basic document with no links, pictures, or other embellishments. However, adding a few pictures gives your page a personalized touch, and can be extremely useful for providing information. Even if your browser does not allow you to view the pictures directly, you will be able to download the image for later consideration. Here's another more complex example, followed by the image the HTML program creates:

```
<HTML>¶
<HEAD><TITLE>Katie's Home Page</TITLE></HEAD>
¶
<BODY>¶
<P><IMG SRC = "kati.gif" ALT = "[Picture of K
atie]">¶
<H1>My name is Katie Nelson, and I like...</H
1>¶
<UL>¶
<LI> Cats <IMG SRC = "cat.gif" ALIGN = BOTTOM
ALT = "[Picture of Cat]">¶
<OL><LI> Grey Cats <LI> Black Cats <LI> Orang
e Cats </OL>¶
<LI> C64 Computers¶
<LI> Making Lists¶
<LI> Reading <A HREF = "http://www.the-spa.co
m/cmd/"> <EM>Commodore World</EM> </A>¶
<LI> Reading <A HREF = "http://omnipresence.c
om/Amiga/News/AR"> <EM>Amiga Report</EM> </A>
</UL>¶
<P> A <A HREF = "simplepage.html"> sample HTM
L document </A> is also available.¶
<HR><P>Katherine Nelson<BR><EM>kati@cup.porta
l.com</EM>¶
</BODY>¶
</HTML>¶
```



Netscape: Katie's Home Page

Location: http://www.xnet.com/~jcompton/katipage.html

My name is Katie Nelson, and I like...

- Cats
  1. Grey Cats
  2. Black Cats
  3. Orange Cats
- C64 Computers
- Making Lists
- Reading *Commodore World*
- Reading *Amiga Report*

A sample HTML document is also available.

Katherine Nelson
kati@cup.portal.com

The translated version (with "Katie's Home Page" in the title) looks like the graphic above. The first new tag is the <IMG SRC = "filename" ALT = "text"> tag. Here, "filename" is the name of the picture to be displayed (.GIFs and JPEGs are the most common), and "text" is the alternate text to be displayed for a text only HTML viewer. Both need to be enclosed in quotation marks. In the example with the cat picture, there is also ALIGN = BOTTOM. This is actually the default alignment. The others are ALIGN = TOP, where the top of the picture is aligned with the text, and ALIGN = MIDDLE, where the picture is vertically centered with respect to the text.

Next is the <UL>, </UL> set, and the stand-alone <LI> tags. <UL> stands for Unordered List, and with its </UL> counterpart, encloses all members of a list. Each member of a list begins with <LI>, as shown above. <OL>, Ordered List, and </OL> may be substituted for <UL> and </UL> respectively to obtain an ordered, numbered list (<LI> remains the same). Each of these two list types may be embedded within another list as shown. Notice again how spacing is unimportant within the code. It is nice to have orderly spacing, however, to make reading the source easier.

Another important type of tag is the link tag, also known as an "Anchor" tag. Text surrounded by the <A HREF = "URL"> and </A> is generally highlighted and underlined. Selecting this text will take you to the address URL. If the linked document is on another computer, a full address is required. However, assuming that a document is in the same directory (as with simplepage.html), only a "partial URL" is required, that being only the name of the linked document. Generally, on graphical browsers this text will appear in a different color and will be underlined, while on Lynx you will find that the text highlighted in reverse video.

Graphics may be incorporated into the links, by placing the <IMG SRC> tag inside the <A HREF> and </A> tags. The picture included in this manner will become part of the link, and clicking on the picture will lead to the given document. This is useful for creating "icons" to follow different links.

Using the anchor links, one page can lead to many others of varying topics. Most people have links to pages about their favorite music, books, computer, and other such information.

For particularly long documents, links are also possible to different places within the same document. Use <A NAME = "label"> and </A> to denote the destination of the link, and the link area is designated by <A HREF = "#label"> and </A>. Any text can be substituted for "label", but it must be consistent and contained in quotes. Be sure to remember the # sign before the label in the link.

This use of tags, to refer to locations within the same document, is often used on what are sometimes called "flat" hypertext pages. That is, while there are several different areas of interest within a single document, the tags and links provide a convenient way to jump around the document without having to load a new one into memory.

Now, to actually get your page up and running, check with your internet provider. Sometimes, they have a script that executes all necessary commands for you. Also, some machines may have different methods of setting up a page. On our UNIX internet provider, however, we execute the following commands:

```
mkdir ~/public_html
chmod -R a+rX ~/public_html
chmod a+x ~/
```

Then, anything in the public_html directory should be able to be read by outside viewers. If you test your site and you get an error message, try entering the last two commands again.

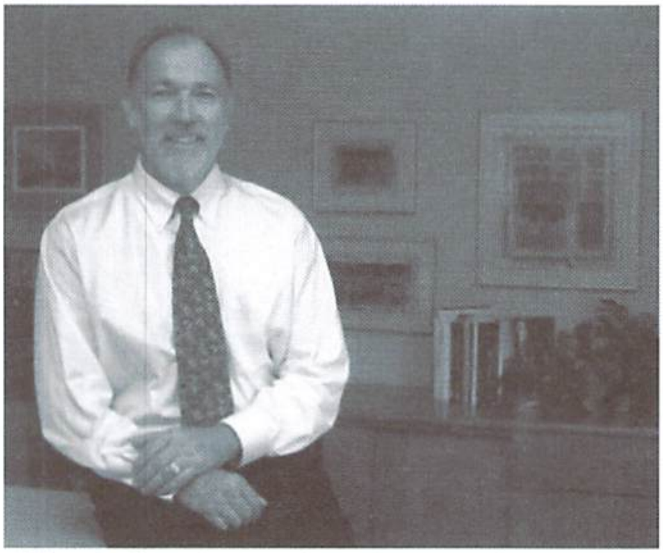
If you don't take our pictures' word for it, point your browser (more lingo!) at this URL — http://www.xnet.com/~jcompton/ and you'll find the two sample pages used in this article, as well as future examples to be used later in this article series. Or, if you feel that you're ready to branch out, try the CMD home page at URL http://www.msen.com/~brain/guest/cmd/ (Or, you can just use the link provided on our example page!)

The tags and instructions listed here should provide you with enough material to create a functional home page. Next issue, we'll pick up where we left off here with more HTML commands and other useful applications of the Web.

Historical Perspective:
The Mensch Connection

By Jim Brain

WDC founder William Mensch not only designed the W65C816S processor, but also played a vital role in creating the technology at the heart of our favorite computers.



William D. Mensch, Jr., chip designer and founder of The Western Design Center, Inc.

- Photo courtesy of The Western Design Center, Inc.

The Internet is buzzing with the news. Creative Micro Devices is planning to introduce a 20 MHz accelerator cartridge for the Commodore 64 and 128 (in 64 mode) in early 1996. The topic has generated much renewed interest in accelerator cartridges, which are special units that purport to speed up the operation of programs executing on your Commodore machine. Many are wondering how compatible the unit will be with GEOS, games, and productivity software. Others are comparing the planned unit with earlier designs, including the Schnedler 4MHz accelerator, introduced in the late 1980's, and the FLASH 8 cartridge, which runs at 8 MHz and was introduced just last year. Some of these designs used a faster version of the 6502 (the CPU found in the VIC-20), while others used a new CPU, the 65C816, which is a 16-bit version of the 6502 that can emulate a 6502. This is the chip that CMD plans to use in its product. Some of the discussion on the "Net" revolves around this latter CPU design.

All of the speedy CPUs used in these accelerators are the product of a small company in Arizona: The Western Design Center, Inc. That may not surprise you, but the way these chips are related to the 65XX CPUs in the Commodore computers just might.

Let's begin in the early 1970's, when Motorola started work on a CPU based on the Digital Equipment Corporation (DEC) PDP-8 minicomputer system. The designers worked diligently to create a one-chip CPU that shared similarity with the multiple chip CPU in the PDP-8, thought of by many as DEC's first successful computer system. While designing the chip, however, DEC came out with the very successful PDP-11 minicomputer. The designers wanted to design a CPU based on the PDP-11, but Motorola was not ready to commit to the newer technology in the PDP-11. The designers finished the original CPU, which became the Motorola 6800 CPU. In the following years, Motorola enhanced the basic design and introduced the 6805, the 6809 (used in the TRS-80 Color Computers and as a secondary processor in the Commodore SuperPET), and the 68HC11 (found at the heart of some American automobile computers). Nonetheless, the designers never forgot the PDP-11 idea.

One of the principal designers of the 6800 was Chuck Peddle, whom many Commodore owners remember as the father of the Commodore PET. Another was William D. Mensch, Jr., who is very important to this story.

After the 6800 was completed, the designers got a chance to work on their PDP-11 CPU derivative design. The engineers left Motorola for a small calculator chip manufacturer named MOS Technologies. The company embraced the idea of this CPU, and Chuck Peddle, Bill Mensch and others set to work to realize their idea. By 1975, the chip was completed. In September of 1975, MOS introduced the MCS6500 series CPU in two forms. One version of the CPU was pin compatible with the Motorola 6800 and was advertised as the MCS6501 for $20.00. The other, the MCS6502, did not share the pin compatibility, and was priced higher at $25.00.

In November, 1975, Motorola filed suit with MOS Technologies over the 6501, which began a long series of negotiations. Eventually, Motorola agreed to drop the suit if MOS would cease marketing the 6501. MOS complied, and the 6502 became the only CPU offered for sale.

Many Commodore owners know the name MOS Technologies as the producer of all of the ICs in the Commodore computer systems. For years, Commodore had contracted with MOS for calculator chips, which was Commodore's big market prior to computers. As the 6502 was entering the market, Jack Tramiel of Commodore started continually asking for more volume on the calculator ICs, forcing MOS to add more staff for increased production. Then, almost overnight, Commodore told MOS no more chips were needed. MOS, over-staffed and having just lost a large customer, became

The Mensch Computer™
- Photo courtesy of The Western Design Center, Inc.

financially unstable, and Commodore stepped in and bought the company for a fraction of its worth. In late 1976, MOS Technologies and all of its employees became part of Commodore Business Machines.

Now comes the tale of the two engineers. Both started working on the new computer system to be offered by Commodore, the Commodore PET. One engineer, Chuck Peddle, decided to stay with the new company and finish the machine, while the other, Bill Mensch, saw that Commodore had no intention of furthering the 6502, and left the company for greener pastures. As Commodore owners know, the Commodore PET system was the beginning of a long line of successful computers, including the VIC, the C64, and the C128. The story should end here... or should it?

In 1978, Jack Tramiel sent word that Commodore needed to re-engineer its calculators to use the new CMOS fabrication process (earlier MOS ICs used the older NMOS process). Bill Mensch was asked to come back and work for Commodore as a contractor. Jack met with Bill, and the two struck a deal. Mensch would start a business whose sole customer would be Commodore. The new business would engineer CMOS calculator ICs for CBM, and royalties from the sale of the chips would go to Mensch. The deal was inked in 1978 and the designing began.

The new company was named The Western Design Center, Inc., and was in business solely to create CMOS designs for Commodore. While the calculator designs were underway, Mensch tried to interest Commodore in a CMOS version of the 6502. True to their earlier stance, Commodore was still not interested in any new work on the 6502. However, Bill Mensch was very interested. The project was put on hold and the calculator chip designs were engineered. As the designs for Commodore neared completion, WDC acquired facilities in Mesa, Arizona, where they remain to this day. However, part of the payment for the building was supposed to come from the royalties of the chip designs, which didn't look too promising. No sooner had WDC finished the designs for CBM, than Toshiba halved the price of its calculator chips, and CBM stopped work on its calculator chip production. WDC was left to suffer the same fate as MOS.

However, Mr. Mensch was not about to let that happen. In October 1980, shortly after the CBM/WDC deal was signed, Xerox Corporation had sought to contract Mensch for some design work, but because of the CBM deal, WDC couldn't have Xerox as a customer. Xerox offered to give CBM access to its latest fabrication techniques in return for the ability to contract Mensch, but Commodore wasn't interested. After Commodore and WDC parted ways, Mensch contacted Xerox. They were still interested, and dumped their entire R&D budget into WDC's account to keep them alive until the beginning of 1981. At that time, Mensch and WDC helped design the CPU for Xerox's Star Systems Computers. These computers were the fabled work of the Xerox Palo Alto Research Center (PARC) that included the first graphical user interface and introduced the world to that amazing input device, the "mouse".

Mr. Mensch established WDC to implement his philosophy of design simplicity for high technology and to concentrate on design improvements for application proven W65C products. That philosophy evolved into a product line that includes the W65C02S and W65C816S CMOS microprocessors, the W65C134S and W65C265S CMOS microcomputers, the W65C22S Versatile Interface Adpter, and the Mensch Comptuer™, (above) a system that encourages new applications for W65C products and the platform for a wide range of consumer computers.

After that work was completed, WDC was contracted by Mitel Corporation to create a CMOS version of the 6800, which Mensch had helped design not even a decade prior. Evidently, Mycel had created a design, but the design contained a serious flaw that prevented the unit from achieving a speed over 250 kHz. WDC helped Mitel redesign the part, and Mensch saw that Mitel's fabrication process suited his needs for a CMOS version of a 6502, the project he had tried to interest Commodore in earlier. So as quickly as possible, WDC engineered a CMOS version of the 6502, called the 65C02, in 1982.

The new design was completed and initial prototypes ran at 4 MHz, 4 times the speed of the MOS/CBM NMOS version. Immediately, Rockwell and GTE licensed the unit and started production. Commodore, being the way they were, sued WDC over the design, claiming it used trade secrets acquired at MOS. Once again, the 6502 design faced legal problems. Luckily, this suit (as the first) was settled out of

Important Events in the Timeline

- The September 8, 1975 Issue of Electronic Engineering Times contains the first ad for 6501/02 from MOS.
- The November 3, 1975 Electronic News has first mention of Motorola/MOS suit.
- Mensch left MOS in March of 1977.
- Mensch was contacted by CBM in February of 1978.
- Mensch and Tramiel met on Mother's Day, 1978 to strike the deal. They talked after Jack had won a couple thousand at the blackjack table (I assume in Vegas).
- WDC came into being on May 26, 1978.
- Mensch had 10 MHz NMOS 6502 CPUs in operation in 1978. They were due to a mistake at the fab, and ran very hot, but also ran very fast.
- Commodore and WDC parted ways in October of 1980.

Other Tidbits

- Mensch says Commodore put many companies out of business, but put him *in* business.
- A 6600 CPU was designed at Commodore and included the 6502/RAM/ROM on a single chip. It was canned.
- Mensch helped in designing the PET, the VIC-20, and the KIM-I.
- Mensch wrote a game for a test machine used to test 6502 chips. It used a 6522 to run the screen. The game was similar to "BarnStorming".

court, since it was shown that the timing for the two units was not identical. However, the most important piece of evidence for WDC was a particular operation of the CPU that very few programmers ever use.

The 6502 can treat data in the CPU two different ways. The first is called binary mode, in which a byte holds a quantity from 0 to 255. The other mode (which most programmers stay far away from) is called decimal mode, in which each byte is treated as two 4-bit nybbles which can hold a value from 0-9. Thus, a byte can express values from 00 to 99. Under arithmetic operations, decimal mode causes the CPU to perform extra work. In binary mode, adding $01 to $19 yields $1A. However, in decimal mode, the same operation looks different ($01+$19=$20). Notice the difference. In decimal mode, the low nybble is computed first ($1+$9=$0), and the "carry" is added to the high nybble addition. This is called high nybble fixup. While at MOS, Bill Mensch designed a way of doing the fixup elegantly and had been issued a patent for it. In designing the 65C02, he purposely performed the fixup in a different way which required an extra clock cycle. Therefore, Commodore couldn't claim patent infringement. So Commodore, without much evidence to take to court, settled the suit. As part of the settlement, WDC allowed Commodore to license the design.

After the 65C02 started taking on a life of its own, WDC and Bill Mensch set its sights on a 16-bit version of the 6502. Again, Mensch tried to interest Commodore in the idea, but CBM was not interested. In fact, Commodore thought the idea was ludicrous. WDC contacted Apple Computer Corporation, who came to take a look at the design. The Apple engineers were impressed, but flatly stated that Apple had no interest in such a CPU. Mensch was disappointed, but vowed to complete the design, believing that Apple would use the unit someday. How right he was!

To complete the design of the 16-bit 6502, Mensch worked on the plans at home on a game table. He kept it distinct from his business ventures, since the design could not be funded. Then, 1983 saw the birth of the 16-bit 6502, dubbed the 65C816. Initially, the unit ran at 4MHz and came in two versions. One, called the 65C816S, was the full blown design. The other, called the 65C802, was a 65C816 that was pin compatible with the 65C02. To ease migration of code from the 65C02, both versions could emulate a 65C02 CPU.

Not long after the prototypes were produced, WDC got a call from Apple. It seems Steve Wozniak, the famed designer of the Apple II, had come out of semi-retirement to head up the design of the Apple IIX, and the "Woz" wanted to use the 65C816. Bill Mensch's prophecy had come true as Apple introduced the Apple IIX, called the Apple IIGS when introduced. Furthermore, all that time spent designing the unit at home had paid off for Bill Mensch.

The rest, as they say, is history. The success of the Apple IIGS kept money rolling into The Western Design Center. As the years progressed, both the 65C02 and the 65C816 saw their die size made smaller, resulting in increasingly faster clock speeds, and finally culminating in the 20 MHz unit induced in 1995. Even after the Apple IIGS was discontinued, each CPU found customers who were familiar with the 6502 architecture, and impressed with the speeds of WDC's CPUs. One such customer is more familiar to many of us. At the turn of the decade, a new genre of game consoles came on the market, one of the first being the Nintendo Entertainment System. Its successor was the Super NES, which uses a 65C816S as the main CPU.

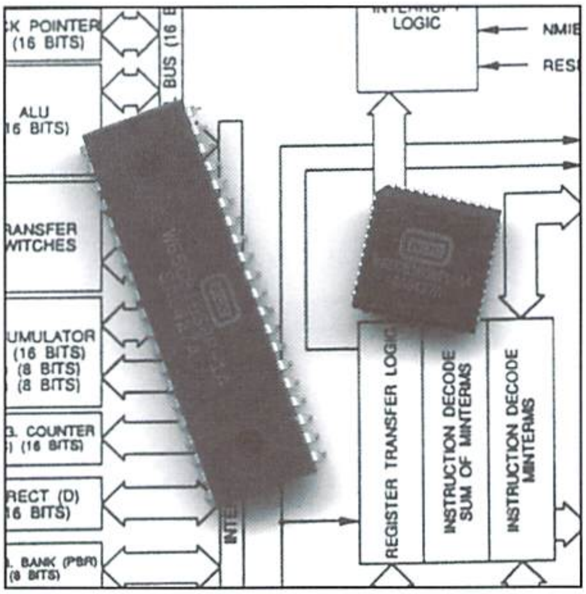
Commodore has seen some use of this CPU as well. While Commodore never bought into the 16-bit CPU idea, third party manufacturers have brought out numerous accelerator cartridges for the Commodore 64, including either the 65C02 or 65C816S as the heart of the units. As earlier noted, the FLASH-8 cartridge introduced in Europe last year uses the 8 MHz 65C816S, and the CMD unit planned for 1996 will use the newly introduced 20 MHz version of the chip.

So, if you purchase a CMD SuperCPU accelerator cartridge, remember that when you hook the unit up to your Commodore 64 or 128, you will be reuniting two technologies connected to a single man: William Mensch.

The Western Design Center, Inc.
2166 East Brown Road
Mesa, Arizona 85213
Phone: (602) 962-4545
Fax: (602) 835-6442
E-mail: wdesignc@indirect.com

THE SUPER PROCESSOR

Western Design Center's W65816S



By Jim Brain

It has been announced on the Internet and FIDONet, written about in this magazine and many others like it, and I am sure some of you may have already ordered it. What am I referring to? The CMD SuperCPU accelerator cartridge. At a maximum speed of 20 MHz, the unit will be over twice as fast as the current crop of accelerators. In addition to speed, the unit contains a processor that is more than just a souped-up 6510 CPU. This article will touch on the differences between the 6510 and this new processor, the W65C816S from WDC, The Western Design Center, Inc.

The W65C816S from WDC is an interesting processor by itself, but most programmers and users are probably wondering how this change will affect them and the performance of the SuperCPU cartridge, and many are interested in the additional features this CPU contains.

Emulation Mode

When the W65C816S CPU (called the '816 from here on) is first powered up, it behaves just like a 65C02, which is a CMOS version of the 6502 as found in the VIC-20. The 6510 in the Commodore 64 and the 8502 in the Commodore 128 are just 6502 CPUs with an internal I/O register and data direction register for that register. Since this set of registers can be added externally to a 6502 without any loss of compatibility, and a 6510 and 8502 only differ with respect to maximum processing speed, we will use the 6510 for comparisons. As the CPU will be in this mode most of the time, we should evaluate how it compares to a stock 6510.

Onboard I/O

As earlier noted, the '816 does not contain onboard I/O, but that can be addressed with some simple hardware in the accelerator unit. Therefore, this problem can be overcome and will not affect compatibility.

CPU Speed

The '816 can run at speeds from DC to 20 MHz. Obviously, this is a drastic change from the 6510, which runs at approximately 1 MHz. Programs which rely on the fact that instructions will be executed at 1 MHz should be expected to fail. Notable examples include many "demos", and programs that contain some disk speedup utilities. Programmers who want to ensure compatibility with accelerated systems should try to use the raster interrupt for video timing, and the CIA timers for general purpose timing, instead of "cycle counting". Where more precise timing is required, programmers could use a routine to check the number of cycles executed over a specific period of time via one of the CIA TOD clocks. [Note: the SuperCPU also has a register that can be used to slow the clock down to 1 MHz for routines which require timing based on the standard clock rate.]

Undefined or "Illegal" Opcodes

The original 6502, as designed by MOS Technologies, contained only 151 opcodes, but it used an entire 8-bit byte to represent each instruction. Therefore, there are 255 opcode possibilities, leaving 105 (256-151) undefined opcodes. In the case of the 6510, MOS didn't "mask" the effects of unused opcodes.

These opcodes are often called "illegal" because the CPU manufacturer reserved the right to change any or all of them at will, and advised against their use. They usually perform a function determined by others opcodes in the same row and column. Some programmers considered certain of the 6510's undefined opcodes useful in certain instances, and have utilized them over the years.

Since the '816 defines all of the 256 opcodes, any program utilizing a undefined 6510 opcode will most likely fail. Since the number of programs using these opcodes are few, this should not pose a significant problem. It may even be possible to patch some important programs that contain these offending instructions, but programmers would be well advised to avoid using undefined opcodes, or at least make some attempt to determine if the target computer can support such opcodes at run-time.

Zero-Page Page Crossing

In the 6510, issuing the instruction LDA ($FF),X would retrieve the contents of the address in (High:Low) $00:$FF. This behavior comes from the 6510 CPU's inability to cross the page boundary when using zero-page. The '816 will retrieve the data from the address in (H:L) $100:$FF, which is the more correct behavior. Very few programs should suffer from this difference, since most programmers don't use address $00 (which is where the onboard I/O resides) as a regular RAM location.

Indirect Jump Across Page Boundary

Also in the 6510, trying to do a JMP ($XXFF) will jump to the location in (H:L) $XX00:$XXFF. The '816 jumps to the more correct location in (H:L) $XX00+$100:$XXFF. Again, very few programs should fail to execute due to this difference, but the possibility is always there. Programmers should strive to prevent this scenario from occurring.

Decimal Mode Status

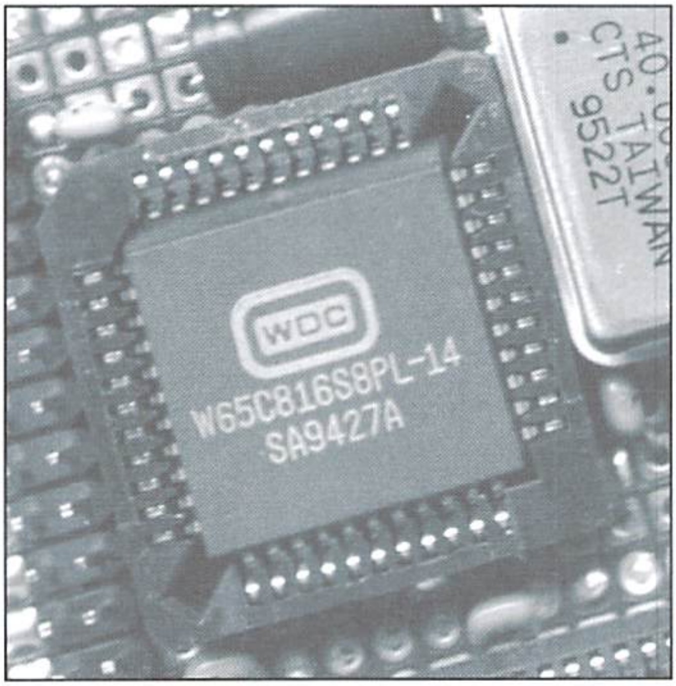
In the 6510, the status of decimal mode (rarely used mode on the 6510) is unknown after reset. Also, the status of this mode is not changed after an interrupt occurs. On the '816, both actions cause the Decimal Flag to be reset to 0. Since notable authorities like Jim Butterfield and most programming books advise against using this mode at all, the effects of this are likely moot.

Flags in Decimal Mode

The '816 ensures that the Z (Zero), N (Negative), and V (Overflow) flags are correctly set and reset when in Decimal mode. The Western Design Center claims that the 6510 does not correctly set these flags while in decimal mode. However, alternate sources indicate that the 6510 does indeed set these flags correctly when in decimal mode. As noted, so few programmers use this mode, the issue is most likely nonexistent.

RDY Pin Influence

On the 6510, if the RDY pin is pulled low during a write, the write will complete before the processor stops executing. On the '816, the processor will stop immediately. This is why both the VIC-II and the REC chip in the Commodore REU series wait 3 cycles after signaling the processor to stop before making any attempt to use the data bus during the CPU time slice. This should only affect those programs which use severe mutations of nonstandard video modes. [Note: While this difference would cause a problem if the 6510 were replaced by an '816, this isn't the case with the SuperCPU. VIC-II memory operations will occur only in the main computer, using the internal bus. REU access with the SuperCPU is handled by custom logic which mimics the described characteristics of the 6510.]



Interrupt After Fetch of BRK Vector

If an IRQ is triggered after the 6510 loads the vector for a BRK condition, the 6510 will ignore the pending BRK and load the IRQ interrupt vector and jump to it. On the '816, the BRK will continue to completion, and the IRQ will be handled after the BRK is handled. This incompatibility relies on these two events happening within a few clock cycles of each other. Since BRK is normally used only while executing a program in a monitor, this incompatibility should not affect typical users in normal situations. The '816 behavior does not pose a problem for the programmer, since he or she will always catch the BRK situation now, whereas the 6510 might allow it to slip through the cracks. This also applies to other interrupt scenarios, as in NMI after IRQ, NMI after BRK, etc. The 6510 might lose the first condition, while the '816 will always handle all exception conditions.

All other aspects of the CPU are identical to the 6510 with respect to function and form. It is stated by The Western Design Center that the 65C816S can be inserted into an Atari 2600 game console and the user can play all but a handful of the games ever marketed for the machine. However, that was when the CPU ran at the same speed as the original 6502 it replaced. Although the above conditions do exist in emulation mode, the single most problematic issue is simply speed. Some programs will refuse to run at the increased CPU clock speeds, while others (mostly games) will simply execute too quickly to be useful.

Native Mode

The '816 is actually two microprocessors in one. As noted, it emulates a 6502/6510 very nicely, but wait, folks—there's more! The '816 contains an impressive set of extensions to the 6502 architecture that enable it to perform 16-bit operations. That's right! The '816 is a 16-bit CPU. Here is a quick rundown of the many features available in 16-bit, "native" mode:

- Internal 16-bit Arithmetic Logic Unit (ALU)
- 16-bit stack pointer (stack space increased from 256 bytes to 64KB)
- 16-bit accumulator and index registers
- 24-bit address bus allows RAM capacity of 16 MB
- Includes original 6502 addressing modes and opcodes
- Additional opcodes to support 16-bit operations
- Support for Virtual Memory
- Extra pins on IC indicate internal conditions like interrupt vector fetch, valid data and valid program address
- Instructions like WAI (Wait for Interrupt) and STP (Stop Clock) further decrease CPU power usage

Internal support for co-processor

For the end user, it is apparent that most of these enhancements will more than likely only be useful if software is written to use these new features. Other features may require that additional hardware be developed to take advantage of the many additional modes and properties. So in the remainder of this article, I will introduce the Commodore programmer to these new features and suggest some possible uses for them with respect to Commodore computers.

16-bit Architecture

The '816 is a 16-bit version of the 6502. Internally, the unit can support either 8- or 16-bit data as single units. The programmer can switch between 8- and 16-bit functions via a bit in the status register. In 8-bit mode, the accumulator is 8 bits wide (A), while in 16-bit mode the accumulator is 16 bits in width (A + B = C). In 8-bit mode, the upper 8 bits of the accumulator, although not accessible to the programmer, can be used as temporary storage with the Exchange Accumulator (XBA) instruction. The index registers (X and Y) can also be 16 bits in length, allowing the programmer to index across 64KB of data.

> ***"Because the CPU provides so many new features, the Commodore computer owner and/or programmer may be initially overwhelmed by all the new things that can theoretically be accomplished."***

16-bit Stack Pointer

Stack oriented programming languages like 'C' and Pascal are more suited to the '816 because the '816 contains a 64KB stack page. In contrast, the 6502/10 has but one-sixteenth of that space. Machine language programmers will also appreciate the added stack area for multiple or recursive subroutine calls.

24-bit Address Bus Allows RAM Size of 16 MB

With the address bus increase to 24 bits in width (the 6510 has a 16-bit address bus), the '816 can directly access 16 MB of memory. Since all of the area can be seen by the processor at one time, the concepts of bank switching, REU DMA, and other memory expansion tricks are no longer necessary. Of course, the added memory must be present in order to make use of this feature, but the possibility is there.

Includes Original 6502 Addressing Modes and Opcodes

Programmers can start porting applications and utilities to the '816 immediately. All of the familiar instructions and addressing modes found in the 65XX architecture are included, as are the familiar registers .A, .X, and .Y. In fact, programmers may opt to not learn any new instructions at all.

Additional Opcodes to Support 16-bit Operations

To support the new features found in the '816, the CPU contains over 90 new opcodes. The ability to transfer the stack pointer to C register (TSC), store zero bytes into memory (STZ), and perform block moves with a single opcode (MVN, MVP) ensure that the '816 programmer has the tools needed to simplify programming tasks.

Support for Virtual Memory

With the '816, extra hardware outside the CPU can alert it to an exception condition on the memory currently being accessed. The ABORT pin can be brought low to force the CPU from updating any registers affected by the currently executing instruction. This facility, coupled with some additional hardware, permits the creation of more sophisticated operating system designs that force programs to request memory and disallow access to memory areas not owned by that program. This allows designs of multitasking operating systems and multiple user systems. Imagine a Commodore 64 being used by multiple people. (This is my personal favorite feature.)

Extra Pins on IC

These indicate internal conditions like interrupt vector fetch, valid data and valid program address. The serious programmer will appreciate the extra outputs available on the CPU. These extra logic pins will reflect what is happening inside the processor during internal operations. These pins could be used for debugging large programs, or external CPU enhancement cartridges. Other uses include co-processor operation, and multiple CPU environments.

Instructions like WAI (Wait for Interrupt) and STP (Stop Clock) further decrease CPU power usage. Although not necessary for a home computer like the C-64 or C-128, these features could prove useful for individuals taking machine into the field or on the road. Each of these opcodes puts the CPU into "sleep" mode, which requires very little current draw.

Internal Support For Co-processor

The Co-processor opcode (COP) can be used to support additional or secondary processors, or emulate complex pseudo-opcodes in software. 127 unique user COP opcodes can be defined however the programmer sees fit. This feature provides an easy way for programmers to interface with secondary processors, and emulate the operation of a secondary processor if one is not present.

And More Surprises

There are many other notable features in the '816. In the area of indicator output pins on the chip, one pin indicates the mode the processor is running in (emulation or native mode), while another pin displays whether the accumulator and index registers are currently being accessed as 8-bit registers or 16-bit ones. Finally, one pin can be used to externally synchronize memory shared by multiple '816 processors (The Memory Lock or MLB pin).

Internally, the processor has yet more surprises. The programmer can switch between native and emulation mode on the fly, and the stack of the CPU can be accessed via relative addressing. With the 16-bit index registers, programmers can read and write 64KB chunks of data in one loop, and the block move opcodes make moving memory impossibly easy. Since all opcodes are present in both modes, one can take advantage of many of these features without having to reprogram existing 6510 machine language.

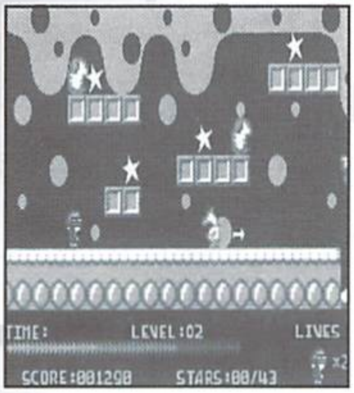
Conclusion

It is beyond the scope of this article to completely address all of the features of the '816. Because the CPU provides so many new features, the Commodore computer owner and/or programmer may be initially overwhelmed by all the new things that can theoretically be accomplished. I am convinced that only a short time will pass before many of these features will be exploited to further the power and usability of the Commodore computer system. However, let us always remember to walk before we run, and apply that principle to our computing as well.

MORE NEW GAMES

EXCITING NEW PRODUCTS AVAILABLE FROM CMD IN 1996...

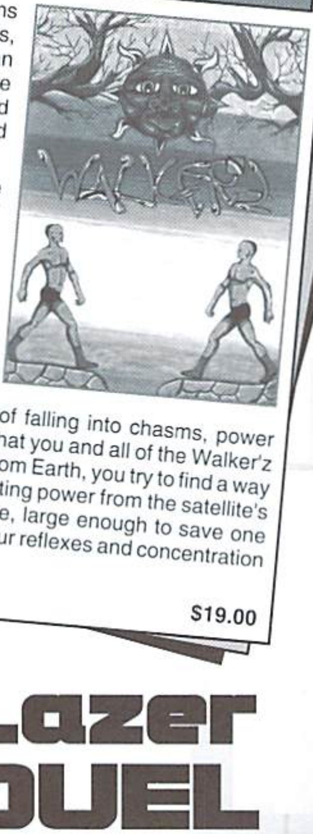
THRESHOLD PRODUCTIONS *presents* SLATER MAN



Armed with an unlimited supply of ammo, you set out to retrieve all the stars and terminate or avoid the creatures who try to stop you, while humming the catchy tune featured in this game. SlaterMan features five unique screens for each level, passwords for moving to a given level, plus the ability to enable both music and sound, just music, just effects or silence. SlaterMan is a well designed game promising a great escape from everyday stress; but best of all, you can play it without reading volumes of instructions!

SLATER SlaterMan **$14.95**

WALKER'Z



Walker'z are genetically engineered life-forms programmed to reconstruct dying planets, and turn them into class M planets which can support human life. Walker'z do not require oxygen, can go weeks without food and water, are 10 times stronger than a man, and have a life span of more than 200 years.

In the year 2101, Walker'z are sent to the planet Talia, which is about to implode and create a black hole. Only 72 hours remain when the Walker'z land on the planet. However, shortly after they take up position, a freak magnetic storm completely erases their memory, causing them to run amok across the planets surface.

You watch in horror from your sattelite as the Walker'z wander aimlessly; running the risk of falling into chasms, power generators, or off the edge of cliffs! Realizing that you and all of the Walker'z could be destroyed by waiting for technicians from Earth, you try to find a way to save the Walker'z until help arrives. By diverting power from the satellite's shields, you can create a shield on the surface, large enough to save one Walker at a time. With everything on the line, your reflexes and concentration must be keen!

WALKERZ Walkerz by EBES **$19.00**

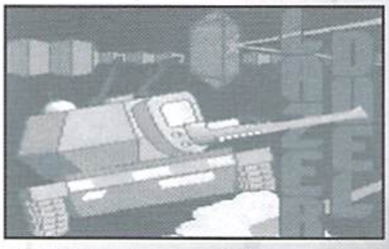
COMING IN WINTER '96
SWORD of HONOR

The dynasty sword, known as the Sword of Honor, has been in Shogun Yuichiro's family since the beginning of time. Tradition says that a Samurai's sword is his soul and that no one else can touch it, which is why the Yuichiro sword has been guarded for centuries.

The power-hungry damion, Toronaga, ruler of northern Japan, would do anything to defeat his rival, Yuichiro, ruler of southern Japan. To further his quest to rule Japan, Toronaga has stolen the Yuichiro sword. Upon finding that his sword has been stolen, Yuichiro calls you, his most dangerous ninja warrior, to reclaim his sword and family's honor! Despite your power, you must beware—rumor has it that the evil damion has mysterious powers!

If the story line didn't get you, the spec's on this phenomenal game will. Try 2.5 megabytes of compressed data - that's right, four disk sides or one 3.5" disk! Outstanding graphics - over 90 colors on screen! Outstanding Sound! It also features: lethal weapons, vicious hand-to-hand combat, and awesome gameplay! And if that's not enough - there's even a special 3.5" version for1581, CMD FD, HD and RAMLink owners! So keep your eyes open for the Winter '96 release of this outstanding action game!

THRESHOLD PRODUCTIONS *presents* Lazer DUEL



Fusion powered laser tanks firing through a field of geometric shapes that deflect or absorb each laser blast make up the field of play in this game of skill. Only those who have mastered the technique of banking shots in the direction of their enemy will survive! Single or Two player, Lazer Duel offers 40 screens and 80 levels - each victory bring's a greater challenge, making it difficult to beat the game!

LAZERDUEL Lazer Duel **$14.95**

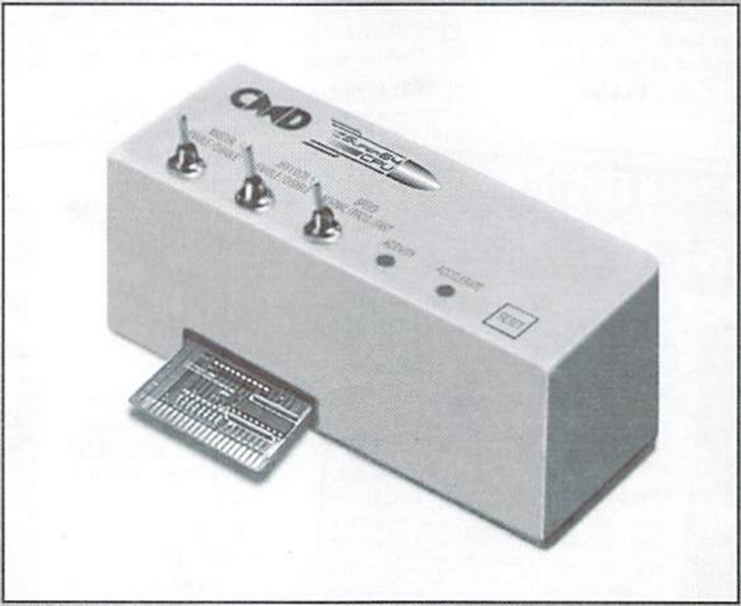
To Order CALL CMD
(800) 638-3263
See our general ad for shipping details

A SNEAK PREVIEW *of the* SUPERCPU

With the upcoming release of CMD's SuperCPU, there have been numerous calls, letters, and E-mail messages from users asking about this new product. What is it? What does it do? Where does it plug in? How will it effect the operation of your computer?

Hopefully, many of these questions will be answered by this exclusive preview of the product. It is important to note, however, that the SuperCPU is a work in progress, and that some of the details simply aren't available yet. As they become known, we'll endeavor to keep our readers informed, and up-to-date.

Let me also point out that CMD has already altered some of their plans for the SuperCPU product line. If you haven't yet read *On The Horizon* in this issue, CMD has announced that the SuperCPU 64/10 (the 10 MHz version) has been cancelled due to insufficient interest in that model. At the same time, there has been an official announcement of a 128 version. This latter decision by CMD was based on the large response from the 128 user community for such a product. But this latter announcement comes with a price: time. In order to make the 128 version feasible, CMD will attempt to co-develop parts of the 128 version along with the present 64 version, which they expect will now be delayed for an additional two months. For further details of these announcements, see this issue's installment of *On The Horizon*.



The SuperCPU from CMD *(Artist's rendering)*

What is the SuperCPU?

The Super64 CPU is an accelerator module that speeds up the operation of your C-64/128 computer. It plugs directly into the cartridge port, and not only gives your computer added speed in calculations, but also provides the computer portion of JiffyDOS (CMD's disk speed enhancement speed). At its heart is the W65C816S microprocessor running at 20 MHz and a CPLD (Complex Programmable Logic Device IC) which contains the custom logic required to mimic the characteristics of the Commodore 64. The C-64 version also contains 128K Bytes of high-speed Static RAM (the same type of high-speed cache memory found in 486/Pentium systems), and 64K Bytes of ROM. The 128 version will include additional RAM, ROM and custom logic to allow it to operate in both 64 and 128 modes.

Physical Characteristics

The SuperCPU is contained within an enclosure that measures approximately 6" wide by 3" high and 2" deep. The enclosure has an opening for the Cartridge-Port Pass-Thru connector at the rear, precisely in line with the computer's Cartridge Port. The unit is about the same width as a RAMLink or Commodore REU. The main circuit board inside the SuperCPU mounts vertically, keeping the depth of the unit to a minimum. This helps to prevent devices plugged into the pass-thru port from extending too far back from the computer. Three easy-to-use toggle switches line the upper front edge of the unit: a Master Enable/Disable switch, a JiffyDOS Enable/Disable switch, and a Speed selection switch. While the current prototype lacks a reset button and LED status indicators, these options are being considered for the final design.

Installation

Installing the SuperCPU 64 is simple: just plug it into the Cartridge port. No additional wiring or jumpers need to be installed inside of the computer. (The C-128 version may, however, require a jumper clip.)

Operation

Operating the SuperCPU requires no special knowledge. The user can perform all standard computer tasks in exactly the same manner as with a stock system, gaining the benefit of greatly accelerated speed. The toggle switches on the unit are clearly labeled, easily accessible, and can be used without having to refer to a manual in order to determine their function. As with CMD's RAMLink, the SuperCPU has been designed to work out-of-the-box with any C-64 or 128 computer—no manual "tuning" adjustments will be required.

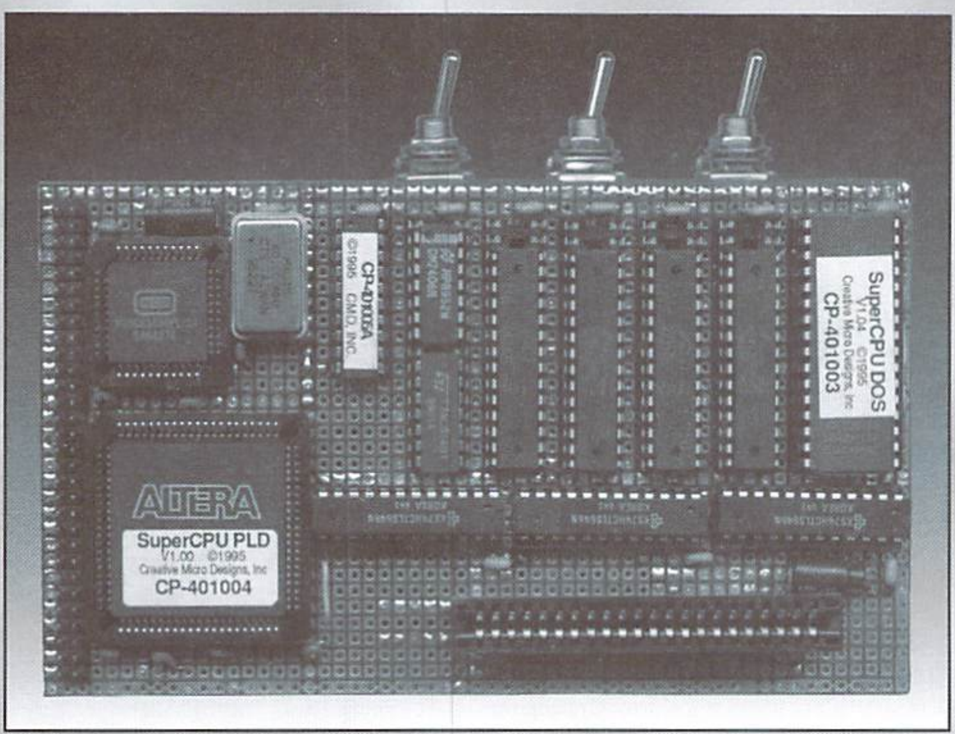
Compatibility

The SuperCPU 64 operates with C-64 and C-64c computers as well as the C-128/128-D in 64 mode. The W65C816S supports all legal 6502/6510/8502 opcodes, and will therefore provide a very high level of compatibility with existing software. The C-64 memory map is emulated exactly by the CPLD, and proprietary write-thru logic in this chip ensures full compatibility with all VIC graphic modes and memory mapping.

REU's: The CPLD chip in the SuperCPU includes special DMA transfer logic to provide 100% compatibility with all types of data transfers to and from Commodore REU's. The unit does not have to be slowed down to 1 MHz in order to initiate a DMA transfer to/from the REU.

RAMLink: The Super CPU is 100% RAMLink compatible, and contains its own version of RL-DOS, which runs from the SuperCPU's high-speed Static RAM. The faster RL-DOS, along with the efficiency of new 65C816S opcodes allows faster data transfers to and from RAMCard memory and also speeds up data transfers to/from CMD HD Series hard drives connected to RAMLink's parallel port.

GEOS: Version 2.0 will be 100% compatible. Special software will be provided with the SuperCPU allowing GEOS to run at full speed.



The current prototype board for CMD's Super 64 CPU

Performance

To help achieve its high level of performance and to accommodate the 20 MHz speed, all operating system routines are downloaded into 64K of the high-speed Static RAM during power-up. This feature eliminates the bottleneck caused by ROM-based operating system code (on PC's, this technique is called ROM-BIOS shadowing). 64K of high-speed SRAM remains free for programs. In addition, buffered write-thru circuitry (similar to PC cache controllers) eliminates the need to slow down when writing data into computer.

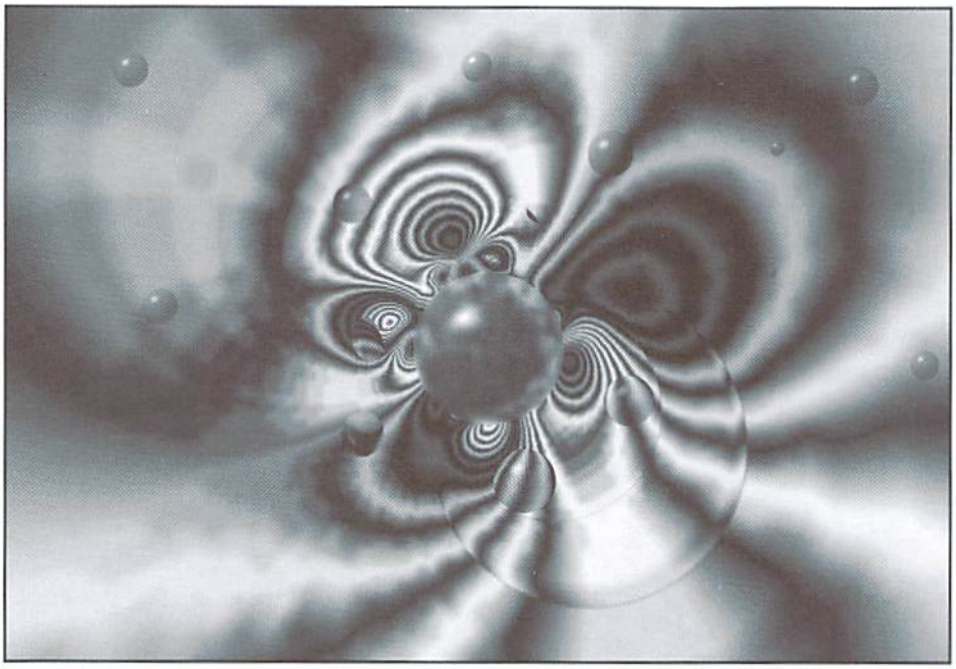
Additional Features

Built-in JiffyDOS Kernal ROM: The computer portion of JiffyDOS is built into the SuperCPU, providing DOS-Wedge commands and enabling high-speed serial transfer rates to and from JiffyDOS-equipped disk drives. The JiffyDOS Enable/Disable switch is designed to function on the fly without disabling the accelerator itself.

Cartridge Port Pass-Thru: Enables the full use of most plug-in hardware devices such as REU's, RAMLink, Swiftlink, SID Cartridge, GEORAM, and some software cartridges. These devices will work while the accelerator is running at 20 MHz.

GEOS Software: Custom software to optimize GEOS operation is provided. Additionally, an intelligent write-thru hardware circuit designed especially for GEOS effectively eliminates major performance bottlenecks associated with previous accelerator designs. Coupled with the 20 MHz clock speed, this feature will boost the performance of GEOS far beyond anything previously available.

Speed Selection: The speed selection switch provides three options: 1) Slow (1 MHz) mode for compatibility with programs (such as some games) that may operate too quickly in the turbo mode; 2) a "soft" Turbo mode which operates at 20 MHz and provides programs with a software speed selection register for switching to Slow speed when necessary; and 3) a "forced" Turbo



mode which keeps the CPU in 20 MHz mode regardless of the setting of the software register. This mode is useful for providing compatibility with existing programs that unintentionally alter the software register. (Note: The SuperCPU performs disk access functions properly in all three speed modes.) The user can change speed modes by toggling the speed selection switch while the SuperCPU is operating without quitting from the current program.

Expansion Capabilities

The SuperCPU includes an internal connector (called the "Rocket Socket") which will enable the user to plug in a CMD-supplied RAM Expansion card. This card will have on-board SIMM sockets and can contain from 1 to 16 Megabytes of Dynamic RAM (DRAM). The RAM on this card can be used as either data or program storage. In addition, some or all of this RAM can be configured for use as an ultra-high-speed RAM disk drive.

Development Support

CMD will be actively working with developers willing to create new commercial software and/or modify existing software to take full advantage of the SuperCPU. A comprehensive developers package will be available, and will include an assembler which supports all 65C816S opcodes and addressing modes. In addition to the assembler, extensive documentation will provide pertinent technical specifications, guidelines, and sample code. This package will enable developers to produce programs which take advantage of the 65C816S's enhanced instruction set, 16 MB addressing capability (for users with the optional SuperRAM Expansion Card), as well as other SuperCPU capabilities.

Programmable Functions

The SuperCPU contains a special register which allows programmers to control the clock speed. An additional register is used to control the SuperCPU's WriteSmart™ function (see the sidebar), which determines what areas of the SuperCPU memory are mirrored in the host computer. Applications which can use reduced mirroring can significantly increase their overall operating speed.

Temporary Conclusion

As mentioned previously, the SuperCPU is a work-in-progress, and it will take a little more time before everything is known, and before everything that appears to be 'known' now is really considered to be 'carved in stone'.

If anything can be concluded, I believe that CMD is creating a powerful new engine to power our C-64's and 128's, for those of us who either want or desire that power. Clearly, the features outlined here show that this design goes far beyond other previous accelerators in the areas of performance, expandability, compatibility, and features.

Furthermore, the creation of an assembler and the developers kit hints at the possibility of new applications in the future that will take advantage of the power and resources this new device brings to the C-64/128 platform. Naturally, we'll keep you informed as further developments unfold!

Some SuperCPU Technical Notes

The SuperCPU operates independently of the host computer's clock, but can synchronize with the host when necessary. This allows the SuperCPU to process data at a full 20 MHz while the VIC chip continues normal operation. Slowdowns can occur, however, because the host computer's memory and I/O locations must be updated when the associated SuperCPU memory locations are changed. In addition, serial bus timing routines must be performed at the standard 1 MHz speed.

Since the host computer's clock and memory operate at a slower speed than the SuperCPU, memory transfers between the two must be synchronized. Such transfers are the result of using LOAD and STORE instructions (such as LDA or STX) when the memory referenced is part of mirrored memory.

By nature of its design, the SuperCPU is able to maintain a minimum effective speed of 4 MHz during these memory transfers. By optimizing code for the SuperCPU, higher transfer rates are possible.

Programmers can also take advantage of the WriteSmart™ and CacheWrite™ features within the SuperCPU to further optimize their code. Doing so allows optimum speed during memory transfers, and avoids transfers when they aren't required. Here's a brief overview of these features.

WriteSmart: This feature determines which areas of SuperCPU memory and host computer memory must be mirrored. A register within the SuperCPU allows setting of four different configurations: all memory, BASIC (for default screen and color memory), GEOS (GEOS screen and color memory), and none. I/O memory is always mirrored unless I/O is switched off.

CacheWrite: This feature works in conjunction with STORE instructions, and holds a byte that needs to be placed into the host's memory until the host can accept it. If no further STORE instructions occur while the cache is full, no clock-stretching will be required in the SuperCPU, and optimum speed will be maintained.

EXPAND YOUR SYSTEM

3 SLOT CARTRIDGE PORT EXPANDERS

The EX2+1 and EX3 Cartridge Port Expanders bring new expandability to you Commodore 64 or 128. Combine the use of compatible cartridges. Disable cartridges not in use. Change the address your computer finds a cartridge at. Avoid the constant changing of cartridges that puts extra wear on your cartridge port. The EX3 offers 3 vertical expansion slots, while the EX2+1 provides 2 vertical slots and 1 horizontal slot. Both units offer 7 switchable signals per slot, address mapping on slot 2, and a reset button.

CALL 1-800-638-3263

EX3

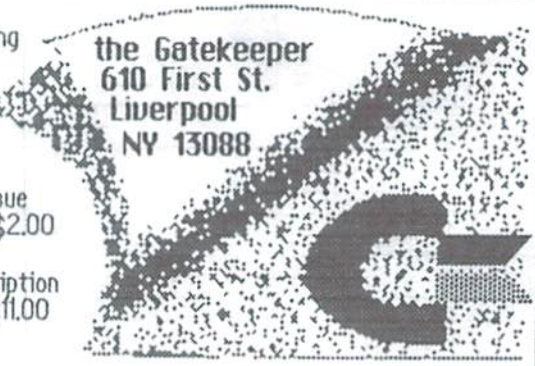
$29.95

EX2+1

$39.95

Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow MA 01028

- programming
- projects
- reviews
- news
- and more!



the Gatekeeper
610 First St.
Liverpool
NY 13088

Sample issue
(20 pages): $2.00

1 year subscription
(6 issues): $11.00

The final page in commodore computing.

Moving?

Don't forget to notify COMMODORE WORLD!

Please call or write with your change of address 6 to 8 weeks prior to your move so that you won't miss a single issue!

Click Here Software's

geoFAX

Available NOW

Send and Receive faxes on your C-64/128

Features

- **SEND & RECEIVE FAXES to/from any fax machine or computer running a fax modem**
- **User-friendly and easy to operate**
- **Auto-answer mode for unattended operation**
- **Manual send/receive mode lets you share phone line**
- **Built-in functions convert faxes to geoPaint files**
- **geoPaint files may be faxed without conversion**
- **Built-in printing functions for printing fax documents and geoPaint files**
- **Printer drivers may be selected directly from within the program**
- **Full-page scanning into geoPaint is possible using a standard fax machine as a scanner**
- **Built-in simple terminal program lets you browse telecommunications services**

System Requirements

- **Commodore C-64, C-64C, C-128, 128-D computer**
- **GEOS 64 or GEOS 128 version 2.0**
- **SwiftLink RS-232 cartridge**
- **Group III, Class 2 fax modem**
- **GEOS compatible disk drive (large capacity drive or RAM disk recommended but not required)**

Prices & Special Bundles

geoFAX........... $39.95	*geoFAX & SwiftLink $75.00*
SwiftLink $39.95	*geoFAX, SwiftLink & BOCA 14.4 Fax Modem......... $195.00*
BOCA 14.4 Fax Modem .. $129.95	

Creative Micro Designs

See our main ad in this issue for complete ordering information.

1-800-638-3263

CMD • HARDWARE • ORDERS: 1-800-638-3263 • SOFTWARE • CMD

HD Series SCSI Hard Disk Drives

HD-40, 42 MB (Limited Supply)	**$249.00**
HD-340, 340 MB (Special Edition)	**CALL**
HD-500, 500+ MB (Special Edition)	**$399.00**
HD-1000, 1 GB (Special Edition)	**$499.00**
HD-2000, 2 GB (Special Edition)	**$899.00**

FD Series 3.5" Floppy Disk Drives

FD-2000 (800K and 1.6 MB) (Special Offer)	**$159.00**
FD Real-Time-Clock Option/Kit	$20.00/$29.00
Box of 10, High Density Disks (1.6MB)	$14.95
Box of 10, Enhanced Density Disks (3.2 MB)	$29.00

JiffyDOS

(Specify computer serial number and drive model)

JiffyDOS C64/SX-64 System	$49.95
JiffyDOS C-128/128-D System	$59.95
Additional Drive ROM's	$24.95

Miscellaneous Hardware

80-column Monitors (Refurb)	CALL
Aprotek 3-Way User Port Expander	$30.00
Aprotek User Port Extension Cable	$19.00
C-64/64-C Computers (Refurb, w/JD)	$89.00
C-64, C-64C Power Supply (Repairable)	$39.00
C-128 Power Supply (Repairable)	CALL
Cannon BJC-4100 Color Bubble Jet Printer	**$399.00**
Commodore 1541 Disk Drives (NEW, w/JD)	$119.00
Commodore 1541 Disk Drives (Refurb, w/JD)	$75.00
Commodore 1541-II Disk Drive (Refurb, w/JD)	$89.00
Commodore 1571 Disk Drive (Refurb, w/ JD)	$99.00
Commodore 1802 40-column Monitor (Refurb)	$149.00
CMD EX2+1 3-Port Cartridge Port Expander	$34.95
CMD EX3 3-Port Cartridge Port Expander	$29.95
CMD Gamepad/Joystick Controller	$24.95
CMD GeoCable II Print Cable 6ft./15ft.	$29.00/$34.00
CMD SmartMouse (1351 Compatible Mouse)	$49.95
CMD SmartTrack (1351 Compatible Trackball)	$69.95
Inkwell Light Pen Model 170C	**$75.00**
Monitor Cables	CALL
Mouse Pad	$2.95
MW-350 Printer Interface (ØK/8K Buffer)	$49.00/$60.00
Panasonic KXP-1150 9-pin Printer	**$169.00**
Panasonic KXP-2023 24-pin Printer	**$229.00**
Panasonic KXP-2135 24-pin Color Prinnter	**$319.00**

Utilities

Big Blue Reader V4.10 (SOGWAP)	$39.00
CMD Utilities	$24.95
JiffyMON-64 (ML Monitor)	$19.95
The Compression Kit '94 (Mad Man)	$39.00

GEOS

Collette Utilities (Handy Geos Utilities)	$19.95
Desk Pack Plus	$29.00
Dweezils Greatest Hits (NewTools2,Stamp,Label128)	$30.00
FONTPACK Plus	$25.00
gateWay 64 or 128 (Specify Version)	$29.95
geoBASIC	$20.00
geoCalc 64/128	$40.00/$45.00
geoChart	$29.00
geoFAX	**$39.95**
geoFile 64/128	$40.00/$45.00
geoMakeBoot (Makes Bootable copies)	$12.95
geoProgrammer	$45.00
geoPublish	$40.00
GEOS 64 v2.0	$44.00
GEOS 128 v2.0	$49.00
geoSHELL V2.2 (CLI for GEOS)	$24.95
International FONTPACK	$25.00
Perfect Print LQ for GEOS (Laser-like output)	$49.95
RUN GEOS Companion	$20.00
RUN GEOS Power Pak I or II (Specify)	$20.00

Books

Anatomy of the 1541	$12.00
Basic Compiler Design for the C-64	$12.00
C-64 Science & Engineering	$12.00
C128 Computer Aided Design	$12.00
C128 BASIC Training Guide	$12.00
Cassette Book for C-64 and Vic 20	$12.00
Commodore 64 Tricks and Tips	$12.00
GEOS Programmers Reference Guide	$35.00
Graphics Book for the C-64	$12.00
Hitchhikers Guide to GEOS	$35.00
Ideas for Use on Your C-64	$12.00
Printer Book for the C64	$12.00
Mapping the C64	$12.00
Simple Internet	$16.95
Superbase - The Book	$15.00

Scanning & Video

Handyscanner	$249.00
Pagefox	$139.00

RAMLink Power-Backed RAM Disk

RAMLink Base Model (ØMB, No RAMCard)	$149.00
RAMLink w/1 MB RAMCard	**$229.00**
RAMLink w/4 MB RAMCard	**$349.00**
RAMCard RTC Option/Kit	$10.00/$19.00
RAMLink Battery Back-up (Optional)	$24.95
Parallel Cable (RAMLink to HD)	$14.95

MIDI & Sound

Digimaster	$34.95
SID Symphony Stereo Cartridge	$44.95
Sonus 64 Sequencer, MIDI Interface, Cables	$99.00
Sonus 128 Sequencer, MIDI Interface, Cables	$99.00

Languages & Compilers

BASIC 64 Compiler (Abacus)	$17.00
BASIC 128 Compiler (Abacus)	$25.00
Blitz! 64 Compiler (Skyles)	$30.00
Blitz! 128 Compiler (Skyles)	$30.00
Buddy 64/128 Assembler	$39.00
Cobol 64 (Abacus)	$17.00
Fortran 64 (Abacus)	$17.00
Pascal 64 (Abacus)	$17.00
Power C 64 (Spinnaker)	$14.00

Productivity

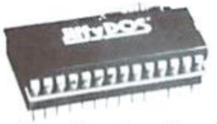
Cadpak 64 (Abacus)	$22.00
Cadpak 128 (Abacus)	$25.00
Chartpak 64 (Abacus)	$17.00
Chartpak 128 (Abacus)	$25.00
I Paint v1.5 (128, 80-col., 64K VDC) (Living Proof)	$39.00
I Port v1.54 (128, 80-col., 64K VDC) (Living Proof)	$29.00
Outrageous Pages (Batteries Included)	$19.00
Personal Portfolio Manager (Abacus)	$16.00
Pocket Writer 1 (64) Digital Sol.)	$35.00
Pocket Writer 2 (64/128) (Digital Sol.)	$65.00
Pocket Writer 3 (64 or 128) (Digital Sol.)	$70.00
Pocket Planner 2 or Pocket Filer 2 (Digital Sol.)	$35.00
PowerPlan 64 (Abacus)	$16.00
RUN Productivity Pak I, II, or III (Specify)	$15.00
RUN Super Starter Pak 1541 or 1581	$20.00
RUN Works	$20.00
SEC Check Register 128	$29.00
SuperScript 64 (Precision)	$15.00
SuperScript 128 (Precision)	$20.00
Suberbase 64 Version 3.01 (Precision)	$35.00
Superbase 128 Version 3.01 (Precision)	$35.00
SwiftCalc 64 (Timeworks)	$16.00
Perfect Tax 64 (Free '95 Upgrade)	$69.00
Perfect Tax 128 (Free '95 Upgrade)	$79.00
TWS 64 w/Speller (Busy Bee)	$29.00
TWS 128 w/Speller (Busy Bee)	$39.00
TWS Modules (HD/RL/Illustrator)	each $5.00

Telecommunications

Aprotek MiniModem C-24 (C= ready, 2400 baud)	$69.00
Aprotek MiniModem C (C= ready, 1200 baud)	$50.00
Aprotek Modem adapter (C= to Ext. PC Modem)	$20.00
BOCA 2400 Baud Modem	$69.00
BOCA 2400 w/SwiftLink and Cable	$99.00
BOCA 14.4K bps FaxModem	$129.00
BOCA 14.4K w/SwiftLink & Cable	$159.00
BOCA V.34 28.8K bps FaxModem	**$199.00**
BOCA V.34 w/SwiftLink & Cable	**$229.00**
Dialogue 128	$29.00
SwiftLink RS-232 Cartridge (Up to 38.4K baud)	$39.95
SwiftLink Modem Cable (DB9-DB25)	$9.95

Games

Atomino	$17.00
Ballistix	$13.00
Blood Money	$15.00
Chomp!	$10.00
Day in the Life of Prehistoric Man	$19.00
Escape Route	$19.00
Heavenbound	$19.95
Island of the Dragon	$19.00
Laser Squad	$13.00
Lions of the Universe	$19.95
Mainframe	$13.00
Menace	$15.00
Navy Seal	$10.00
Rings of Medusa	$16.00
RUN C128 Funpak	$10.00
RUN C64 Gamepak	$10.00
Skate or Die	$13.00
The Amazing Spider-Man	$15.00
The President Is Missing!	$10.00
The Three Stooges	$10.00
Tie Break Tennis	$16.00
Total Eclipse	$10.00
Ultima V	$17.00
Wings of Circe	$19.00
Wizardry 5: Heart of the Maelstrom	$24.00



JiffyDOS

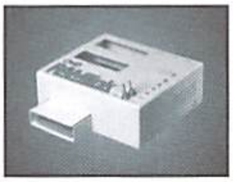
Increase Speed Up to 1500% while retaining 100% compatibility

- Speeds up Loading, Saving, Verifying, Formatting and Reading/Writing of Program, Sequential, User and Relative files unlike cartridges which only speed up Loading and Saving of PRG files
- Built-in DOS Wedge plus 17 additional features including file copier, text dump, printer toggle, and redefinable function keys make using your computer easier and more convenient
- ROM upgrade installs easily into most computers and disk drives. Supports C-64, 64C, SX-64, C-128, 128-D, 1541, 1541C, 1541-II,1571, 1581 and more.

FD Series

1.6 MB 3.5" Floppy Drives

The FD-2000 disk drives utilize today's latest 3.5 inch disk drive technology. FD-2000's support 800K (1581 style) and 1.6 MB (High Density) formats. Fast and reliable, they support 1541, 1571 and 1581 style partitions, Native Mode partitioning and can actually read and write 1581 disks. FD's feature built-in JiffyDOS, SWAP button and optional Real Time Clock. High capacity, speed and compatibility make the FD right for every application, including GEOS.



RAMlink

Power-Backed Expandable REU

The fastest possible form of storage, RAMLink provides instant access to files and retains data while your computer is turned off. Easy to use and expandable up to 16 MB, RAMLink utilizes the same powerful operating system found in CMD Hard Drives. Unlike Commodore REU's which are compatible with less than 1% of commercial software, RAMLink supports more than 80% of the commercial titles. RAMLink also offers built-in JiffyDOS Kernal, SWAP feature, reset button, enable/disable switch, pass-thru port and RAM port for REU/GEORAM use. RAMLink offers maximum speed, expandability and compatibility with all types of software and hardware including GEOS.

HD Series

SCSI Hard Drive for the C-64/128

HD Series Hard Drives are available in capacities up to 1 GB, are fully partitionable, and can emulate 1541, 1571, & 1581 disks while Native partitions utilize MS-DOS-style subdirectories. HD's connect easily to the serial bus or parallel via RAMLink. Includes built-in JiffyDOS, SWAP feature and RTC. HD's offer superior compatibility with most commercial software including BBS, Productivity and GEOS. And with new pricing, HD Series drives offer the lowest cost per megabyte of any C64/128 storage device.

Shipping and Handling Charges

Use the chart below to match your order subtotal with your shipping zone and method

	Continental United States				
Order Subtotal	UPS Ground	2nd Day	Next Day	AK HI, PR	Canada
$0.01 - $19.99	$3.00	$8.00	$17.00	$12.00	$5.00
$20.00-$29.99	$5.00	$9.00	$18.00	$14.00	$7.00
$30.00-$59.99	$6.00	$10.00	$20.00	$15.00	$9.00
$60.00-$149.99	$8.00	$12.00	$23.00	$19.00	$12.00
$150.00-$299.99	$10.00	$14.00	$27.00	$21.00	$20.00
$300.00-$799.99	$15.00	$20.00	$34.00	$27.00	$25.00
$800.00 +	$20.00	$25.00	$40.00	$32.00	$35.00

UPS C.O.D. add $5.00 (U.S./PR only)

WE ACCEPT ALL MAJOR CREDIT CARDS

Creative Micro Designs, Inc.

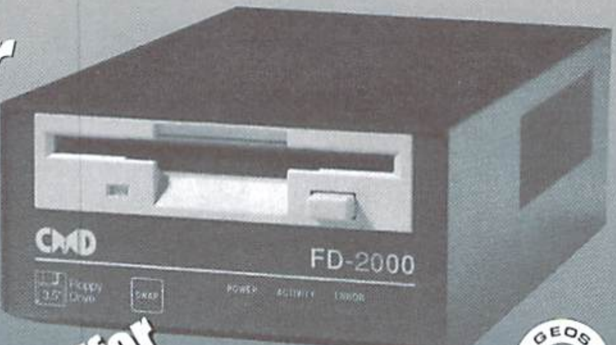
P.O. Box 646
East Longmeadow, MA 01028

Info: (413) 525-0023
Fax: (413) 525-0147

Policies: CMD accepts: Major Credit Cards, Money Orders, COD and Personal Checks. Personal Checks are held 3 weeks. Customer is responsible for shipping charges on refusals. Most items are stock, contact CMD for firm delivery. Returns for merchandise credit only within 30 days with prior authorization. Credits are less shipping, handling, taxes, customs, duties, and a 10% restocking charge. No refunds or credits on opened software. All prices and specifications are subject to change without notice.

Today's Most Advanced Mass Storage Devices Just Became More Affordable

Our lowest price *ever* on the FD-2000 3.5" Floppy Disk Drive!



FD-2000 Drive Specifications

- Amazing 1.6 MB capacity, plus reads & writes 1581 disks
- Up to 20% faster than a 1581 on stock computers, and up to 1500% faster on JiffyDOS-equipped systems
- Emulates 1541/71/81 disk formats for better compatibility, plus CMD Native partitions for greater flexibility
- Configurable as devices 8 through 15, plus front panel SWAP button for instant device 8 or 9 selection
- Optional Real-Time Clock (RTC) stamps time and date on all files and automatically sets GEOS clock
- Reads, Writes & Formats MS-DOS diskettes using Big Blue Reader utility program (available separately)

Special Offer Extended!

ONLY $159^{00}

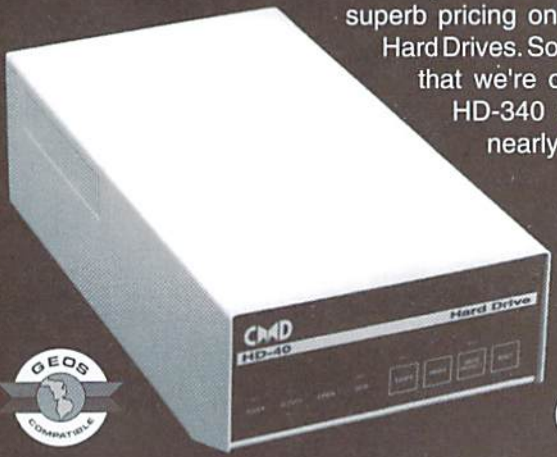
Optional Real-Time Clock (RTC) add $20^{00}

This offer valid only in U.S. and Canada through February 29, 1996. May not be combined with any other offers. Prices do not include shipping and handling. Please see our main advertisement on the opposite page for shipping prices and policies.

To Order Yours, Call 1-800-638-3263

If you thought the HD-40 was affordable before, just look at this...

Recently, during one of our routine searches for hard disk drives, we were offered a significant number of 40 MB Conner and Quantum SCSI hard disk drives at a price so low that we just couldn't refuse. Additionally, were offered some superb pricing on the larger Hard Drives. So low, in fact, that we're offering the HD-340 at a price nearly $1,500.00 lower than the original HD-200 back in 1990. If you've ever considered adding a CMD HD Series hard drive to your system, **now is the time to do it!** These outstanding prices will only last as long as our supply of these specially priced drives. So don't hesitate—add the power and flexibility of a CMD hard drive to your system today!



CMD HD-40 JUST $249^{00}

HD-340	**340 Megabytes**	**CALL**
HD-500	**540 Megabytes**	**$399^{00}**
HD-1000	**1 Gigabyte**	**$499^{00}**
HD-2000	**2 Gigabytes**	**$899^{00}**

Prices do not include shipping and handling. See our main advertisement on the opposite page for shipping prices and policies.

CMD Creative Micro Designs, Inc.

GAMES

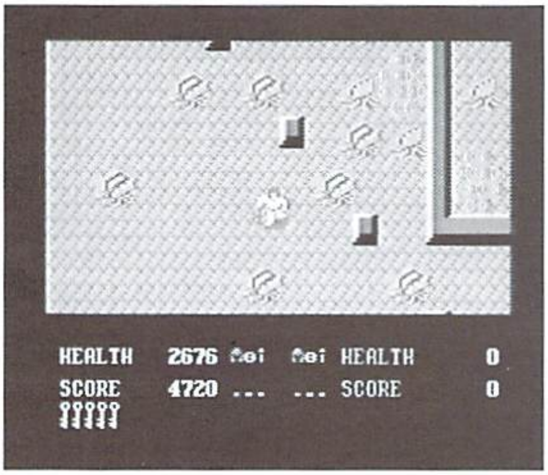
Demon Stalkers: Raid on Doomfane

Electronic Arts

I deliberately waited until Halloween to review Demon Stalkers: Raid on Doomfane. This adventure game from Electronic Arts puts you in a dungeon filled with gruesome creatures; perfect for the year's most haunting holiday!

You have been elected to rid your kingdom of Calvrak, the most dreaded monster in the land, who was awakened by the construction of the 100 level dungeon he now inhabits. You've been given an eternally-armed crossbow to aid you on your quest. Use extreme caution as you wander through the dungeon, because it's loaded with many other dangerous monsters and traps. Also, keep your eyes open for treasures, clues and keys.

Demon Stalkers can be played with one or two players. Before you can begin game play, the manual suggests that, to prevent any damage to the master disk, you create a "scenario" disk, which takes approximately a half hour to make. I felt that this was a major detraction from the game because like most people, I wanted to start game play immediately upon opening the package.



```
HEALTH   2676             HEALTH   0
SCORE    4720 ...     ... SCORE    0
```

Once you've created the scenario disk, be sure to give yourself a nice chunk of time to play. Demon Stalkers is so much fun that it's hard to tear yourself away; after all, there are 100 levels! If you manage to make your way through all of them, a construction kit is included allowing you to create all new mind boggling levels.

As you enter the dungeon, you are immediately greeted by a host of rats. The rats are the least dangerous of the many monsters. During your search for Calvrak, you'll encounter a variety of rats, snappers, dervishes, and ghosts. The darker the color of the monster, the more harm it can cause you and your health. Also, dervishes can rob you of your hard earned possessions. You need to discover the clues, keys, treasures, and magic spells scattered throughout each level to complete the game. In order to advance to the next level of the dungeon, you must survive the monsters, and/or solve the rooms puzzle, and then locate the exit steps leading down into the next floor of the dungeon. The monsters and puzzles become increasingly more difficult with each new level.

The first time I played Demon Stalkers, I made it to level 16, which consists of a series of paths filled with keys. Unfortunately, you can only hold so many keys. Once your pockets are filled, you can't pick up any more keys, and your path ultimately becomes blocked by them. The only ways to dispose of the keys are to unlock doors or run into a dervish who, unfortunately, will also steal away your health. Needless to say, I am still stuck on level 16.

The game screen looks much like every other role playing adventure game. The playing screen consists of a maze filled with monsters, treasures, etc., and occupies most of the top of the screen. The bottom part of the screen is your status indicator. Look here to see how much health you have left, the number of keys, points, armor, and other treasures you may possess.

The game is fully equipped with a save and restore feature. You should become a pro at using these commands, because unless you're an expert game player or have a lot of free time on your hands, the only way to finish Demon Stalkers is by regularly saving the game.

One feature I didn't like in the game is the fact that after completion of every fifth level you are asked a question. The answer can only be found on the "wheel" supplied with the game. I think for copy protection purposes one question per game would suffice.

Demon Stalkers does not have out-of-this-world graphics or an extraordinary soundtrack. In fact, there is very little music in the game at all. But you will instantly love this game for the multitude of challenges awaiting you on each level and the constant variety of interesting monsters. Be careful, you may never see the light of day again!

-Sherry Freedline

GAMES · GAMES · GAMES · GAMES · GAMES · GAMES
· GAMES · GAMES · GAMES · GAMES
GAMES · GAMES · GAMES · GAMES · GAMES · GAMES
· GAMES · GAMES · GAMES · GAMES
GAMES · GAMES · GAMES · GAMES · GAMES · GAMES

Alcon

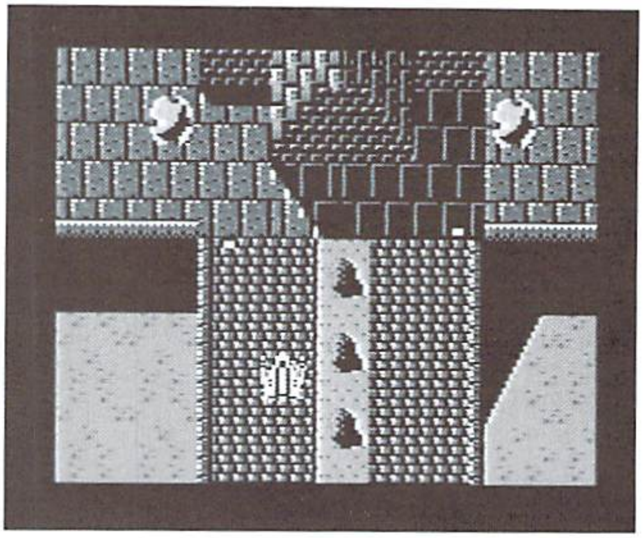
Taito
Available from ??

Many of my all-time favorite Commodore games are among those produced by Taito. Who can forget the experience of playing Arkanoid, Arkanoid II, or even Bubble Bobble? What special quality do these games possess that makes them so immensely enjoyable, even after owning them for so many years? Personally, I think it is the challenge these games provide, as well as their amazingly bright, colorful graphics and unique soundtracks that keeps me and many others coming back again and again. All of the Taito games I have played to date feature many levels of difficulty, making them very hard to outgrow or master.

Alcon is one Taito game I never had the opportunity to play. Since I have played other games from Taito, my expectations for Alcon were understandably quite high. And, just as I expected, Alcon did not disappoint me.

Alcon is Taito's venture into the land of space games. Alcon's game play whisks you ahead to the year 2059 to the planets Theon and Orac. In the year 2055, humans were forced to exile to Theon due to an alien invasion of their home planet Orac. Since this time, aliens have had total control of Orac and have been keeping some human hostages. Now, four years later, it's high time someone did something to end this alien reign and free the human hostages; thus, the Allied League of Cosmic Nations (ALCON) was formed.

The ALCON has elected you, Major Astor, to fly their most advanced starfighter, the SW475. The starfighter is capable of possessing a selection of seven different weapons/protection devices to aid you in your mission to liberate Orac: speed, shots, side (shoot from the side), wing (enlarge wings & firing range), bomb, laser, homing missiles, and shields.



The first thing I noticed is how the Alcon game screen scrolls from top to bottom, unlike the normal left to right scrolling common in most games. The game screen consists of your SW475 starfighter flying up your screen over various space-type landscapes. Of course, you'll frequently meet up with a variety of alien ships. The second thing I noticed was the challenging game play Alcon presents. Unfortunately, I found this game to be very difficult right from the start. I'm embarrassed to admit that it took me three attempts before I was able to knock off the first two aliens of the game. An explanation here will reduce my embarrassment and maybe prevent you from making the same mistakes I did. The first two aliens fly right at you, one on either side. If you concentrate on shooting both aliens, one of them will nail you with their incredible aim (they have homing missiles). I finally managed to get past them by dodging their missiles and flying to the right of the alien on the right side of my screen. Once I managed to get by these first two aliens, the game got a little easier. The next alien does not have a companion and is easily destroyed. The game continues in this manner.

Initially, the game furnishes you with three SW475 starfighters. You are awarded an additional starfighter after obtaining 50,000 points. Then, additional starfighters are awarded after every 70,000 points. Your starfighter does not come equipped with enhanced weapons and protections devices; these valuable items must be earned by destroying aliens. The death of certain species of aliens results in the award of a yellow star. You must fly over the star to collect the weapon or device it represents. The weapon/device is then activated by pressing the spacebar. However, use caution in using these aids. For instance, the wing function augments the width of your ship and its firing range. This can be good as well as bad since it also gives your opponent a larger target.

Alcon is indeed one of the most challenging space games I have played to date. Unfortunately, if your are an easily frustrated game player, Alcon is not for you. It takes a great deal of patience to master this game, but patience pays off. The stimulating graphics make this game a great diversion from everyday stress. I would like to know if anyone has discovered any cheat codes or has made it through to the end of this truly awesome game!

-Sherry Freedline

HARDWARE IN REVIEW

THE CMD FD-2000

Introduction

Reputation. This is one of the only things I can think of which is acquired easily, but not lost quickly. I have personally found this to be true. As an online Commodore resource, people think of me as someone who knows what I'm talking about. Boy, have I fooled them! But seriously, people ask me about very technical issues concerning Commodore computers, and for some inexplicable reason, I come up with (somewhat) correct technical information in responses. How it happens, I have ABSOLUTELY no idea. Maybe it's all in my name...

CMD is another group I think I must have fooled. Case in point: They sent me one of their CMD FD-2000 and asked me to do a technical review of it, claiming that they "appreciate my honesty and technical prowess." So, one of the drives arrives at my house, and my reputation dictates that I review it on technical merit. So here goes.

Physical Appearance

The CMD FD-2000 comes in four pieces. The purchase will yield 1 disk drive, 1 Commodore serial cable, 1 small 9 volt wall power supply, and one disk of utility programs. The cable and power supply are pretty ordinary, but the drive is distinct from other drives for the Commodore. The entire unit is painted black, which creates a striking appearance for the unit. (Personally, I happen to like black as a technology case color, but that's another story.) The drive appears to be a standard Teac 3.5" microfloppy drive mechanism, and is the traditional IBM clone beige color. The front panel contains 1 switch and 3 LEDs in addition to the customary manufacturer logo and drive model number. The LEDs are labeled "Power", "Activity", and "Error". Turning the unit around, we see the back of the unit is pretty simplistic as well. Two serial bus connectors are provided, as is a connector for the power supply and a power switch.

The drive can be plugged into the serial bus chain of devices at any point, although I would suggest locating it towards the beginning of the chain. Anyone who has ever plugged in a serial device will find connecting the drive a breeze, as CMD uses standard connectors for the serial bus hookups. The drive was up and operational in a matter of minutes, awaiting my grueling tests.

Also supplied is a manual, which covers all of the features of the FD-2000, how to use the supplied utility disk, and how to use the FD drive with GEOS. Even though as a man, it is against my principles to read a manual, it is very clear and well-written, presenting all of the commands available and what each does. Since the drive emulates the Commodore DOS functions very well, users will be able to get things going quickly. Still, one should read the parts of the manual detailing formatting and partitions, as these are new ideas to many Commodore users.

Features and Operation

The CMD FD-2000 disk drive packs a lot of stuff into a pretty small case. Below are just some of the features the FD-2000 sports. The unit can do the following:

- Format 800 KB 1581 style disks
- Emulate the standard Commodore DOS
- Commands, except BACKUP (found on early IEEE drives)
- Work with GEOS and CP/M
- Format 1.6 MB high density (HD) disks and utilize them with the CBM computer
- Logically "split" disks into multiple 1541, 1571, and/or 1581 partitions
- Logically organize partitions into subdirectories
- Allow time and date stamping of files with the Real Time Clock (RTC) option
- Support burst commands on the C128
- Automatically provides speed enhancement when used with Jiffy-DOS enhanced computers

One feature that bears special mention is the GEOS compatibility. Without any special drivers, the FD drive will work with GEOS as a 1581 drive. GEOS can also take advantage of 1.6 MB disks by formatting them as two 1581 partitions. When used with GateWay, the replacement for GEOS' DeskTop, the user can use and abuse native 1.6MB disks in the FD. Since GEOS enjoys much popularity, the FD's GEOS compatibility is a welcome sight.

Since the test computer has JiffyDOS installed, the computer and drive could communicate at JiffyDOS speeds, since the FD series of drives comes pre-installed with JiffyDOS. My first test involved a 1581 formatted disk. I loaded data and programs, performed utility chores like copying scratching files, renaming files, and initializing the disk. The only differences in operation from my 1581 were much faster access (my 1581 is not JiffyDOS equipped), and some operations done within the drive itself, like appending a file, were also faster. This bodes well for the unit, as it did almost everything my 1581 does, only more quickly. The one thing this drive didn't do that my 1581 does is corrupt disks. I have a 1581 with the Western Digital 1770 drive controller IC, which Commodore offered a replacement for many years ago. So, in this respect, what the FD-2000 didn't do was welcome.

The next test involved high density floppy disks. I grabbed a spare blank disk (It happened to be an AOL disk that came in the mail that day) and formatted it on the drive in the usual way. I then loaded a directory and saw that only 3160 blocks were available. This bothered me, as I could format that much space in a 1581. So, before wrapping the unit back up and marking it as non-functional, I looked in the user's manual entry for formatting and found that in order to format a fresh HD floppy for maximum space, you must specify a ",HDN" after the drive ID. So with book in hand, I followed the instructions and the drive went through its format process once more. After the drive stopped, I asked for a directory and found 6336 blocks available for use.

With the newly formatted HD floppy, I proceeded to work my way through the standard DOS operations. I saved files, loaded them back in, performed a save-with-replace, scratched some files, copied a few files around on disk, and then verified a file. All operations performed as expected, with no surprises or mishaps, and all operations were completed somewhat faster than the same operations on a 1581. Formatting still takes the most time, as I suppose it should.

After the standard DOS commands were exercised, I tackled the CMD specific extensions. I'm told these commands are similar or identical to those used on other CMD devices, so not everyone will think these are as new as I do. For those who don't know, some of the most used ones are listed in Table 1.

TABLE 1

Command Type	Description
Format Extension	,(format) after ID in regular (N)ew command dictates format density & partitioning options
Drive Number Extension	Partitions can be numbered. (LOAD "2:JIM",8 loads from partition 2)
New	R-H command allows changing directory header name
New	G-P command returns info on specified partition
New	C-P command changes partitions
Directory Extension	$=P views partitions, $=T views time
New	R-P renames partition/drive name.
New	MD command creates CMD style subdirectories.
New	CD moves between CMD style subdirectories
New	RD removes a CMD subdirectory
New	L locks (or unlocks) a file
New	T-RA reads date and time from optional on-board clock.
New	T-WA writes new date & time into clock

All of those commands functioned as described in the owner's manual, providing easy access to all of the extension and additions in the FD drive. Some of the commands, like G-P and C-P, have a version for direct mode usage and a version for program usage, making development of FD-aware software even easier.

1
0
Manufactured in the United States by:
Creative Micro Designs, Inc.
15 Benton Drive, PO Box 646
East Longmeadow, MA 01028
POWER
DEVICE
SERIAL
SERIAL

Rear view of the CMD FD-2000.

The subdirectory commands are what really make the FD drive shine. Before, Commodore users had to group files on separate disks or live with all their files in one directory. Now, with the MD/CD/RD commands, users can organize files more effectively on the same disk. To explain, say a user has 9 files (f11, f12, f13, f21, f22, f23, f31, f32, f33) and wants to group all f1* files into one place, all the f2* into another place, and all the f3* files yet another place. On the FD, the user could create three subdirectories (dirf1, dirf2, dirf3), and place each of the files into the associated directory, like so:

dirf1:	f11
	f12
	f13
dirf2:	f21
	f22
	f23
dirf3:	f31
	f32
	f33

When the user requests a directory listing of the disk, all he/she sees is the directory names. After a CD command is issued, a directory listing will show the files in that directory. The files are all on the same disk, just more organized. Directories can be nested and can be named just like regular files.

Since a drive may have more than one partition, the FD allows partitions to be named and renamed. However, a partition name is NOT the same as the top level directory name returned when a user does a directory listing of that partition. This will confuse some, so it is best to try these commands and view the results. Also, the same applies to subdirectories. The name of a subdirectory as seen from a parent directory does NOT have to be the same as the name displayed at the top of the directory listing when inside that directory.

C128 users will be happy to know that one can autoboot with the FD drive, and GEOS users can do so as well, although a special program must be used to make GEOS autoboot disks.

To increase the usefulness of the FD with those programs which will only recognize drive #8 or drive #9, CMD has provided a SWAP button on the front of the case. Hitting this button once will temporarily change the unit's device number to 8. Hitting the button twice changes it to 9, while hitting it 3 times will change it to the default device number. This operation can also be done via software with the "S-8", "S-9", and "S-D" commands.

One of the first things that users who purchase the RTC option for the FD should do is set the clock. Setting and recalling the time and date is easily accomplished from BASIC and can be done in PETSCII, BINARY, or BCD. Time and date stamping of files is useful to help track dates of the last update on files, as well as which files are the newest revisions. Also, RTC users who have GEOS should take the time to install the supplied files on the utility disk that will automatically set GEOS' clock with the clock in the FD.

Compatibility

One cannot talk about a third-party Commodore peripheral without at least mentioning compatibility issues. And so it is with the FD-2000. By now, most people know that Commodore drives all have their own CPU and memory map—a drive computer system. Over the years, programmers and others have taken advantage of that extra CPU and RAM in the drives to realize fast load routines or implement copy-protection schemes, among other things. When 1541 clones were prevalent, a successful clone needed to duplicate not only the physical drive characteristics, but also the computer system inside the drive. Commodore's 1581 was a different story, as fewer software titles were released for the 3.5" disk format, and very few titles used any of the features of the internal drive computer system. The 1581 was and is mostly used as a data storage drive. So, 1581 compatibility can be viewed in two different ways.

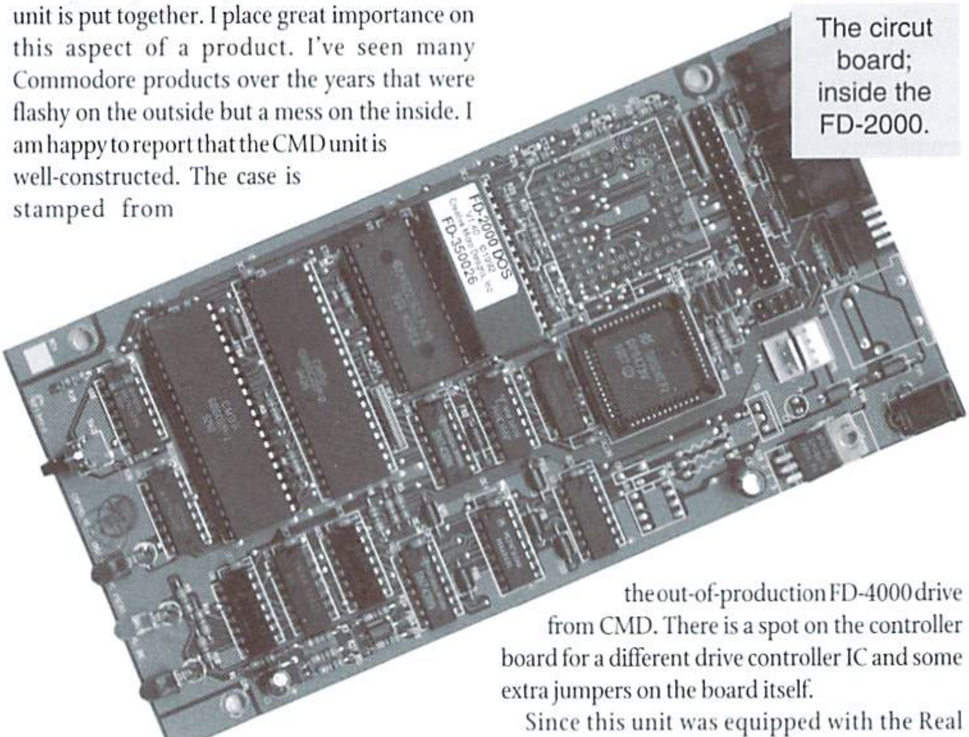
If one views compatibility as the complete duplication of the 1581, the FD-2000 fails, as the internal drive computer system is radically different. If, on the other hand, one views 1581 compatibility as duplicating the 1581 disk format and layout, the FD-2000 shines. In practice, one has to take both issues into account. For people who want to use the FD-2000 as a data storage unit, the FD-2000 will work admirably. For others with copy-protected 1581 software, speed enhancement software, or non-JiffyDOS hardware based speed enhancement systems, the FD-2000 will have problems.

As the FD literature states, the drive can also emulate 1541 and 1571 disk formats, called emulation partitions, but note that this implies only the disk format and layout will be emulated. Running copy-protected software out of a 1541 or 1571 partition is an iffy proposition. In addition, attempting to use any 1541 or 1571 undocumented ROM routines will most likely fail.

One possible compatibility issue with all uses of the FD-2000 involves programs that attempt to "discover" the drive type and capacity. For some such programs that were developed prior to the introduction of the FD, the "discovery" code might fail if the code relies on undocumented ROM routines or special ROM/RAM locations in the disk drive memory.

Internal

As is customary with any product I receive, I don't give my seal of approval until I see how the unit is put together. I place great importance on this aspect of a product. I've seen many Commodore products over the years that were flashy on the outside but a mess on the inside. I am happy to report that the CMD unit is well-constructed. The case is stamped from standard gauge sheet metal, and is covered with what looks like a baked-on, black finish.



The circut board; inside the FD-2000.

I took the FD apart for an internal look the day after my initial tests. The printed circuit board is cleanly laid out, with standoffs to maintain distance from the case and appropriate masking and screening applied (masking is that green non-conductive stuff sprayed on boards, and the screen is the white lettering and outlining. Screening is unnecessary, but masking protects against shorts and errant electrical behavior.) I noticed only a single hand-wired trace on the board, which appears to ground out an unused oscillator pin. The almost non-existence of rework on the board indicates a well-thought-out design, which means fewer problems for the end user.

As for I/O, all connectors are securely mounted, which will come as good news to those who regularly plug and unplug their systems. The SWAP button on the front is securely mounted the board, and protrudes through the case to rest against the back of the front bezel decal work. The LEDs are mounted in front of cut-outs in the case, and show through "windows" in the decal.

While inside the case, I noticed a cutout on the rear of the case which could accommodate a DB-25 connector or similar. Possibly included for future expansion needs or to adapt the drive case for other uses, the hole is covered by the decal work on the rear. Also, it is apparent that the controller board used in the FD-2000 is also used to control the out-of-production FD-4000 drive from CMD. There is a spot on the controller board for a different drive controller IC and some extra jumpers on the board itself.

Since this unit was equipped with the Real Time Clock option, a Dallas Semiconductor RTC module was installed in the ROM socket. For units without the RTC option, that module will be missing. The implementation of the RTC module makes adding an RTC option to a FD-2000 unit a snap, and directions are printed in the manual for this purpose.

For the programmer, CMD has included information on the FD memory map, the layout of the drive tracks and sectors, the important functions and memory locations in the FD ROMs, and other technical information. Such openness is characteristic of CMD and should be useful to any programmer needing in-depth knowledge of the CMD drive.

Issues

As you might have already guessed, I am very impressed with the FD-2000. CMD has produced a fine piece of hardware, and packed it full of features. When I started looking at the drive, I was expecting a unit that would format 1581 disks and a "super-1581" disk, with twice the capacity. I was not expecting 1541/1571 partitions, Native mode subdirectory support, and a SWAP button. Of course, I expected JiffyDOS support, which is included as well. Nonetheless, I have the following concerns and requests:

I'm confused by the ability to name a partition/subdirectory from one place, and then assign a different name to that same partition/subdirectory in another place. I was expecting to change the name of the partition and see the change reflected in a top-level directory listing. Such is not the case. For partitions, this is not a huge problem, since few will request partition listings, but the ability to give a subdirectory two different names just adds unneeded complexity to the already foreign construct of subdirectories. I suggest either not renaming commands (the initial creation of a partition or subdirectory assigns the same name to both places), or making very sure that the names are in sync with each other.

The SWAP command/button takes control of the serial bus and issues commands to other drives to perform its trick. (Example: if the FD is device 10, and the user hits the SWAP button, the FD changes any device 8 on the serial bus to device 10, and then changes itself to device 8.) Users with more compatible drives will see no problem, I'm sure, but I wonder about less Commodore compatible drives (Indus GT, MSD SD-1 and SD-2, etc). Will the SWAP command work with them on the serial bus?

If CMD ever upgrades the DOS for the FD series, I would like to see an option to format and utilize MS-DOS disks. I know Big Blue Reader and other utilities exist to read and write such disks, but nothing can compare to using MS-DOS disks with native Commodore DOS commands.

For users who opt to install the RTC option after initial purchase, be aware that, as part of installation, the drive mechanism must be removed. There isn't any problem with that, but be careful, as the Phillips screws holding it in are of aluminum, and can strip easily, rendering installation much harder.) Since CMD has made it possible for users to do the installation themselves, a switch to a screw that is more forgiving to typical user tools would be beneficial.

Through no fault of CMD, some BASIC 7.0 disk commands in the Commodore 128 work less than ideally with the partition concept of the FD. Evidently, BASIC 7.0 will only accept drive 0 and 1 as correct choices for drive commands, meaning that issuing COLLECT,D2,U10 will NOT validate the second partition on device 10. C128 users will need to re-acquaint themselves with the BASIC 2.0 equivalents of the BASIC 7.0 disk commands to get around such limitations.

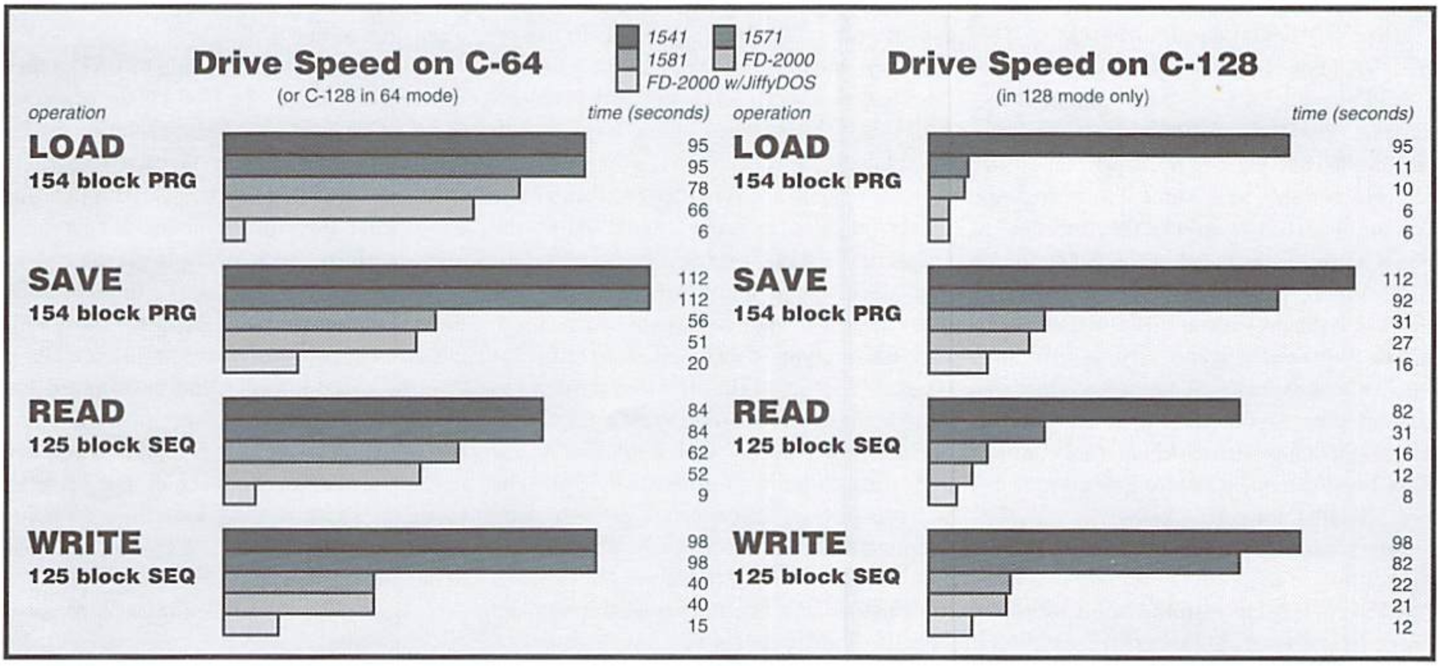
Even as well written as the manual is, it will probably intimidate some users. There are just so many features to describe! Thankfully, CMD supplies a utility disk that takes most of the work out of using the drives.

Conclusion

The CMD FD-2000 realizes the promise of a disk drive that betters the venerable 1581. The ability to format High Density disks, work with GEOS, segregate disks into 1541/71/81 partitions, provide faster operation, and maintain the correct time and date make it immensely useful for the serious Commodore user. As for programmers, the rich set of commands, myriad of options, and overall stability of the drive makes supporting it in programs a rather easy task.

However, the drive is not utopia. The drive will best fit software developers, users who have the need for more removable storage capacity, and those searching for a better alternative to the out-of-production 1581. My only hesitation comes from the physical limitations of the FD drive. Its 1.6 MB capacity is only so much, and users needing more and not needing the ability to transport files should probably look into a CMD HD series hard drive instead. Of course, the FD makes backups of an HD easier, so the above caution may have no bearing. But hey, what do *I* know?...

- Jim Brian



BOOKWARE IN REVIEW

Three titles that may ease your entry to the Internet

The Complete Idiot's Guide to the Internet
The Complete Idiot's Guide to Unix
Everybody's Guide to the Internet

The Complete Idiot's Guide to the Internet
Second Edition, by Peter Kent, © 1994 Alpha Books (367 Pages, $19.95)

The Complete Idiot's Guide to Unix
by John McMullen, © 1994 Alpha Books (375 Pages, $19.99)

I really hate being called an idiot, but I hate feeling like one even more. After discovering that some on-ramps to the Internet are different from others, I found myself needing to learn about accessing the Internet via a UNIX shell account. I felt the need to find a few sources of information.

After a brief hesitation due to the titles of "The Complete Idiot's Guide" books, I finally decided to read a couple to see if they could offer me guidance. The books aren't written totally from the IBM-PC users perspective, which make them suitable for my use, since I'm using my Commodore 128 to access the Internet. A Windows disk is included in the Guide to the Internet book, which I promptly removed and reformatted for use with my FD drive. Both books include a tear-out reference card, but the only one I've actually torn out to use was from the Internet guide. The books and the reference cards are a bright orange with dark blue in color, which make them extremely easy to find even in the most cluttered situations.

Both books use clear examples and a good dose of humor to get the information across. The Internet guide assumes some working knowledge of how to load a term program and how to dial out, but makes very little in the way of assumptions about which type (there are a few) of Internet on-ramp you're using, whether it's dial-in to a UNIX system, a menu driven system, or for PC users, a direct connection. The UNIX guide at times makes the assumption that you are using UNIX on a system which is available to you locally, such as your place of work. It spends a chapter or two on printing files, something you wouldn't need for use when using a UNIX shell account to access the Internet, but aside from this, most of the book discusses commands which are usable by those of us who are accessing a UNIX system with a remote computer, such as when I use my Commodore to access the Internet. As well, the Internet guide devotes a chapter or two on Windows, but luckily it didn't devote the entire book to the subject, and the majority of it deals with things the average Commodore user will find of use.

In each book, each chapter ends with "The Least You Need To Know", summing up the chapter as a list of information, which can be a very quick way to skim through the book. If there are other methods which can be used, this information is included. Four different types of "additional" information are given in various places of the book in the margins. These are highlighted in grey, with a different type of caricature graphic to go with it. One type explains phrases and shows a geeky looking character I call the "Geek Speak" guy, another has an older fellow who gives background information (the book actually says you can skip this fodder unless you're interested). A scared looking kid at a computer gives help for when things go wrong, and an intelligent looking woman gives notes and tips to make things easier.

The Complete Idiot's Guide to the Internet has subjects such as: "Your First Trip to the Internet", "Menus and Shells, Oh My!", and "A Unix Survival Guide" and then gets to more detailed subjects such as EMail, Newsgroups, FTPing, World Wide Web, Archie, Veronica, and WAISs. It includes a list of "Things to Do, Places to Visit", as well as the complete archive of "Speak Like a Geek". The index is accurate, something that I've come to depend on to quickly find information which isn't included on the reference card, such as the correct syntax to use when uuencoding a file.

The Complete Idiot's Guide to UNIX takes the time to explain why UNIX is the way it is, and then explains why this information can be useful to the average user. Written with wit, it covers the basics and then deals with tips and tricks, using text editors, and mastering some of the mysteries, such as how to get the system to do more than one thing at a time. Most of the information in it is suitable for those who must use UNIX to access the Internet, including ways to make it easier and customized for personal use by changing specific files. It also has a chapter devoted to E-Mail and another devoted strictly to the Internet, but overall I found most chapters had something useful to remote users such as myself. I found the book to be quite beneficial.

- Gaelyne R. Moranec

Everybody's Guide to the Internet
by Adam Gaffin, ©1994 Electronic Frontier Foundation, published by The MIT Press, Cambridge, MA and London, England (211 Pages, $14.95)

Not everything good must come in a large, brightly colored package. "Everybody's Guide to the Internet" is a relatively small book, slightly larger than a 5x7 photo in size, but packed with useful Internet related items, and is written by someone who has definitely had experience with the various Internet software, such as the different types of mail readers.

It begins with information about computers, modems and terminal programs, most of which can be skimmed over since it covers only MS-DOS and MAC computers. Some of the information, such as the suggestion to use VT-100 emulation is useful if we simply forget the type of computer being discussed. The rest of the book covers E-Mail, Newsgroups, Mailing Lists, Telnet, FTPing, Gophers, WWW, advanced E-Mail, then spends time explaining what MUDs and IRC's are (with the comment that they're more fun than they sound). It also gives listings of commands, and examples are given from a practical point of view.

The book includes two appendices, one explaining the terms used, and another with information about joining the Electronic Frontier Foundation, finishing with and a very comprehensive nine page index. While not as flashy or bold as "The Complete Idiot's Guide to the Internet", it offers very straightforward information. While it's not quite as humorous, it does have it's moments, such as when the difference is explained between ELM and Pine, which are UNIX programs used to read and reply to mail with, the book covers both subjects individually and refers to the latter as "an even better way". These two programs are barely mentioned in "The Complete Idiot's Guide to the Internet" but are given comprehensive coverage in "Everybody's Guide to the Internet", including a list of the basic commands for both ELM and Pine, as well as the pros and cons of each.

Other Internet/UNIX software is given the same treatment, making this book a very useful reference. It may seem odd than an electronic version of the book is readily available, but there is a distinct difference between having a text file and having a book you can refer to away from the computer. The frequently updated electronic version of this book is available via anonymous FTP to ftp.eff.org, in the pub/Net_info/Guidebooks/EFF_Net_Guide/netguide.eff (please note, this is case sensitive) directory, or you can send an Email message to info@eff.org. The electronic version is entitled "EFF's Guide to the Internet".

- Gaelyne R. Moranec

Mastering the *DON'T DO IT* Key

by Jim Butterfield

The RETURN key is sometimes called the ENTER key; sometimes I call it the DO-IT key. That's because every time I press it, the computer does something. But I'd also like to tell you about another important key: the DON'T-DO-IT key.

This key is not documented in any Commodore manual, but everybody knows about it. Everybody, that is, except YOU... it's so well-known that nobody thought to tell YOU about it.

The DON'T-DO-IT key (I'll tell you where to find it in a moment) does the reverse of the do-it RETURN key: it causes the cursor to leave the line it's working on without doing anything else. The cursor just hops to the next line.

Many of us enjoy doing nothing. But why would you want the computer to do nothing? There are two basic reasons: you might have muddled up a line you're typing; or you might just be doodling on the screen. Either way, you don't want the computer to try to do whatever you have just typed.

Let me give you an example: you want to use the screen to type a message. Perhaps the message is DON'T TOUCH THIS UNTIL YOU HAVE CLEANED YOUR ROOM, or BACK IN 10 MINUTES, or even FEED THE CAT. If you typed such a message and then pressed RETURN, the computer would respond ?SYNTAX ERROR, which is its own way of saying it doesn't know how to feed the cat. You could rename your cat SYNTAX ERROR, but that's just avoiding the problem. Best to type the message, and then press the Don't-Do-It key; the computer will know that you don't really want it to feed the cat, and move along to the next line.

I'd better tell you how to find that key before you get too impatient: it's Shift-RETURN. That is, hold down the shift key and press RETURN and your computer will move the cursor to the next line without trying to do anything else. By the way, if you ever touch those "other" machines (PCs, Amigas, and the like) the equivalent key combination there is CNTRL-X. But let's stay with our neat little 8-bit Commodore jobs, where Shift-RETURN is the way to do nothing.

So after you type FEED THE CAT, press Shift-RETURN, and then you can type in other tasks, such as WATER THE LAWN.

The other situation where Don't-do-it, Shifted-RETURN, is useful—almost indispensable—is when you're typing in a program line and somehow mess it up. The most common way to mess up a line is to get into "programmed-cursor" mode. That's sometimes called "quotes mode", because one way of getting stuck there is when you're typing stuff in quotations. When you try to move the cursor around, it won't go: instead, you get those peculiar reversed characters. They are wonderful when you want them, but horrible when you don't. If you have a wrecked line, RETURN would put it in your program, but Shift-RETURN abandons it on the screen and lets you try again.

Graphic Interpretation

by Steve Vander Ark

GETTING STARTED WITH GEOBASIC: PART 3

In the last issue, we started writing our program. Here is the rest of that program listing.

```
400 @quit
410 CLS
420 END
```

(That was easy. When a geoBASIC program ends, the deskTop is automatically reloaded and run.)

```
500 @colChoice
510 DBSTRN"Enter color
number (Ø-15)",c$
```

(I don't have to use the dialog box editor for this one; the DBSTRN command automatically creates one in order to accept user input and places the text string inside.)

```
52Ø IF VAL(c$) > 15 THEN 51Ø
```

(This is standard stuff in BASIC, trapping possible erroneous input.)

```
53Ø SETCOL VAL(c$)
```

(We are actually setting the background color here. We'll draw our colored rectangle using that color. The foreground color, which we would use if we wanted a frame around our rectangle, would require some additional math. As it is, the foreground color will remain black, which we set it for back in line 320.)

```
54Ø RETURN
6ØØ @pattChoice
61Ø DBSTRN"Enter pattern
number (Ø-33)",p$
62Ø PATTERN VAL(p$)
63Ø RETURN
7ØØ @drawRect
71Ø DBSTRN"<C>olored or
<P>atterned rectangle?",r$
72Ø IF r$="C" THEN 74Ø
73Ø IF r$="P" THEN 77Ø
74Ø GOTO 71Ø
75Ø COLRECT 2Ø,12,35,22
```

(The color used for COLRECT is the background color we set with the SETCOL command.)

```
76Ø RETURN
77Ø RECT 2Ø,12,35,22
```

(The pattern used for RECT will be whichever one was set with the PATTERN command.)

```
78Ø RETURN
8ØØ @drawLine
81Ø LINE 4Ø,3Ø TO 2ØØ,15Ø
82Ø RETURN
9ØØ @sayHi
91Ø WINDOW Ø,5Ø,1ØØ,1ØØ
```

(We're opening a new text window here. If we wanted to get fancy, we could use RND to place the text randomly around the screen, but for now we'll just use the same spot each time.)

```
92Ø PRINT"Hi there!"
92Ø RETURN
1ØØØ @scrnClear
1Ø1Ø DIALOG "clrD",A
```

(The dialog box editor has pre-defined buttons which return certain numbers when clicked. We'll test for a "YES" button, which will return a 3, and other wise assume the answer was "NO.")

```
1Ø2Ø IF A=3 THEN 1Ø4Ø
1Ø3Ø RETURN
1Ø4Ø SETCOL 15
```

(You never know what the color may have been changed to, but we want a filled rectangle using the background color, light gray.)

```
1Ø5Ø RECT Ø,25,319,199
1Ø6Ø RETURN
11ØØ @about
111Ø DBSTRN"Program by Steve
Vander Ark. Press any key
and press 'return'",A$
```

(I could have defined a nice dialog box to display my name, but this will work too.)

```
1120 RETURN
```

So just what does this program do? Nothing particularly useful. But it does demonstrate some of the basic techniques you will use when you write a geoBASIC program. There are plenty of commands I haven't even mentioned, including ones for sound and disk access. With a little imagination and some time trying things out, you could easily create very exciting GEOS programs this way.

GEOS Dreams...

Most of the programs I've written over the years with BASIC have been simple games and the like. I enjoy the challenge of writing programs and wish I had more time to spend at it. But I really don't consider myself a programmer at all, especially when I start talking to people like Maurice Randall and Doug Cotton. One area that I would like to explore more in depth is disk access using geoBASIC. I have a feeling, though, that I'll never get to it.

There are plenty of folks out there, however, who have put in the time and know the tricks and shortcuts for using geoBASIC. I used to correspond with a few of them on QLink; people who had written truly powerful, exciting programs. I have since lost track of them, I'm afraid. I would really like to get a hold of a few more of these experts, so if you're one of them, why don't you share some of your expertise with the rest of us? I would be happy to include geoBASIC tricks and tips in this column. Please contact me; you'll find addresses in the box at the end of this column.

If you're stuck for ideas of what programs to write, let me offer a few suggestions. These are ideas that I've had over the years for programs that I think would be perfect for the GEOS environment. Give them some thought. I'm sure some of them are outside of the scope of geoBASIC, but what the heck, I'll toss them out for you to consider anyway. And if you've got some great program ideas of your own you'd like to suggest to the experts out there, why not send them to me for this column? Meanwhile, here are a few ideas of mine:

Multimedia Authoring Tool

This is a natural for GEOS. The point and click interface is made for this kind of thing, and the Commodore computer is capable of all kinds of multimedia output. This program would allow even a novice user to put together a multimedia presentation with clickable buttons on full-color graphic screens and all manner of music, sounds, and even animation tied in. The authoring tool itself could use a metaphor similar to HyperCard for the Mac, where the various elements are stitched together all with the mouse and a variety of attribute dialog boxes. The program should be able to incorporate SID music files, RAW digitized sounds and voices, sprites, high-resolution and medium-resolution screens, and animation files such as those created by Album Animator. How would this be used? I'd use it in my classroom, letting the kids create displays about their science topic, for example. Some people might use this to create store window displays or informative presentations. A lot of people, myself included, would just have fun playing around with it.

Web Browser

Yeah, right. I know, browsing the World Wide Web takes a lot more speed than your average Commodore can handle. Those detailed graphics take a while to come in even on my Pentium at 14.4 baud! What can anyone hope to do with a Commodore? The answer to that question is absurdly simple. You just browse! Maybe you won't see all the pretty pictures, but that doesn't mean you'll miss everything. As a matter of fact, I've recently been turning the images off when I roam the web using Netscape on my PC. They may look nice, but at the slow rate them come creeping onto my screen, I'm not so much browsing as stagnating. Hey, if I'm not going to be looking at the pictures anyway, why can't I use my Commodore? Surely it could interpret HTML, the programming language of the World Wide Web, and display a formatted text page. Then with GEOS we could point and click our way around. In other words, we could use "browse" as it's real meaning!

I know it isn't quite that simple. I realize that there are scads of little obstacles to overcome before a Commodore running GEOS accesses the Web in anything other than put text mode. But I also know that the Web is becoming a major form of communication, growing larger and more exciting every day. I don't want GEOS and the Commodore to be left behind.

Some Really Cool Games

Now, I do love geoTiles and other strategy games for GEOS. I love to pop up a game of blackjack every now and then. But I'm talking about other kinds of games here. Where are the graphic adventures like the Bard's Tale series? Where are the war games and the flight simulators? And speaking of simulations, where is the GEOS version of SimCity? Any of these types of games could work under GEOS. We need more games like Cluster Wars along with our Tetris and Solitaire.

Connections

This isn't one program; it's more of an idea, really, but one which will translate into many different programs. I'd like to see connections in GEOS for all sorts of devices, including CD-ROM drives and scanners. These types of devices are becoming the norm in computing these days, and the Commodore should take advantage of them. Here's an example: I use a small, hand-held computer called a Newton in my classroom. I have to trot on down the hall to the library to hook it up to a desktop computer in order to store information or load new programs onto my Newton. But there is absolutely no reason why this Newton couldn't communicate with my Commodore instead, storing data in a geoFile database, for example. That Newton would become quite a nifty peripheral for my 64, wouldn't you say?

And So On...

There are always new and exciting ideas coming along in computing. Bill Gates and Company would have you believe that the way to computer happiness is through ever bigger and more powerful desktop machines. Many of the new computer users these days are buying their first machines on which to run multimedia software for the kids. The Internet is now referred to by commentators as a full blown craze in this country. All of this new high-powered, high-tech stuff is great, and I would love to see the GEOS user tap into some of it. After all, you don't need new equipment every six months to be innovative and on top of things, you just need software that takes you places and let's you connect. I see GEOS as having that potential.

Maybe all those big ideas won't come through geoBASIC. But if you've got ideas and a hankering to get started with programming, geoBASIC isn't a bad place to start. Who knows where it might take you?

Addresses for Steve Vander Ark

E-Mail on the Internet at:
stevevark@aol.com.

You may also send letters to:
Steve Vander Ark
6730 Brad Ave SE
Grand Rapids, Michigan, 49548.

The SysOp's Corner

By Max Cottrell

SO YOU WANT TO RUN A BBS?

Submitted For Your Approval...

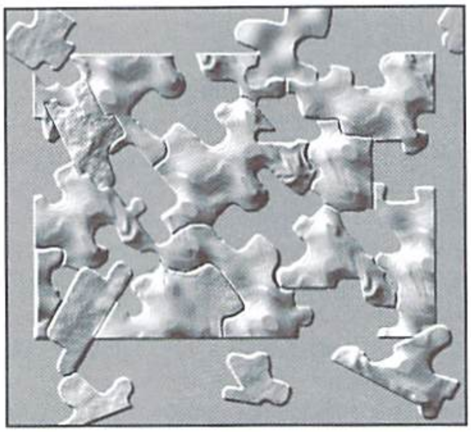
Many of you Commodore users out there realize that with the right ideas and set up, you can have a successful business based on a Commodore 64. But other people are so used to the IBM and Macintosh platforms that they forget about the smaller, older ones. When they are confronted with the existence of them, they are momentarily amazed; jolted out of their mundane, overproductive high speed programs to a supposed bygone era of small, inexpensive, yet useful computers.

So it has been with me. I am the owner and operator of a small online newspaper/support BBS in Ypsilanti, Michigan called "The Ypsilanti Community Information Service" (YCIS). My BBS/newspaper has been featured on the front page of the local newspaper and in five other articles, including an editorial about the system. I've been interviewed twice by a local radio station, and the Freenet which is starting in this area has asked me to help design the system and do graphics for them.

Running my BBS takes a lot of time and energy, but it has also taught me a lot about running my photography company. I learned some important lessons along the way, which is the basis of this column. I'll touch on some subjects here and then expand on them in future issues.

Publicity

When you start to set up your system, you will want to keep in mind that the most important thing to decide is what the theme of your system will be. Publicity goes hand in hand with that. The careful planning you do when setting up will mean little if you have no attention in the media. I suggest looking around your community; read the local paper and see if you can pick out what the interests are in your area. Is it community events, sports, debates, or political disassociation? I doubt if you need to make a decision right away, so take your time. Dual themes are okay if you can do it. My focus is local news (with the help of city hall... more on that another time), and support for my photography company.



Menu Design

Here's where the software that you choose comes into play. I did several weeks worth of research before I settled on the software I currently use called "Does The Job". Most will allow you to create whatever you want in the way of menu design. Color 64 will do it for you automatically, Omni 128 (which requires a 128 to run), needs to have menus created before you can start.

Does The Job has no real menu system, but allows you to create whatever kind of menu you want. I have found that after the 1,000-plus log-ons I have had since starting the YCIS, that menus are where it's at. What is 'it'? A user interface. That is what a menu is. It is the face of your system. The files may be the personality of a BBS, but the menus are the face that the public sees. Too many colors, too much writing, difficulty in getting to the menu will add a feel of 'klunkiness' to your BBS. When I design a menu, I start simple. No ANSI colors, no wacky or cute entries, just a screen clear and the selections. Then I log onto the system and see what it looks like from as many platforms as possible. I always log on with my C64 and my IBM to see what it looks and performs like online. Then wait for feedback. Let that decide where you go from there.

Naming the System

Here is where publicity, advertising, and the theme come into play. The online newspaper I run is community based, so I named the BBS 'The Ypsilanti Community Information Service'. If the theme were chess for example, I might have named it 'Chess City USA'. Your imagination is the driving force.

I can't stress enough the importance of two things: First, make sure that the name you choose is not already taken by another business, and second, make it something you like and can live with, and which is descriptive of the system you are running. But let me add a twist here. What if you decide to change the theme if it turns out that nobody is interested in what you are offering? (It might not happen, but be prepared.) Don't make your title too narrow; I suggest making the name generic. Here in Southeastern Michigan, we have BBS names like "HAL9000", and "Motor City BBS". These systems can all change themes without changing the name.

Advertising

Put your system everywhere you can; on other BBSs, in classified ads, door to door flyers, telephone poles, computer stores, and in the all-important press release. These are the least expensive avenues you can take. Start cheaply and see what happens. Continue if it generates interest, or try something else if it doesn't, and see what happens next. I was able to get over 40 new users by posting a flyer in a local computer store, which had an area where you can put flyers or advertisements. Advertising takes time to generate the interest that will get users online, so don't be disappointed if you get users who don't call back. As always, patience and feedback from the users is the key, as is cost.

Press Releases

Go to a book store or the library and find a book on starting a small business. Look at the press release examples. Copy them. Make your BBS sound like it is the greatest thing since peanut butter! Ham it up and write it in a friendly manner. Show it around and ask your friends what they think. Then send it to every media outlet that deals with small news items. Sending it to CNN makes little sense, but sending it to the local radio station makes all kinds of sense! Post it on other BBSs, send it to newspapers, and send it to stores that might be interested in getting involved. This is free advertising. If you do it right, you might be on the front page!

Modems

Ah, here is my favorite subject. What good is a car without an engine? A bicycle without a chain? A BBS without a good modem? I think you see the connection. However, this can get expensive very quickly. You don't want to start out at 300 baud, and there is probably no need to have 28,800 baud either. Most systems still use 14,400. Error correction, speed matching, compression, and DTR detection all are important.

Recently, I switched from a Aprotek 2400 minimodem to two different state of the art 14.4 modems in one week. I started out with a US Robotics Courier, and had some error correction problems with it, so I switched to an off brand modem that works better. Remember that you will want to choose one that is designed with running a BBS in mind.

All modems have an auto answer feature, but many do not offer the ability to automatically change speeds, autotrain, compress data, and detect what kind of modem is calling. When you call GEnie or whatever online service you use, the modem at the other end will ask your modem what speed it is, what error correction it is, if it compresses data, if it is synchronous, asynchronous, autoasynchronous, error correction asynchronous, uses MNP, LAPN, V.42, V.32, V.21, and a ton of other questions that are important to modems. You have to keep in mind that when you are the answering party, you are expected to be able to accept signals from just about any modem. This is especially true if you are using a speed greater than 2400 baud. The faster the speed, the more things that the signal requires to stay 'clean'.

I have found that letting the modem do it all is the easiest way. I set mine so that it will ask the calling modem what is going on and what it wants to do. I suggest one that will use as many protocols as possible. My current modem uses a few kinds of error correction, and other features that are needed these days. This has proven to be a problem for my users, and generates a lot of calls to my customer support line. Don't be afraid to ask for advice from other BBS owners.

When I call a BBS, I always ask what kind of modem that they are using. This has really helped in my selection of modems. Ask other Commodore BBS owners what they use. Some BBS software programs don't support some modems. I'm glad that I got mine up and running with the Aprotek before I started messing around with faster speeds. This gave me a chance to get the BBS running well before I gave myself the stress of wondering if users were having problems getting online.

Openness of the System

What kind of users are you trying to attract? Will you allow aliases or require real names? Do you want to keep a mailing list? Are you planning on charging for an account? Again, relating this to my system, I have two levels of accounts. I have a guest account that everyone is welcome to use, and then I have actual accounts that my users can use for E-mailing each other and to have a name attached to the articles that they upload. I require an account be held to do uploads, so this works out well for me. But it might not for you.

Some systems allow the users to choose their own password, and some assign them for you. Will you force the user to give their real name? If not, they can start trouble on your BBS by doing or saying something illegal or immoral. There won't be any way to stop them because even if you kill their account, they can just get a new one under a different name. The software that I use has a swear checker that will dump the user offline if they use words that I don't want on my system. All of this is up to you, the owner of the BBS. Remember, you have control over what users do online and what they have access to.

Hours Online

This one is easy: 24 hours. If you don't, it doesn't matter how great or wonderful the system is. You will find that there will be no users who will schedule their online times around your schedule. This has been proven time and time again.

Beta Testing

Have friends (who will give you an honest opinion about what your system looks like to them) call your BBS several times a week. As you get more and more menus and files set up, more and more glitches and bugs will creep into the system. You will be better off having a friendly voice tell you that something is wrong than a irate customer (if you charge a fee to get online). This will also allow you to find out if the system is easy to use. Since you are so close to the subject, you might think that it's easy to use, but your users might find it difficult or confusing. A good beta tester is nice to have.

That's about it for this time, but I will expand on each of these subjects in future columns. Now, for a BBS kind of idea. My system is user driven. What the users want, I try to do. So, E-mail me at mcphoto@sojourn.com or mc.photo@genie.geis.com with your ideas and comments about what kind of BBS related subjects you would like to see here. I have contact with a lot of other SYSOPs on other systems, and we frequently match wits to solve problems that we all share. Do you have questions? Feel free to ask. I might not have all the answers, but I can point you in the right direction. If there is a mistake to make, I have made it; from erasing all the rooms on the system to forgetting to turn on the modem. These kinds of things WILL happen when you run a BBS. Expect to learn a lot, work a lot, and get a lot of attention when you run a BBS.

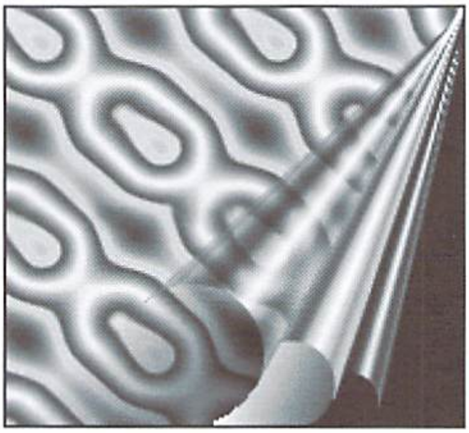
Carrier Detect

By Gaelyne R. Moranec

DOWNLOADING MADE SIMPLE

The Case of the Invisible Shrink-Wrap

You've downloaded the latest and greatest from your favorite BBS but now that you have it, how do you get the program to run? Have you ever seen funny looking messages that seem like they might have a file in them, but you didn't know how to change these from text back into program form? For some reason, the idea of "codes" reminds me of Sherlock Holmes, grappling with mysteries to solve. So, let's go on an adventure and find out about what to do with files once we've downloaded them.



The Shrink-Wrap

Mastering the mysteries of downloaded files is a matter of following the clues to find out where to peel the edge of the shrink-wrap it comes packaged in. Cool Commodore stuff that you find online doesn't come in plastic shrink wrap of course, but you'll often find that it does come packaged in one way or another. Many program files you'll find when BBSing or checking out the online services are in archived format. Archiving accomplishes the same thing as if the programmer had created a shrink-wrap package for the software, since he or she will usually include all the things you'll need to use the program such as supporting files and the instructions, so you'll know what to do with it once you have it.

Most archival programs will perform a bit of magic and let you fit more into the package than you'd normally be able to. This is called compression. When you download the software, you'll need to unpack it from its packaging by using a program to "dissolve" the archive. Luckily, many of the newer files you'll find online come in environmentally friendly 'SelF-dissolving' packages. These are usually called SDA (SelF Dissolving Archives) or SFX's (SelF eXtracting) files. This type of archive is easy for the end user because you simply make sure you have a blank disk (or more) handy, then run the file with the .SDA or .SFX suffix.

Files ending with "SDA" don't give you much of a choice as to what disk drive you want to use, as it will only dissolve to device #8. When you run the SDA, you will get a prompt to insert a blank disk and to press any key when ready. The disk you use doesn't have to be blank, but it does need to have enough space to hold the archived files on it. There are some SDA files that can be dissolved in 128 mode, but most I've run into have to be dissolved in 64 mode.

When you load and run SFX files, you may have a pleasant surprise— these sometimes display text or a graphic which may be an advertisement for the program, or a BBS. Some programmers use this screen to tell the user how to load or use the software. It always pays to read this before proceeding with dissolving the file. After showing this screen (if there is one), you will be prompted for the drive number (0, 1 or a partition number for CMD devices) and then for which device number (8, 9...) that you want to use when dissolving the files. Sometimes you need a little more to dissolve a file, as some archived files require you to use a utility program to dissolve their files. These archives have endings like .ARC, .LNX, .ZIP, .ARK, and .LZH.

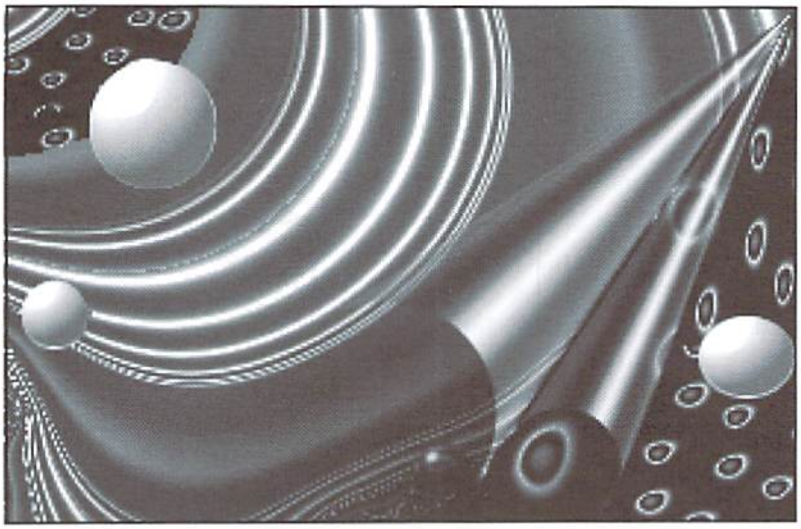
One utility worth mentioning is the OMEGAQ-II program from the now defunct Q-Link. This is a program which lets you do a lot of things, such as read text files, copy files, and even dissolve many of the different archive types (.ARC, .ARK, .SDA, and .LNX to name a few), and lets you use more than one disk drive too. It's an easy to use menu program and if you belong to a user group which had a Q-Link account, chances are OMEGAQ-II will be in the club library.

A program called ZIPUNZIP takes a whole Commodore 1541 disk side and compresses it into four files which are named A!DISKNAME-D!DISKNAME. For this, you'll need a blank disk. The utility program you'll need is ZIPUNZIP.SDA.

Files which end with .ARC, .LZH and Zip may have been created on another platform computer. I say "may", because we have utilities which can create and dissolve most of these type archives. There are two different .ARC files, both of which can be created on the Commodore. Of these, one is strictly Commodore and the other is an MS-DOS style. The name .ARC is the same, but the method of compression is very different, and the software you use to dissolve it depends on which type it was made with. It's not hard to figure out which program you need to dissolve it, however, as if one doesn't work, you'll know to try the other.

ARC250.4 is the C64 utility (it's really a small operating system) which is used to create Commodore style .ARC files. You can find it as ARC250.SDA. ARC250 has been called many things, but it's never been called "user friendly". You don't have to use this program to dissolve .ARC files though, as you can use other utilities such as the above mentioned OMEGAQ-II, or depending on the size of the ARC file you could use a program called ARC-SDA2.SDA to turn the ARC into a SelF dissolving file.

The C64 doesn't have the niche in creating ARC files, as both Commodore and MS-DOS style ARCs can be created (and dissolved) using C128 software. Chris Smeets created an operating system for the 128 and named it after himself (if I wrote one, I'd name it after myself, too). CS-DOS is sometimes thought of as a utility to create or dissolve archives such as .ARC, .LZH, .SFX, but really it's an operating system with a lot of utilities, a few of which have to do with archiving files. CS-DOS has had many "add-on" utilities for it come out over the years. Look for CS-DOS15.ARC and be sure to pick up any supplementary files like CS-ASST1.SFX, CS-ASST2.SFX, LHARC128.ARC, or CSXTRA45.LZH.



CSX01.SDA is a stand alone program for both the C64 and 128 which was also written by Chris Smeets. It allows you to dissolve MS-DOS style ARC's from either the 64 or 128.

You may even run into Commodore files which have been Zipped using PKZip 1.x - and there are two utilities which you can use to unZip These files. The C64 utility is UNZIP101.SDA, and the 128 utility is NZP12817.SFX. These programs will only dissolve Zip files which were created using PKZip version 1.x, not the later v2.04g version.

CBM To GEOS and Back Again

Sometimes after you've dissolved a file, you may find you still need to do more with it before you can make use of it, especially if it's a GEOS file. GEOS uses its own file structure which means they have to be converted to Commodore format before they can be shared via modem.

When you download a GEOS file, it will often have a file extension of .CVT, which is a reminder that you need to convert it back to GEOS format before it can be of any use to you. Here's a classic example of a Catch-22: the program you need to convert the files is a GEOS application so after you've downloaded it, it too needs to be put back into GEOS format again. Luckily, it's not all that bad, as some (not all) versions of Convert are capable of converting themselves. These programs are run first from BASIC, and then you load GEOS to use them. Two such versions are CONVERT.SDA and CON25DA.ARC.

The File's in the Mail

It's easy to send a disk to someone through the regular postal mail, but how do you go about sending someone a program (or other file that's not text) through Internet E-mail? What do you do if someone sends you a file in E-mail or you find one in the message conferences that you read? And what do they look like anyway?

U-U-WHAT?!

Uuencoding a file means changing it from an 8bit binary file to a 7bit ASCII file. The ASCII file is text, and can be put into a message. It looks like code because that's what it is—and the process of converting it back (decoding it) into a binary file is called uudecoding. It sounds like a horrible and difficult process, (that's what I thought when I first heard of it; it sounded too mysterious to be easy) but it's not really that hard once you know what to do with them. This what a uuencoded file looks like:

```
begin 640 sample.prg
M'1PI''H'F2+915,L(%1(25,@4D5!3$Q9($E3($S@4D5!3"!&24Q%+B('81P4
M')DBR2!#04XG5"!"14Q)159%(%E/52!!0U15040,62!465!%1"!42$4@0T]$
*15,@24XN(@"""
'
end
```

Changing this back to its original form can be done with a 64 utility called uuxfer (uuxfer.sfx). This program also lets you create uuencodes. It's a very easy program to use; it asks which you want to do, uuencode or uudecode, then does the work for you. You can also use ACE, an operating system written by Craig Bruce, which has utilities to create and decode uuencoded files. Look for ACE14A.SFX through ACE14E.SFX.

Registered QWKRR128 users can decode uuencodes they receive in their mail by exporting the message. If a uuencode file is detected, the user is prompted if they want to decode the file and given the option to change the file name. Multiple uuencodes in a message can be decoded too. First, export the message then import it into a "reply". Select <W>rite and then <d> for decode and the first file is decoded. Next, change the word "begin" on the first uuencode in the message and <W>rite the message to disk again, this time the second uuencode is decoded. If there are any other uuencoded files in the message, continue this process until you've decoded them all. If you use the ACE operating system or uuxfer all uuencoded files get decoded automatically.

Sometimes a large uuencode can take up many messages. When this happens, you'll need to combine the messages into one text file, then put the code back together again so there are no breaks between the lines of code. To decode the file, use either uuxfer or the utilities in ACE. ACE can seem a little disconcerting if you've never used it, but it's very simple to use for uuencoding or uudecoding files. The command to uuencode a file is:

```
    uuencode program.name >uuencode.name
```

The ">" key ensures that the uuencode ends up in the text file, otherwise it will end up "writing" the uuencode only to the screen. To uudecode files, the ACE command is:

```
    uudecode textfile.name
```

If you're using a UNIX system, you can uudecode and uuencode files while online. The commands are very similar to ACE. To uudecode a file, the command is:

```
    uudecode textfile.name
```

Uuencoding requires that you give THREE names altogether: what the name is NOW, what you want the name of the file to be after it's decoded, plus the name you want the uuencoded file to be called. If I wanted to uuencode a program called "program.prg", the command would look like:

```
    uuencode program.prg program.prg >program.uue
```

When done, the uuencode text is in the file "program.uue".

BCODE

Bcode is another utility of the ACE operating system. This is a Commodore method of encoding 8bit files into 7bit ASCII similar to uuecode, but in some ways it's a little easier to undecode these files, as they do not need to be in any specific order, and they don't have to be "put back together again", like long uuencoded files require. The only drawback is that because bcode isn't available to other platform users, the files can only be shared between Commodore users. The command in ACE to create a bcoded file is:

```
    bcode program.name >textfile.name
```

To unbcode a file, the instruction is:

```
    unbcode filename.txt
```

Bcoded files look like:

```
-bcode-begin 1 sample.prg
ARwpHAoAmSLZRVMsIFRISVMgUkVBTExZIElTIEEgUkVBTCBGSUxFLiIAYRwUAJkiySBDQU4n
VCBCRUxJRVZFIF1PVSBBQ1RVQUxMWSBUWVBFRCBUSEUgQ09ERVMgSU4uIgAAAA==
-bcode-end 1 100 90155dd9
```

After The Shrink-Wrap is Off

Once you've got the shrink-wrap off, there are still a few things that can go wrong. A few that I've experienced seem trivial now, but they weren't when I had problems loading and running new files. To save you from the same mistakes, I'll share a few of my better blunders.

When I bought my first C128, I'd been modeming with my 64 for several years, and thought I knew all the ropes when it comes to downloading files and getting new programs to run for me. Well, the C128 had a few new tricks to teach me about files! Since the 128 can function in either 40 or 80 column modes, the first time I loaded an 80 column program while in 40 column mode, I had no indication that the program had loaded correctly, and I certainly couldn't SEE any changes. New and maybe even experienced C128 users should remember to take note of which mode the program is to be run in. Sometimes this info is in the filename, such as the number "80" or "40" somewhere in the name, other times it may be in the program description. I've also been known to absent-mindedly try to load a C64 program while the computer is in 128 mode, and if I've done it, I'm sure I can't be the only one!

Wrap Up

Once you've managed to dissolve an archive or uudecode a file you've received, don't forget to read any documentation that comes with it. Some programs won't begin unless you send a SYS command to start it, which was a problem I ran into recently. Another program may require that you have other files on the disk along with it. Now that the shrink-wrap mystery has been solved, you can solve many more mysteries by reading the docs you find after peeling back the shrink-wrap.

Don't wait until it's too late...

RENEW EARLY!

Is your Commodore World subscription close to running out? Here's an easy way to check: look at the mailing label on the front of your copy. You'll find your subscription number and the expiration issue number. For example:

```
James Smith  12345EXP14
123 Home Street
Grand Rapids, MI  49502-0123
```

Jim's subscription will run out with Issue 14 as indicated by the EXP14 in his subscription code. Jim would be wise to re-subscribe early to avoid missing a single issue of Commodore World!

GEO PROGRAMIST

GET IN LINE

by Maurice Randall

Let's begin this time by taking a look at one of the routines which is built into the GEOS Kernal. The routine is known as DoInlineReturn. While developing a program, you might find it very handy and use it quite often.

The purpose of this routine is to provide a method for returning from a subroutine call. Normally, a simple RTS will do this. But the beauty of this routine is that you can use the calling routine to supply parameters to the subroutine which immediately follows the JSR to the subroutine. The parameters don't have to be placed in a specific location, they merely get placed 'inline' to your code. The subroutine can find the parameters because there is an address located on the stack that helps the subroutine find them. When the subroutine is finished, it will call DoInlineReturn so that your program may continue as though an RTS was encountered. The program counter will wind up at the first instruction following the parameters.

DoInlineReturn merely increments the program counter by the correct amount. How does it know how to do this? Your subroutine must pass a couple of parameters to it also. An example makes it all look easy:

```
;here's a segment of the calling routine.
 ...
  jsr iShowDiskError
  .byte "Write Protect On",0
 ...
;code continues on here.

;and this is the subroutine that gets called.

iShowDiskError:
  PopW returnAddress
  LoadW r11,#100
  LoadB r1H,#50
  ldy #1
  sty stringCounter
 10$
  lda (returnAddress),y
  beq 80$
  jsr PutChar
  inc stringCounter
  ldy stringCounter
  bne 10$
 80$
  iny
  tya
  php
  jmp DoInlineReturn
stringCounter:
  .block 1
```

Here's what this example does: the subroutine displays a text string on the screen 100 pixels from the left and 50 pixels down from the top. Notice that it begins by loading r11 and r1H with those values. The Kernal routine PutChar will use this as a starting point for the first character to be shown. PutChar will automatically increment the horizontal location for us so that each subsequent character will be properly displayed. We could call PutString for this operation, except we need to keep track of how long the text string is. When we encounter the zero byte at the end of the string, we know how long the string is by the value in y.

The subroutine's first job was to pull a two byte value off the stack. This is the return address that the processor would normally use if we ended the routine with an RTS. We'll use it instead and place it in a location called 'returnAddress'. This location is defined by GEOS and is in zero page. DoInlineReturn will make use of this when called. Our routine can also take advantage of it since it is a zero page location. No other routine will disturb it. Just be sure not to call another inline routine from your own inline routine. The GEOS Kernal has several of them. If you need to use one of them, use the non-inline version instead. For instance, use MoveData instead of i_MoveData. The GEOS inline routines will use returnAddress and mess your routine up good.

DoInlineReturn expects the accumulator to hold the number of bytes that needs to be incremented, plus one. In our example, notice that y gets incremented near the end. This value will be used along with the address in returnAddress to get the program counter where it belongs. Also, we must perform a PHP, because DoInlineReturn will do a PLP so that the processor flags may be passed back to the calling routine.

Think of other ways you could use inline routines. GEOS uses them to pass values into various registers. If you do something like this often, it could save you a few bytes here and there. The calling routine only needs to supply the values and the subroutine can load them into the appropriate registers.

Filling With Nothing

Can you fill something with nothing? It doesn't make sense, does it? Remember this when you use the routine called FillRam. You can't use FillRam to fill an area of ram with zero bytes. There seems to be of some sort of bug that will cause problems. Always call this routine to fill with a value other than zero. If you need to fill an area with zeros, use ClearRam instead.

With either routine, first load r0 with the number of bytes you wish to affect. Then load r1 with the starting address of the desired area. For ClearRam, that's all that is needed, just call the routine now. For FillRam, we need to load r2L with the value that we want placed in each of these bytes. Just don't put a zero into r2L. If you do, you will have some debugging to do and you might not catch the fault for awhile.

Catching Our Own Faults

I make programming mistakes just like everyone else. It's relieving when I find where I made my mistakes. Debugging a program sometimes involves discovering where certain things just won't work, as in the case with FillRam, but most often, it is our own mistakes that cost us time and trouble. If you suspect a bug in your program, find it now before proceeding any further, or make a note of it and fix it as soon as possible. With some bugs, you have no choice but to fix them right away, especially if the program won't run at all.

Here's an example of a mistake I made; I'm working on a program that can load in external programs. The main program has a jump table at the beginning. This gives an external program access to many of the routines contained in the main program. In order to pass along this information to the external programs, I create an equate file that is included in the programs.

Here's how part of the jump table might look:

```
StartOfJumpTable:
   jmp ColorOn
   jmp ColorOff
   jmp FillVRam
   jmp iFillVRam
   jmp MoveVData
   jmp iMoveVData
   jmp SetTo64K
   jmp ColorUp
   jmp FillBytes
   jmp ColorRectangle
;...etc
```

Now, the portion of the equate file would look like this:

```
StartOfJumpTable=$2000
ColorOn=StartOfJumpTable+3
ColorOff=ColorOn+3
FillVRam=ColorOff+3
iFillVRam=FillVRam+3
MoveVData=iFillVRam3
iMoveVData=MoveVData+3
SetTo64K=iMoveVData+3
ColorUp=SetTo64K+3
FillBytes=ColorUp+3
ColorRectangle=FillBytes+3
```

I can easily relocate the jump table and only need to change the value of StartOfJumpTable in the equate file. Or, while I'm still developing the program and haven't yet finalized it, I can easily insert or delete entries from the jump table. If I put in the actual addresses for each location, it would be tedious to change all the entries that followed where I made the change.

Now, study the equates again and see if you can find where I made a mistake. Notice how the location for MoveVData is defined? I mistyped it and left out the '+' sign. MoveVData is now defined to be located at the same address as iFillVRam3. But there is no such address as iFillVRam3. GeoAssembler doesn't flag this as an error because only the first 8 characters of the symbol are valid. In other words, as far as GeoAssembler is concerned, iFillVRa exists and so MoveVData, or MoveVDat will be considered to be at the same location. Because of this mistake, every entry in the jump table following MoveVData is also pointing to the wrong location.

Most mistakes in programming are simple typing errors like that. When you're working on a large project with many routines and symbol values, it's also easy to duplicate labels; maybe not in their entirety, but perhaps within the first 8 characters. Being able to use more than 8 characters helps us remember what a routine is used for, but it also makes it easier to end up with duplications in the first 8 characters. Watch out for that one. A program that ends up being twice as large as another probably requires 4 times the development time. This is largely due to debugging time that must be spent to correct simple errors that could be avoided with some careful thought.

BASIC INSTINCTS

CLEVER SPEED

By Gene Barker

If you've been following the last couple issues, you may have noticed that BASIC programs lack the speed of their machine language counterparts. Last time we conceded that sometimes we must incorporate machine language in our BASIC programs to achieve desired speed. In this issue we finish with an overlooked method for achieving greater speed: clever programming.

Shuffling Cards

Let's define a deck of cards. Using our BASIC Instincts, we code:

```
100 REM *** CREATE DECK OF C
ARDS ***
110 C$="A23456789TJQK"
120 DIM DK$(52)
130 FOR I=Ø TO 3
140 : FOR J=Ø TO 12
150 :   DK$(J+(I*13))=MID$(C$
,J+1,1)
160 : NEXT J
170 NEXT I
```

Now we have a deck of ordered cards inside a BASIC array. The question is, "How do we shuffle the deck?" At first thought, we code:

```
2ØØ REM *** SHUFFLE METHOD O
NE ***
21Ø FOR I=1 TO 2ØØ
22Ø : A=INT(52*RND(1)):REM P
ICK RANDOM CARD
23Ø : B=INT(52*RND(1)):REM P
ICK ANOTHER RANDOM CARD
24Ø : T$=DK$(A):REM NOW SWAP
 THEM
25Ø : DK$(A)=DK$(B)
26Ø : DK$(B)=T$
27Ø NEXT I
```

In method one, we randomly select two cards and swap them. We repeat this process several times. In the demo program, you'll find that this method is slow and does a poor job. Since method one leaves something to be desired, let's try the following clever method:

```
3ØØ REM *** SHUFFLE METHOD T
WO ***
31Ø FOR I=Ø TO 51:REM STEP T
HROUGH CARDS 1 BY 1
32Ø : B=INT(52*RND(1)):REM P
ICK RANDOM CARD TO SWITCH WI
TH
33Ø : T$=DK$(I):REM NOW SWAP
THEM
34Ø : DK$(I)=DK$(B)
35Ø : DK$(B)=T$
36Ø NEXT I
```

Notice how we only step through the deck of cards once, yet we completely shuffle the deck. Not only did we speed up the shuffle, we improved it! Study method two; if you plan on programming for games, you will use it often.

Sorting

Let's say for some reason we wished to sort a list of numbers in descending order. Assuming we had a list of numbers in an array A(N) where N is the length of the list, some manuals recommend:

```
4ØØ REM *** BUBBLE SORT ***
4Ø5 FOR I=1 TO N-1
41Ø : IF A(I)>A(I-1) THEN NE
XT I:GOTO 46Ø
42Ø : T=A(I-1):REM SWAP ADJA
CENT NUMBERS TILL SORTED
43Ø : A(I-1)=A(I)
44Ø : A(I)=T
45Ø : I=N-1:NEXT I:GOTO 4Ø5
46Ø REM DONE
```

To me, a bubble sort seems silly. A more natural method would be:

```
5ØØ REM *** SELECTION SORT *
**
51Ø FOR I=Ø TO N-2
52Ø : LV = I:REM FIND THE LO
WEST VAL LEFT IN LIST
53Ø : FOR J=I+1 TO N-1
54Ø :   IF A(J)<A(LV) THEN LV
=J
55Ø : NEXT J
56Ø : T=A(I):REM NOW SWAP LO
WEST VAL TO RIGHT POS
57Ø : A(I)=A(LV)
58Ø : A(LV)=T
59Ø NEXT I
```

A selection sort is similar to how we might sort our checks: Find the check with the lowest check number, and set it a side. Then find the check with the next lowest number, and set it side. Repeat until you have a sorted pile of checks. This sort makes more sense and is faster than that evil bubble sort.

There are several techniques for sorting; many which are faster than the ones mentioned. But the above sorts serve as good examples of how clever programming can speed up your BASIC programs.

Conclusion

Often times the speed we gain with clever techniques is all we need to achieve adequate speed in our programs. However, there will be times when you have no option but to incorporate machine language into your programs. This will especially be the case with sequential file access and graphics programming. But don't fret. You have the master himself, *Commodore World's* Jim Butterfield to teach you how in each issue. I can't recommend a better way to get your feet wet in machine language.

Notes

As you type in this issue's program, take it one section at a time. Try to get a general idea of what the section is trying to accomplish. See if you can follow how the section is using its variables. If you see an unfamiliar BASIC statement, take a look at it in your BASIC manual. If you are still confused, move on to the next section; often times the next section helps explain the previous one. Above all, don't forget to BACKUP your work frequently. Also, before entering this issue's program, load and run the CHK-LIST utility (located elsewhere in this issue). CHK-LIST insures that you enter the program correctly the first time. Also, remember to SAVE the program before you attempt to RUN it. It never hurts to be safe. And finally...

Gene's Last BASIC Instincts

That's right, this is my last column for Basic Instincts. But don't worry; we have an excellent replacement who can give this column the time it deserves. I'd like to welcome David Pankhurst to our team. You will find David to be an excellent road guide on your Commodore BASIC journey. As for me, I'll be concentrating on special projects which will appear every few issues or so.

CLEVER.BAS

```
5ØØØ  1ØØ rem-------------------------------
6dc2  1Ø5 rem commodore world magazine
c67c  11Ø rem basic instincts w/gene barker
Ø6e7  115 rem clever speed demo
Øe6b  12Ø rem
2d24  125 rem (c)1995 creative micro designs
92ad  13Ø rem-------------------------------
c1cc  2ØØ rem-
ca77  2Ø5 rem setup menu
da6b  21Ø rem-
c2d7  215 poke 5328Ø,11:rem set gray border
ba17  22Ø poke 53281,Ø:rem set black screen
45e7  225 print"{CLEAR/HOME}{LGRN}{CRSR DN}{CR
      SR RT}{RVS ON}clever speed demo"
1Øa1  23Ø print"{CRSR DN}{CRSR RT}[f1] compare
       shuffle methods"
1955  235 print"{CRSR RT}[f3] compare sorting
      methods"
9be3  24Ø print"{CRSR DN}{2 CRSR RT}[_] exit p
      rogram"
da6b  25Ø rem-
6c32  255 rem get users key press
5291  26Ø rem-
aead  265 get x$:if x$="" then 265
5Ø58  27Ø if x$="{F1}" then 1ØØØ:rem f1 check
14d1  275 if x$="{F3}" then 2ØØØ:rem f3 check
41Ø3  28Ø if x$<>"_" then 265
9ba2  285 end
d85a  1ØØØ rem------------------------------
82aa  1ØØ5 rem compare shuffle methods
8b23  1Ø1Ø rem------------------------------
b365  1Ø15 print"{CLEAR/HOME}{CRSR DN}{CRSR RT
      }{RVS ON}compare shuffle methods"
6edØ  1Ø2Ø if c$="a23456789tjkq" then 11ØØ
82Ø2  1Ø25 : rem *** create deck of cards ***
d7a9  1Ø3Ø : c$="a23456789tjkq"
8Ø38  1Ø35 : dim dk$(52)
6f1Ø  11ØØ print"{CRSR DN}{CRSR RT}shuffle met
      hod one..."
Ø672  11Ø5 gosub 13ØØ
baeØ  111Ø tm=ti
Ø1e9  1115 for i=1 to 2ØØ
b11b  112Ø : a=int(52*rnd(1))
9a98  1125 : b=int(52*rnd(1))
9942  113Ø : t$=dk$(a)
f1ff  1135 : dk$(a)=dk$(b)
Ø493  114Ø : dk$(b)=t$
2d77  1145 next i
a53c  115Ø gosub 14ØØ
fØ55  12ØØ print"{CRSR DN}{CRSR RT}shuffle met
      hod two..."
ffcf  12Ø5 gosub 13ØØ
a6ca  121Ø tm=ti
af68  1215 for i=Ø to 51
32bc  122Ø : b=int(52*rnd(1))
dc21  1225 : t$=dk$(i)
e11d  123Ø : dk$(i)=dk$(b)
445d  1235 : dk$(b)=t$
8726  124Ø next i
e2ef  1245 gosub 14ØØ
3c2b  125Ø print"{CRSR DN}{CRSR RT}(press any
      key)"
8e76  1255 get x$:if x$="" then 1255
d5cb  126Ø goto 1ØØ
4444  13ØØ rem *** init deck of cards ***
Ø1a4  13Ø5 for i=Ø to 3
1Ø24  131Ø : for j=Ø to 12
6339  1315 :   dk$(j+(i*13))=mid$(c$,j+1,1)
Øf9f  132Ø : next j
c7f7  1325 next i
fc12  133Ø return
```

CLEVER.BAS *(cont.)*

```
628d  14ØØ rem *** print shuffle results ***
      14Ø5 tm=abs(int((tm-ti)/1ØØ))
48bd  141Ø print"{5 CRSR RT}shuffle time (seco
971Ø  nds) ="tm
4fd9  1415 print"{5 CRSR RT}deck:":print
681c  142Ø for i=Ø to 3
97e3  1425 : print"{5 CRSR RT}";
ac5d  143Ø : for j=Ø to 12
c384  1435 :   printdk$(j+(i*13));
2f86  144Ø : next j
d5f2  1445 : print
92ac  145Ø next i
6ada  1455 return
124e  2ØØØ rem------------------------------
8bØe  2ØØ5 rem compare sorting methods
      2Ø1Ø rem------------------------------
259Ø  2Ø15 print"{CLEAR/HOME}{CRSR DN}{CRSR RT
aacd  }{RVS ON}compare sorting methods"
12e8  2Ø2Ø if n=25 then 21ØØ
592Ø  2Ø25 : rem *** create number list ***
ba8b  2Ø3Ø : n=25
84ØØ  2Ø35 : dim lt(n)
3Ød6  21ØØ print"{CRSR DN}{CRSR RT}bubble sort
8895  ..."
Ø32f  21Ø5 gosub 23ØØ
b783  211Ø tm=ti
      2115 for i=1 to n-1
85e8  212Ø : if lt(i)>lt(i-1) then next i:goto
       2145
      2125 : t=lt(i-1)
      213Ø : lt(i-1)=lt(i)
      2135 : lt(i)=t
      214Ø : i=n-1:next i:goto 2115
      2145 gosub 24ØØ
      22ØØ print"{CRSR DN}{CRSR RT}selection s
      ort..."
      22Ø5 gosub 23ØØ
      221Ø tm=ti
      2215 for i=Ø to n-2
      222Ø : lv=i
      2225 : for j=i+1 to n-1
      223Ø :   if lt(j)<lt(lv) then lv=j
      2235 : next j
      224Ø : t=lt(i)
      2245 : lt(i)=lt(lv)
      225Ø : lt(lv)=t
      2255 next i
      226Ø gosub 24ØØ
      2265 print"{CRSR DN}{CRSR RT}(press any
      key)"
      227Ø get x$:if x$="" then 227Ø
      2275 goto 1ØØ
      23ØØ rem *** init list w/random #'s ***
      23Ø5 for i=Ø to n-1
      231Ø : lt(i)=int(1ØØØ*rnd(1))
      2315 next i
      232Ø return
      24ØØ rem *** print sort results ***
      24Ø5 tm=abs(int((tm-ti)/1ØØ))
      241Ø print"{5 CRSR RT}sort time (seconds
      ) ="tm
      2415 return
```

Peripheral Vision

By Jim Butterfield

DEVICES 0 AND 3

Most users know that devices that carry a number of 4 and greater are located on the serial bus. Typically, device 4 is the printer, device numbers 5 through 7 are set aside for other printers or special devices, and devices numbered from 8 and up are disk drives. These numbers for devices 4 and above are assigned for convenience; the serial bus logic is the same.

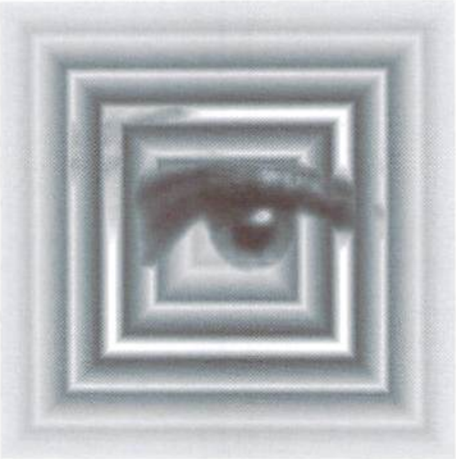
We will look at devices below 4; specifically, device 0 (keyboard) and 3 (screen). The input stream connects by default to the keyboard and screen (GET looks to the keyboard, INPUT looks to the screen), and the output stream (PRINT) connects by default to the screen. But when we address the keyboard or screen as a device, the rules change, and the devices start to behave in new ways.

First, let's complete the picture by identifying the two missing numbers. Device 1 is the cassette tape; the interface is still there on all 8-bit Commodore machines, although customized ROM kernal packages often remove Datasette logic in favor of other features. Device 2 depends on the model of computer. The earliest Commodore computers (the PET, CBM8032 and similar units) assigned device 2 to a second cassette tape drive. From the VIC-20 onwards, device 2 has been used for the RS-232 interface.

A Note on Files

You open a file with the OPEN statement. This must contain at least two values: a logical file number, and the device number. Notice that I used the word "values"; you don't have to supply an actual number. It can be a variable or mathematical expression.

The logical file number comes first. You pick this number: it can be any number from 1 to 127 that's not in use at the moment. After you have set this number in the OPEN statement, it's the ONLY one you will use in subsequent GET#, INPUT#, PRINT#, and CLOSE statements. (There may be an exception when you're using relative disk files).



The device number comes second. In the context of this article, we're interested in device number 0 for the keyboard and 3 for the screen, but I'm sure you are aware of other popular numbers such as 4 for printer and 8 for disk drive. There can be other information following the device number: a secondary address, and perhaps a string. These don't apply to devices 0 and 3.

The point is that if you free your thinking from using numbers, you may discover amazing things your programs can do. For example, suppose the device number in the OPEN statement was a variable rather than a number. The program could open whatever device it chose by setting that variable value. To be more specific, suppose that you have a program that might want to send data to the printer, or might want to send it to the screen. Your code could look like this:

```
150 PRINT "WANT TO PRINT THE
 DATA (Y OR N) ?";
160 P=Ø
170 GET P$
180 IF P$="Y" THEN P=4
190 IF P$="N" THEN P=3
200 IF P=Ø GOTO 170
210 OPEN 1,P
```

Depending on what the user types, we would open our output path to the screen (3) or printer (4). Our data would be sent with PRINT#1 in either case. And a neat program will CLOSE 1 at the end, of course. By the way, the screen will ignore any secondary address you send, so if your printer requires secondary address 7 for text, you could change line 210 to OPEN 1,P,7 and all would be well. The logical file number could be a variable, too, so that a program could pick which logical file to send to. Suppose we've opened the screen and printer with the following code:

```
100 OPEN 1,3
110 OPEN 2,4
```

Now as our program runs we can decided whether to send selected data to the screen or the printer. The code, looping through a set of records, might run as follows:

```
500 L=1
510 IF M$="MAY" THEN L=2
520 PRINT#L,M$;":";N$
```

If the record being handled shows a month of"MAY" the program will send it to the printer, otherwise, it will go to the screen. By the way, this is a great method for splitting a file into two or more smaller files (in which case, both logical devices would specify a write-to-disk file). But that's getting away from our discussion of devices 0 and 3.

The Keyboard: Device 0

You read characters from the keyboard buffer by using GET X$ (any string variable name may be used in place of X$). In the same way, you can open the keyboard as a device and get characters with GET#.

There is a small difference, even with the simple GET. As we discussed above, the use of a logical file number would allow us to select input from the keyboard or some other device. Keep in mind that the keyboard returns CHR$(0)—or a null string in Basic—when no character is waiting, which we discussed in the last column. For many applications, it's safe to discard those unwanted characters. Thus, you could construct code that would receive input, character by character, from the keyboard or from some other device such as a disk file or RS-232 connection.

Input From Device 0

There's a noticeable change in style when you switch from the INPUT statement to INPUT# from device 0, the keyboard. It's a handy way to get rid of that annoying question mark that Basic always puts on the screen on your behalf. When you lose the question mark, you also lose a couple of other things: the "prompt", and the movement to a new line after input is received. These are not hard items to correct. Replace the prompt with a PRINT statement and give another PRINT after the input. You'll lose other things too, so run this example.

```
100 OPEN 1,0
110 PRINT "YOUR NAME: ";
120 REMARK: NOTE SPACE AND S
EMICOLON ABOVE
130 INPUT#1,N$
140 PRINT
150 PRINT "HELLO, ";N$
160 CLOSE 1
```

The above program will show other aspects of receiving input through a logical file. No cursor! And here are a few things to try: press RETURN instead of supplying a name (you may need to press it twice), and move the cursor to a different part of the screen where there is text and then press RETURN. As a sidelight to just pressing RETURN, try inserting an extra line:

```
105 N$="NOBODY"
```

Normally, pressing RETURN in response to INPUT leaves the input variable, N$, untouched. With INPUT#, you'll see that entering nothing gets you... nothing.

Despite the fact that we specified device 0, keyboard, a little experimentation will show you that the INPUT# command is in fact reading from the screen. INPUT# from device 0 has its uses, but you can also see its disadvantages, such as loss of a cursor. There are kloodges to get around this, but it seldom turns out really well. Still, it's an option.

Output To Device 3, Screen

This is the most popular combination. As our first example above shows, using PRINT# to device 3 sends the same data that would be sent by PRINT, or that would go to a data file or communications line. But there's a small difference in the way numbers are printed. Try the following code and see the difference on the three lines.

```
100 X=1
110 Y=20
120 Z=300
130 OPEN 1,3
140 PRINT X;Y;Z
150 PRINT#1,X;Y;Z
160 PRINT#1,STR$(X);STR$(Y);
STR$(Z)
170 CLOSE 1
```

The first line of output a standard PRINT; the second is PRINT#. The third line is how these same numbers go to a file, even when we don't use the STR$() function. The difference is that when sending to the screen, the computer automatically adds a cursor-right operation after each number. As a result, the numbers on the first two lines are spread more widely than that of the last line. Moral: if you want to exactly match screen output to file output, don't use numeric values; instead, change everything to strings as we have done in line 160 above.

Reading From Device 3 (Screen)

This is the least known combination, but it can be useful. In particular, GET# from device 3 can be a good way to copy the entire screen to a file or printer. The following program prints a couple of lines at the top of the screen, and then performs a hundred GET# operations. It prints the hundred characters both as an ASCII numeric value, and then as characters.

```
100 DIM X(100)
110 PRINT CHR$(147);
120 PRINT CHR$(19);"MY DOG H
AS FLEAS."
130 PRINT " .. LOTS OF THEM!
"
140 PRINT CHR$(19);
150 OPEN 1,3
160 FOR J=1 TO 100
170 GET#1,X$
180 X(J)=ASC(X$)
190 NEXT J
200 CLOSE 1
210 FOR J=1 TO 100
220 PRINT X(J);
230 NEXT J
240 PRINT
250 FOR J=1 TO 100
260 PRINT CHR$(X(J));
270 NEXT J
```

When you run this program, look closely at the numeric values that are printed. Each character on the screen is delivered to the GET# routine as ASCII (not as a screen code). Thus, a letter A is delivered as character 65, not its screen code of 1. But there's an extra twist: the RETURN character is inserted at the end of each line! On 40-column screens, you'll see RETURN (character 13) as the 40th character. There exception is that if you have a double line (or more-than-double multiple on some machines), the RETURN will not be shown until you reach the true end of the line. This is an elegant way of dumping a screen: GET# a thousand characters (or two thousand, for an 80 column screen), and you can restore it, analyze it, or send it to another computer.

INPUT# from device 3, in contrast, is a problem. It works after a fashion, but has all kinds of difficulties. If you want to play around with it, try the following rules. First, position the cursor carefully before performing each INPUT#, and don't trust the cursor position to be where you think after the operation. Next, don't try to get information from a blank line; INPUT# doesn't like it. Another factor is that the INPUT operation always trims leading spaces from any string it finds, so you won't always get accurate representation of a screen line. Finally, don't be surprised if you sometimes get the RETURN character as part of the string read by INPUT.

Summary

Each of the above access methods has its own quirks. I've picked the three most useful below.

- PRINT# to the screen delivers screen output in the usual way. Remember that, where numbers are involved, screen output is not exactly the same as output to other devices.

- GET# from the screen can be an effective way to scoop a whole screen's contents.

- INPUT# from device 0 can help eliminate the question mark prompt, which is sometimes annoying. But you also lose the cursor.

ASSEMBLY LINE

by Jim Butterfield

The three data registers will start to take on individual characteristics. And we'll print a phrase on the screen using a simple loop to walk through each character.

A, X, and Y Registers

All three data registers can load or store data, a number from 0 to 255, an ASCII character, or whatever else will fit in eight bits. But each register has its own style. For example, the A register, or accumulator, is where the computer stages ASCII characters for input and output. Later, you'll see that A is also good for addition and subtraction, among other things.

The X and Y registers are good for counting and pointing. They are called index registers, and we'll use one of them (either one would do) in our simple string output example. The powerful thing about input registers is not simply that they can count, but they can also be used to modify an address.

Here's how "indexing" works: you already know that an instruction will often specify an address in memory. With indexing, this address will be "adjusted" by whatever value is in the selected index register, X or Y. "Adjusted" means that the instruction address will have the register contents added to it, making an effective address. Figure 1 illustrates the idea: the instruction address is a fixed "anchor" pointing somewhere in memory. The contents of the index register is an adjustable offset that the program can change as needed.

This sounds ideal for our project, which is to print a message. The instruction will point at the first character of the message. An index register will "walk" along, allowing us to pick out the characters one by one and print them.

The PRINT call: address $FFD2

Any time we have a character in the A register ready to be printed, we can simply call subroutine CHROUT at $FFD2 and the character will be sent to the output stream. That connects to the screen display unless we change it. Special characters like cursor movement, color change, and even screen-clear will perform their actions when sent to the screen. We'll make a point of printing the standard RETURN character at the end of our output string. The CHROUT call has an uncommon benefit: it will preserve the contents of all three data registers. That's often not the case with other system calls. When we use those, it will be wise to save the values in A, X, and Y if they will be needed again.

Loops And Incrementing

Registers X and Y are convenient for use as counters. We can increment the value in either register with command INX or INY (increment X, increment Y). The contents of the register will be increased by one. There are corresponding commands to decrement the registers, DEX and DEY, but we won't use them here. As a general rule, counting up is better than counting down. We want to go around a loop until the value in the index reaches a certain value. This is easy. We can compare X or Y with CPX or CPY, following which BNE, Branch Not Equal, will take us back if we haven't reached our target.

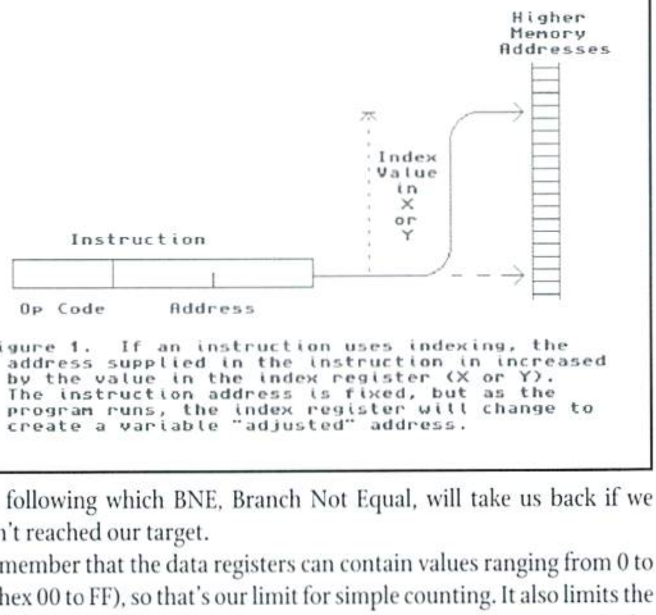


Figure 1. If an instruction uses indexing, the address supplied in the instruction in increased by the value in the index register (X or Y). The instruction address is fixed, but as the program runs, the index register will change to create a variable "adjusted" address.

Remember that the data registers can contain values ranging from 0 to 255 (hex 00 to FF), so that's our limit for simple counting. It also limits the "reach" of indexing. If our message were 300 characters in length, for example, an index wouldn't be able to reach all the characters. We'd need to use another method ("indirect addressing", coming up later).

The Program

It seems that every first program says "HELLO WORLD". Let's expand our thinking and say HELLO UNIVERSE!, complete with a RETURN character at the end. Fixed information such as this string is usually placed behind the program code, but we don't know where our code will end yet.

If we let a symbolic assembler figure that out for us, our program would take the following form:

```
*        =    $2ØØØ           ; start the program at this address
CHROUT   =    $FFD2
START    LDX  #Ø              ; load zero into index X
LOOP     LDA  MESSAGE,X       ; load a character from the message
         JSR  CHROUT          ; call the output routine
         INX
         CPX  #16             ; have we sent 16 characters?
         BNE  LOOP            ; if not, go back
         RTS                  ; if so, return to BASIC
MESSAGE  .ASC "HELLO UNIVERSE!"
         .BYTE $ØD            ; here's the RETURN character
         .END
```

Even if you have no symbolic assembler, you can see the program flow. The first time we execute the instruction at LOOP, the value in X is zero, and so we load the first character at whatever location MESSAGE is found. The next time around, X is equal to 1, and so indexing causes us to load the next character, and so on.

An experienced user would also ask the symbolic assembler to calculate the length of the message, but we'll just put the count of 16 in there for the moment. There would likely be even more comments than those I have supplied here.

Coding with the Machine Language Monitor

One problem we will need to solve here is that of a "forward reference". At the point where we need to insert the address of MESSAGE, we won't yet know where it is. There are two ways to solve this problem. The easy way is to leave extra space. The slightly harder method is to go back and fix the code when you find out where the address will be. That's what a symbolic assembler does automatically. We'll use the easy method here, but you're welcome to take extra steps.

Enter the Machine Language Monitor program and get ready for the first line.

```
A 2ØØØ LDX #Ø
```

The next line is where we need to enter the address of MESSAGE. We can take a fair guess: our program is unlikely to be over 32 bytes long, so $2020 would be a reasonable place for the message (32 decimal is hex 20).

```
A 2ØØ2 LDA 2Ø2Ø,X
```

Try to think of that comma as a plus sign. That's really what happens with indexing.

```
A 2ØØ5 JSR FFD2
```

Now we're printing the character. And we rejoice in the fact that the contents of X will not have been disturbed, so we can continue by counting.

```
A 2ØØ8 INX
A 2ØØ9 CPX #$1Ø
```

There are sixteen characters in our message, and decimal 16 is hexadecimal 10. Your MLM may allow you to code #+16, in which case you'll see it changed to hex as you press RETURN.

```
A 2ØØB BNE 2ØØ2
```

This is a backwards branch, so we won't need to do any forwards-guessing on the address.

```
A 2ØØD RTS
```

You'll see a prompt of A 200E, but we have no more code to input, so just press RETURN. You can see that $200E is our first free address, so we could put the message right here. If you do so, remember to go back and change the instruction at $2002 so that it shows the correct address. But it's also all right to leave the address as it is, with a few empty bytes between the program and its data. Now we need to switch mental gears. The last piece of information we need to put into memory is the message. It's not a program. You can not assemble it, you just put it into memory. To do that, first display memory with command:

```
M 2Ø2Ø 2Ø3Ø
```

You'll get two or three lines of data, showing what's in memory. The memory values are those two-character hexadecimal numbers. The four-digit number of the right is the address that starts the line, and the stuff on the left is corresponding ASCII when appropriate. The data you see might be anything. Now the task is to change it to our message.

Go back to the line that starts with address 2020 and type over the two-character hex numbers. When you have filled in the eight or sixteen values on a line, press RETURN. You may need to keep going on the next line, so don't forget to press RETURN again if you do. The bytes you enter should be:

```
48 45 4C 4C 4F 20 55 4E 49 56 45 52 53 45 21 ØD
```

When you press RETURN, look at the right-hand side of the line and you should see the correct characters for HELLO UNIVERSE! there. It's a good check on your data entry.

Running The Program And Saving It

Return to BASIC (with command X, exit), command SYS 8192, and you should see your hello message. When you have confirmed that it runs properly, re-enter the MLM and we'll save the program just written onto a disk file. Type the following command:

```
S "Ø:HELLO.MSG",Ø8,2ØØØ,2Ø3Ø
```

You have probably figured that command S stands for Save, and that the following string is the file name. The next value is the device (8 for disk, of course), followed by the start address and the last address plus one.

If you've closed up your message so that it's down around $200D, you can lower the second address in the save command accordingly. Remember, it's not the address of the last byte, it's one higher. Later, if you want to load this program from BASIC, remember that on most machines you must command:

```
LOAD "HELLO.MSG",8,1
```

That final 1 is to insist that the computer must not relocate the program as it loads it. If it were relocated, you wouldn't know what address to SYS to!

How to Type In Program Listings Appearing in Commodore World

While *Commodore World* currently doesn't make it a habit of publishing type-in programs, a number of our columns do require entering sample routines. For this purpose, we have created our CHK-LIST utility for the Commodore 64 and 128. This utility uses a 16-bit CRC checksum method to verify that you have correctly entered each program line, and that each of the characters in the program lines are in the correct order.

You'll notice that program listings appear with a column of values to the left of the program lines. These values are the CHK-LIST values and are not to be entered as part of the program. A similar set of values are generated by the CHK-LIST utility to allow you to verify that everything has been entered correctly.

Enter the CHK-LIST program from BASIC. You can use either a C-64 or a C-128 computer. If you use a C-128, it can be in either 64 or 128 mode. Be sure to enter each line carefully to avoid mistakes—until you actually have CHK-LIST working, finding errors in program entry won't be easy. After you have finished entering the program, be sure to SAVE a copy to disk before you attempt to RUN it, just in case. If you aren't familiar with how to save a program to disk, you can use the following command:

```
SAVE"CHK-LIST",8
```

To use CHK-LIST, load it into your computer and type RUN. Make sure that any program you are currently working on is saved first, or start CHK-LIST before you begin typing in a new program. After you have CHK-LIST in memory and running, type NEW. You may now either load or begin typing the program you wish to have CHK-LIST check on. Whenever you want to check your program, type in the appropriate SYS command given below:

C-64 or C-128 in 64 mode:	`SYS49152`
C-128 in 128 mode:	`SYS4864`

Note that when typing in listings, some special characters will appear in braces. For example, {CLR/HOME} means that you should enter the Clear key, which is done by holding down the SHIFT key while you press the HOME key. Other times you may see a number ahead of the key name, such as {3 SPACES} or {5 CRSR L}. This means you should press the key indicated the number of times shown. Most special keys are easy to identify, since the text shown will generally match the text on the key. Exceptions are the space bar {SPACE}, and cursor keys which include directions ({CRSR UP}, {CRSR DN}, {CRSR L} and {CRSR RT}). Be sure to use the correct key combinations for color keys, such as <CTRL><2> for {WHT}.

CHK-LIST

```
A454  1Ø F=ABS(PEEK(65533)=255):M=49152:IFFTHE
      NM=4864
6E2F  12 C=Ø:PRINT"{CLR/HOME}WORKING";
E35Ø  2Ø READD:IFD=-256THEN4Ø
AD2Ø  3Ø C=C+D:IFD<ØANDF=ØTHEN2Ø
3316  31 IFD<ØTHEND=Ø-D:M=M-1
Ø7FØ  32 POKEM,D:M=M+1:PRINT".";:GOTO2Ø
578A  4Ø PRINT:READCK:IFC<>CKTHENPRINT"ERROR I
      N DATA STATEMENTS!":END
Ø679  5Ø PRINT"DONE.":END
8D92  6Ø :
E7FE  49152 DATA 165,43,-45,133,251,165,44,-46
      ,133,252
B2AE  4916Ø DATA 169,Ø,141,36,193,-2Ø,169,147,
      32
CD5Ø  49168 DATA 21Ø,255,32,194,192,-19,16Ø,Ø,
      14Ø
C9CD  49176 DATA 37,193,-2Ø,177,251,133,253,2Ø
      8,3
2Ø58  49184 DATA 238,37,193,-2Ø,2ØØ,177,251,13
      3,254
EA9C  49192 DATA 2Ø8,3,238,37,193,-2Ø,173,37,1
      93,-2Ø
6C15  492ØØ DATA 2Ø1,2,2Ø8,1,96,2ØØ,177,251
E7ØE  492Ø8 DATA 17Ø,2ØØ,177,251,32,2Ø5,-5Ø,18
      9,-142,169
6795  49216 DATA 6,133,211,-236,169,61,32,21Ø,
      255
F8ØF  49224 DATA 169,32,32,21Ø,255,16Ø,2,177
9735  49232 DATA 251,32,213,192,-19,2ØØ,177,25
      1,32
Ø734  4924Ø DATA 213,192,-19,2ØØ,177,251,24Ø,6
      ,32
D99D  49248 DATA 213,192,-19,76,9Ø,192,-19,173
      ,191,192,-19
AC3Ø  49256 DATA 32,167,192,-19,173,19Ø,192,-1
      9,32,167
```

CHK-LIST *(cont.)*

```
B343  49264 DATA 192,-19,169,13,32,21Ø,255,165
      ,253
DF3A  49272 DATA 133,251,165,254,133,252,238,3
      6
A6E2  4928Ø DATA 193,-2Ø,173,36,193,-2Ø,2Ø1,2Ø
      ,24Ø,3
936E  49288 DATA 76,18,192,-19,162,Ø,189,1,193
      ,-2Ø
8C3A  49296 DATA 24Ø,6,32,21Ø,255,232,2Ø8,245,
      32
EB74  493Ø4 DATA 228,255,2Ø1,13,2Ø8,249,32
6Ø95  49312 DATA 228,255,2Ø8,251,76,8,192,-19,
      72
AØØ1  4932Ø DATA 1Ø6,1Ø6,1Ø6,1Ø6,32,18Ø,192,-1
      9,1Ø4
FAA2  49328 DATA 32,18Ø,192,-19,96,41,15,17Ø,1
      89
EBFD  49336 DATA 2Ø,193,-2Ø,32,21Ø,255,96,Ø,Ø
E9Ø7  49344 DATA Ø,Ø,169,Ø,141,19Ø,192,-19,141
E8EA  49352 DATA 191,192,-19,169,33,141,192,19
      2,-19,169
A7D7  4936Ø DATA 16,141,193,192,-19,96,162,8,7
      2
6Ø4Ø  49368 DATA 41,127,77,191,192,-19,141,191
      ,192,-19
D24B  49376 DATA 24,14,19Ø,192,-19,46,191,192,
      -19,144
D52F  49384 DATA 18,173,192,192,-19,77,19Ø,192
      ,-19,141
DCA6  49392 DATA 19Ø,192,-19,173,193,192,-19,7
      7,191,192,-19
6Ø32  494ØØ DATA 141,191,192,-19,1Ø4,1Ø,2Ø2,2Ø
      8,215
37C5  494Ø8 DATA 96,13,8Ø,82,69,83,83,32
9A2A  49416 DATA 6Ø,82,69,84,85,82,78,62
AC9Ø  49424 DATA 13,13,13,Ø,48,49,5Ø,51
FE71  49432 DATA 52,53,54,55,56,57,65,66
Ø17E  4944Ø DATA 67,68,69,7Ø,Ø,Ø,-256,37944
```

* CLASSIFIED ADS *

Cable and software allowing two C-64s to share memories. Effectively the two become one computer. $36.95 + $3. S&H. Marcus Electronics, P.O. Nox 620028, Doraville, GA 30340.

C64/128 PUBLIC DOMAIN. REQUEST FREE CATALOG OR SEND $2 FOR A DEMO & CATALOG. CALOKE IND., P.O. BOX 18477, RAYTOWN, MO. 64133. VISA-M/CARD ACCEPTED.

C-64 FOREIGN-AMERICAN Utilities, Graphics, Hacker, Arcade. 32¢ stamp gets catalog. Home-Spun Software, POB 1064-CW,Estero,FL. 33928

For sale: C128 system with software. Send SASE for a listing to Don Cochenour, 1750 Patrick Place #211, Library, PA 15129.

GEOS Publication. One Year Subscription $8.50; two years $16. 713 E. Main Street, Independance, KS. 67301-3726. Monthly.

COMMODORE WORLD

Classified Advertising

Commodore World subscribers may place non-commercial classified advertising in Commodore World at a cost of $10.00 per issue. Your advertisement may contain up to 150 characters (including spaces). Send your advertisement with payment to: CW Classified Advertising, c/o Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow MA 01028-0646.

CW & RUN BACK ISSUES

For years, *RUN* Magazine provided Commodore Users with a great source of information, and now CMD has given you Commodore World. Don't let this valuable information slip away—fill in the voids in your library now!

RUN Magazine Back Issues

Any 3 issues for $12.00, any 6 for $18.00, or any 12 for only $24.00

RMJAN88	January 1988	**RMOCT89**	October 1989
RMFEB88	February 1988	**RMSP89**	Special Issue 1989
RMMAR88	March 1988	**RMJAN90**	January 1990
RMAPR88	April 1988	**RMMAY90**	May 1990
RMMAY88	May 1988	**RMJUN90**	June/July 1990
RMJUN88	June 1988	**RMDEC90**	December 1990
RMJUL88	July 1988	**RMJAN91**	Jan/Feb 1991
RMFEB89	February 1989	**RMJAN92**	Jan/Feb 1992
RMMAY89	May 1989	**RMMAR92**	March/April 1992
RMJUN89	June 1989	**RMMAY92**	May/June 1992
RMJUL89	July 1989	**RMJUL92**	July/August 1992
RMAUG89	August 1989	**RMSEP92**	Sep/October 1992
RMSEP89	September 1989	**RMNOV92**	Nov/December 1992

Shipping: 3 or 6 issues - U.S. $3.00, Canada $5.00, Foreign $15.00; 12 issues U.S. $5.00; Canada $7.00; Foreign $20.00.

Commodore World Back Issues

CW1	Issue 1, Volume 1, Number 1	**CW6**	Issue 6, Volume 2, Number 1
CW2	Issue 2, Volume 1, Number 2	**CW7**	Issue 7, Volume 2, Number 2
CW3	Issue 3, Volume 1, Number 3	**CW8**	Issue 8, Volume 2, Number 3
CW4	Issue 4, Volume 1, Number 4	**CW9**	Issue 9, Volume 2, Number 4
CW5	Issue 5, Volume 1, Number 5	**CW10**	Issue 10, Volume 2, Number 5

Shipping: U.S. and Canada $2.00 for first issue, plus $1.00 per additional issues.; Foreign $5.00 per issue.

TO ORDER CALL 1-800-638-3263

ADVERTISERS INDEX

Statement of Ownership, Management and Circulation

1. Title of Publication: Commodore World Magazine. **2.** Publication No.: 1078-2516. **3.** Filing Date: 09/29/95. **4.** Issue Frequency: Bi-Quarterly. **5.** No. of Issues Published Anually: 8. **6.** Annual Subscription Price: $29.95. **7.** Complete Mailing Address of Known Office of Publication: Creative Micro Designs., Inc., P.O. Box 646, 15 Benton Dr., East Longmeadow, MA 01028-0646. **8.** Complete Mailing Address of Headquarters or General Business Office of Publisher: P.O. Box 646, 15 Benton Dr., East Longmeadow, MA 01028-0646. **9.** Full Names and Complete Mailing Address of Publisher, Editor and Managing Editor: Publisher: Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow, MA 01028-0646. Editor: Douglas Cotton, P.O. Box 646, East Longmeadow, MA 01028-0646. Managing Editor: Charles R. Christianson, P.O. Box 646, East Longmeadow, MA 01028-0646. **10.** Owner: Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow, MA 01028-0646. Partners: Ruth A. Christianson, P.O. Box 646, East Longmeadow, MA 01028-0646; Charles A. Christianson, P.O. Box 646, East Longmeadow, MA 01028-0646; Mark A. Fellows, P.O. Box 646, East Longmeadow, MA 01028-0646. **11.** Known Bondholders, Mortgages, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securites: None. **12.** For completion by nonprofit organizations authorized to mail at special rates: Not Applicable. **13.** Publication Name: Commodore World Magazine. **14.** Issue Date for Circulation Data Below: Issue 10 Issues 5 through 10. **15.** Extent and Nature of Circulation: (X) Average No. Copies Each Issue During Preceding 12 Months. (Y) Actual No. Copies of Single Issue Published Nearest to Filing Date. a. Total No. Copies: (X) 3997 (Y) 5300. b. Paid and/or Requested Circulation: 1. Sales Through Dealers and Carriers, Street Vendors, and Counter Sales: (X) 148 (Y) 163. 2. Paid or Requested Mail Subscriptions: (X) 3158 (Y) 3852. c. Total Paid and/or Requested Circulation: (X) 3278 (Y) 3987. d. Free Distribution by Mail: (X) 200 (Y) 1050. e. Free Distribution Outside the Mail: (X) 14 (Y) 17. f. Total Free Distribution: (X) 214 (Y) 1067. g. Total Distribution: (X) 3492 (Y) 5054. h. Copies Not Distributed: 1. Office Use, Leftovers, Spoiled: (X) 529 (Y) 246. 2. Return from News Agents: (X) 0 (Y) 0. i. Total: (X) 4021 (Y) 5300. Percent Paid and/or Requested Circulation: (X) 93.87% (Y) 78.88%.

C-64 AND C-128 OWNERS!

SHOOT FOR THE STARS WITH CMD's

Now C-128 owners can share in the excitement of CMD's new SuperCPU accelerator for the Commodore C-64 and C-128.

Our decision to produce the C-128 version of the SuperCPU was based on the overwhelming number of requests we received from C-128 users.

The specifications of the C-128 version will be virtually identical to its C-64 counterpart, and will support both C-64 and C-128 modes. Naturally, the SuperCPU 128 will have additional RAM, ROM and custom logic to support these capabilities, but will only be available as a 20 Mhz model.

The development of the C-128 version does, however, come at a price. In order to co-develop the two versions, we have had to push back our original delivery date of February, 1996 for the SuperCPU 64. Our new release date for the C-64 version is now April, 1996; the C-128 version will be released approximately 90 days later. At this time we cannot project a retail price with any accuracy. However, we would like to keep the price below $300.00. Therefore, we will NOT be accepting SuperCPU 128 deposits until such time as we can offer an accurate estimated retail price. For additional information on the SuperCPU, please see the article on page 24 of this issue of *Commodore World*.

**Enclosed is my advanced order for the SuperCPU 64/20 (20 MHz).
I understand that a deposit of $50.00 per unit is required.**

Name ____________________

Address ____________________

City ____________________ State/Prov. ____________________

Country ____________________ ZIP/Postal Code ____________________

Home Phone ____________________ Day Phone ____________________

Payment: □ Credit Card □ Money Order □ Personal Check □ COD

Credit Card # ____________________ Exp. Date __________ Bank Name __________

We prosecute anyone placing fraudulent credit card orders

MAIL ORDERS TO:

Creative Micro Designs, Inc.
P.O. Box 646
E. Longmeadow, MA 01028-0646

PHONE ORDERS:

1-800-638-3263
(9:30 AM - 5:30 PM EST, Mon. - Fri.)

American Express Cards · DISCOVER NOVUS · MasterCard · VISA