

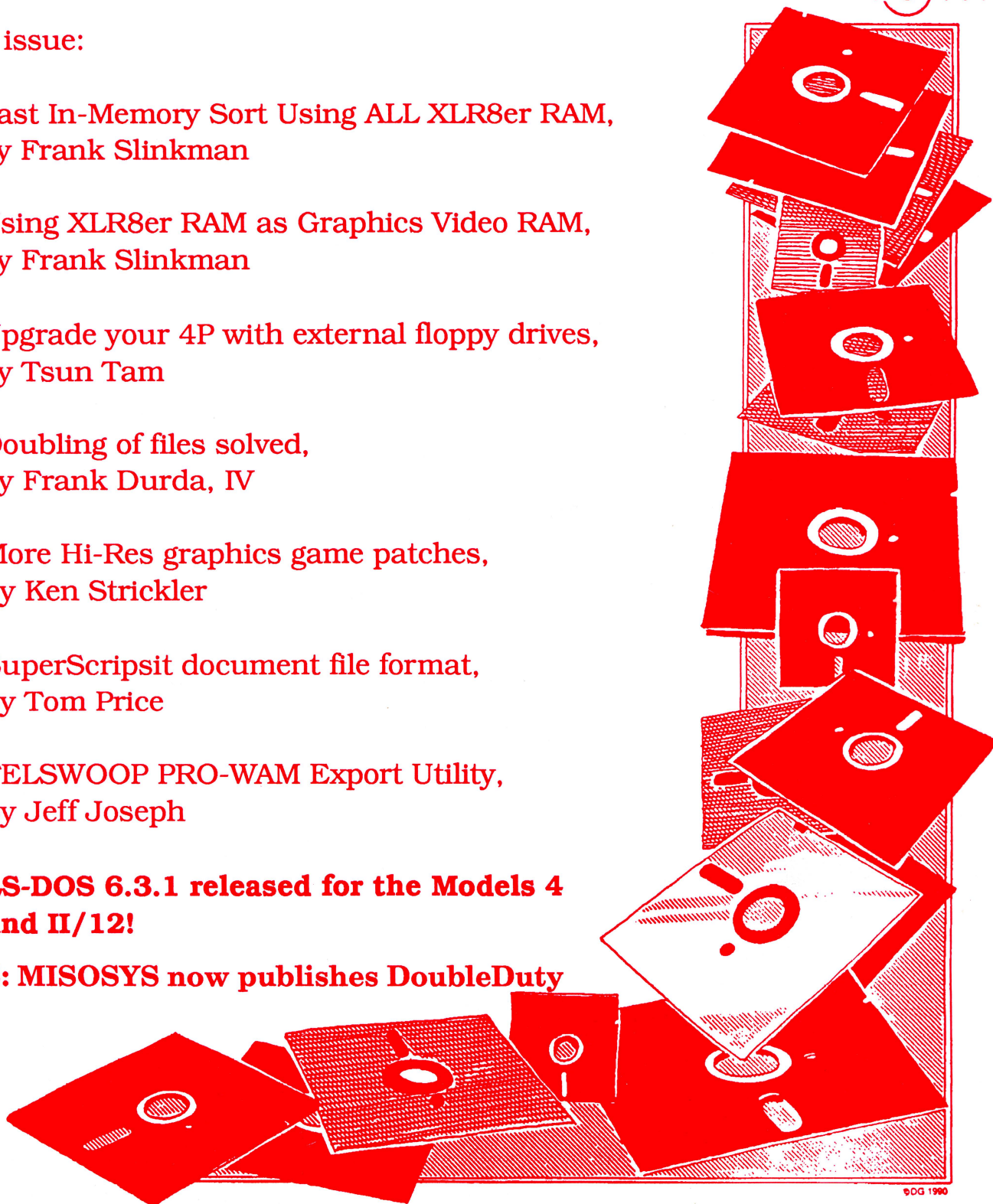
# THE MISOSYS QUARTERLY

## Information worth saving...

In this issue:

- ✎ Fast In-Memory Sort Using ALL XLR8er RAM,  
by Frank Slinkman
- ✎ Using XLR8er RAM as Graphics Video RAM,  
by Frank Slinkman
- ✎ Upgrade your 4P with external floppy drives,  
by Tsun Tam
- ✎ Doubling of files solved,  
by Frank Durda, IV
- ✎ More Hi-Res graphics game patches,  
by Ken Strickler
- ✎ SuperScripsit document file format,  
by Tom Price
- ✎ FELSWOOP PRO-WAM Export Utility,  
by Jeff Joseph
- ✎ **LS-DOS 6.3.1 released for the Models 4  
and II/12!**

**NOTE: MISOSYS now publishes DoubleDuty**



©DG 1990

## Books by Christopher Fara



**MOD-4 BY CHRIS** for TRS/LS-DOS 6.3, 232 pages

**MOD-III BY CHRIS** for LDOS 5.3, 234 pages

**MOD-III BY CHRIS** for TRSDOS 1.3, 210 pages

\$24.95 each, \$39.95 any two, \$59.95 any three

Complete Owner's Manuals for Models 4/4P/4D and Model III, fully updated for all current DOS versions. These beautifully designed books replace obsolete and confusing Tandy and LDOS manuals and addenda. Mod-III editions combine both the "Basic Operations" and "Disk System" manuals in one book. Mod-4 edition has chapters on DOS SuperVisor Calls previously not accessible without a separate "technical" manual. No more fumbling between pages: each subject is contained under a logical, bold heading on one page or on pages facing each other when the book is open, with plenty of blank space for notes.

Written in plain English, the manuals are better organized, with more and better examples for use of DOS, JCL and BASIC; include chapters with examples on interfacing of DOS and BASIC with assembly language; describe in detail popular ROM, RAM and DOS subroutines; and provide lots of useful extra information never before published in the Model III and Model 4 manuals.

"... no matter how long one is using a system, there will be times to look up the manual ... nothing easier than looking into Chris' comprehensive, beautifully arranged and printed treatise ... the organization is exceptional good ..."  
[ Review by Henry H. Herrdegen ]

"... excellent alternative ... not only does it offer information I have not been able to find in the regular and BASIC manuals, it explains in better detail what some of the more arcane commands are good for, or not good for ... here is a manual where you can find it all ..."  
[ Review by Henry A. Blumenthal ]

**JCL BY CHRIS** 30 pages, \$7.95

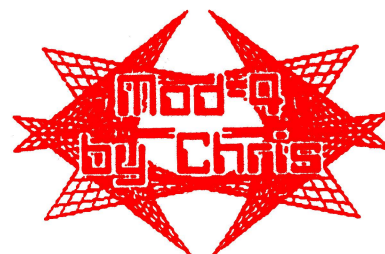
Job Control Language for Mod-III LDOS and Mod-4 TRS/LS-DOS doesn't have to be so confusing as the 'official' manuals made it. Our remarkable, well-organized booklet includes step-by-step explanation how to design, build, DO and compile JCL files, plus a description of other JCL features, and a reference section with examples. We've got rid of the jargon and JCL turns out to be simple, easy, useful and fun.

"... the investment for this instruction booklet was small compared to the welcome education on the expanded use of my computer ... thanks to Chris and his way of explaining things in a simple and logical fashion ..."  
[ Ray Stanley ]

**Z-80 TUTOR I** 40 pages, \$9.95

Introduction to assembly language programming for beginners, based on Chris' popular essays in Computer News 80 (1989 volume) and revised in a book form. Covers memory and machine code concepts, typical steps in assembling, and application of basic Z-80 instructions and subroutines in Mod-III and Mod-4. Plain talk will quickly ease you into useful programming, and practical examples will give immediate satisfaction.

**Published and Distributed by**  
**COMPUTER NEWS 80 PO Box 680 Casper, WY 82602**



S & H Add:  
\$4.00 for one in US,  
\$6.00 for two in US,  
\$8.00 for three in US.

\$6.50 for one to Canada  
\$9.00 for two to Canada  
\$12.00 for three in Canada  
Overseas write for S&H Charges



*The MISOSYS Quarterly* is a publication of MISOSYS, Inc., PO Box 239, Sterling, VA 22170-0239, 703-450-4181.

*The MISOSYS Quarterly* is published four times a year in February, May, August, and November. Published by Roy Soltoff

## THE MISOSYS QUARTERLY

### subscription rate information

Each issue of TMQ has information on MISOSYS products, programs and utilities, patches, significant messages from our CompuServe forum, and articles on programming. Not only that, TMQ will keep you up to date with information, news, and announcements concerning our entire product line and related machine environments. Subscription cost varies by rate zone as follows:

A = \$25; United States via 3rd class bulk mail  
 B = \$30; Canada, Mexico, or United States via 1st Class  
 C = \$32; Colombia, Venezuela, Central America via AO Air  
 D = \$35; South America, Europe, & North Africa via AO Air  
 E = \$40; Asia, Australia, Africa, Middle East via AO Air

### TMQ Toolbox

*The MISOSYS Quarterly* is published using the following facilities:

The hardware used for development of the "camera ready" copy consists of an AST Premium/386 computer (20 MHz) equipped with 5 Megabytes of RAM, a Seagate ST4096 80-Megabyte hard drive, a Colorado Memory Systems DJ10 tape backup device, a NEC Multisync II color monitor driven by a Video Seven VGA card, an AST TurboScan scanner (Microtek MS300), and a NEC LC-890 PostScript laser printer.

Text is developed, edited, spell-checked, and draft formatted using Microsoft WORD Version 5.0; Submissions on paper and letters are scanned and converted to text using Read-Right optical character recognition software by OCR Systems. Final page composition is developed using PageMaker 3.0 by Aldus. Cover art and clip art comes from CLIPPER, a product of Dynamic Graphics.

## Table of Contents

<b>The Blurb</b>	2
FIX631A for LS-DOS 6.3.1	4
PD Software Librarian	4
Out of print TMQ's	5
DISK NOTES 4.3	5
<b>Letters to the Editor</b>	6
Floppy drive resistor packs	6
DoubleDuty doesn't support hires	8
DISKCOPY needs a patch	9
Resource: Companies	13
<b>DOS Subjects</b>	14
Doubling of files solved, by Frank Durda, IV	14
Restoring erased files using Toolbox/Toolbelt	18
HDBOOT, by Phillips modified by Toenniessen	19
More hires graphics game patches, by Strickler and Soltoff	20
SuperScripts document format, by Tom Price	21
Patches to 6.3.1 for Alpha Tech, XL8er, and MHDT34 clock	24
<b>Contributions</b>	27
Upgrade your 4P with external floppy drives, by Tsun Tam	27
Fast in-memory sort using all XLR8er RAM, by J.F.R. Slinkman	34
Using XLR8er RAM as a graphics video RAM, by Slinkman	39
FELSWOOP PRO-WAM export utility, by Jeff Joseph	44
Using DoubleDuty with PRO-WAM, by Roy Soltoff	50

### List of Advertisers

Computer News 80	IFC
Gooth Software	18
HPB VECTOR WSO	13,51
M.A.D. Software	58,59
MISOSYS, Inc.	48,49,52-57,RC
Pacific Computer Exchange	IRC
T/Maker Research Company	60
TRSTimes magazine	60

## Points to Ponder

**Base 0 counting:** Last issue I stated, "Look at the various topics covered in the seven-teen sections in our forum.

- 0 General/NewUplds
- 1 Languages
- 2 MISOSYS Products
- 3 M1/3 Programming
- 4 Mod4 Programming
- 5 LDOS 5 Support
- 6 TRSDOS 6 Support
- 7 Applications
- 8 Hardware Hacking
- 9 MS-DOS Topics
- 10 Programmer's Jrnl
- 11 Communications
- 15 Beta-Testing
- 16 Open Forum
- 17 Sysop Business"

Nobody caught me on that one; there's **eighteen** sections. And I thought I had some very careful readers!

**Losers and Winners:** Here's an interesting statistic. According to various industry publications, Unisys lost \$648 million dollars in the third quarter of 1989. Just how does a company stay in business losing that much money! They're not the only ones, lots of hi-tech companies are losing many millions. Does that make me feel better when I make but a few bucks and the future looks dim? Not necessarily, because when the big guys lose, they seem to gain.

If you or I get behind in the mortgage, we just might lose the house. When a big guy loses hundreds of millions, the banks just seem to be there to help them out. But not all big companies are losers. Here's some interesting figures: The top computer-based advertising spenders in December 1989 were (1) DEC at \$1,545.7K, (2) DELL at \$1,244.7K, (3) Northgate at \$1,224.7K, (4) Toshiba America at \$1,081.3K, and (5) IBM PC at \$987.9K. Tandy was eighth on the list at \$787.7K. When you look at printer-based, you find HP up there at \$1,874.5K!

## The Blurb by Roy Soltoff

A quick peek at numbers of outlets reveals the following: Tandy (6994), Inter-Tan (2488), Intelligent Electronics (793), ComputerLand (771), Microage (540),... MISOSYS (1). We're hanging in there. Note the power of Tandy in those figures.

Based on a poll of 162 PC Users, the preferred applications fall into the following categories:

- 98% spreadsheet
- 98% word processing
- 96% mainframe access
- 90% database management
- 85% graphics
- 84% desktop publishing
- 76% project management
- 74% accounting
- 63% electronic mail
- 63% CAD (computer aided design)
- 61% PIM (personal info. mgt)

With spreadsheets taking up 98%, and Lotus taking 80% of the market, no wonder they're rolling in dough. Care to forward your own list of preferred applications? If I get enough input, I'll summarize it for the next issue.

**Floppy drives:** Manufacturers expect to ship about 31 million 3.5" floppy drives in 1992, about 77% of all floppies shipped that year. The 5.25" drive is on the way out. Don't forget that we sell 720K 3.5" drives in a 5.25" 1/2 height form factor housing at competitive prices.

**Speedy Z80:** According to a recent issue of *Electronic News*, Zilog is now shipping a 20MHz version of its Z80 processor claimed to have a peak execution time of 5 Mips (millions of instructions per second). Samples are available for \$18. Any hardware hacker want to beef up their TRS-80?

**Credit Card Orders:** Just a reminder to anyone submitting an order using a credit card, I need the expiration date! Also, don't forget to include all of the digits:

MasterCard account numbers are 16-digits in four groups of four; the last four digits are usually in the position of the hologram and sometimes hard to see, but they're there. VISA account numbers are either four groups of four or one group of four followed by three groups of three. Errors in account numbers usually delay your order. Also, please include a daytime telephone number (that's daytime referenced to Eastern Time). Phone numbers are absolutely essential for foreign hard drive orders.

**Calculators:** Hewlett-Packard may have finally bitten the bullet. I read where they recently debuted a new calculator, the 19BII, which uses both reverse Polish notation and the more traditional algebraic entry method. The calculator contains over 450 preprogrammed functions, has a four-line 23 character dot matrix display and an infrared link to a companion printer.

**Games:** If you're in to games, don't forget that I stuck my neck out to acquire licensing rights to five powerful action games previously published by The Cornsoft Group: **Frogger (tm)**, **Scarfman**, **Bouncezoids**, **Crazy Painter**, and **Space Castle**. These are exceptional games with great video and joystick support (even the new MISOSYS joystick). All games are for Model III/I (or 4 in III mode). I also have available the three game sets sold by PowerSoft: **Leo's Greatest Hits**, **Kim Watt's** greatest hits, and **Lance Miklus'** greatest hits.

**Tandy Software Products:** Tandy is getting closer to firming up a distribution procedure for the discontinued software they own the rights of. But I really stirred up a pot full. Apparently, other folks have been asking about other types of products no longer available. So it appears that Tandy is working on bringing to availability many "ancient" and "not-so-ancient" products that are no longer economical for store retail efforts. It looks like their



Consumer Mail department will be growing considerably. As soon as they iron out some kinks relating to product support, the program should be in effect. So hang in there a little longer if you are trying to find some older product not currently available. I'm predicting that the program should be underway before TMQ IV.iv hits the streets. Incidentally, Consumer Mail's order number is 800-351-3133.

**TMQ Revision:** I took a slightly different tack in this issue of THE MISOSYS QUARTERLY. To begin with, I have noted over the past that TMQ takes considerable effort on my part. I had to reduce the time I spent on producing an issue so I could go back to some **real programming!** I have discontinued publishing excerpts from our CompuServe forum. That's because it used to take me about a week of effort to review three months of downloaded messages, parse them into topics, restructure and edit threads, then groom the items into the various chapters I used in TMQ. I reduced this issue to articles, and correspondence. It typically takes less time to scan letters (some folks even provide long letters on disk making it even easier to get them into the Quarterly). So for now, this issue's style is the style I'll use for the next issue.

Based on the current subscription level, I also have to continue to justify the effort. So I will have to make a conscious decision to continue publishing past TMQ issue IV.iv. Right now, that issue will be the last; if I become convinced that TMQ can be produced with minimal effort to a sufficient number of subscribers, then I'll probably continue it. I've put out some pleas for column contributors, and have gotten some input. What's needed is some regular contributions. So let's see what happens over the next three months.

Make a note that this issue has no coupon. It turned out to be too costly. So don't start thinking that it fell out. Also, I dropped the 25% discount for the DISK NOTES disk. Out of the dozens of orders I get for the disk, most are just for the disk. It's too costly to process an order for \$7.50!

**Late Breaking Patch:** Did anyone really think I would be able to release a new version of the Model 4 operating system

without having to quickly come up with a patch to fix something?

The source code used to generate LS-DOS 6.3.1 constitutes 207 files taking up 1331K of disk space. I conveniently keep the DOS modules divided into three 720K Subdisks; one each for system, library, and utility files. That makes it useful to backup the three subdisks as 80D2 images to a 3.5" floppy drive (one of the Fujitsu's that I sell).

The generation of a 6.3.1 disk is controlled by a 567-line Job Control Language procedure file which generates a complete 6.3.1 diskette in a little less than one hour running on a XLR8er'd Model 4D with a 40 Megabyte MISOSYS Hard Drive; one command line starts the JCL file. I used to joke back in the LDOS Model I days that I had a KSM file which used the "L" key to generate the one command line needed to initiate the JCL file for LDOS 5.1 generation; that was my LDOS key!

In spite of the automated procedure, the DOS is so complex, with so many interrelationships, that it is virtually impossible to make a change in one module without causing some adverse effect in another. That's why every change has to be carefully considered. In spite of careful consideration, no amount of testing can catch everything that could go wrong.

Stan Slater reported a problem of getting an "Unknown error code" when using DISKCOPY to copy a 2-sided diskette made with the 6.3.1 FORMAT utility; even the 6.3.0 DISKCOPY gave the same error! After isolating the problem, it turned out to be one of those "gotcha's"; it wasn't a change in DISKCOPY, but in FORMAT. I even used DISKCOPY frequently to make backups of the source file subdisks - which are 2-sided images, so why the problem? Turns out that I didn't try it on a 2-sided floppy formatted with 6.3.1!

LS-DOS 6.3.0 had a "bug" in the FORMAT command which made the ERN of DIR/SYS identical to the cylinder size. That's okay for single-sided floppies, but when the cylinder size exceeds 34 sectors (as it does with 2-sided floppies or hard drives with a cylinder composed of more

than one track), the directory ERN should remain at 34 sectors. Although nothing in the DOS would be upset at a ERN larger than 34, other programs which "read" the DIR/SYS as a file would read past the end of the actual directory. Also, if a hard disk system which used the DOS formatter to add system information would have a 256-sector cylinder size, the ERN of DIR/SYS would show up as 0!

One of the changes I made in FORMAT was to correct the ERN of DIR/SYS if the number of sectors per cylinder exceeded 34. But I also changed it to write only a number of sectors equal to the size of the directory. Since the directory sectors are written with a code to generate a Data Address Mark (DAM) on the floppy diskette different from a normal data sector, and a 2-sided floppy diskette has 36 sectors, the last two sectors of the cylinder which contained the directory would not have been written to generate the directory DAM. Nothing is actually used in those two sectors, anyway. And BACKUP handles it perfectly because BACKUP can accept either "No error" or "Attempt to read system sector" as error codes when it reads from the source disk.

Unfortunately, DISKCOPY uses the @RDSSC service call which demands the latter error. So when DISKCOPY was copying the directory of the 2-sided disk formatted by FORMAT 6.3.1, it got to sector 22H - which is the 35th sector of the cylinder - and received an error from @RDSSC. The error code was actually X'FA', a strange, but easily explained error.

The @RDSSC service call should be used only to read the directory sectors. It makes a dual attempt in order to provide additional security in case of random failure. The error code used to report the "Attempt to read system sector" is a "6".

To keep the code at a minimum, @RDSSC subtracts 6 from the expected error-6 so that a ZERO result would prevail without any errors. If @RDSSC is used to read a standard data sector, the FDC driver returns a "0" to indicate no error. But @RDSSC subtracts 6 resulting in X'FA'! Thus, an "Unknown error code" prevails. There's few modules in the DOS (backup,

format, etc) which use @RDSSC; and when they encounter an error code, it is translated to GAT error, HIT error, or Directory read/write error, depending on what sector of the directory is being referenced. But DISKCOPY doesn't translate the error; that's why it generates the "Unknown error code".

The following JCL file can be invoked to change FORMAT to continue to write all sectors of the directory cylinder but still keep the DIR/SYS ERN correct. This will counteract DISKCOPY's usage of @RDSSC. Note that the patch also updates the "Level" in BOOT/SYS to level "1B". Note also that the BOOT patch will only be valid for a disk formatted with 6.3.1's FORMAT command as that is the only way the password becomes "SYS-TEM6".

```
. FIX631A/JCL - 03/08/90 -
Cause FORMAT to write all
sectors of DIR cyl
. Apply via, DO FIX631A
(D=d) where "d" is drive to
patch
//if -d
//. Must enter drive to
patch!
//quit
//end
patch boot/sys.system6:#d#
(d02,1f=42:f02,1f=41)
patch format/
cmd.utility:#d#
(d03,7f=21:f03,7f=32)
//exit
```

## BBS' and Clubs

I used to set aside space in TMQ as a service to the Model I/III/4 community of users to publicize three things: a list of **phone numbers** of companies still servicing and supporting this market, a list of **public computer bulletin boards**, and a list of **computer clubs** which support the TRS-80 user. No one has ever remarked on the lists, pro or con. I have received a few updates on BBS' and clubs. I've

decided that perhaps a monthly or bi-monthly publication may be the better source for those lists; for now, I'll keep the list of companies.

## TMQ Schedule

Our target for mailing the *THE MISOSYS QUARTERLY* is the last week of the respective month as follows: Spring issue in February, Summer issue in May, Fall issue in August, and **Winter issue in November**. Due to the efforts to get LS-DOS 6.3.1 generated and released, this issue is a little behind target!

Note that your mailing label usually has the expiration date of your subscription. For instance, those with "90/05" complete their subscription with this issue. If you want to save me the cost of mailing a renewal notice, send in your renewal fee quickly. I usually wait about a month after TMQ is mailed before sending out renewal notices.

Because of the uncertainty of continuing publication past volume IV, issue iv, I attempted to have last issue's expiring subscriptions renew for just two issues. If you want to renew for the next issue, send in one-fourth of the standard subscription fee: \$6.25, \$7.50, \$8.75, or \$10. **Note that I will not be sending out renewal notices to those folks with a subscription expiring with this issue.** Subsequent to the publication of IV.iv, I'll be deciding on the question of continuing into Volume V.

## TMQ advertising

If you are interested in reaching a dedicated TRS-80 audience, consider *THE MISOSYS QUARTERLY*. If you have a TRS-80 Model III or 4 related product to sell, you can reach these buyers by placing your advertisement in our publica-

tion. TMQ is read world-wide. Our subscribers are predominantly in the United States; however, we do have a significant number in Canada, Europe, and Australia.

Current space rates are as follows:

Full page	\$125
Half page	\$75
Quarter page	\$50
Ninth page	\$20

Note the ninth-page ad layout designated 'The Marketplace', which would have been the last page of this issue had I received any Marketplace ads. I compose this so you have no artwork charge. Just submit your text. We accept only black & white ads; however, ads for our inside covers are printed in the same color as the cover (TMQ alternates between PMS colors: green 354, purple 266, blue 293, and red 199). If you would like to place your ad in *THE MISOSYS QUARTERLY*, give me a call.

Subscribers, please mention TMQ when you call or write to these advertisers (I know, there's fewer these days); they deserve your support.

## PD Software Librarian

Vic McClung has volunteered to be the librarian for the collection of TRS-80 public domain diskettes. Henceforth all requests and contributions be directed directly to him at:

Vic McClung  
914 Crescent  
Sikeston, MO 63801  
USA

Note that if you upload a "public domain" file to our CompuServe forum [PCS-49], and want it to receive general distribution, please also mail a copy on disk to Vic. There is no legal provision for downloading files from CompuServe and redistributing them. Some of our readers



who do not have access to our forum have an interest in those submissions. So if you want to help out the most numbers of fellow users, don't limit your submissions to just one source.

### DISK NOTES 4.3

Each issue of *THE MISOSYS QUARTERLY* usually contains program listings, patch listings, and other references to files we have placed onto a disk. DISK NOTES 4.3 corresponds to this issue of TMQ. If you want to obtain all of the patches and all of the listings, you may conveniently purchase a copy

DISK NOTES is priced at \$10 Plus S&H. The S&H charges are \$2 for US, Canada, and Mexico, \$3 elsewhere. I have dropped the 25% discount for purchasing the current issue with the coupon; the coupon has been dropped, too.

### Out of print TMQ's

For out of print issues, we are providing back issues of *THE MISOSYS QUARTERLY* via copier reprint. The price is \$12.50 plus \$2.75 S&H in the U.S. and CANADA. For foreign zone D, the S&H rate is \$5.50; zone E is \$6.50. The price for regular back issues still in print is \$10 + S&H. We are currently out of print on all issues of Volume I and Volume II. **Special deal now in effect for all four issues of Volume III: just \$24 + S&H \$5 (US), \$6 (CAN), \$14 ZoneD, \$20 (ZoneE).** Here's a synopsis of past issues:

**Volume I** See the index in issue III.i.

**Volume II** See the index in issue III.iii.

**III.i** Reading NEWDOS/80 disks; An LB archival utility; Popup Application Window; XMODEM in

C; Getting into computer math, part I; TMQ Volume I index.

**III.ii** Getting into computer math Part 2; Writing interactive RATFOR/FORTRAN programs; PRO-EnhComp: a review; Desktop publishing and the Model 4; A better TERM/APP; adding floppy drives; and a new XLR8er interface.

**III.iii** The CRC program; PG: a page display program; Locating high memory routines; FIXMA3; Jumbo tape backup for PC clones; New style for TMQ using Pagemaker; and an Index to Volume II.

**III.iv** Checking for a file from Model 4 BASIC; Surviving the Hard Disk crash; An "interview" with Niklaus Wirth; Keep your printer clean and oiled; On-line HELP with PRO-WAM; MISOSYS announces availability of Hard Drives; Logic in the C language.

**IV.i** Cataloging files with a word processor; Page display PRO-WAM application; File undating with FUNDATE; Array load routine for BASIC; XLR8er and the GT-180 graphics board.

**IV.ii** Printing from BASIC without cutting words; LOAD100 for Model 100; Generating date/time stamp; Favorite recipes; Some BASIC routines.

### New Product News

#### LS-DOS 6.3.1 released

Here's a new version with something for everyone. See the ad on the rear cover! Note that a Model II/12 version is now available.

#### MISOSYS Hard Drive

Since September 1989, MISOSYS has been shipping its 20 and 40 megabyte hard drive package with Hardware real time clock and joystick options. Details on this package and its various options were printed in TMQ 4.1; specific infor-

mation is available on request. We also sell piece parts: H/A @ \$75, HDC @ \$95.

### DRAM and PALs

DRAM prices have bottomed out - but now on the upswing. I keep a supply of 256K-150ns DRAMs on hand for our XLR8er board, and 64K-150ns DRAMs for motherboard replacement. I'm making these chips available for separate purchase. Note that we generally stock DRAM "Pulls"; chips used previously in sockets but pulled, refurbished, and tested. All DRAMs are 100% tested again by us before shipping. In addition, MISOSYS has acquired a BP Microsystems Logic Programmer to program the Programmable Logic Device (PLD) for our SCSI host adaptor. As this piece of hardware can also handle a wide assortment of PALs and PLDs, I can be a source for the memory expansion PAL (U72) needed to upgrade a 64K 26-1069 Model 4 to 128K. Prices as follows:

**64K-15 DRAM @ \$2/chip**  
**256K-15 DRAM @ \$4/chip**  
**U72 PAL16L8-25CN @ \$8/chip**

### Ribbon Cable Assemblies

MISOSYS has also acquired a Cirris Systems cable tester. This interesting piece of equipment performs a 100% test for shorts and opens on cables. It handles many different kinds of connectors based on switchable test assemblies. I have acquired an assortment of test assemblies to support the kinds of connectors typically associated with the TRS-80 micro-computer. Thus, I'm custom fabricating low-volumes of cables according to specifications, as well as providing standard replacement cables for your needs. These are all using unshielded ribbon cable. I can provide cables using DB-25 M/F, 34-pin edgcard M/F, 36-pin printer, 50-pin edgcard F, 50-pin SCSI M/F, 34-pin Header M/F, as well as DB9 Male. Need a replacement RS232 or printer cable? Probably about \$10-\$15.

**Kel-AM 34-pin male edgcard @ \$8**  
**Kel-Am 34-pin female edgcard @ \$5**

## Floppy Drive Resistor Packs

**Fm Lawrence Rossiter, Victoria, BC, Canada:** Dear Roy, I would like to point out an error in Pete Granzau's reply to Walter Sullivan on Page 30 of TMQ Vol IV.ii concerning the terminating resistor pack used in disk drives. This pack is not used with the internal drives, only on the last external drive. See the TRS-80 Model 4 Technical Reference Manual (26-2110), Section V, 5.1.7, Page 83 in my copy. It states in part "Note. The internal drives in the Model 4 computer are not terminated".

I use three drives, 2 internal and one external, the resistor pack being in the external drive 2. About a year ago I swapped drives 0 and 2 (0 was somewhat noisy when it contained a disk) but forgot to change over the pack so it was then in drive 0. Before and after reassembly of the computer I put disks in all drives and got directories so assumed everything was OK. My next task was to make up a JCL file but after typing BUILD drive 0 ran for a few seconds then the computer locked up. It would not reboot using that disk. I did the same with another disk and had the same results. Finally remembering the resistor pack I moved it back to drive 2 and everything has since worked correctly.

Neither of the disks were write-protected, neither would boot and when placed in another drive would not list a directory. Both contained the only copies of a 27K programme that I valued very highly but did not look forward to having to retype. I was able to restore one disk with the help of SU+ and learned a lot about using that programme, about disks and about putting my backups safely out of reach!

## Feedback on "Printing From BASIC"

**Fm Murray W. Diller, Phoenix, AZ:** Re: TMQ IV.ii PP 40 Mr. Ainsworth's Article: Gentlemen, May I extend my very sincere thanks and appreciation for Mr. Ainsworth's article PRINTING WITH JUSTIFIED RIGHT MARGIN in the subject issue. It provided the necessary nudge to do something wished

for for  
some time.

My bag of commercial programs contains a most excellent word processor which is never used because of the lack of desire and retentive capability of committing some 70 pages of instructions to memory when it is used only a few times a week.

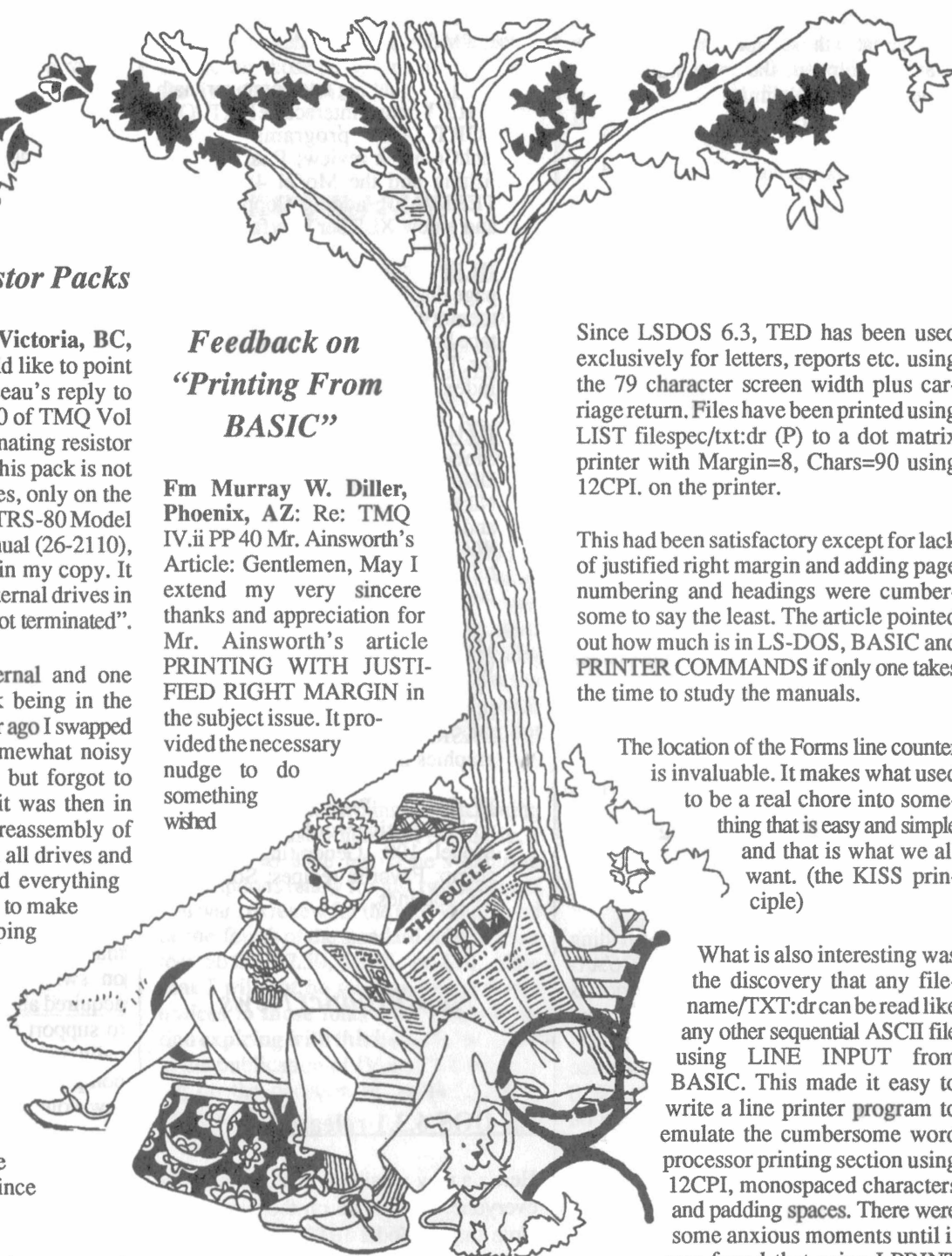
Since LSDOS 6.3, TED has been used exclusively for letters, reports etc. using the 79 character screen width plus carriage return. Files have been printed using LIST filespec/txt:dr (P) to a dot matrix printer with Margin=8, Chars=90 using 12CPI. on the printer.

This had been satisfactory except for lack of justified right margin and adding page numbering and headings were cumbersome to say the least. The article pointed out how much is in LS-DOS, BASIC and PRINTER COMMANDS if only one takes the time to study the manuals.

The location of the Forms line counter is invaluable. It makes what used to be a real chore into something that is easy and simple and that is what we all want. (the KISS principle)

What is also interesting was the discovery that any filename/TXT:dr can be read like any other sequential ASCII file using LINE INPUT from BASIC. This made it easy to write a line printer program to emulate the cumbersome word processor printing section using 12CPI, monospaced characters and padding spaces. There were some anxious moments until it was found that using LPRINT CHR\$(27);CHR\$(1 to 9); on

DMP133's was great except for 6 resetting the forms filter line counter and 9 which has another priority. Both made the line printer do some strange things but this was easily overcome. For some reason as yet unknown it will not work in NLQ mode.





TED may have never been designed to be a full featured word processor but it does very well when used with this printer program. In my field you go with what works! And this does.

Now if I can only tie it in with a spelling checker it will be perfect!

Thanks again to you and Mr. Ainsworth and please continue with articles such as this one.

### *On MISOSYS' Acquisitions*

**Fm Kenneth M. Strickler, Stanwood, WA:** Dear Roy, Congratulations on the GREAT 1989 COUP! Getting the rights to distributing DOUBLE-DUTY and the POWERSOFT software insures continued availability for the MODEL III and 4 users of some excellent software! (How about DOSTAMER from George Fischer, P.O. Box 252, Temple City, California 91785-0252 - and possibly the Versa Series Business package from COMPUTRONICS. DOSTAMER CRASHES if you try to have both PROWAM and DOSTAMER on line together, and VERSA SYSTEMS needs a module to replace the CMD'O sort routine plus some shortening of some of the command lines to work with LS-DOS, but it is written in basic. I see that EnhCOMP has a replacement for the CMD'O routine, but I haven't finished the conversion to MOD 4 yet! It's only been a couple of years in the works - PARTIME- maybe soon!)

For those looking to keep up the MAGNETIC DISK strength, TOOLBOX (MOD III) AND TOOLBELT (MOD 4) contain a Power Verify Utility (PVU) and a Power REFORMAT (PREFORM) which really does the trick. It is supposed to be run on FLOPPIES, but it seems to work on DISKdisk partitions as well. The PVU program tries to read an entire disk - only 1 time! - and any errors are duly noted. If you have any errors, depending on how long the disk has been since re-magnetized, the PREFORM program can usually correct it. I have had several disks that the

PVU command has indicated as READ ERRORS, and the PREFORM has corrected each one. Apparently, the PREFORM program tries multiple reads, and from the way it really tries to find the missing data, tries RE-SEEKS, and I would guess XOR of the data read. I have both TOOLBOX and TOOLBELT in my UTILITY partition. There are a bunch of other utilities included, making the package truly a GREAT VALUE.

For those who really have to get into TROUBLE, the SUPER UTILITY package is another powerful addition. It has to be run from floppies, and is pretty tightly locked up, software-wise, requiring kind of learning another language, but surely demonstrates the ability of the TRS-80 under to control of a SUPERB Programmer - Kim Watt. The features are way to extensive to list here!

As for DOUBLE-DUTY, it allows a 128K model 4 to be partitioned into 2 - 64K machines and a machine which can run any of the LIBRARY commands. It requires a LOT of the LOW MEMORY storage area, so cannot be loaded if you need to many drivers. Some experimentation will have to be done in order to get just the configuration that you might need. I found the program particularly valuable when I was experimenting with TBA (The Basic Answer) writing FULLY DOCUMENTED programs in the TBA structure, with MUCHO COMMENTS, GLOBALS et. al., and compiling in one partition, saving, switching partitions to one containing BASIC and running the program without having to repeatedly load TBA or BASIC! I am sure that running EDAS or MRAS in one partition and running the NEW PROGRAM in the other would work equally as well. Another FINE UTILITY. I wonder if DOUBLE-DUTY could be structured to run a 'tasking' environment, with a 'foreground' and 'background' task? Maybe the real time clock in the HARDDISK could be used to change partitions? Maybe with an XLR8er and the extra Memory? Plenty of opportunities here!

Again - ROY - A REAL COUP!!!! I hope that everyone will have the forethought to sign up for your BUY a POWERSOFT product and get a MISOSYS, Inc. product

of equal value for FREE. By the time that this letter finds its way into the TMQ, if it does, the FREE will be GONE.

Actually - now I have gotten to the part of the letter I am hot to trot on! I remember that when I got the last LS-DOS 6 from you for resale, there was a new updated price sheet enclosed for the MISOSYS products. I did not keep a copy of it, and I was wondering if you would send me a copy ASAP, so that I won't miss the hot deals. I already have most of the POWERSOFT and MISOSYS products, but might be able to get all of the rest! also I need to know if the dollar amounts are accumulative, ie \$100 worth of POWERSOFT products allows \$100 credit toward MISOSYS products? can the difference be made up in cash? enclosed are my LB beta disks for the upgrade - ah, thank you very much !!!!!. I will keep the blue card until I place my order, after I get the new price sheet.

Also, I assume that I need a POWER 'Y' cable and an XT cable, plus a 'BUBBLE' to upgrade to 80 MEG, am I right?

This WINTER 89/90 TMQ is REALLY CHOCK FULL 'O GOODSTUFF! Just seems to be getting BETTER 'N BETTER.

### *LB needs [ENTER] before [F3]*

**Fm Danny C. Mullen, APO Miami, FL:** Dear Roy, I just got done adding about 50 records to "CN80/LB" using beta version of LB. Upon review, I found that field #7 was not being written to disk most of the time. This started around record 44 and goes on until the end of the file.

I tried the old LB version 1.0 and all is well, so I don't know what it is. This is my first try at extended use of beta version. No other filters, drivers, etc. loaded when using it. A copy of the data file is enclosed. As you can see, fields 1-6 are on

the disk, but 7 is mostly blank. Though some that were written only had part of the data stored.

I will continue using the original LB version 1.0 until I hear back from you. Other than this, beta version looks pretty good.

**Fm MISOSYS, Inc:** Dear Danny, I got around to checking into the LB beta problem you reported, concerning "losing" the data in field 7. I can find no problem. My suspicion is that for those records where you "lost" the last field, you saved the record with <F3> before depressing <ENTER>. Remember that LB requires you to complete your edits (or entries) for a field via the <ENTER> key, then depress <F3> to save it. If the field is still in edit mode, no changes you typed for that field will be made to that field. See page 86 of the LB manual, "Be sure and press the <RET> key after changing data in a field. When all changes are complete, press the <F3> key to save the record on disk."; this is listed under Update or Delete records, but it is again noted at the bottom of page 81 for Add records.

### ***DoubleDuty Doesn't Support Hires Graphics***

**Fm Don O. Coffin, Mesa, AZ:** Gentlemen, I recently purchased a copy of DOUBLEDUTY (S/N #00040) in response to your ad in Computer News 80. I am very pleased with the program, but I am confused about a statement in the Operator Manual for this software.

On page 16 it says "Double Duty will not access a graphics card" and that "you cannot perform graphics operations while Double Duty is loaded" This was a great disappointment to me, as my principal reason for purchasing this program was to address my Microlabs Hi-Res Board simultaneously through the two partitions provided by Double Duty — via PRO-DRAW in Partition 1; via GBASIC and

some home-brew assembly language routines in Partition 2.

Despite the manual's statement that this would not work, I tried installing Double Duty and then loaded the two programs above. This arrangement appears to work just as I had hoped it would. The only eccentricity occurs when I am in PRO-DRAW (TM) and press the [CAPS] key with one of the Function keys to change partitions. This has the same effect on the cursor as pressing the Up-Arrow. However, this action of the [CAPS] key always occurs in the PRO-DRAW program, whether or not it is used in conjunction with DOUBLE DUTY.

Perhaps the phrase Graphics "Card" refers to something other than a HiRes Graphics "Board", but I don't know what that could be on a Model 4. Is it possible that DOUBLEDUTY can address the Microlabs Hi-Res board but not the Tandy board? Or does this limitation apply to early versions of either the Model 4 hardware, TRSDOS 6.x operating system, or the Tandy-marketed version of DOUBLEDUTY? In any case I am happy to report that DOUBLEDUTY does work with my late Model 4P (gate array) LS-DOS 6.3, and the Microlabs Hi-Res board and related software.

I believe you would be well served to look into this statement and correct it in future revisions of the manual. Otherwise you are discouraging a very useful application of your excellent software for those Model 4/4P/4D owners who have a Hi-Res board installed.

**Fm MISOSYS, Inc:** Dear Don, Concerning DoubleDuty used on a machine with a hi-res graphics card: You are confused about what is meant by the statement, "DoubleDuty will not access a graphics card". That does not say that DD will not run on a machine with a graphics card, only that DD will not do any access of the hires graphics memory when partitions are swapped. Don't forget that video memory must be saved and swapped when a partition is swapped out. There is not sufficient spare memory to also save and swap hires graphics video memory; thus,

if you are running more than one program in partitions which are using the hires graphics, they will both be accessing the same video RAM and will corrupt each other's screen image. I would gather that the display of hires graphics - especially on the Graphyx Solution board which can simultaneously display both hires (from its own RAM) and low res (from the Model 4's video RAM) would also be a problem as any hires graphics images left on the screen from a graphics program swapped out would remain on the screen. That is sufficient to say that "DoubleDuty will not access a graphics card".

As far as the CAPS key interfering with PRO-DRAW, that program may well be doing something very odd with the CAPS key. DoubleDuty does do everything in its power to inhibit the generation of CAPS once it is determined to be depressed simultaneously with a function key. If PRO-DRAW were to use an interrupt task to check for CAPS, then that would certainly be the culprit as DD must pass the CAPS indication until it senses both CAPS and a function key during its interrupt task process.

### ***Directory Pointers; Floppy Installation***

**Fm David J. Kelton, Richmond, VA:** Dear Roy, As a faithful reader of TMQ, I would like to wish you a Merry Christmas and a Happy New Year. I hope you and your family are doing fine considering the weather we've had over the past few days.

After I last wrote you on October 9 regarding my not being able to get the BACKUP command in LDOS 5.3 to work on my hard drive, I decided to do a few things to try to locate the problem. Although these were unsuccessful, I'd like to tell you about them in the event they can be of any help.

First, I went back to my original 5.1.3 disks; made a backup; installed 5.3.0 as per the instruction sheet; reviewed all my /FIX files against the data in TMQ; and



finally installed those patches. BACKUP would still not work. But this time there was another curious development. When I tried to BACKUP the /SYS files from my system floppy to the HD "system" disk (which at the time was :6), the BACKUP command issued an "Unknown error" command and quit. Using DEBUG, I observed the DCT at 4700H both before and after the attempted BACKUP. The contents before were:

```
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9
C3 83 45 54 61 0C 27 11 45 14 Floppy #1 :0
C3 83 45 44 42 FF 27 11 45 14 Floppy #2 :1
C3 70 FE 0C 12 00 98 1F 2F 4C Hard Hd#3 :2
C3 70 FE 0C 13 00 98 1F 2F 4C Hard Hd#4 :3
C9 83 45 00 00 00 27 00 00 00 (Disabled) :4
C9 83 45 00 00 00 27 00 00 00 (Disabled) :5
C3 70 FE 0C 10 00 98 1F 2F 4C Hard Hd#1 :6
C3 70 FE 0C 11 00 98 1F 2F 4C Hard Hd#2 :7
```

On return from the BACKUP command, all information was the same for drives except :6. Its data was:

```
C3 70 FE 0C 30 00 3A 1F 2F 4D Hard Hd#1 :6
```

Unlike the first situation I described in my first letter, neither the FREE or the DIR command would work. I took a look at the meaning of the DCT codes and I think I can guess that the info is too screwed up to make sense to any of the programs.

Also, I'd like to comment on installing the Fujitsu drives in a Model III. On page 73 of TMQ IV.i, it was made to sound like a simple pop-out pop-in installation. I would advise anyone putting one in a Model III to read Charles Ainsworth's article in TMQ III.ii — it's right on the money. I too decided to replace the original RS floppies with the Fujitsu models. I had been using them as external drives, but the internal ones were getting so bad, I decided to install at least one of the Fujitsu drives as an internal drive. After getting everything disassembled, I discovered I could not fit them in the original mount — they are too long. So I wound up installing them as Charles did. The two external drives are now :0 and :1. I did two things differently. First, since I was in the middle of the job late one night when I discovered they were too long, I decided to use the external cable which you provided. I carefully removed the connector from the flat cable, threaded the cable through one of the air vent holes in the

bottom of the case and then re-installed the connector. Secondly, I was able to remove the old cable and install the new one without removing the motherboard shield. Other than those differences, one should carefully read Charles' article — its very good and very complete. I also had to disconnect the power supply line to the original drives. When I first restarted the computer, the drives wanted to run all the time. There's only one thing you need to be prepared for that Charles didn't point out. If you're used to hearing the whirl of the original drive motors, you will be greeted by an eerie silence when the new Fujitsu drives kick in. For a while you're going to be worried that something is wrong. After a bit, you'll just appreciate it.

Looking forward to the next TMQ. Thanks for all your services and information and just being there when I and a lot of other folks needed you.

**Fm MISOSYS, Inc:** Dear David, From the "before" and "after" DCT information you provided, my guess is that either (1) the BOOT record of that hard drive partition has a corrupted pointer to the directory, or (2) the directory has a bad GAT. I lean more to the first problem as the DCT shows that the cylinder number of the directory has changed after trying to use BACKUP.

The third byte of sector 1 on cylinder 0 is a pointer to the cylinder containing the directory cylinder. If your drive works normally with the directory on cylinder 4C, check out the BOOT/SYS file. You can list it in hex via the command, LIST BOOT/SYS.SYSTEM:6. Then examine the third byte of the second sector. If it is 4D instead of 4C, there's your trouble. The BOOT/SYS file may even be totally corrupted. You may want to just correct the byte using FED, then BACKUP the files from that drive. If more than just the one byte is wrong, correct it, then reformat the drive after backing up the files.

## DISKCOPY does need a Patch; KelAm Bus Connectors

**Fm Hans de Wolf, Haarlem, The Netherlands:** Dear Roy or Brenda, A few months ago I have reported to you an error in LB 1.2.0 (the same as detected by Mr. Tollini in TMQ IV.ii), and a problem with DISKCOPY in LS-DOS 6.3 (DISKCOPY keeps asking to insert a system disk after a copy from floppy :6 to floppy :7 when drive :0 is a DiskDISK on my hard disk). Up to now, I have received no reply from you. Maybe my letter is still in that stack on your desk you mentioned in The Blurb, but if you did not receive my letter and need extra information to solve the DISKCOPY problem, please tell me so.

About the custom cables you can produce: do you sell cables which allow more than one piece of equipment (hard disk plus Orchestra/Joystick/...) connected to the model 4(P) card edge? By this, I mean a cable of which one side plugs on the Model 4's 50-pin card edge, and contains at other side (and somewhere along the cable) copies of this card edge.

**Fm MISOSYS, Inc:** Dear Hans, The problem with SORT rejecting the "valid" invocation was fixed in LB 1.2.2; a copy is included for you.

Your right about a previous letter; probably still in the queue. Nevertheless, I took a look at your report. On the surface, I was not able to reproduce your trouble report - that of using a DiskDISK as the system disk and having DISKCOPY continuously prompt for the system disk at its conclusion. On the other hand, I found what I believe to be the bug. DISKDISK incorrectly examines the FOURTH byte in the first sector of the BOOT/SYS file (sector 0) as being the pointer to the directory. It should examine the **third** byte! DISKCOPY then attempts to read the first sector of the cylinder pointed to by that byte. It then examines the byte at offset X'CD' to determine if the disk is a "system" disk (bit 7 must be a '0'; the byte examined is

supposed to be the configuration byte in the GAT. If the pointer contained an "invalid cylinder number", or if valid but the byte at offset 'CD' had bit-7 set, then you would be constantly reprompted.

Here's a patch to DISKCOPY which corrects the "bug":

**PATCH DISKCOPY.UTILITY**  
(D05, 4F=02:F05, 4F=03)

As far as the cable is concerned, I could custom manufacture such a cable; however, whether or not your machine would work properly with multiple peripherals is questionable. There generally is conflict among the peripherals unless they are well designed. What you need is a ribbon cable with one female card edge and multiple male card edge connectors. As it is a 50-pin bus, you need 50-pin card edge connectors. The male connectors are expensive, probably retailing in the range of \$15. I do have a source for them as I use the 34-pin male edgcard connector. Just decide on what your needs are. I would suggest a price of \$12 for the cable (12"-18") and female connector, then \$15 per male connector.

Note that the only source for the connectors is a company called PCD. They used to be KelAm, but went through either a buyout or a name change. The connectors have an orientation for pin 1 which is reversed for every other known manufacture of edgcard connectors; thus, you have to match a 50-pin PCD (KelAm) female connector to the male connectors. I buy these connectors from a PCD distributor. I believe the high prices for the connectors are due to the sole source manufacturing.

### *Please Read the Documentation!*

**Fm [name withheld], Miami, FL :** Dear MISOSYS, With all due respect, I have had nothing but trouble with the disk dame that I have just purchased from your

company. The name of the disk game I purchased is "Lair of the Dragon". I must apologize for not replying sooner but I spend eighty percent of my time on the road and I have little, usually none, time to spend at my terminal.

In any event, my first complaint is this; following the instructions within, it is near impossible to load megadv1 to any Model 4 system disk, that is as stated in your manual. The programs must be copied to a data disk.

My second complaint is; following the on screen instructions, one is asked to remove the master copy of Lair of the Dragon and insert a system disk. Who's system? Every time I attempt to use 6.2 TRSDOS or 6.3 LSDOS I get an error message and the entire process stops. The error message given is "disk error! copy withdrawn". Sorry, I just ran out of ideas on how to beat this one.

My third complaint is this; after loading the games to a data disk, loading the data disk in drive 1, and using a 6.3 LSDOS system disk in drive 0, I get the following error;

```
DO MEGADV1/CMD
Line 1 ...garbage...
Line too long
LS-DOS Ready
```

Sorry fella's, but if I understood machine language I would be selling the games to you people and not buying them.

Well now that I had my say, what kind of help can you offer besides two aspirin and a glass of cool water for my aching head? I was going to call but I figured that if there was an assembly problem you might want to see it for yourself. I thought maybe a letter for starts in case I'm not doing something right, which happens to be very likely. Let me know if you want the disk back (your disk), in which case if you do I will send along my data disk and system disk so you can see if it was me or not.

Much thanks and have a happy holiday.

**Fm MISOSYS, Inc :** Re: Complaint of "nothing but trouble" with the Lair of the Dragon game disk. I fail to understand why you are having so much trouble. The manual states, "TRS-80 Model III/4 users can copy the files or, if your system does not support LDOS/LS-DOS formats, boot the distribution disk in Drive :0 and allow it to transfer them to your system or data disk". So since you are using a Model 4 DOS, you DON'T use the LAIR's internal copy facility! Simply copy the MEGADV1/DAT file and the MEGADV1/MD4 file to your system disk or your data disk. Then RENAME the MEGADV1/MD4 file to a /CMD extension. Instructions for using the COPY utility of your DOS is printed in your DOS manual. But if you cannot locate them, just put a minimum TRSDOS 6.2 or LS-DOS 6.3 SYSTEM disk in drive :0 and the LAIR disk in drive :1; then type

```
COPY MEGADV1/DAT:1 :0
COPY MEGADV1/MD4:1 /CMD:0
```

which copies both files and does the renaming at the same time.

How do you create a minimum system disk? Use the PURGE utility documented in your DOS manual to purge all files except the SYS files. A command such as

```
PURGE :0 (I,Q=N)
```

does the trick.

Your second complaint is related to the first. You are not supposed to use LAIR's internal copy utility if you are using LDOS, LS-DOS, or a DOS with a compatible format. If you were using Model III TRSDOS 1.3, then you would use the LAIR internal copy utility.

Your third complaint originates from your failure to understand the use of the DO command of your DOS. "DO" is only used to invoke a Job Control Language file (JCL). Nothing in the LAIR manual says anything about a JCL file. Your clue should have been when you typed "DO MEGADV1/CMD" as /CMD files are always invoked under your DOS by simply typing their name, such as "MEGADV1"!



Sorry fella, but you absolutely do NOT have to understand machine language to play Lair of the Dragon; but you certainly do have to understand simple functions of your operating system, such as how to invoke CMD files. It couldn't be simpler. The Lair manual, on page 5, even tells you how to do that when it states, "from the DOS system prompt type *MEGADV1* and press the [ENTER] key."! Nothing says to type "DO MEGADV1/CMD"!

You should now be on the right track to an "adventure of your dreams..."

### ***TRSCROSS Does Handle SuperScripts***

**Fm Peter Meloy, Helena, MT:** Dear Sirs, I have recently purchased TRSCROSS and I am having trouble. I want to put TRS-80 files on Wordperfect.

I use a Tandy 1000 TX with a hard drive. I have on the hard drive Wordperfect 4.1, Wordperfect 4.2 and Wordperfect 5.0. Memory 640.

I put the Trscross on the hard drive and am successful in getting the menu. The program will allow the files to be transferred to a disk and by using List Files in Wordperfect it properly lists all the files. But when I retrieve the files the file is filled with garbage interspersed with part of the file.

I am enclosing a print of what I get with some remarks I have made. You will note that it works best with Wordperfect 5.0. Can you help me?

**Fm MISOSYS, Inc:** Dear Peter, Re: the conversion of SuperScripts document files to MS-DOS using the PowerSoft TRSCROSS program.

My impression from reading your letter is that you may or may not have specified the F10 mode for the type of the files to be converted; this specifies SUPERS mode. Page 11 of the TRSCROSS user manual states, "This mode converts files to ASCII

or RFT/DCA (Revisable Form Text)." I suspect that you did convert to RFT files, but neglected to import them as RFT into WordPerfect. TRSCROSS does not convert SuperScripts files to WordPerfect files, but to either plain ASCII (losing formatting parameters) or to RFT.

I suggest that you confirm your invocation of TRSCROSS as:

TRSCROSS /RF

as noted on page 6. Then specify the mode as SUPERS for your SuperScripts files. This will create Revisable Form Text files. You then will need to consult your WordPerfect manual for the method of either importing RFT files or converting RFT to WP. Older versions of Word Perfect used to have a separate utility for converting RFT/DCA to WP.

### ***Radio Shack printer/plotter***

**Fm Richard Von Heim, 1850 Hanover Drive, Apt # 208, Davis, CA 95616:** Gentlemen, Happy new year, ... The very best for 1990

I have been a subscriber to Computer News 80, and have watched your ads. (Great work - keep it up, we all need your support!)

I am getting started on my new year's resolution of adding to my Model 4 & Model 12 computer systems. So, please put me on your mailing list for products and sales.

I have been stuck with a Radio Shack printer/plotter FP-215 Cat # 26-1193. My attempts to do graphics on the video screen has been slow but determined I have used the graphics characters to print side margins, i.e. # 128 to # 191. But my screen dumps have show the letters okay, but the "special characters" do not print. They are printed as dots or periods!

According to Computer News 80, you are "very experienced in oddball drivers",

which I presume Tandy somehow overlooked. Do you have such a driver that I could use to get the doggone thing to print what takes me hours to put on the video screen? The plotter costs too much to just not use! Please, please help me with the situation. Thank you.

**Fm MISOSYS, Inc:** Dear Richard, MISOSYS acquired the PowerSoft line of products as of November 1989; your letter was forwarded to me.

As far as printing graphics directly onto a printer, two things need to be dealt with. First, the printer must be able to directly support the reproduction of TRS-80 graphics characters given the appropriate print codes, and the DOS must be told that the printer has such capability. The Model 4 LS-DOS is given that information when you set the printer graphics mode via the command, SYSTEM (GRAPHIC=YES); the normal default is NO. Whenever the graphic mode is not established, the DOS printer screen facility will always convert graphics character codes to periods.

I am not familiar with the Radio Shack plotter FP-215, so I don't know whether it directly supports TRS-80 graphics. I do know that early Epson dot matrix printers, such as the MX-80 did provide that support. However, once the IBM character codes became the defacto standard, printer manufacturers shifted to support those codes.

Even if your printer does not provide that support, a printer driver could be written (if it hasn't already been done) to print all character codes in the TRS-80 character set. Such a programming project may be useful for you to pursue; neither MISOSYS nor PowerSoft has (had) such a driver.

### ***Hard Drive Needs Startup Time***

**Fm W. Dayton Sumner, Phoenix, MD:** Dear Roy, Just a little feedback on the new 40-Meg Hard Drive. I'm well pleased

with it and thought you might be interested in knowing about the problems I had in getting it up and running.

First, I tried running the HD40INIT/JCL which, as we know, had an error in line 105. Before I called you and found out about that correction, however, I tried FSCSI6 since the JCL had produced messages indicating the drive wasn't formatted.

After I called you and got the correction, I ran the JCL and got a puzzling result. It appeared to be starting to verify tracks, so I left my desk for a few minutes. result. When I came back the screen was full of asterisks and numbers as if a lot of tracks were locked out (I presume). At the bottom it said:

Directory will be placed in Cylinder 030

Initializing System Information - Can't  
Disk Pack Not Formatted

Puzzled, I ran the JCL again. This time it ran correctly until it had verified 1-202 on all four drives, did the Backups and Swaps, and then asked me to "Insert SOURCE Disk" which puzzled me because I had a disk with LS-DOS and your MSCSI6 programs in the machine. I moved it to Drive 5, it did some more backups and then stopped,

At that point I found I had two Floppy Drives as 0 and 1, and four Hard Drives as 2, 3, and 4. I did some manual swapping and Sysgened the Boot Disk and I've been in business since then.

I don't know whether I did something wrong or whether there's another bug in the JCL. But thought you might be interested in the sequence of events.

The only problem I'm having now is that when I Boot Up, using the Clock to provide date and time, I get "Error 11h" repeating and have to reset once or twice. Is this just a question of the drive getting up to speed? I believe I applied SYSOCLK/FIX to SYS0 on both the boot disk and the system (Hard) Disk. Is there a way to fix this?

**Fm MISOSYS, Inc:** The message, "Note: drive appears to be unformatted" refers to high-level formatting. I admit that with both low-level and high-level formatting, the message is not clear. I should have changed it to be more descriptive; it was a message in the high-level formatter existing from a previous version where only one formatter existed. Of course, if there was no low-level formatting performed, the MSCSI/DCT driver would have told you of that error by means of another message.

I can't guess at what was occurring when you ran the JCL, but asterisks are generated whenever a granule is locked out during the verification process. Since you are operating okay now after re-running the JCL, whatever was at play is no longer. The prompt you got to "Insert source disk" occurred because backups are made from both drive 0 (the then bottom floppy drive) and drive 5 (the top floppy drive). Drive 5 was supposed to contain the MSCSI diskette (or a backup copy of it).

As far as the repeating "Error 11H" during boot, the problem is a simple one, but not easily corrected. As you can guess from the "noises" which are generated when you first power up the hard drive, the controller is going through a series of diagnostic tests of the drive as well as reading the configuration information stored on the drive. That takes time; first for the drive to come up to speed, and second for the controller to complete its tasks. Unfortunately, while the controller is busy doing this, it doesn't show that it is busy, it just generates a "seek" error. So the driver returns with an error, rather than waiting for the controller to be not busy. What happens during the boot process is that the booting is done from the floppy, and the configuration file is loaded. This switches the DOS over to the hard drive as the system drive. But the hard drive controller is not ready yet - it is still initializing. So when the DOS tries to load SYS1/SYS to present the DOS Ready message, it gets a seek error. It reports this error then re-tries to get SYS1/SYS loaded as it has nothing else to do. What is needed is to be able to recognize the seek error and always re-try on that error. But I don't think that kind of solution is wise universally. Only during booting should

the driver (or what's calling the driver) re-try seek errors.

One solution is to ignore the stream of errors at boot; nothing detrimental is happening. Another solution is to wait until the hard drive has "settled down" before turning on the computer. Another solution is to leave the hard drive powered up all the time. If I think of any other solution, I'll get the word out.

### *What's the name of that patch???*

**Fm Patrick H. Larkin, Bedford, TX :**  
Dear Roy, Here are a few comments, some regarding the Winter 1989/90 issue of *The Misosys Quarterly*. No response is expected.

It appears you may have re-assigned a previously-used name (SYS7I/FIX) to the patch on Page 3 (see the enclosed information from MY patch list).

Your "Bonanza Special to 12/31/89" looked good to me (I'd like to have Super Utility 4/4P/4D) until I tried to find a MISOSYS product I can use for equal or less value, since I already have Tandy's TRSDOS 6.2 Utilities and Double Duty, as well as your EDAS and DSMBLR (LDOS versions) and LDOS-530 with HIK.

Each of the GO: series has at least one interesting item I'd like, but there are some duplications of TRSDOS 6.2 Utilities, and each sells for more than SU4.

I'm not into games (my wife thinks I spend too much time at the computer as it is), and I already own "Leo's Greatest Hits".

LB would be nice, but even at the TMQ bargain that "expires February 28, 1989 (sic)" is more than SU4.

PRO-WAM (also more expensive) would be nice, but to me, unless I could afford to get an XLR8er, would be of somewhat

limited usefulness to me, as my primary use of the Model 4 is VisiCalc, which uses the other 64K bank of RAM.

Speaking of VC, I have encountered, from time to time, a problem with some of my files. VC would not retrieve a /VC file saved to my data disk (with VC). The last time it happened, I listed the file from DOS, and found what appeared to be (part of) a SPOOLer file overlaying some of what should have been a /VC file. I have my bootup JCL set up a 24K SPOOL file on my data disk ("SPOOL \*PR VCFILE:1 (BANK=0, DISK=24)"). Have you any reports of conflicts between SPOOL and VC?

Again, I don't expect a reply from you. I know you're busy and have to earn a living doing those things that generate revenues.

```
.SYS7I/FIX - Patch to LDOS
5.3.0 SYS7/SYS
. Corrects FREE with no
parameters when all 8 drive
slots
. are active. See TMQ
II.iv, p. 24.
. Apply via, PATCH SYS7/
SYS.SYSTEM using sys7i
D0C,59=C3
F0C,59=CD
. Eop
```

```
.SYS7J/FIX - Patch to LDOS
5.3.0 SYS7/SYS
. Corrects ATTRIB
(LOCK|UNLOCK)
. See TMQ IV.ii, p. 3.
. Apply via, PATCH SYS7/
SYS.SYSTEM using sys7j
D1E,03=00 00 00 00
F1E,03=2C 73 2C 72
. Eop
```

Fm MISOSYS, Inc: You appear to be right about that patch nomenclature: the patch on Page 3 of TMQ IV.ii should be named SYS7J/FIX. I reprinted the "correct" SYS7I/FIX and renamed SYS7J/FIX as you related to avoid confusing any other folks.

I have not heard of any conflict between

Visicalc and SPOOL. But make sure you don't switch disks in drive :1. With the SPOOLer operating out of bank 0, there should not be any external memory conflicts (shouldn't anyway). But the SPOOLer keeps the spool file open and will access direct disk reads/writes even if you inadvertently switch disks.

### Resource: Companies

Aerocomp, PO Box 223957, Dallas, TX 75212 [214-637-5400]

Anitek Software Products, PO Box 361136, Melbourne, FL 32936 [407-259-9397]

Computer News 80, PO Box 680, Casper, WY 82602

Computer Reset, PO Box 461782, Garland, TX 75046 [214-276-8072]

Cornucopia Software, Inc., 1625 Beverly Place, Berkeley, CA 94707 [415-528-7000]

GRL Software, Suite 209, 1051 KLO Rd., Kelowna, BC V1Y 4X6, CANADA

Howe Software, 64 Windmill Road, Armonk, NY 10504 [914-273-3998]

Hypersoft, PO Box 51155, Raleigh, NC 27609 [919-847-4779]

M.A.D. Software, P.O. Box 331323, Fort Worth, TX 76163

Microdex Corp., see Computer News 80

Micro-Labs, Inc., 7309 Campbell Road, Dallas, TX 75248 [214-702-8654]

MISOSYS, Inc., PO Box 239, Sterling, VA 22170 [703-450-4181: Orders to 800-MISOSYS]

Pacific Computer Exchange, 1031 SE Mill, Suite B, Portland, OR 97214, [503-236-2949]

Powersoft: Contact MISOSYS effective

11/1/89

RANTECH Computer Systems, PO Box 1101, Clackamas, OR 97015 [503-771-0390]

Storage Power, 10391 Oakhaven Dr., Stanton, CA 90680 [714-952-2700]

Tandy National Parts - Hardware [817-870-5600]

T/Maker Research Company, 812 Pollard Road, Suite 8, Los Gatos, CA 95030, [408-866-0127]

TRSTimes magazine, 20311 Sherman Way, Suite 221, Canoga Park, CA 91306

Try-o-Byte, 1008 Alton Circle, Florence, SC 29501 [803-662-9500]

## SECURE PROGRAMS

AND ASCII FILES  
WITH  
**CODE 4**

COMES WITH  
LIST SOURCE

**\$19.95**

SEND CHECK OR  
MONEY ORDER

**HPB VECTOR WSO**

607 DUNSMUIR AVE, #104  
LOS ANGELES, CA 90036

ALLOW TWO WEEKS DELIVERY



## Doubling of Files

### Solved

Published in *THE MISOSYS QUARTERLY* Volume IV.i (pages 26-30) was a lengthy discussion of a problem exhibited as more than one instance of the same filename appearing on a diskette. The problem was tested and tested at MISOSYS and considered prevalent only on gate array Model 4s (26-1069A, 26-1070, or 26-1080A). MISOSYS firmly believed that the problem had nothing to do with whether or not an XLR8er board was installed. Furthermore, MISOSYS strongly suspected a hardware problem.

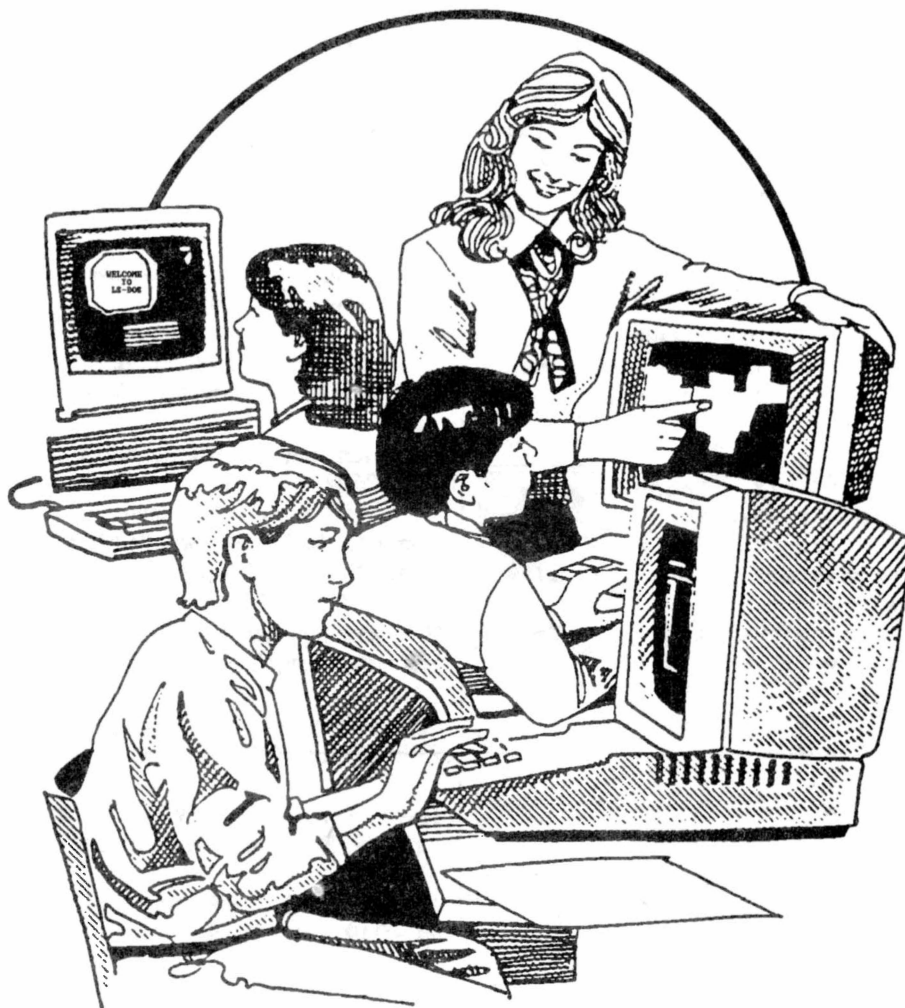
There were no resources at MISOSYS to continue to investigate such a random occurrence; I enlisted the assistance of others. Readers of TMQ and users of the LDOS Forum (CompuServe PCS-49) were asked to run a test program on their machine to aid me in gathering additional empirical data.

I also asked Frank Durda if he had any idea of the root cause. He proceeded to do a bang up job of not only investigating the problem, but in arriving at a reasonable workaround solution. Here's his response to the problem.

**From Frank Durda:** Dear Roy, I finally was able to reproduce the problem with the 1773 that we discussed. Apparently I had set some value in the program incorrectly. Once I had a failing program, I have been able to view the problem on two different Model 4D gate-arrays, both with all mods and stock Z8OAs. I also viewed the problem numerous times on a 4P gate array and had a plain 4P run over 6000 iterations without a fault.

Based on your assumption that there is a problem with the 1773, I have compared the data sheet for the 1773 and the 1793 FDC and discovered two things: (1), Western Digital claims that the 1773 is

## DOS Subjects



"fully software compatible with the 1793", and (2), The 1773 is not really fully software compatible with the 1793. It is close, but not identical.

Let me say up front that the problem we are seeing IS NOT caused by the 1773. However, the differences in the chip could cause other problems that you should be aware of. Here is a discussion of the key differences.

### Difference I

In the Write Sector command on the 1793, bit 1 controls Side Compares. If set (1), bit 3 of the command ('S' bit) would be used to determine if the correct 'side' of the media was about to be written to. Most systems including LS-DOS set this bit in both Read and Write Sector commands. (See "The Source", page 51, @ WRCMD1:)

In the Write Sector command on the 1773, bit 1 controls Write Precompensation! If set (1), Write Precompensation is dis-

abled. If not set (0), Write Precomp is enabled. The Side Compare bit does exist in Read commands as before. I am told by an engineer on this project that Western Digital had eliminated side compare capability entirely on 1773 prototypes because Radio Shack did not sell/support sided drives (this was 1983 when this took place), but the RS engineer demanded that it be put back and side compares were partially restored.

Note that in the 1773 mask (1772 and 1770 are the brothers), precompensation can be enabled/disabled by a pin on the chip and this is the way the gate array boards handle this. The FDC-assist chip routes the value programmed in port 0xf4 to the FDC so that precomp works as before.

Based on a study of the standard floppy driver, this programming difference does not appear to affect LS-DOS. However programs that perform their own disk I/O (like alien disk read/write utilities) or other operating systems are at risk from this difference if side flags are used unnaturally.

### Difference II

The 1793 and 1773 have recovery specifications regarding access to the FDC. These require software to wait a period of time before giving additional commands to the FDC, or before data is read, etc.

In the 1793, table I was used; however, the 1773 changed the values significantly to those in Table II. Because of this change, systems using timing loops or code-flow to provide a delay between issuing a command and reading any status may not be providing sufficient time for the 1773 if they were optimized for the 1793 timing requirements. Fewer wait states or higher speed CPU's will impact this issue further.

LS-DOS 6 does use a timing loop to delay operations following the issuing of a command, but for double-density operations (MFM), it should be sufficient, even at XLR8er speeds. However, at single-density, it is possible that it could be exceeded. I have not tested this area since single-density is not used much anymore.

Table I			
Operation	Next Operation	Delay Required	
		FM(SD)	MFM(DD)
Write to Command Reg.	Read Busy bit (Status Bit 0)	12 usec	6 usec
Write to Command Reg.	Read Status (Status Bits 7-1)	28 usec	14 usec
Write Any Register	Read from Same Register	N/P*	N/P*
Write Any Register	Read From Different Register	0	0
* This value never officially published, but verbally was stated as less or equal to the Write Command-Read Status Bits timing.			

Table II			
Operation	Next Operation	Delay Required	
		FM(SD)	FM(DD)
Write to Command Reg.	Read Busy bit (Status Bit 0)	48 usec (was 12)	24 usec (was 6)
Write to Command Reg.	Read Status (Status Bits 7-1)	64 usec (was 28)	32 usec (was 14)
Write Any Register	Read from Same Register	32 usec	16 usec
Write Any Register	Read From Different Register	N/P **	N/P **
** Not published, but tests imply that the old value (0) is probably still valid.			

### Difference III

The minimum required index pulse width on the 1793 is 10 usec. On the 1773, it is 20 usec. Since the typical width of an index pulse from a drive can go up to 4 msec, this should not be a problem.

Those are the key differences between the 1773 and the 1793. I have attached the data sheets for your information.

### The Real Problem

Because none of the above problems seemed to be the true cause of the problem you have encountered, I obtained a Z80A processor emulator and logic state analyzer. After a weekend of back-tracing, I determined that the problem does not rest with the index pulse detection circuitry or the software that checks it. The problem actually occurs much earlier and goes undetected until it reaches the index pulse code.

The problem is that the FDC-Assist chip, which is one of the custom chips that RS created for the gate-array models, fails to work as expected every once in a while.

When port 0xf4 is written to, a one-shot timer is used to enable the floppy drive motors and will allow them to run between three and four seconds. Each time port 0xf4 is written to, the timer starts over. In the original FDC design, a real one-shot was controlled by a resistor and capacitor which determined "how long" the motor would stay on. In the gate-array, a series of divide-by-n counters are used to divide an 8 MHz clock down to a value that provides about four seconds of motor on time.

The problem is that once in a while, the access of port 0xf4 (DRVSEL) is NOT detected and the counter does not start running, so the drive motors do not start.

This problem is fatal when an @SEEK is performed because the code path only accesses port 0xf4 once. In fact what happens is that the drive is selected. In this case, the access is not detected and the motor does not start. The code is unaware of this and issues a SEEK command WITHOUT VERIFY. The code does this when it "knows" that the drive is already at the desired cylinder. (That info comes from DCT+5.) Because a SEEK command without verify can be performed on a drive that is not ready, the presence of a NOT READY signal in the FDC status port is not meaningful. Any code that runs after the @SEEK command that assumes the motor is running will fail if the motor isn't.

So we have issued a seek (no step pulses were actually emitted), and the @SEEK code calls TSTBSY to see if the BUSY bit is off. Because we delay before we get here (to meet FDC requirements) and no head movement or verify is performed, the seek commands ends immediately, and by the time we reach TSTBSY, BUSY is already off. (If the FDC was still BUSY, TSTBSY would reselect the drive by writing to port 0xf4 and would cause the motor to start.) The @SEEK call now returns the caller. In the case of the @OPEN code, we now enter the index-pulse detection code. This fails, because the motor is

not turning.

### *What do we do now?*

I have been looking at the internal circuitry of the FDC-assist chip (the schematics still exist) to see what would happen if we bypassed the MOTOR ON portion of the circuit and use a fool-proof one-shot to provide the MOTOR ON signal.

This solution will work IF only the motor timing part of the FDC-assist chip is the only part that is malfunctioning. If other parts of the FDC-assist chip that must respond to DRVSEL also glitch, then a separate one-shot will make things worse. This is because the FDC-assist chip also latches the drive selects, density and wait-state control associated with port 0xf4. If these fail to latch the correct information and the motor starts anyway, we could seek on the wrong drive. Read/Writes should not be affected since they repeatedly write the correct information to port 0xf4, although it is possible the selected drive could change in the middle of a write for one byte and then revert back to the correct drive. OOPS! (I don't think this is happening. See below.)

Because of the uncertainty about how widespread the malfunction in the FDC-assist is, I am looking into other ways to force the chip to recognize the select signal and act on it. At this time, I have nothing to report but hope to have something in the next few weeks.

### *Is there a software solution?*

Well maybe. If the one-shot is the only part of the FDC-assist that is messing up, the OUT to port 0xf4 in the select code could be done twice in a row. Since the test program is only able to get a failure every once in a while, the chance of the chip failing to detect two selects in a row is remote. Obviously this can't be done in the read/write code, but the motor should be on at that point and if the timer misses one write to port 0xf4, another will occur in 32usec.

If all circuits in the FDC-assist chip are missing the access, the double-out solu-

tion will not work reliably.

I have composed a crude patch to verify this software solution. At NOPCMP+5 lines (0xed4), the statement:

```
LD (PDRV$),A
```

was patched to:

```
CALL ADDR
```

At location ADDR:

```
ADDR: LD (PDRV$),$
;Current drive select
OUT (DSELECT),A
;Second select
RET
```

This patch ran on two of the failing systems for over 6,000 iterations each before I stopped them.

Sometime back I wrote some code improvements to the floppy driver during 6.3 development and I believe they may provide the space needed for an inline patch. The changes ended up not getting used (too late as I recall.) I will be looking into my records to see if they are applicable.

### *Why does LDOS work?*

I suspect that an examination of the LDOS code will show that it writes to port 0xf4 more than once before or during the time while it expects the media to be in motion and that is why it does not exhibit the problem. The fact that it seems to work flawlessly also supports the theory that only the motor timing portion of the chip is malfunctioning since LDOS apparently exhibits no problems that would be caused by mis-reading drive selects, density or wait-state enable commands. LS-DOS does not seem to exhibit these potential problems either.



### Follow-up and solution

I have previously sent you a package with a considerable amount of material regarding the floppy drive problem on gate array Model 4/4P/4D systems. If it has not yet arrived, this letter may not make a lot of sense.

Based on what I have discovered to date, it does appear that only the one-shot emulator in the FDC-assist chip is malfunctioning. Other tests designed to catch the chip failing to perform other tasks have so far indicated no failures.

Therefore, I am more confident that a second OUT of the value written in the drive selection area of the floppy driver should cover the problem. I have devised the following inline patch that fixes the problem and shows no side-effects.

The patch command and breakdown of the patch is shown in Table III.

An analysis of the driver indicates that no one actually uses the value that is returned in A from SELECT, so this code change was possible.

The above patch was applied (using FED) to a previously-failing system and run

continuously for 20,000 iterations without failure. (So much for that floppy.)

I hope that this resolves this problem. If there is any other information you need, please feel free to call or send me mail.

### Restoring Erased Files

**Fm W. J. Russell, Woodville Park, SA AUSTRALIA:** Dear Sir, I noticed your advertisement in the fall edition of The MISOSYS Quarterly for a number of software products. Among those advertised was Model 4 Toolbelt. A couple of years ago I purchased a copy from Breeze/QSD; my disk is serial number 1239.

I would appreciate knowing if there is an update since this purchase. Toolbelt seems useful but the one thing the manual is not clear on is how one can UNdelete a file from the hard disk.

I have a Model 4 with 128K and a 15MRS hard drive plus three DSDD 5 1/4 floppy drives.

If there is an update since my copy, could you please advise me of the cost of upgrading?

**Fm MISOSYS:** Thanks to the efforts of M.A.D. Software, the problem with occasionally doubling of files on gate array model 4s has been traced to a glitch in the FDC gate array logic chip. When the glitch occurs, the gate array floppy support chip fails to generate the MOTORON signal when an OUT is directed to the DRIVE SELECT port. This fails to turn the floppy motors on. It is rare, but the glitch happens. Analysis has failed to disclose any other problem. A software workaround is to generate the OUT instruction twice in the floppy driver's drive select code. The following patch is mandatory!

```
. BOOT4/FIX - 12/08/89 -
For Level 6.3.0 L+ BOOT/
SYS.LSIDOS - Patch #18
. Patch works around a
failure of gate array
Model 4s to occasionally
. not start the floppy
drive motors on drive se-
lect.
. Apply with the command,
PATCH BOOT/SYS.LSIDOS
USING BOOT4/FIX
D0C,B4=FD 7E 03 F5 C5
F0C,B4=F5 C5 FD 7E 03
D0C,D7=D3 F4 F1 D0 CB 57
F0C,D7=F1 D0 FD CB 03 56
. Eop
```

Table III

The patch (correct for LS-DOS 6.3 L+, newest version I have) is:

PATCH BOOT/SYS.LSIDOS:0 (D0C,B4=FD7E03F5C5:F0C,B4=F5C5FD7E03)

PATCH BOOT/SYS.LSIDOS:0 (D0C,D7=D3F4F1F0CB57:F0C,D7=F1D0FDCB0356)

Here is a breakdown of the patch:

Part 1. Old				New			
EB4	F5	PUSH	AF	EB4	FD7E03	LD	A, (IY+3)
EB5	C5	PUSH	BC	EB7	F5	PUSH	AF
EB6	FD7E03	LD	A, (IY+3)	EB8	C5	PUSH	BC
EB9				EB9			
Part 2. Old				New			
ED7	F1	pop	AF	ED7	D3F4	OUT	(DSELECT), A
ED8	D0	RET	NC	ED9	F1	POP	AF
ED9	FDCB0356	BIT	2, (IY+3)	EDA	D0	RET	NC
EDD				EDB	CB57	BIT	2, A
				EDD			

**Fm MISOSYS, Inc:** Dear Dr. Russell, As a subscriber to *THE MISOSYS QUARTERLY*, I'm sure you are aware that MISOSYS has acquired the line of PowerSoft software products. Thus, your letter of December 26th was forwarded to me.

MISOSYS publishes a product to restore files inadvertently removed; it is the UNREMOVE utility included as part of the GO:MTC package which has programs to test memory, repair directories, map files, monitor disk I/O, and alter Drive Control Table information. The UNREMOVE program directly restores files from either floppy or hard disks with virtually no user fussing, other than entering a number in the menu of files capable of being restored.

On the other hand, if you want to employ the PowerSoft TOOLBELT for this purpose, I am including some information entitled "Restoring erased files on a hard disk using Toolbox/Toolbelt" which Doug Hogarth wrote when he worked for PowerSoft. This may provide the information you need. Incidentally, to my knowledge, there are no updates to TOOLBELT. The most recent file date on the disk is 26-Mar-87 for the PKILL6/CMD program. The other files have dates from 1982 through early 1986.

### *Restoring erased files on a hard disk using Toolbox/ToolBelt*

The ToolBox for LDOS and the ToolBelt for TRSDOS 6 can be used to restore erased files, though not through a menu option like Super Utility (which only supports floppy diskettes). The procedure described will be for the Model 4 - the only change necessary for Model 1/3 is to drop the 6 from the program name in the examples. For any restoring of files, it is usually required that the disks have not been written to since the erasure of the files.

First, we must use PMOD to manually check the directory. Enter **PMOD6 :x,@**, where x is the drive number with the erased files. A screen should display with the GAT table. There is not much understandable information on the screen except that the disk name and date should be readable in the lower right.

Press the right arrow once. Again, this display is not of much use. It is the HIT table.

Press the right arrow. This is a directory sector. Up to eight filenames can appear here, with the first two often being "system" files. The filenames can be seen on the right hand side of the display. If you are looking for a specific filename, press the right arrow until you locate it, noting that you should stop looking after 32 times since the directory is a maximum of 32 sectors in length. A final note is that there is the possibility of a filename occurring more than once - the first occurrence is usually the most recent.

After locating a filename, look at the byte on the left hand side of the display, just right of the graphic bar on the same line. This is usually 10 for normal files, and 00 for erased files. To restore a file, change the 00 to a 10. Do this for each file desired.

Now we use PFIX to complete the process. For the least amount of changes on the disk, it is only necessary to enter **PFIX6 :x,H**, and the files should be readable. For a complete restore of the disk to the original condition, use **PFIX6 :x,G,H**. For your information, the G is GAT and H is HIT. After that fix, the files should be available for all operations such as directory, copy, etc.

One final option worth mentioning is that PMOD can be used to view all the information on the disk, ignoring the directory. This may be necessary to show information without writing to the hard disk in any way. Just enter **PMOD6 :x,0,0**, and continually press the right arrow. Any readable information will be shown on the right hand side of the display. Screen prints can usually be made using a feature like **<CTRL> \***.

To restore file using Super Utility, first configure the disk types, use Super Utility to make a standard backup, and then use the recover killed files feature. Deleted files show differently on the screen. More files can usually be seen by pressing **<SHIFT> <UP-ARROW>**. The manual should describe this procedure fully. The display sectors feature can be used like the final option of PMOD to view all data on the disk, and **<SHIFT><CLEAR>** can be used to make screen prints.

### *Quick fix for "tight" disks*

**Fm MISOSYS, Inc:** Occasionally I get a disk returned with the complaint that BACKUP (or DISKCOPY) gives a "source disk read error" when attempting to make a backup copy of the disk. Most of the time when I test such a disk, I get no errors (this is attempted on multiple machines). When I do get an error, I happen to notice that the edge of the disk is "crimped" which increases the friction inhibiting rotation of the flexible media in the sleeve. A floppy disk drive with poor hub pressure may not generate torque sufficient to rotate the disk at the proper speed. One cure is to clamp a second hub ring to the diskette - provided you can even get those these days. Another quick fix is the old ball point pen cure. The following letter sent with a diskette being returned back to a customer summarizes this fix.

The "refreshed" LS-DOS 6.3 diskette which you returned with a "source disk read error" was readable on our drives. Unless the hub pressure springs are up to snuff in a disk drive, a diskette which has more than usual jacket pressure may not be perfectly readable. I did notice that there was a crimp in the diskette jacket about one inch down from the corner of the side where the LSI logo appears. This crimp put more than normal pressure on the diskette media. I just rubbed the edge of a ball point pen barrel across the crimp to alleviate some of the pressure. This little quick fix may be useful to you in the future when a diskette appears to not turn as freely within the jacket as you would

expect it too.

I don't know whether the crimp was there before or after the disk was refreshed, however, it is not all that unusual for disks to get slightly crimped if bent. So keep in mind the quick fix mentioned above. It has restored 100% readability to many crimped disks I have come across over the years.

## @BANK patches

&

## HD Boot

Scott Toenniessen  
3330 Lockport-Olcott Road  
Newfane, NY 14108

Michel Houdé was right about the LeScript 1.7 patches I provided in the TMQ III.iii, p.51; they are junk. While they did seem to "work fine" during the time I used them with LeScript showing over 80,000 bytes free with PRO-WAM resident in bank 1, and the spooler in bank 2, as soon as the text buffer fills bank 0 and the use of bank 1 is required by LeScript, a system crash becomes highly likely, especially when an attempt is made to use PRO-WAM. As Michel discovered, LeScript uses @BANK to check for the availability of banks 1 and 2, then directly accesses the hardware for all further bank switching operations. Anybody using them should YANK them immediately or start using a fresh copy of LeScript. I apologize for any problems this may have caused. I don't know what I was thinking about, must have switched a LDB,0 with a LDB,2 or something like that..... I urge caution when using the VisiCalc patches supplied. As I noted originally, these patches weren't working with PRO-WAM. Maybe I made the same mistake, I'll have to take a look when I get a chance.

I have enclosed a patches that have been in use for 8 months on my Model 4P

running LS-DOS 6.3 level L+. It allows a 4P to boot from a R/S Hard Drive with Michel Houdé's XLR8er patches installed. To do this, a slight modification needed to be made to patches written by Gary Phillips to allow 4P self-booting without Michel's patches. This modification needed to be made at patch location D0D,07 in SYS0/SYS. Michel's patches place a CD 94 21 (CALL 2194) in this location which, according to "The Source", pg 107, is a 3 byte location saved for a jump. Gary's patches place a C3 8F 1E (JP 1E8F) in this location. Gary's JP simply skips over a small section of code needed for booting from floppies, but unneeded for booting from a hard disk. All I did was move this JP to patch location D0D,0A which is at the beginning of this skipped code. Upon RETURN from Michel's CALL, this JP is executed and system initialization continues, but from the hard drive.

I have a few comments to add to the documentation in the patch. First, I use the patches with PowerSoft's WD Series R/S Hard Disk Drivers. According to Gary Phillips, the patches should also work with MISOSYS' and Radio Shack's drivers, but I have not tried them. Second, the entire system partition (logical drive 0) must be under head 1 and on this partition logical cylinder and sector numbers must equal physical cylinder and sector numbers. Keep in mind that LS-DOS can support a maximum of 203 logical cylinders. This means that if the hard disk being used has more than 203 physical cylinders, 2 physical cylinders will be used for each logical cylinder. I use a 10 meg. hard drive which has 4 heads and 306 cylinders per head. My first head is split into 2 partitions. The first one is 153 cylinders (0-152) and is used as the system partition. The second partition is 153 cylinders (153-306) and is used for data. By splitting the head into two partitions, I can overcome the 203 logical cylinder limit. The system partition has 153 physical cylinders which the DOS allocates as 153 logical cylinders. This allows the HD booting patch to work. Each of my other 3 heads are used as one 306 physical cylinder partition (0-306) which the DOS allocates as 153 logical cylinders; 2 physical cylinders per logical cylinder.

WHEW! Was that confusing! Here's a

couple other hints from my experience. I have never been able to self boot off the hard drive when the hard drive was connected to one of those 50 pin I/O bus 'Y' adaptors (Alpha Products used to sell them). I have one which allows up to three peripherals to be plugged into the bus at the same time. Even when the only thing plugged into it is the hard drive, I have never been able to get the 4P to boot from the hard drive when using it. It will boot when plugged into the bus extender on the Micro-Labs Joy-Mouse interface, though, as long as the Joy-Mouse AC power supply is plugged in. My advice is when first trying to use these patches that the hard drive be connected directly to the 4P I/O bus. After the patches are working, use of 'Y' adaptors or other peripherals can be tried.

Before trying these patches, BACKUP the HARD DRIVE! Then, if needed, re-format the drive as discussed above to get each logical cylinder to equal one physical cylinder on the system partition. Install the hard disk drivers and copy the LS-DOS 6.3 system files to the system partition. Then apply Michel's patches followed by the HDBT63FY/TXT patches (be sure to replace xx, yy, and zz in HDBOOT/FIX with the proper values; I used xx=20, yy=04, zz=0C 90 00 98 1F E3). SYSGEN your configuration with the hard disk drivers resident to the hard drive and, I would suggest, to a floppy disk. Then reset the computer and it should boot from the hard drive. If it doesn't, you can use the floppy as a boot disk to gain access to the hard drive.

Remember, get rid of LESBANK/FIX and VISBANK/FIX and again I apologize for the screw-up. Hopefully, the enclosed patches will make up for the difficulties caused to some of your readers.



```

. HDBT63FY/TXT
.March 10, 1989
.This file contains patches for LS-DOS 6.3 to allow a Model 4P to boot
.directly off of a hard disk drive. These patches were originally
.written by Gary Phillips. The HDSYS0 patch has been modified
.by me to work in conjunction with Michel Houde's XLR8er board patches
.for LS-DOS 6.3. I made the following change to the patch:

.D0D,07=C3 8F 1E changed to D0D,0A=C3 8F 1E
.F0D,07=00 00 00 changed to F0D,0A=3A 9D 43

.The modified patch will work even if Michel Houde's patches are
.not used. No change needed to be made to the HDBOOT patch for
.use with Houde's patches.

.Gary's original documentation is included below.

. P.S. use a text editor to extract HDSYS0/FIX and HDBOOT/FIX
. Then apply them using PATCH sys0/sys.lsidos USING hdsys0/fix
. and PATCH boot/sys.lsidos USING hdbboot/fix

. Scott Toennissen 72057,3415

.
.
.December 22, 1987

.The following patch files can be used to permit a model 4p to boot
.directly from a hard disk that uses the standard Radio Shack 8X300
.or WD1010 controller board. They use the code and technique that
.was originated by Adam Rubin for TRSDOS 6.2, but have been modified
.for LS-DOS 6.3. I have tested them with a standard RS 5 Meg drive,
.using either the original TRSHD6 driver or the Misosys RSHARD6 driver
.software. You should also read Adam's original HDBOOT.DOC file
.before applying these two patches. (See DL6 for that file.)

.The HDSYS0/FIX patch should be applied as it stands, to the SYS0/SYS
.file on your hard disk. The HDBOOT/FIX patch must have the xx, yy,
.and zz bytes filled in from your DCT before applying it. Be sure that
.your hard disk is completely backed up before applying these patches,
.because if you have difficulty you could lose data!! Remember, too,
.that once you have applied these patches, you should NOT use backup
.to copy /SYS files to or from the hard disk. Your 4P will automati-
.cally
.boot from the hard disk when it is powered on or reset, unless you
.hold
.down the F2 key while booting. (If you have an XLR8er board, you may
.need
.a couple of tries on a cold boot because the cold start speed of the
.XLR8er
.is a bit slow.)

.Note that the xx and yy values must be entered in hexadecimal. For a
.RS 5 Meg drive, xx is 32 sectors/cylinder or hex 20, for example.

.If you have your HD system partition configured as drive 0 (by
.booting
.from a configuration floppy) you can obtain the zz values by using
.DEBUG to look at address x'0473' through x'0478', which is DCT+3 for
.drive 0 under LS-DOS 6.x for the model 4.

.I will be happy to try to answer any questions, but remember, I can't
.be responsible for any problems, so please be SURE you have a good
.backup before trying this modification! -Gary Phillips, 72425,354

.
. HDSYS0/FIX - Patch to SYS0/SYS of LS-DOS 6.3 to permit direct
.booting
.of model 4P from RS hard disk -originated by Adam Rubin

```

## HiRes Game Patches

Ken Strickler  
P.O. Box 773  
Stanwood, WA 98292-0773

Enclosed is the patch for CHESSL/EXE, the low resolution version of T-CHESS. It is the same patch as for the HIGH resolution, but in a different place [see *THE MISOSYS QUARTERLY* II.iv, pages 42-48]. It took me a little time to find a UTILITY which would allow reasonable access to CHESSL/EXE, as the PDS format that was used wouldn't allow me to step to the next record. I didn't have success with PROCESS, I suspect because of their file format. I tried DISKDUMP (Dos-Plus) and SUPER UTILITY (Powersoft), and finally found the DISK ACCESS mode of FED2. By configuring my XLR8er to the size of the program to look at plus a little for DIR/SYS and BOOT/SYS, I could start looking into the program at 0200X. Searching for the matches on the fixes that you wrote for CHESSHX/FIX, found them and that was easy to fix.

Next I loaded GAMMON (Backgammon) and tried the same thing. Well 2 of the patches showed up (#1 and #3) but no match on #2, well maybe they didn't use that sequence in GAMMON. I patched what I found, but keyboard is still locked up. (Found out later that FED2 - FILE MODE worked on this.)

I did try the patches for T-CHESS and they work fine, but I don't think that the 64180 is set at FAST. Maybe a patch to do that would help. The programs which would benefit the most from the speedup are the formula intensive programs like CHESS and FRACTALS. In the final analysis however, to patch the programs so that they run under LSDOS would be far better! I had a terrible time trying to find DISKS that met RAPIDOS expectations! the FORMAT program doesn't know LOCKOUT. Since I use 'EL-CHEAPO' disks, I figured that it was the media, so I

```

. -adapted for 6.3 level K by Gary Phillips -08/12/87, rev. 08/22/87
. modified for use with Michel Houde's XLR8er patch for 6.3 by
. Scott Toennissen 03/10/89
. Apply via PATCH SYS0/SYS.LSIDOS using HDSYS0/FIX
.
D0C,B9=00 00 00
F0C,B9=31 80 03
D0D,0A=C3 8F 1E
F0D,0A=3A 9D 43
D0E,A0=00 00 00
F0E,A0=C2 A0 19
D0E,E2=00
F0E,E2=C0
D0F,14=FE
F0F,14=0C
. eop
.
. HDBOOT/FIX - Patch to BOOT/SYS of LS-DOS 6.3 to permit direct
booting
. of model 4P from RS hard disk -originated by Adam Rubin
. -adapted for 6.3 level K by Gary Phillips -08/12/87
. Apply via PATCH BOOT/SYS.LSIDOS using HDBOOT/FIX
. Before using this patch, you must substitute:
. for xx, the number of sectors per cylinder of drive 0 (hexadecimal)
. for yy, the number of sectors per gran under your driver (hexadecimal)
. for zz, the values of DCT+3 to DCT+8 for drive 0
.
D00,4D=20 00 00 00 00 00 00 00
F00,4D=12 FD CB 04 6E 28 01 87
D00,5A=04
F00,5A=06
D00,9F=C3 43
F00,9F=70 04
D01,41=18 0D F9 43 0C 90 00 98 1F E3 00 00 00 00 00
F01,41=1E 00 CD 74 43 3A CD 12 E6 20 21 74 04 B6 77
D01,6F=18 2C 00
F01,6F=C3 38 02
D01,7C=C3
F01,7C=96
D01,80=00 00
F01,80=E6 1C
D01,96=00 00 00 00 00 00 00 00 21 43 43 11 71 04 01 08 00
F01,96=01 F4 81 ED 41 0D 3E 18 ED 51 CD D9 43 DB F0 CB
D01,A6=ED B0 C3 38 02 F3 CD C3 43 3A 78 00 D3 84 FB 3E
F01,A6=47 20 FA 7B D3 F2 3E 81 D3 F4 D5 11 02 C1 3E 80
D01,B6=05 C0 3A 79 04 92 3E 06 28 01 AF A7 C9 E5 D5 3E
F01,B6=CD D9 43 3E C0 D3 E4 DB F0 A3 28 FB ED A2 7A D3
D01,C6=85 D3 84 D3 9C 4B 5A 16 00 06 0C 3E 02 CF F5 AF
F01,C6=F4 ED A2 20 FA 18 FE D1 D1 AF D3 E4 3E 81 D3 F4
D01,D6=D3 9C 3E 86 D3 84 F1 D1 E1
F01,D6=DB F0 C9 D3 F0 06 18 10 FE
D01,F9=3E 09 90 28 AD AF C9
F01,F9=00 00 00 00 00 00 00 00
. EOP

```

opened some of my 'RESERVE' BASF FlexyDisk 2D - Double sided - Double Density. They were NO BETTER! RAPI-DOS continues to remind me of the FIRST CP/M operating systems! If it wasn't for the GAMES that run under it, I wouldn't have it! (Of course if I didn't have a computer, I would have any of them!)

I have also enclosed my MASTER BACK-GAMMON DISK AND MANUAL, (I wouldn't want to be accused of 'pirating' software. Please return when 'WE' are finished), and the patch that I found so far for GAMMONH/EXE is on this disk (GAMMONHX/FIX). I will take a look at 3D TIC-TAC-TOE next, and let you know.

**Fm MISOSYS, Inc:** Thanks for the patches. I worked one up for the low-resolution version of GAMMON, as well.

Incidentally, the Rapidos games are not "PDS" files; they are core-image binary files. It just so happened that the file you looked at started with 04H, which FED thought was the end of an ISAM directory. A little trick, which happens to be documented in the FED manual, is to force FED to ignore typing a file under edit by prefixing the file specification with an exclamation point, "!". FED then doesn't try to test the file for a load module structure.

## SuperSCRIPT Document File Format

Over the years many tales of corrupted SuperScript document files have been heard. When the problem occurs, it is a difficult task to re-capture the text material contained in the document. A few years back, Tom Price put together a white paper outlining the structure of the document file maintained by SuperScript. Using the material contained in this document, and a good file editor such as our FED2 or Super Utility, it should be possible to restore the "integrity" of the document structure enabling the repaired document file readable by SuperScript. The following information is what was prepared and released by Tom Price some years ago. It is reprinted for your use, just in case you may not have come across it.

## SuperSCRIPT Documents

This article will explain in detail the structure of SuperSCRIPT document files, which may be of assistance in repairing damaged files with the aid of a 'zap' program or a file editor.

Each SS file has four distinct areas:

**Record 0:** Document header and



. CHESSLX/FIX - 04/11/89 - Patch to Tournament Chess  
 . Note: This is a Model 4 version. For details see TMQ 3.4  
 . Removes interference of joystick port access with 64180 CPU  
 . internal register access. Joysticks are still supported.  
 . Apply via PATCH CHESSL/EXE CHESSLX.  
 . THIS IS FOR THE LOW-RESOLUTION DISPLAY!

D48,12=3C DB 00 2F A7 20 17 21 80 F4

F48,12=DB 00 2F A7 20 18 21 80 F4 AF

D4B,03=57 3C DB 00 2F 87 87 87 21 40 F4 B6 E6 78 37 C8

F4B,03=DB 00 2F 87 87 87 21 40 F4 B6 E6 78 37 C8 0F 0F

D4B,13=0F 0F 0F 5F

F4B,13=0F 5F 16 00

D6B,77=06 07 B0 DB 00 2F 87 87 87 21 40 F4 B6 20 0D 05

F6B,77=DB 00 2F 87 87 87 06 07 21 40 F4 B6 20 0E 06 06

. GAMMONHX/FIX - 04/07/89 - patch to Backgammon  
 . Note: This is a Model 4 Version. For details, see TMQ 3.4

. Removes interference of Joystick port access with 64180CPU

. internal register access. Joysticks are still supported.

. Apply via, PATCH GAMMONH/EXE GAMMONHX

D2C,BD=3C DB 00 2F A7 20 17 21 80 F4

F2C,BD=DB 00 2F A7 20 18 21 80 F4 AF

D71,74=06 07 B0 DB 00 2F 87 87 87 21 40 F4 B6 20 0D 05

D71,74=DB 00 2F 87 87 87 06 07 21 40 F4 B6 20 0E 06 06

. GAMMONLX/FIX - 03/02/89 - Patch to Backgammon V1.2.1  
 . Note: This is a Model 4 version. For details, see TMQ 3.4

. Removes interference of joystick port access with 64180CPU

. internal register access. Joysticks are still supported.

. Apply via, PATCH GAMMONL/EXE GAMMONLX

D2C,BA=3C DB 00 2F A7 20 17 21 80 F4

F2C,BA=DB 00 2F A7 20 18 21 80 F4 AF

D61,65=06 07 B0 DB 00 2F 87 87 87 21 40 F4 B6 20 0D 05

F61,65=DB 00 2F 87 87 87 06 07 21 40 F4 B6 20 0E 06 06

other vital information

Records 1-4: Disk block index

Record 5: List of new page markers

Records 6-EOF: Disk blocks containing the text.

Before getting into the details, it is necessary to define the meaning of a 'disk block'. It is 1K in size, consisting of 4 256 byte records. Each block contains a 7 byte header (explained later), up to 985 bytes of text, paragraph and control information, and 32 bytes of overflow space to accommodate minor changes without starting a new block. Each of the four possible header/footer pages will occupy its own block, if present. A document may not contain more than 174 blocks. Blocks are numbered from 0 to 173, with block 0 starting at document record 6.

#### Document Record 0 - Header

Byte	Description
00	ID - Always E0, identifies a SS document
01 - 18	24 bytes for the document name
19	Maximum lines per page in half-line increments
1A	Pitch - PS=00
1B	Line spacing in half-line increments
1C - 23	8 bytes for printer driver filename
24-25	Page number to start footers
26-27	page number to start headers
28	Odd footer length in half-lines
29	Odd header length in half-lines
2A	Even footer length in half lines
2B	Even footer length in half-lines
2C	Horizontal cursor position on video display at document close
2D	vertical cursor position on video display at document close
2E	Column position of cursor at document close time
2F - 30	Document line number of cursor at document close time
31 - 45	Tab line 0 (21 bytes)
46 - 5A	Tab line 1 (21 bytes)
[NOTE] - The 21 byte tab line contains 168 bits, each bit representing a column position. If a bit is set, there is a tab stop	



	at that position.
5B-71	Bit map of disk block allocation (23 bytes) [Note] - The first 174 bits of this map represent disk blocks 0-173. If a bit is set, that block is allocated. If the bit is reset, the block is available.
72	Disk block number of odd footer, (if any)
73	Disk block number of odd header, (if any)
74	Disk block number of even footer (if any)
75	Disk block number of even header (if any)
76	Disk block number of tab line table (FF = none) [NOTE] - This block has space for 48,21 byte tab lines, which added to the two available in this record, make up the maximum of 50 tab lines per document.
77	Number of tab lines currently assigned to the document.
78 - 97	Name of author (32 bytes)
98 - B7	Name of operator (32 bytes)
B8 - D7	Comments (32 bytes)
D8 - FF	Not currently used

#### Disk Block Index - Records 1 through 4

Byte 00 of the index contains the number of active text blocks in the entire document, not including any blocks assigned to headers, footers, or tab lines. Starting with Byte 01 of Record 1, there is room for 174, 5 byte group, each group representing a disk block containing text. The groups are arranged in the actual order of the documents text as printed. The meaning of each byte in the group is as follows:

Byte	Description
00	Disk block number (range 00 to AD). If the value is FF, it denotes the end of the index and all following bytes have no meaning. The actual record number in the file can be found by multiplying the block number by 4 and adding 6.
01 - 02	Actual length of valid text in the block
03 - 04	Number of text lines in the block, The upper nybble of byte 04 is used to contain block

control information as follows:

Bit 7 - set if the first line of the block is a whole line, not part of a line from a previous block.

Bit 6 - set if the block contains an open marker '['

Bit 5 - set if the block contains a close marker ']'

Bit 4 - set if the block has been changed (edited)

#### Disk Block Structure

Each disk block starts with a 7 byte header. Bytes 00-01 contain the number of text bytes following the header. This value must be identical to bytes 01-02 of the index group for this block. Bytes 02-06 of the header comprise the default paragraph format for the block as follows:

Byte	Description
02	Column containing the left margin
03	Column containing the right margin
04	Column containing the indent tab
05	Number of the tab line in use.
06	Control byte - following bits are used:

Bit 4 - Set if the paragraph is frozen

Bit 3 - Set if the paragraph is centered

Bits 2 through 0 - used to indicate line spacing in half lines.

After the header comes the actual text. If the block starts with a complete paragraph (line), there will be a 5 byte paragraph group identical in format to the default group in the block header followed by an EF control byte indicating the end of the paragraph info. Then comes the actual text of the paragraph, terminated by an FD, denoting a hard carriage return. This will be followed by a 5 byte paragraph info group for the next paragraph, followed by an EF, and so forth until the number of bytes shown in Bytes 00-01 of the block has been reached.

Everything after this point is MEANINGLESS; the text continues on the next block shown in the disk block index. In the text, the following control codes may be encountered:

Byte	Description
E5	Null - deleted text
EC	Soft page marker
ED	Hard page marker
EF	End of paragraph control info
F0	Start block marker (shows as '[' on screen)
F1	End block marker (shows as ']' on screen)
F2	Normal tab
F3	Align tab
F5	'Code' for printer control (underline, bold, etc)
F6	Filler bytes for insert mode
F7	Space compression for two succeeding spaces (delta)
F8	Soft carriage return replacing a space
F9	Soft carriage return replacing a double space
FA	Hard hyphen
FB	Hard space (for hyphenation)
FC	Hard carriage return during inserting
FD	Hard carriage return
FF	End of file/text.

#### List of New Page Markers - Document Record 5

This record contains the location of any hard page breaks (^) in the document. Byte 00 is the number of markers in the document, followed by 127 pairs of bytes (byte FF is not used). Each non-zero byte pair contains the line number, relative to the start of the document (line 0), where a page break is located.

Armed with the above information, a user with a clobbered file may be able to zap it to a point where it will load properly and allow final repair with the normal SuperSCRIPSIT editing functions. The information in the disk block index MUST agree with what is actually contained in the disk blocks themselves. For example, when the FF byte is encountered in the index, it is a sign that the previous block should contain the FF end of text code somewhere among the valid bytes of that block.

## LS-DOS 6.3.1 Released

MISOSYS has released version 6.3.1 of LS-DOS (see back cover, if you haven't already noticed). This release carries LS-DOS beyond the twentieth century for dating support; the release supports dates from 1980 through 2011 inclusive. Other tweaks have been made to the DOS based on a compromise of requests versus needs.

The release required some changes to SYS0/SYS, and other modules. We did make every attempt to avoid any change to low-memory, even to the point of avoiding a change to the address origin of everything in the BOOT/SYS file. But some enhancement patches previously applied to LS-DOS 6.3.0, such as the Houdé XLR8er patches and the patches MISOSYS had published for the Alpha Tech memory expansion board, needed to be altered. The following patches are versions of LS-DOS 6.3.0 patches previously-published but revised for LS-DOS 6.3.1; make note of the change in the password for system files!

### For MHDT34 CLOCK

```
. SYS3CLK6/FIX - Patch to
LS-DOS 6.3.1 SYS3/SYS - 02/
22/90
. to update system time
from the DS1287 clock
. on every @CLOSE operation
. Apply via, PATCH SYS3/
SYS.SYSTEM6 SYS3CLK6
. BUT NOT TO THE CURRENT
SYSTEM DISK!!!!!!!!!!
.
D00,37=CD 37 21
F00,37=1A CB 7F
D03,6C=67
F03,6C=37
X'2137'=3E 0D F3 D3 B0 DB
B1 FB FE 80 20 1E E5 C5 21
2D 00 01 B1 0A F3 0D ED 41
0C ED 78 E6 80 20 F6 06 03
D3 B0 C6 02 ED A2 20 F8 FB
C1 E1 1A CB 7F C9
. Eop
```

```
. SYS0CLK6/FIX - Patch to
LS-DOS 6.3.1 SYS0/SYS
. to load system date time
from the DS1287 clock
. Apply via, PATCH SYS0/
SYS.SYSTEM6 SYS0CLK6
.
D0D,2D=3E 0D F3 D3 B0 DB B1
FB FE 80 C2 EA 1F 21 A0 1E
F0D,2D=3A C2 04 B7 21 33 00
4E 36 00 23 46 36 00 23 7E
D0D,3D=3E 07 1E 01 CD C6 1E
2B 7E 21 9F 1E 77 0E 03 11
F0D,3D=36 00 C2 AD 1F 2E FF
77 2B 70 2B 71 EB 3D FE 0C
.
. If INTL version of LS-DOS
6.3, the above line should
be
. F0D,3D=36 00 C2 Ad 1F 2E
FF 70 2B 77 2B 71 EB 3D FE
0C
.
D0D,4D=33 00 ED B0 C3 0A 1F
01 B1 0A F3 0D ED 41 0C ED
F0D,4D=38 0E 21 1B 15 11 5C
21 01 30 08 CD D6 20 30 F2
D0D,5D=50 CB 7A 20 F6 06 03
0D ED 79 83 0C ED A2 20 F7
FB C9
F0D,5D=1A FE 0C 30 03 C6 64
12 4F D6 50 FE 20 30 E3 E6
03 3E
.
D0E,3F=21 2D 00 E5 3E 00 1E
02 CD C6 1E E1 C3 E2 1F
F0E,3F=3A C3 04 B7 20 37 06
03 21 FF 00 36 00 2B 10
. Eop
```

### For AlphaTech Memory

```
. BOOTAT/FIX - 10/14/87
. Copyright 1985 Roy
Soltoff, All rights re-
served
. Patch to BOOT/SYS (TRSDOS
6.2/6.3) for Alpha Tech
board
. Apply via: PATCH BOOT/
SYS.LSIDOS BOOTAT (O=N)
. Patch low memory pointer
at 206H
D00,06=19 10
. Patch byte I/O handler at
65EH
D04,5E=88
```

```
. Patch byte I/O handler at
67DH
D04,7D=88
. Patch KCK@ routine at
7F6H
D05,F6=2F
. Patch KCK@ routine at
7F9H
D05,F9=81
D05,FB=57
. Patch ENADIS_DO_RAM and
@BANK routines at 817H
D06,17=CD 4A 08 21 03 0C 39
30 09 E1 ED 73 74 08 31 40
D06,27=03 E5 21 55 08 E3 E5
21 7F 08 23 C5 0E 43 ED 40
D06,37=70 23 47 3A 78 00 30
02 E6 7F 77 E6 FC F6 82 ED
D06,47=41 18 1C F3 22 7C 08
F5 E1 22 77 08 AF C9 CD 4A
D06,57=08 2A 2F 08 7E CB 7F
CB FF 2B C5 0E 43 46 2B 22
D06,67=2F 08 32 78 00 D3 84
ED 41 C1 20 03 31 00 00 21
D06,77=00 00 E5 F1 21 00 00
FB C9 00 00 00 00 00 00 00
D06,87=00 E6 7F FE 1F 30 15
05 FA CB 08 0E 86 28 1D 0E
D06,97=46 05 28 18 05 28 0E
05 3A 02 02 C8 C3 ED 0D E5
D06,A7=21 14 04 18 0B CD B3
08 C0 78 0E C6 E5 21 10 04
D06,B7=CD 10 10 78 E6 07 07
07 07 B1 32 C8 08 AF 3E 08
D06,C7=CB 46 E1 C9 E5 21 05
80 39 E1 38 D0 C5 0E 46 CD
D06,D7=A6 08 78 C1 20 C6 CD
F4 0F E6 7F 32 02 02 A9 B0
D06,E7=4F CB 79 06 00 C8 E3
BF C9
. Patch video driver at
0C8FH
D0A,8F=2F
. Change pointer in FDC
driver to account for patch
at 0E3FH
D0C,3F=18 10
. Add code at end of FDC
driver for @BANK and TASKER
at 0FF4H
D0D,F4=B7 28 01 3C D3 43 3A
02 02 47 79 C9 21 77 00 CB
D0E,04=F6 21 02 02 C9 32 02
02 21 77 00 C9 47 0F 0F 0F
D0E,14=E6 03 85 6F C9
. Eop
```

```
. SYS0AT63/FIX - 02/20/90
. Copyright 1985,86 Roy
Soltoff, All rights re-
served
. Patch to SYS0/SYS (LS-DOS
6.3.1) for Alpha Tech board
. Apply via: PATCH SYS0/
SYS.SYSTEM6 SYS0AT
. Patch the SVC table at
1CCH
D08,6E=88;F08,6E=77
. Patch the disk I/O han-
dler at 19FDH
D09,5B=88;F09,5B=77
. Patch the TASKER at 1C06H
D0B,8C=CD 00 10 7E F5 DB 43
F5 AF 77 D3 43
F0B,8C=21 77 00 CB F6 21 02
02 7E 36 00 F5
. Patch the TASKER at 1C3DH
D0B,C3=D3 43 F1 CD 09 10 CB
B6
F0B,C3=32 02 02 21 77 00 CB
B6
. Patch the TASKER at 1C91H
DOC,17=F1 E3;FOC,17=E3 FB
. Patch system initializa-
tion routines at 1E48H
DOC,D6=D3 84 26 FF 56 01 43
1F 71 C5 48 0D 06 03 3E 66
FOC,D6=47 3E A7 0E 84 ED 41
21 FF FF 22 0E 04 22 1C 00
DOC,E6=EF C1 ED 41 5E 34 7E
BB 73 28 09 C5 48 0D 06 01
FOC,E6=56 36 55 ED 79 5E 77
ED 41 BE 72 ED 79 73 ED 41
DOC,F6=3E 66 EF C1 10 E3 ED
41 72 CD 9A 21
FOC,F6=3E FE 28 02 3E F8 32
01 02 32 00 02
. Patch system initializa-
tion routines at 219AH
D10,34=6C 22 0E 04 22 1C 00
21 10 04 CB 86 11 14 04 0E
F10,34=00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
D10,44=04 ED B0 C9
F10,44=00 00 00 00
. Eop
```

```
. SYS1AT/FIX - 02/20/90
. Copyright 1985 Roy
Soltoff, All rights re-
served
. Patch to SYS1/SYS (TRSDOS
6.2/6.3) for Alpha Tech
board
. Apply via: PATCH SYS1/
```

```
SYS.SYSTEM6 SYS1AT
. Patch cleanup routine at
1E77H and 1E7AH
D00,9C=7F;F00,9C=6E
D00,9F=2F;F00,9F=35
. Patch restoral of bank_0
at 1EC0H
D00,E5=88;F00,E5=77
. Eop
```

### For XLR8er Board

```
*** XLR8A63/JCL *** - 02/
20/90
//.Procedure to patch an
LSDOS 6.3 diskette for use
with an XLR8er
//.Drive to be patched must
be given on JCL command
line.
//.If needed parameter
(O=N) may be included, same
meaning as PATCH
//IF -d
//DO XLR8A63 (D=d [,O=N])
//QUIT
//END
//IF O
.DO XLR8A63 (D=#d#,o=#o#)
//ELSE
.DO XLR8A63 (D=#d#)
//END
.Diskette in drive #d# is
about to be patched
. Press ENTER to continue,
BREAK to abort
//ALERT (7,7)
//IF O
patch sys0/sys.system6:#d#
using xlsys0a/fix (o=#o#)
patch boot/sys.system6:#d#
using xlboota/fix (o=n)
patch sys2/sys.system6:#d#
using xlr8s2/fix (o=#o#)
patch sys12/sys.system6:#d#
using xlr8s12/fix (o=#o#)
patch backup.utility:#d#
using xlr8bu/fix (o=#o#)
//ELSE
patch sys0/sys.system6:#d#
using xlsys0a/fix
patch boot/sys.system6:#d#
using xlboota/fix (o=n)
patch sys2/sys.system6:#d#
using xlr8s2/fix
patch sys12/sys.system6:#d#
using xlr8s12/fix
```

```
patch backup.utility:#d#
using xlr8bu/fix
//END
//ALERT 0,0,1,5,0,2
.eop
```

```
.XLBOOTA/FIX - To use
XLR8er special features
(20-Feb-90)
.PATCH boot/sys.system6
USING xlboota/fix (O=N)
.must also PATCH sys0/
sys.lsidos USING xlsys0a/
fix
.fix lomem pointer @x'0206'
d00,06=6D 10
f00,06=F4 0F
.new @MUL16 svc @x'06c9'
d04,c9=57 5D 6F ED 6C ED 5C
7B 5A 16 00 19 C9
f04,c9=C5 EB 4F 21 00 00 7D
06 08 29 17 CB 01
.TASKER patch1. end of old
@mul16 @x'06d6'
d04,d6=ED 28 38 E3 E5 AF ED
39 38 21 78 00 C9
f04,d6=30 03 19 CE 00 10 F5
4F 7D 6C 61 C1 C9
.@BANK beginning @x'0877'.
XBUR$ @x'087b'
d06,77=79 C3 FD 0F 00
f06,77=E6 7F FE 03 D2
.@VDCTL @x'0b9a' JP to fix
d09,9a=58 10
f09,9a=42 0D
.ROWCOL 2 ADDR @x'0ddf'
multiply
d0b,df=2E 50 ED 6C 00 00 00
00
f0b,df=21 50 00 CD C9 06 65
6F
.change delay in FDC driver
@x'0f26'
d0d,26=24
f0d,26=12
.add XLR8 module @x'0ff4'.
@BANK and @VDCTL
d0d,f4=18 FE 6C 10 04 58 4C
52 38 E6 7F FE 0B 30 2D D6
f0d,f4=00 00 00 00 00 00 00
00 00 00 00 00 6d b6 6d b6
d0e,04=03 30 0A AF ED 39 38
79 E6 7F C3 7E 08 E5 04 05
f0e,04=6d b6 6d b6 6d b6 6d
b6 6d b6 6d b6 6d b6 6d b6
d0e,14=28 30 6F 2C 3E 80 07
2D 20 FC 21 7B 08 05 28 17
```



```
f0a,14=6d b6 6d b6 6d b6 6d
b6 6d b6 6d b6 6d b6 6d b6
d0a,24=05 28 1A 05 28 0A 05
3A 02 02 E1 C8 3E 2B B7 C9
f0a,24=6d b6 6d b6 6d b6 6d
b6 6d b6 6d b6 6d b6 6d b6
d0a,34=ED 34 20 0A B6 18 02
2F A6 77 AF E1 C9 A6 3E 08
f0a,34=6d b6 6d b6 6d b6 6d
b6 6d b6 6d b6 6d b6 6d b6
d0a,44=E1 C9 21 05 80 39 E1
38 E3 07 07 07 C6 38 ED 39
f0a,44=6d b6 6d b6 6d b6 6d
b6 6d b6 6d b6 6d b6 6d b6
d0a,54=38 C3 DB 08 ED 38 38
B7 CA 42 0D F5 AF ED 39 38
f0a,54=6d b6 6d b6 6d b6 6d
b6 6d b6 6d b6 6d b6 6d b6
d0a,64=CD 42 0D E3 ED 21 38
E1 C9
f0a,64=6d b6 6d b6 6d b6 6d
b6 6d
.eop
```

```
.XLSYS0A/FIX - To use
XLR8er special features
(20-Feb-90)
.PATCH sys0/sys.system6
USING xlsys0a/fix
.must also PATCH boot/
sys.lsidos USING xlboota/
fix (0=N)
.new @MUL8 svc @x'190a'
d06,f0=57 ED 5C 7B C9
f06,f0=C5 57 AF 06 08
.tasker patch2. end of old
@MUL8 @x'190f'
d06,f5=ED 29 38 7C 32 02 02
C9
f06,f5=87 CB 23 30 01 82 10
F8
.fix call to modified @bank
instead of SVC @x'1a16'
(disk I/O)
d09,74=CD 77 08
f09,74=3E 66 EF
.tasker (beginning)
@x'1c11'
d0b,97=67 CD D6 06
f0b,97=F5 21 78 00
.tasker (end) @x'1c3c'
d0b,c2=E1 CD 0F 19
f0b,c2=F1 32 02 02
.special registers setup in
initialisation @'1e79' &
x'219A'
d0d,07=CD 9A 21
f0d,07=00 00 00
d10,34=3E 40 ED 39 32 ED 38
```

```
36 E6 FC F6 02 ED 39 36 AF
f10,34=00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
d10,44=ED 39 38 ED 39 39 3E
88 ED 39 3A C9
f10,44=00 00 00 00 00 00 00
00 00 00 00 00
.eop
```

```
.XLR8S2/FIX - Patch to
SYS2/SYS - 02/20/90
.Apply via, PATCH SYS2/
SYS.SYSTEM6 XLR8S2
D00,E4=12;F00,E4=09
D00,EE=40;F00,EE=20
D01,04=40;F01,04=20
.eop
```

```
.XLR8S12/FIX - Patch to
SYS12/SYS - 02/20/90
.Apply via, PATCH SYS12/
SYS.SYSTEM6 XLR8S12
D03,7C=12;F03,7C=09
D03,86=40;F03,86=20
D03,9C=40;F03,9C=20
.eop
```

```
.XLR8BU/FIX - Patch to
BACKUP/CMD of 6.3.1 - 02/
20/90
.Apply via, PATCH
BACKUP.UTILITY XLR8BU
D17,0A=12;F17,0A=09
D17,10=40;F17,10=20
D17,24=40;F17,24=20
.eop
```

## Bug in RSHARD

Fm MISOSYS, Inc: We have received a few reports of folks experiencing problems using the RSHARD driver package when trying to install a second hard drive which has something other than four heads. The problem was traced to a design flaw in the driver initializer which updated the loaded driver instead of the installed driver when the data on the second drive was entered. The number of heads is kept in a data space in the installed driver. The following two patches correct for this problem.

```
. RSH53/FIX - 02/14/90 -
Patch to RSHARD5/DCT
. Corrects initialization
for 2nd drive installed
. Apply via, PATCH RSHARD5/
DCT RSH53
D00,CE=1C;F00,CE=38
D00,EB=C3;F00,EB=CD
D01,05=C9 00;F01,05=18 44
D01,13=C8 00;F01,13=28 09
D01,19=C9 00;F01,19=18 03
D01,41=CD 68;F01,41=36 65
D04,0D=CD FB 60 3A 57 64 B7
CA 0E 61 21 01 02 00 64 45
69 5E 23 56 D5 DD E1 CD E4
60 DD 7E 02 B7 CA 0E 61 C3
3F 61
F04,0D=3A 57 64 B7 28 1F 47
21 45 69 5E 01 02 00 64 23
56 23 D5 DD E1 DD 7E 00 FE
C9 28 0B DD 7E 04 E6 08 4F
3A 73
D05,D0=62;F05,D0=61
D08,F9=CD 36 65 C0 DD 36 02
04 C9
F08,F9=00 00 00 00 00 00 00
00 00
D09,9D=00;F09,9D=04
D09,A0=00;F09,A0=04
. Eop
```

```
. RSH63/FIX - 02/15/90 -
Patch to RSHARD6/DCT
. Corrects initialization
for 2nd drive installed
. Apply via, PATCH RSHARD6/
DCT RSH63
D00,A9=1C;F00,A9=38
D00,C6=C3;F00,C6=CD
D00,E0=C9 00;F00,E0=18 44
D00,EA=C8 00;F00,EA=28 09
D00,F0=C9 00;F00,F0=18 03
D01,1C=28 39;F01,1C=59 35
D04,34=CD D6 30 3A 7A 34 B7
CA E9 30 21 8D 39 5E 23 56
D5 DD E1 CD BF 30 DD 7E 02
B7 CA E9 30 C3 1A 31
F04,34=3A 7A 34 B7 28 1F 47
21 8D 39 5E 23 56 23 D5 DD
E1 DD 7E 00 FE C9 28 0B DD
7E 04 E6 08 4F 3A 96
D05,FE=62;F05,FE=61
D09,58=CD 59 35 C0 DD 36 02
04 C9
F09,58=00 00 00 00 00 00 00
00 00
D09,E5=00;F09,E5=04
D09,E8=00;F09,E8=04
. Eop
```



## Upgrade your 4P with external floppy drives

The following article is an excerpt from **On the Upgrade**, by Tsun Tam, which appeared in the January 1986 edition of *80 Microcomputing* magazine; it has been reprinted with their permission.

Your 4P's "transportable" moniker shouldn't limit your vision. With the modifications I'll describe below, you can give your 4P a power boost that makes it a worthy competitor to its full-blown desktop brothers, the Models 4 and 4D.

I'll show you how to add two double-sided, 40-track internal drives and two double-sided, 40- or 80-track external drives for a total of 2.16 megabytes of disk storage, increase RAM to 128K, incorporate electrical surge protection, and customize your Model 4 DOS (see Table 1 for a complete parts list).

One preliminary note, however: Tandy doesn't sanction these modifications, so you'll void your warranty by making them.

### Two 4Ps

Tandy made two versions of the 4P: The original has a black-and-white monitor, and the later version, which uses gate arrays, has a green screen. Aside from basic circuitry changes, there isn't much functional difference between the two; I'll note any relevant differences.

### Drive Time

You can install two internal 40-track dual headed drives to give your 4P a total internal storage capacity of 720K. In addition, you can add and control two external, double-sided drives. I'll de-

scribe how to do both.

First, remove the 4P's snap-on cover and leave the keyboard inside the computer. Next, put the 4P on a soft surface - you can use several layers of bathroom towels - with the hidden base of the keyboard toward you.

The case is attached to the chassis by six Phillips-head screws. The four beige-colored ones are adjacent to the cover latches; remove them and put them in a safe place. The last two are chrome-plated machine screws hidden under the carrying handle. Tip the 4P up so that the screen rests on the towels. Raise the carrying handle with one hand and remove the screws with the other.

Now gently remove the case by pulling straight up - it should dislodge easily. Be careful, the keyboard may suddenly fall onto your workbench. Should it remain standing with its back to you, lay it flat on the workbench and protect the keys with a towel.

You need to remove the 4P's two original Tandon disk drives from the side of the computer as a single unit. Nine screws hold the drives in place: Four are on top; two are on the base deep inside, below the assembly; one is on the outer side; and

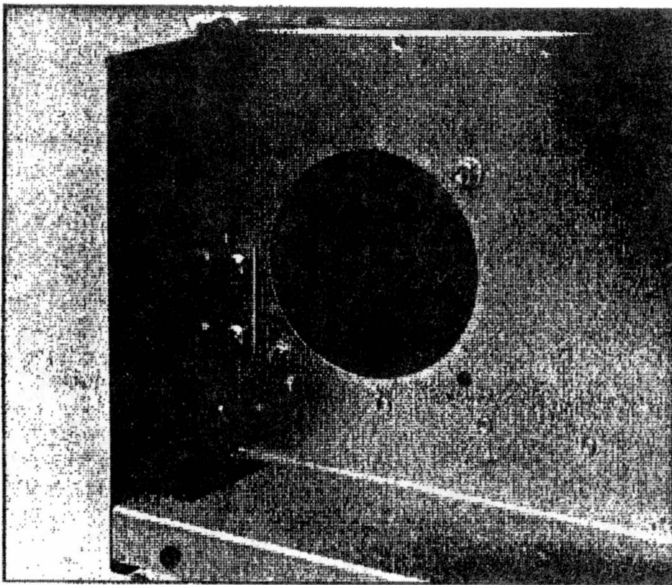


Photo 1. Two of the screws that hold the computer's back plate are located in the disk drive bay near the ventilation fan.

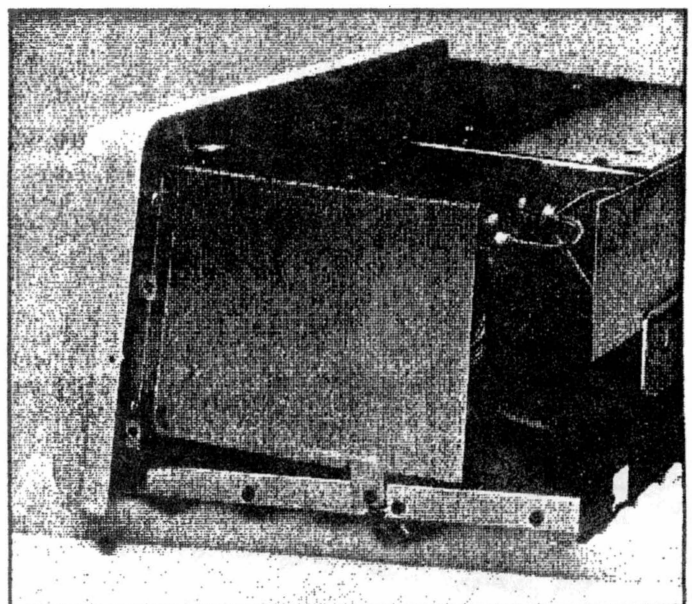


Photo 2. The main PC board is encased in a pan assembly held together with eight screws, four per side.

finally, two chrome-plated screws attach the assembly to the front computer bezel. Be sure to keep track of the two screws at the base of the drives; they tend to fall into the computer. If this happens, remember to take them out when you remove the main printed circuit (PC) board.

Remove the two power connectors and the drive cable from the two disk drives and save the drive cable for later. Gently lift out the two drives. Remove the six machine screws that attach the drives to the subchassis, and remove the drives. If you have either TEAC or Panasonic drives, make sure you orient them with the latch levers down. You should install the Mitsubishi drives with the light-emitting diode (LED) down.

You can measure the exact hole locations for the new drives by inserting the drives in the subchassis. The front edge of the drives must extend from the top front of the subchassis by 1-7/16 inches - too much and the drives won't fit your computer, too little and you'll have a gap between the drives and the front computer bezel. The best thing to do is drill out the holes to a slightly larger size and use washers on the screw heads. Then adjust the position of the drives before tightening the screws. This way, you can position the drives without a lot of fancy measur-

ing and drilling.

Before you install the drives, you need to set the DIP shunts or jumpers according to Fig. 1. Make sure you have the drive-select settings correct. Use DS0 for the drive closest to the monitor and DS 1 for the other. In addition, the drives should load their heads when the computer sends the "motor on" signal.

You should also short out the HM shunt and remove the terminating resistor pack, an integrated circuit-like package usually blue and socketed.

With the drives properly programmed, install them inside the subchassis using your six screws and put the assembly aside.

### ***Power Cables & Drive Connectors***

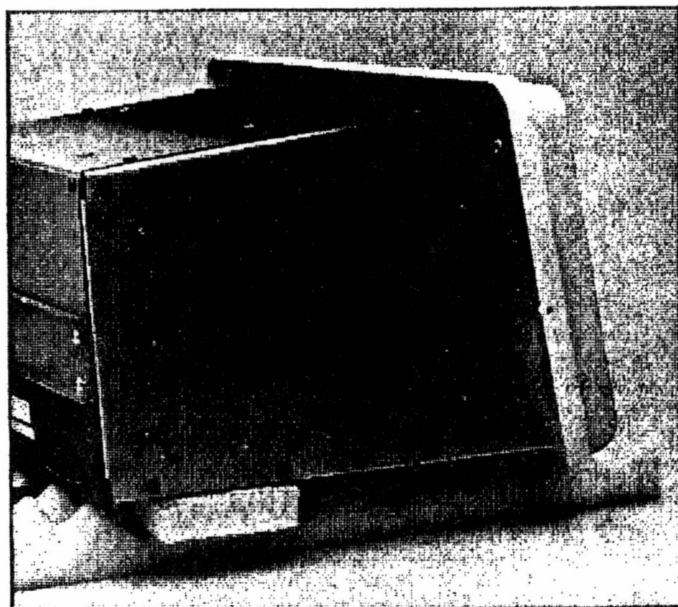
If you use TEAC or Panasonic double-sided drives internally, the 4P's original power cables will work fine. But if you plan to use Mitsubishi drives you must extend the power cable by about 3 inches. For this, you need three 3-inch lengths of wire, a soldering iron and solder, heat-shrink tubing, and a combination wire

cutter and stripper.

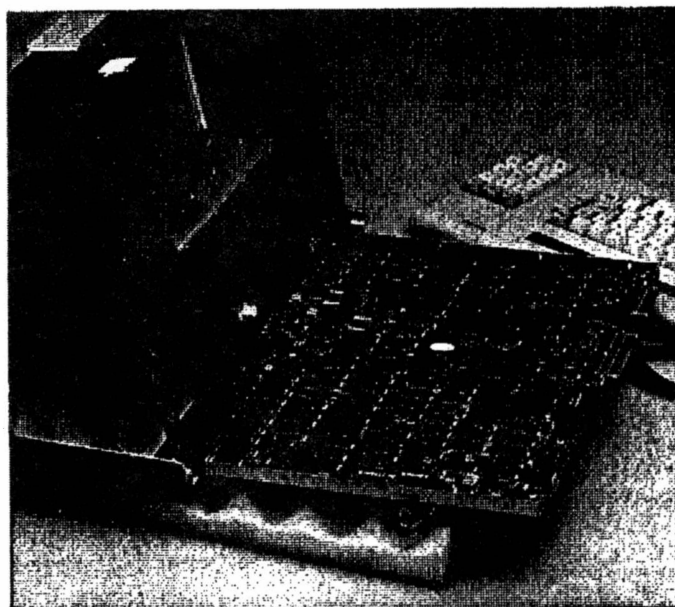
To access the computer's power supply, which is screwed to the top of the main chassis, remove the six screws that hold the top plate to the chassis and then remove the four top screws that attach the back plate. Two connectors attach to the power supply. Remove them by carefully prying them off with your small flat-bladed screwdriver.

The drive power cable feeds to the drives through a small rubber grommet-lined opening. Locate this cable and cut it at a point where you can splice in the three lengths of wire. Use the heat-shrink tubing to insulate the splicing joints. After making the splices, don't replace the power supply yet, as you'll make an ancillary modification to it later.

To make the two new drive controller cables, you can re-use the connectors on the original internal drive cable - a small X-Acto knife is handy here. Make the cables using the new 34-pin edge connectors and parts from a disk drive extender cable. [Note: MISOSYS has these connectors available] The modification requires the use of the "reversed" connectors, with the odd- and even-numbered pins reversed from those of their counterparts. Clamp the male connector onto the



*Photo 3. The left side of the 4P, showing the screws holding the main PC board in the pan.*



*Photo 4. Gently pry the two halves of the pan assembly apart.*

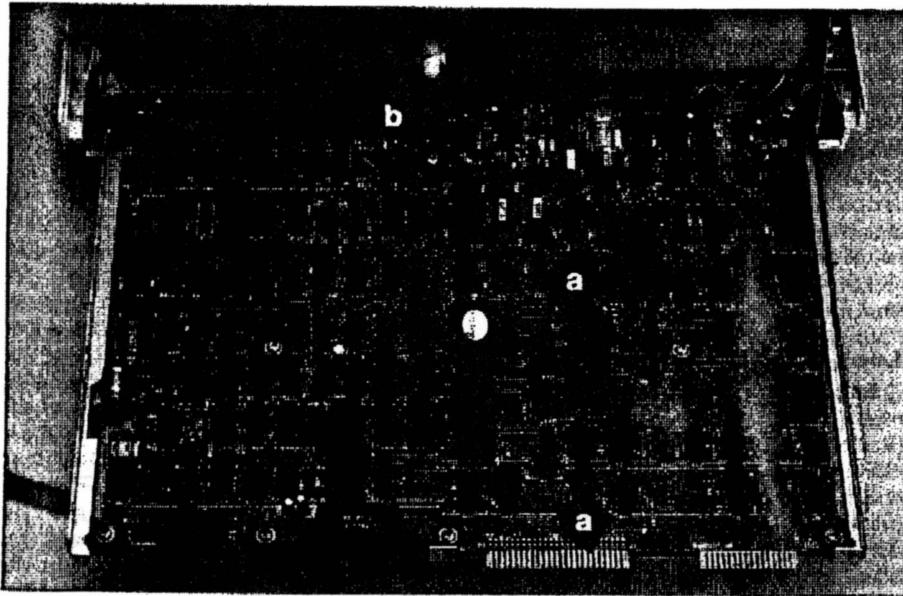


Photo 5. (a) Insert the two black plastic PC board retainers into the main PC board. (b) Add one bank of eight 4164 dynamic random-access memory chips in socket locations U153-U160.

### Mitsubishi Disk Drives Radio Shack Settings

1U ---	15-C: HM ---	DS0 and DS1
Panasonic / Radio Shack		
	HH ---	

Figure 1. Set the DIP shunts or jumpers according to these specifications.

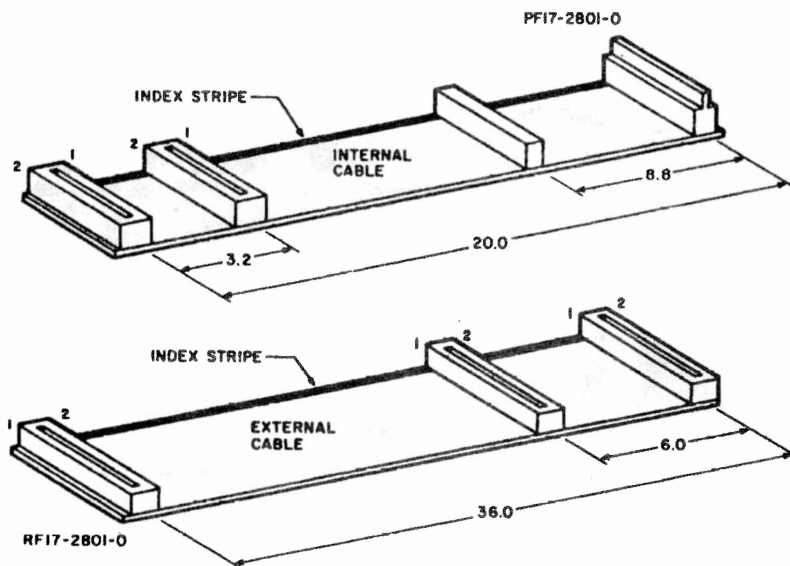


Figure 2. Dimensions for making the new internal and external cables.

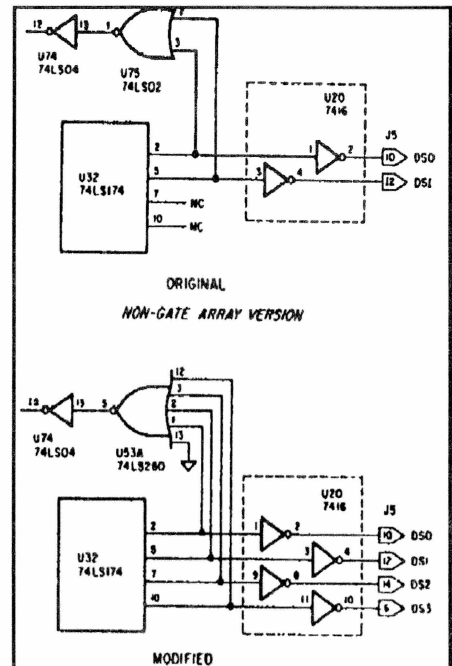


Figure 3. Original 4P version showing the original and modified main PC board.

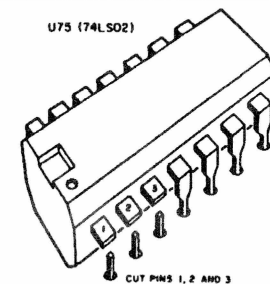


Figure 4. Original 4P version. Make the cuts as close as possible to the IC body.

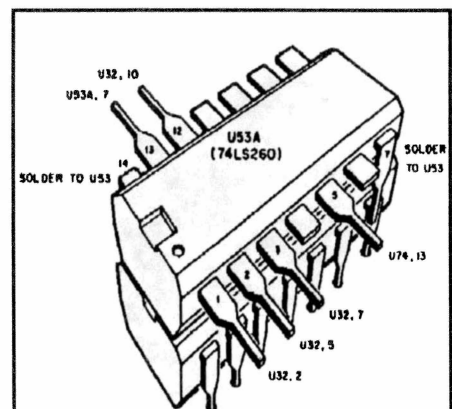


Figure 5. Original 4P version. Modifying the new 74LS260; soldering pins 7 and 14 of U53A to U53.

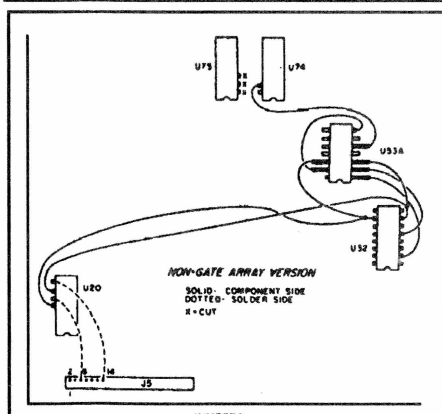


Figure 6. Jumper diagram, original 4P version.

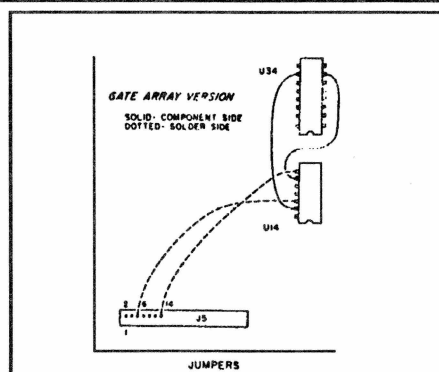


Figure 7. Jumper diagram, gate array version.

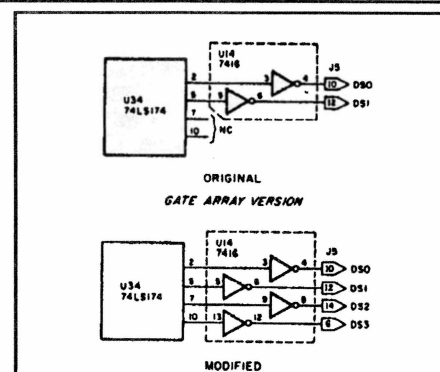


Figure 8. Gate array version showing the original and modified main PC board.

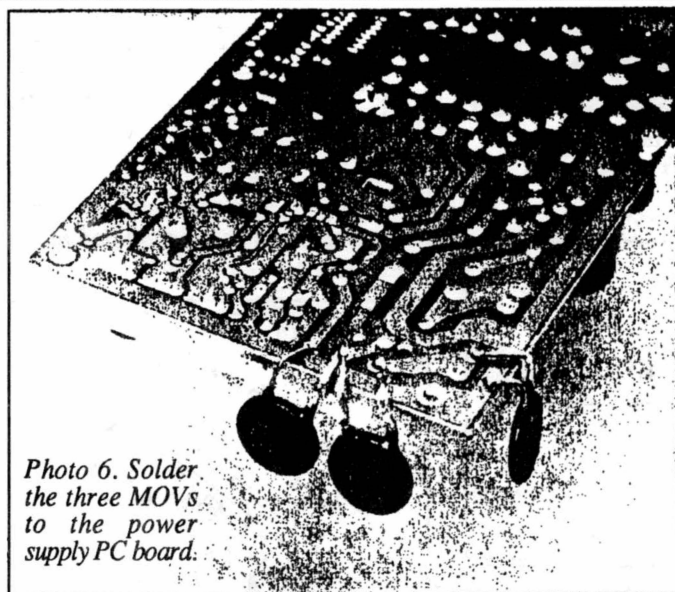


Photo 6. Solder the three MOVs to the power supply PC board.

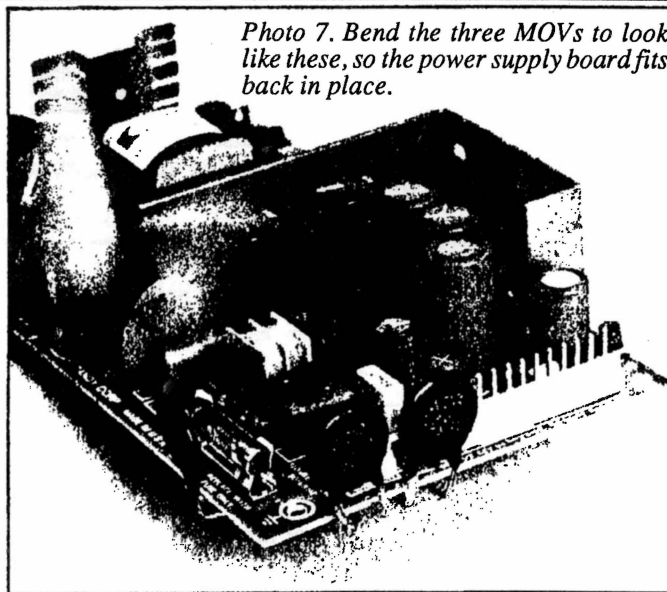


Photo 7. Bend the three MOVs to look like these, so the power supply board fits back in place.

Photo 8 (below). Reconnect the two power connectors to the new disk drives and attach the internal drive cable.

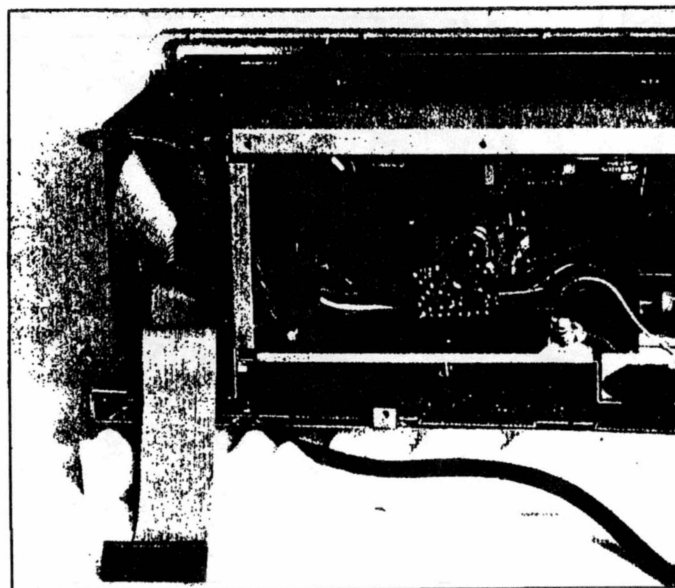
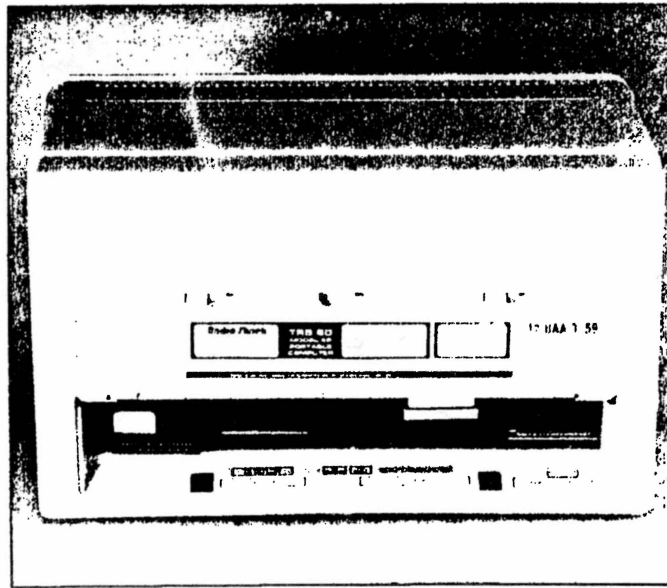


Photo 9 (below). The modified rear of the 4P. Make sure the external drive cable is accessible for proper operation.





internal drive cable and the female onto the external drive cable.

Cut a piece of 34-conductor cable 20 inches long, squaring off the ends. Using a vise or hammer, make the internal drive cable according to the dimensions in Fig. 2. Treat the male connector from the extender cable carefully, and attach a female connector to it to prevent damage when you crimp on the ribbon cable. Use a vice to squeeze the connector to the cable, or a hammer to gently tap the cable onto the connector (see Fig. 2).

After you make the internal cable, prepare the cable for the two external drives, again using the dimensions in Fig. 2. Note that the original internal drive cable may have reusable connectors. You can use these connectors successfully if you dismantle and reassemble them correctly.

If you reuse the old connectors, note that pins are missing from each of the even-numbered rows. This is Radio Shack's way of electrically selecting the two internal disk drives, as both original-equipment manufacturer drives are configured identically. All odd-numbered pins are grounded.

When you install these old connectors on your new external drive cable, position them so that pin 34 becomes the "new" pin 1 and pin 33 becomes pin 2. The odd row must mate with the even numbers on the disk drive PC board, as you need full access to lines 6, 14, and 32 of the drives. The first two select drives 3 and 2 respectively, and line 32 selects a corresponding side.

### Modifying the Main Printed Circuit Board

With the two cables done, your double sided drive conversion is nearly complete. Before the final reassembly, you need to modify the main PC board for controlling the two external drives.

Remove the back chassis plate to access the main PC board. If you have an internal modem connected, remove the modem cable from the computer's RS-232 socket. Remove the back plate by undoing all the

visible black screws and the four screws holding the U-shaped channel or bracket that supports the computer handle. Two of these screws are in the disk drive bay near the ventilation fan (see Photo 1). Lift the back plate off and set it aside.

The main PC board is encased inside a pan assembly screwed together with eight screws, four on each side (see Photos 2 and 3). Remove the screws and put them in a safe place. Gently pry the two halves of the pan assembly apart (see Photo 4). Rest the bottom pan (the part with the PC board) on the keyboard. Four connectors supply the PC board with power and communication data, gently remove them from the board. With the connectors separated (they are polarized for easy reassembly), pull the lower pan assembly from the rest of the computer. Make certain that you are grounded with the proper grounding straps, as you'll be handling CMOS devices.

Remove the keyboard cable from the main PC board, then separate the PC board itself from the lower chassis by removing the supporting screws. Don't confuse these chrome-plated sheet metal screws with the machine screws.

Lift the PC board off, put it on a towel, and look for the correct integrated circuits

(ICs). For the original Model 4P, first locate U75 (74LS02) (see Figs. 3 and 4). Sever the connections to pins 1, 2, and 3 by cutting these pins with your wire clipper. Make the cuts as close to the IC body as possible. Next, modify the new 74LS260 by following the diagram in Fig. 5. Bend all pins straight out except for pins 7 and 14 (the power pins). You may cut off pins 4, 6, 8, 9, 10, and 11, as you won't use them. Locate U53 and piggyback the 74LS260 to it (I'll refer to the top chip as U53A). You must align pin 1 of both chips. Solder pins 7 and 14 of U53A to U53. Follow the jumper diagram of Fig. 6 and solder your connections using wire-

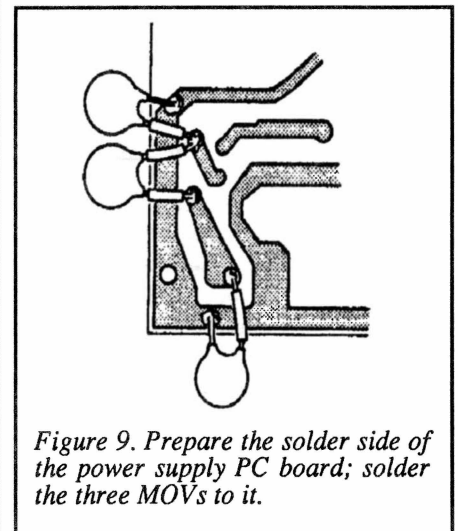


Figure 9. Prepare the solder side of the power supply PC board; solder the three MOVs to it.

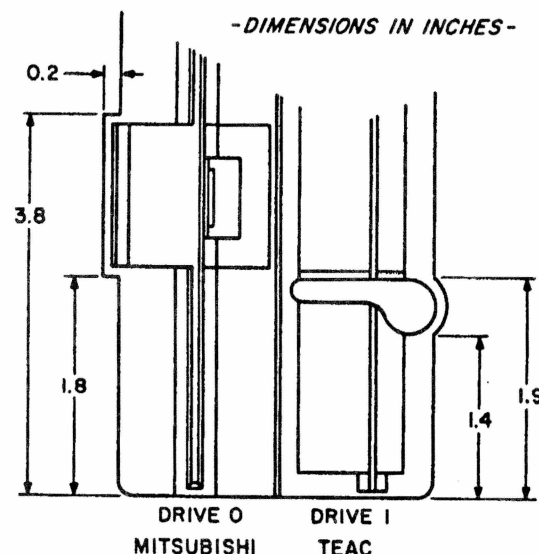


Figure 10. Dimensions for altering Mitsubishi drive zero and TEAC drive 1.

Quantity	Description
2	Double-sided disk drives
6	#4-40 English screws or 6-32 or 3mm metric screws
4	standard 34-pin female drive connectors, and one pair of reversed male and female 34-pin connectors
56 inches	34-conductor ribbon cable
1	74LS260 (original 4Ps only)
3	3-inch lengths of electrical wire (for Mitsubishi drives only)
8	4164 dynamic RAM chips
3	GE or Panasonic MOVs (Radio Shack part number 276-568) or Sprague Transorbs
#2 Phillips screwdriver with at least a four-inch blade	Miniature wire clippers
Portable electric drill with a 3/16-inch bit (or a Whitney-type sheet metal punch)	Wire-wrapping wire; Utility knife
Small flat-bladed screwdriver or IC remover	Hot-melt glue gun or RTV silicone glue
Wire stripper; Heat-shrink tubing	Hacksaw blade; Round file; Hammer
Soldering iron and solder	Small X-Acto knife or single-edge razor blade
	Machine-shop bench vise

Table 1. Parts list

## Original Version

- 1. Cut pins 1, 2, and 3 on U75 (74LS02)
- 2. Prepare the pins on your new 74LS260 (U53A)
- 3. piggyback the 74LS260 on top of U53
- 4. Solder pins 7 and 14 of U53A to U53

	Solder these jumpers		Length of Jumper
	From	To	
— 5.	U32, pin 2	U53A, pin 1	1-1/4"
— 6.	U32, pin 5	U53A, pin 2	1-1/2"
— 7.	U32, pin 7	U53A, pin 3	1-1/2"
— 8.	U32, pin 10	U53A, pin 12	1-1/2"
— 9.	U53A, pin 7	U53A, pin 13	1"
— 10.	U74, pin 13	U53A, pin 5	1-1/2"
— 11.	U20, pin 8	J5, pin 14 (DS2)	1-1/2**
— 12.	U20, pin 10	J5, pin 6 (DS3)	1-1/2**
— 13.	U32, pin 7	U20, pin 9	6"
— 14.	U32, pin 10	U20, pin 11	6"

\*\* denotes that these connections are made on the solder side of the main PCB,  
Unused gates used in this conversion: U20, pins 9 and 8; U20, pins 11 and 10.

## Gate Array Version

	From	To	Length of Jumper
— 1.	U14, pin 9	U34, pin 7	2-1/2"
— 2.	U14, pin 13	U34, pin 10	2-1/2"
— 3.	U14, pin 8	J5, pin 14 (DS2)	2-1/2***
— 4.	U14, pin 12	J5, pin 6 (DS3)	2-1/2***

\*\* denotes that these connections are made on the solder side of the main PCB.  
Unused gates used in this conversion: U14, pins 9 and 8; U14, pins 11 and 10.

Table 2. Connections for the original and gate array Model 4Ps.

wrapping wire (refer to Table 2 for these connections). Check each connection as you make it. Note that the solid lines in both Figs. 6 and 7 are made on the component side of the board. Also note that pins 7 and 10 of U32 have two jumpers connected. The dotted lines are made on the solder side of the board; now make these connections.

If you have the gate array version of the 4P, you only need to add four jumper wires, as Radio Shack supplies all the parts for this conversion. See Figs. 7 and 8 for details. Of the two versions, this one is by far the neatest.

Your 4P is now hardware-ready for up to four disk drives. Reassemble the PC board to the lower chassis pan and reconnect the keyboard cable. Complete this operation by refitting the PC board assembly to the main chassis.

## RAM Charge

You can upgrade RAM to 128K by adding one bank of eight 4164 dynamic random-access memory chips (150 nS or 200nS) in socket locations U153-U160 (see Photo 5). These chips are static-sensitive, so be sure to ground yourself. Insert the RAM chips and move the jumper at E12-E13 to E11-E12 (original version) or E2-E3 to E1-E2 (gate array version). Your machine now has 128K.

## Good MOVE

To relieve your 4P of the Glitch Syndrome, you need three General Electric or Panasonic metal oxide varistor (MOV) surge protectors (Radio Shack catalog number 276-568) or Sprague Transorbs, and four pieces of plastic insulation tubing one-half inch long (you can use the insulation from common 18-gauge zip cord).

Get the power supply you put aside earlier and remove the circuit board from the sheet metal subchassis by undoing the four chrome machine screws. Lift off the power supply and prepare the solder side by referring to Fig. 9. Locate the corner of the power supply PC board where the ac power connector is soldered. On the sol-

der side, scrape off the green solder protection layer at the designated areas in Fig. 9 using a small knife or single-edge razor blade. Tin the affected areas with a bit of solder.

Now slip the four pieces of plastic insulation tubing over the leads of the MOVs. Following Fig. 9 and Photo 6, solder the three MOVs to the power supply PC board. MOVs aren't polarized, so there's no need to orient them in any specific way. Once you do this, your 4P has common and differential-mode glitch protection. Before replacing the PC board on the sub-chassis plate, bend the three MOVs so they look like those in Photo 7.

### Reinstallations

You can now reattach the power supply section to the main computer chassis, making sure to use the correct screws. Don't install the chassis back plate yet, as you still have more to do.

You can reinstall the disk drives now. However, if you have Mitsubishi drives, you'll have to alter the computer's front bezel (see Fig. 10). Carefully mark the location and size of the cut-out area. Make the two horizontal cuts first. As this may take some effort, try using a hacksaw blade without its frame. Next, use your utility knife to cut and remove the plastic material between the two horizontal cuts. Line the area with paper to prevent plastic waste chips and filings from getting inside the computer.

If you're using TEAC drives, also see Fig. 10. Some Model 4Ps require that you trim and alter the front bezel around drive 1. Mark off the center of the alteration with your X-Acto knife and, with a round file, remove enough plastic for clearance of the door latch of drive 1. After cutting and filing, you may want to use a small amount of black model makers' paint color in the exposed beige area.

Now you can reinstall the drive subassembly onto the main computer chassis. To facilitate attaching the two screws to the bottom of the subassembly, use a little adhesive tape to hold the screw to the screwdriver. Reconnect the two power

connectors to the two new disk drives and attach the internal drive cable, using Photo 8 as a guide. A tab on the chassis pan encases the main PC board. Make sure you properly route the cable behind this tab; you may have to bend it.

Now reattach the chassis back plate. Run a bead of hot-melt glue on the back of the male connector of the internal drive cable and attach this connector to the chassis back plate. Reattach the handle support bracket using the four long chrome machine screws.

Before you reinstall the case, position the keyboard so that the keys are up against the chassis where the main PC board is located. Carefully lower the outer case over the computer and position it properly so you can fasten it with the six screws. Photo 9 shows the modified rear of the 4P.

Use the external drive cable you made earlier to connect to the external drive adapter at the rear of the modified 4P. Make sure this cable hangs down for proper operation (see Photo 9). Mate the two drive connectors on the external cable to the drives' PC board edge by observing proper pin orientation. Pin 1 on the connector mates with pin 1 on the PC board.

### Software

You now have a Model 4Plus! Before the internal double-sided drives spring to life, however, you need to modify and patch your operating system. The following procedures let you customize Tandy's own TRSDOS 6.2 so that they access both sides of a disk and up to four drives. If you're using TEAC drives, the labels on the disk must face away from the monitor screen. The labels must face the screen on all other drives.

For TRSDOS 6.2, you need a copy of the original version of the DOS and a blank disk. Insert the copy in drive zero and the blank in drive 1. Invoke the Format command by typing in `FORMAT :1 (SIDES=2)`. Answer the format prompts accordingly and use `DOS62X2` as the new disk name. This alerts you that this version of the DOS is double-sided. After you format the disk, check the drive-1 directory to

see all those added K's.

Transfer the system and all related files from the master copy to your disk formatted for two sides by typing in `BACKUP $:0 :1 (SYS,INV)`. This takes time, as TRSDOS needs to copy by files rather than execute an exact-image copy. With this done, put away your single-sided TRSDOS, put the new disk in drive zero, and boot up. Now check the directory to see the amount of free space available. Check it again with the `DOS Free` command. Your TRSDOS is now usable, but to make it format automatically in double-sided fashion and acknowledge the two external drives, apply the following patches (otherwise TRSDOS treats the drives as illegal):

**PATCH FORMAT/CMD.UTILITY**  
(D09, 5B=02:F09, 5B=00)

Even though you patch the Format utility, you can still format single-sided disks by specifying the sides parameter as 1.

The patch below makes your DOS access the two external drives. At TRSDOS Ready, type in:

**PATCH BOOT/SYS.LSIDOS:0**  
(D02, 84=C3:F02, 84=C9)  
**PATCH BOOT/SYS.LSIDOS:0**  
(D02, 8E=C3:F02, 8E=C9)

After you apply the two patches, reboot the system to test the accessibility of two external disk drives. Notice that drives 2 and 3 are no longer illegal. When you're satisfied that the DOS behaves properly, label this disk as your master double-sided TRSDOS disk and write-protect it. Use it to make working copies and store your master in a safe place.

*Note: MISOSYS, Inc has the following two PCD (previously known as Kel-Am) 34-pin edgecard connectors available (prices do not include shipping):*

**R-KA-P17 - male \$8**  
**R-KA-R17 - female \$5**

## Fast In-Memory Sort Using ALL XLR8er RAM

J.F.R. "Frank" Slinkman  
4108-C Fairlake Lane  
Glen Allen, VA 23060  
(804) 273-0937  
CIS: 72411,650

Can the TRS-80 Model 4 with a MISOSYS 40 Meg Hard Drive and a 256K XLR8er board sort 8192 32-byte records in under three minutes — including disk I/O time? If your initial reaction is, "No way" — then read on.

In the course of my business, I'm involved in computerizing a large portion of the real estate records for the counties of Chesterfield and Henrico — the main suburbs of the City of Richmond.

The two counties have a total of about 130,000 individually owned residential units — about 60,000 in Henrico and 70,000 in Chesterfield — of which I'll eventually have about 60% to 70% on computer.

(No, I am definitely not entering the data myself. I've got home workers with PC's doing that, and then shooting me the data via modem.)

This calls for some fairly substantial data base management capabilities — capabilities most would consider far beyond the reach of the TRS-80 Model 4, even with all the "goodies" installed in and attached to mine.

Sure, the hard drive can easily store the required files of 50- or 60,000 128-byte records, but how do you manage them? For example, using conventional methods, it would take days to sort 50,000 32-byte sort key records, even on an ERAM-DISK!

The accompanying listing entitled

```

;      RAMSORT/ASM      02/24/89
;
; *GET  EQUATES
; LBANK$ EQU      202H
; PTR0  EQU      8000H
;
;      ORG      3000H
;
; DISKBUF DS      100H
; FCB    DM      'TEST/DAT:4',ETX,'
;
;      this shell pointer sort uses 3-byte pointers as follows:
;
;      byte 1: CBR value to bring up bank where data resides
;      byte 2: lsb of data address
;      byte 3: msb " " "
;
;      obviously, this scheme bypasses the @bank routine, and all data
;      addresses in
;      bytes 2 & 3 must be in the range x'8000' thru x'FFFF'
;
; SHELSTRT LD      HL, (COUNT)      ;p/u # of records to sort
;          LD      (STORM), HL      ;and write to M
; CYCLE    EI              ;in case sort completed
;          LD      HL, $-$          ;p/u M count
; STORM    EQU      $-2
;          SRL     H                ;divide by 2
;          RR      L
;          LD      A, H
;          OR      L                ;> 0?
;          RET     Z                ;finished if not
;          LD      (STORM), HL      ;store quotient
;          DI              ;to keep system from updating bank
;          EX      DE, HL
;          LD      HL, $-$          ;p/u # of recs
; COUNT    EQU      $-2
;          SBC     HL, DE           ;and get difference
;          LD      (STORK), HL      ; and store
;          LD      HL, 0            ;start with rec # 0 offset M
;          LD      (STORJ), HL
; AGAIN    LD      HL, $-$          ;p/u lo rec #
; STORJ    EQU      $-2
;          LD      (STORI), HL      ;and make available for back-steps
; REPEAT   LD      HL, $-$          ;lo rec # to HL
; STORI    EQU      $-2
;          LD      DE, (STORM)      ;p/u offset
;          ADD     HL, DE           ;HL is now the hi rec #
;          PUSH    HL              ;save hi rec #
;          LD      HL, (STORI)      ;p/u lo rec #
;          LD      D, H
;          LD      E, L
;          ADD     HL, HL
;          ADD     HL, DE           ;x 3 bytes per ptr
;          LD      BC, PTR0         ;p/u base of ptr table
;          ADD     HL, BC           ;HL -> low rec ptr
;          EX      (SP), HL         ;lo rec ptr addr on stack, hi rec
; # in HL
;          LD      D, H
;          LD      E, L
;          ADD     HL, HL
;          ADD     HL, DE           ;x 3 bytes per ptr
;          ADD     HL, BC           ;HL -> hi rec ptr
;          POP     DE              ;restore lo rec ptr
;          PUSH    HL              ;save pointer addresses for
; possible swap
;          PUSH    DE
;          PUSH    DE              ;save lo rec ptr addr for now
;
;          LD      B, (HL)          ;hi rec CBR value
;          INC     HL
;          LD      E, (HL)          ;hi rec lsb
;          INC     HL

```



"RAMSORT/ASM" is my solution to this problem. Using the pointer shell sort as the foundation, I should be able to sort the maximum possible number of 32-byte keys — 65535 — in about an hour and 15 minutes (estimated, as I don't have nearly that many records in a file — yet).

The concept relies on Michel Houdé's brilliant scheme to handle the HD64180's Memory Management Unit (MMU). Under his method, bank control is as simple as sending one byte to CBR (internal port 38H) and updating the system's LBANK\$.

Full details of his scheme can be found in TMQ III.ii. However, for the purpose of explaining how RAMSORT/CMD works, it is sufficient to know the CBR values for XLR8er banks 3 through 10 are 38H through 70H in increments of 8.

The best way to explain the program is to step through the listing. You may learn a few programming "tricks" along the way.

The program starts, as you might expect, at the label BEGIN. The first thing it does, after clearing the screen, is to put the HD64180 in its fastest possible operation mode, namely 0,1,80 with a 2 T-state refresh cycle instead of the normal 3T. The original settings are stored so the mode can be restored upon program exit.

Next, the sort pointers are set up at addresses 8000H - 0DFFFH in Bank 0. The pointer structure is key to understanding the operation of the sort; so I'll go into this in detail. The 3-byte pointers are constructed as follows:

Byte 1: the CBR value which will switch in the bank in which the data referenced by this pointer will reside.

Byte 2: the lsb of the data's starting address when the proper bank is switched in

Byte 3: the msb of the data address (as above)

Next, bit 0 of SFLAG\$ is set, which allows files to be opened with a Logical Record Length (LRL) different than the LRL in the directory. This is done to allow the data to be read with full sector I/

```

LD      D, (HL)      ;BDE now -> hi rec
POP     HL            ;restore low rec ptr addr
LD      C, (HL)      ;lo rec CBR value
INC     HL
LD      A, (HL)      ;lo rec lsb
INC     HL
LD      H, (HL)      ;lo rec msb
LD      L, A          ;CHL now -> lo rec
;
EXX
LD      B, 30         ;# of bytes to compare
BACK    EXX           ;put ctr in B'
OUT0    CBR, B        ;switch in hi rec's bank
LD      A, (DE)       ;p/u hi rec char
OUT0    CBR, C        ;switch in lo rec's bank
SUB     (HL)          ;subtract lo rec char
JR      C, POPEM      ;swap if lo rec > hi rec
JR      NZ, FINIS     ;no swap if lo rec < hi rec
INC     DE
INC     HL
EXX
DJNZ    BACK
JR      FINIS         ;no swap if lo rec = hi rec
POPEM   XOR     A      ;switch in Bank 0 to access
OUT0    CBR, A
pointers POP     DE      ;DE -> lo rec ptr
POP     HL            ;HL -> hi rec ptr
LD      B, 3          ;# of bytes to swap
SWAP    LD      C, (HL)
EX      DE, HL
LD      A, (HL)
LD      (HL), C
EX      DE, HL
LD      (HL), A
INC     HL
INC     DE
DJNZ    SWAP
LD      DE, (STORM)
LD      HL, (STORI)
XOR     A
SBC     HL, DE        ;is there room to step back?
LD      (STORI), HL
JR      NC, REPEAT    ;go if so
JR      EXITSRT
FINIS   XOR     A
OUT0    CBR, A        ;switch in Bank 0 to access
pointers POP     DE      ;clear the stack
POP     HL
EXITSRT LD      HL, (STORJ)
INC     HL
LD      (STORJ), HL   ;bump base
XOR     A
LD      DE, $-$
STORK   EQU     $-2
SBC     HL, DE
JP      NC, CYCLE
JP      AGAIN
;
; Routine to swap code @ x'19F6' with KEEPBNK to permit disk reads
; directly
; into RAM in banks other than Bank 0, to get around slow LDIR's in
; @ (P)EXMEM.
; First invocation replaces system code; second invocation restores
; it.
;
SWAPCOD LD      HL, WHATCOD ;make flag non-zero on 1st invoca-
tion; zero on
LD      A, -1          ; second
XOR     (HL)
LD      (HL), A

```

```

LD      HL,KEEPBNK
LD      DE,19F6H
LD      B,KPBKSIZ
SWPCD10 LD      C,(HL)
EX      DE,HL
LD      A,(HL)
LD      (HL),C
EX      DE,HL
LD      (HL),A
INC     DE
INC     HL
DJNZ    SWPCD10
RET

;
KEEPBNK PUSH    BC                ;this is replacement for code @
x'19F6'
LD      B,A                    ;function code to B
LD      A,C
LD      (0023H),A              ;put drive # in LDRV$
CALL    1A1EH                  ;get I/O routine address into IY
LD      A,20H
OR      A
CALL    1A1CH
POP     BC
RET
KPBKSIZ EQU     $-KEEPBNK
;
ERROR  EX      AF,AF'           ;save err #
LD      A,$-$
WHATCOD EQU     $-1             ;zero: sys code OK; non-zero:
KEEPBNK is in
OR      A
CALL    NZ,SWAPCOD
EX      AF,AF'
OR      0COH
LD      C,A
SVC     @ERROR
JP      EXIT
;
BEGIN  SVC     @CLS
IN0    A,DCNTL
LD      (PORT32),A
XOR     A
OUT0    DCNTL,A                ;set up for m=0, i=1
IN0    A,RCR
LD      (PORT36),A
AND     OBCH                   ;change refresh cycle from 3T to 2
OR      3                      ;set up for r=80
OUT0    RCR,A                  ;HD64180 at max speed (0,1,80 w/2T
refresh cyc)
;
LD      HL,PTR0                ;initiate pointers
LD      BC,830H                ;B counts banks, C inits CBR
values
BGN010 LD      A,8
ADD     A,C
LD      C,A                    ;next bank's CBR value
LD      DE,PTR0                ;bank starting address
BGN020 LD      (HL),C           ;write CBR val to byte 1
INC     HL
LD      (HL),E                  ;lsb to byte 2
INC     HL
LD      (HL),D                  ;msb to byte 3
INC     HL
LD      A,32                    ;this is the record size
ADD     A,E
LD      E,A
LD      A,D
ADC     A,0
LD      D,A                    ;update data address
JR      NZ,BGN020               ;go if address still <= 0FFFFH
DJNZ    BGN010                 ;else do next bank

```

O, which is much faster than reading it in one 32-byte record at a time.

The file is now opened with an LRL of 256, and the number of 32-byte records is calculated from the contents of NRN and EOF. This is simple, in this case. It would be a little more complicated if the record length wasn't an even power of two, as would be the creation of the pointers, since each record must exist entirely within a single bank.

This sample program aborts if the number of records exceeds 8192 — the maximum number of 32-byte records which can be held in 256K. In "real life," the number of records would be used to determine how many iterations are required to sort more than that number. The number of records is now passed to the sort routine.

Next Bank 3 — the first XLR8er bank — is switched in, and SWAPCOD is called. Normally, the system switches in Bank 0 during all disk I/O. This means that routines like @exmem must be used to transfer that data from some address in the base 64K to the desired location in the desired bank. Unfortunately, this is very time-consuming, since the data must be LDIR'ed at least once and possibly twice to get it where it's supposed to be.

SWAPCOD replaces the system code which switches in Bank 0 with code that doesn't. You'll note that it also manages a flag in the ERROR routine to make sure that any program exit will be made with the correct system code in place. With the unwanted switching to Bank 0 disabled, it is now possible to read directly from disk to XLR8er RAM.

Now the data is read from the hard disk into RAM. Note how FCB+4, the msb of the FCB buffer address, is manipulated and the @BANK SVC is utilized to enable reads directly to the desired location — again avoiding slow LDIR's.

After all data is read, SWAPCOD is called again to restore the correct system code. The file is now closed because, unfortunately, full-sector I/O cannot be easily used to write the sorted data back to the hard drive.

Bank 0 is now switched back in to give the system access to the pointers, and the shell sort is called.

The shell sort itself is a modified version of the one used by the DIR command with changes to handle 3-byte instead of 2-byte pointers and to manage the banks. Except for the bank handling, it's pretty standard stuff; so let's just look at the bank handling.

Notice, though, that this sort runs with the interrupts disabled. This is for speed, but a lot more speed than just cutting out the system's maintenance time. Cutting out this system maintenance keeps the system from switching in what it thinks is the "correct" bank, according to the data it maintains in LBANK\$ and the flag table. The luxury of not having to maintain LBANK\$ makes the sort about 15% faster than when LBANK\$ is maintained, since two bank switches must be made for each character compare operation, plus another bank switch for each string compare operation.

Look at the code section above BACK which is separated by two blank comment lines and starts with "LD B,(HL)."

This code puts the data pointers in the internal registers in such a way that the register combinations BDE point to the higher record to compare and CHL point to the lower record to compare, where B and C are the CBR values, and DE and HL are the data addresses.

Since all the prime internal registers are in use, the A register must be used to make the compares, the index registers are slow, and PUSHes and POPs are also slow, I decided to use alternate register B' to hold the number of bytes to be compared.

Starting at the label BACK, the following operations take place:

1. The bank holding the high record is switched in by writing B to CBR, and the first data byte is read into A.

2. The bank holding the low record is switched in by writing C to CBR. Since HL points to this record's first data byte, (HL) is subtracted from A.

```

;
SVC @FLAGS
SET 0, (IY+18) ;let file be opened with "wrong"
LRL
;
LD B,0 ;LRL = 0 allows reading of eight
32-byte recs ; at a time
LD DE,FCB
LD HL,PTRO
SVC @OPEN
JR NZ,ERROR
;
LD HL,(FCB+12) ;p/u NRN of file (sectors)
LD A,(FCB+8) ;p/u EOF
OR A
JR Z,BGN030 ;all of last sector used if EOF=0
DEC HL ; else make HL = # of full
sectors
BGN030 LD B,3 ;# of shifts to multiply x 8 (256
/ 32 = 8)
BGN040 ADD A,A
ADC HL,HL
DJNZ BGN040 ;HL = # of 32-byte records
LD DE,8192+1 ;8192 is max we can sort at one
time
PUSH HL
SBC HL,DE
POP HL
JR C,BGN050 ;OK if # recs <= 8192
LD A,33 ;'No device space available' code
(if you see JR ERROR ; this error message, disk file
too big)
;
BGN050 LD (COUNT),HL ;pass # of records to shell sort
routine
;
LD BC,3 ;bring up lowest XLR8er bank to
receive first
SVC @BANK ; disk data
CALL SWAPCOD ;stop system from switching in
Bank 0 on system ; calls
;
LD BC,(FCB+12) ;BC will count sectors in file
LD DE,FCB
LD HL,FCB+4 ;-> msb of buffer address
BGN060 SVC @READ
JP NZ,ERROR
INC (HL) ;point FCB to next RAM page
JR NZ,BGN070
LD (HL),80H ;here if bank full (i.e., buffer
addr=x'0000')
PUSH BC ;make buffer addr = x'8000'
LD B,4
SVC @BANK ;get current bank # (rets w/B=0)
INC A ;bump it
LD C,A
SVC @BANK ;switch in next higher bank
POP BC ;restore sector counter
BGN070 DEC BC
LD A,B
OR C
JR NZ,BGN060
;
CALL SWAPCOD ;restore system code
LD DE,FCB
SVC @CLOSE ;all data read - close the file
(must write JP NZ,ERROR ; data back w/LRL = 32)
LD BC,0
SVC @BANK ;switch in bank 0 for access to

```



```

pointers
;
;   CALL    SHELSRT
;
;   LD      B,32
;   LD      DE,FCB
;   LD      HL,DISKBUF
;   SVC     @OPEN           ;re-open file w/LRL=32
;   JP      NZ,ERROR
;
;   CALL    SWAPCOD         ;enable disk writes directly from
XLR8er RAM
;
;   LD      IX,PTR0
;   LD      BC,0           ;record number for @posn
;   LD      DE,FCB
;   DI
;   EXX
;   LD      BC,(COUNT)    ;p/u # of records to write
BGN080 EXX
;   LD      A,(IX)         ;p/u CBR value
;   INC     IX
;   LD      L,(IX)         ;p/u lsb
;   INC     IX
;   LD      H,(IX)         ;p/u msb
;   INC     IX
;   OUT0    CBR,A         ;switch in record's bank of
residence
;   SUB     20H           ;these 4 instructions convert CBR
values of
;   SRL     A             ; 38H thru 70H to a bank number
in the range
;   SRL     A             ; 3 thru 10
;   SRL     A
;   LD      (LBANK$),A     ;just in case the system tries to
update bank
;   SVC     @POSN         ; during time when interrupts on
;   SVC     @WRITE
;   JP      NZ,ERROR
;   DI
;   XOR     A
;   OUT0    CBR,A
;   LD      (LBANK$),A     ;switch in bank 0 for access to
pointers
;   INC     BC
;   EXX
;   DEC     BC
;   LD      A,B
;   OR      C
;   JR      NZ,BGN080
;   EI
;   ;because bank handling over with
;
;   CALL    SWAPCOD         ;restore system code
;   LD      DE,FCB
;   SVC     @CLOSE
;   JP      NZ,ERROR
;
;   EXIT    LD      A,$-$   ;job finished - restore orig
HD64180 settings
PORT32 EQU $-1
;   OUT0    DCNTL,A
;   LD      A,$-$
PORT36 EQU $-1
;   OUT0    RCR,A
;   LD      HL,0
;   SVC     @EXIT         ;and return to LS-DOS 6.3.0
;
;   END      BEGIN

```

3. Based on the states of the C and Z flags, the decision is made as to whether to swap the two records, leave them as they are, or look at the next data bytes.

The rest of the sort is pretty standard stuff.

After the sort is completed, the hard disk file is re-opened, but this time with its correct LRL of 32. SWAPCOD is called again to keep the system from switching in Bank 0 with each disk I/O operation.

The sorted data is now written back to the disk. Again, the interrupts must be disabled to avoid what would amount to double-buffering to move the data from XLR8er RAM to the disk I/O buffer. Even with single buffering (which is what, in effect, is taking place here since each 32-byte record must be LDIR'ed to the disk I/O buffer) it takes more than three times as long to write the data to the disk as it did to read it with full-sector I/O.

In this code, beginning at BGN080, note that LBANK\$ is maintained. This is done as a precaution, in the event the system tries to switch in the "correct" bank during the very brief period between the time the disk I/O code enables the interrupts and this code disables them again.

After all data is written back to disk, SWAPCOD is called to restore the system code again, and the file is closed. It is not necessary to bring in Bank 0 because the @exit to LS-DOS will automatically do that anyway.

The file is then closed, the original HD64180 operation mode is restored, and the program terminates via the @exit svc.

How fast is RAMSORT/CMD? To find out, I stopwatch-timed four test runs, as follow:

1. 107 records consisting of randomly generated alpha characters. Time: 00:02.5
2. 1006 records as above. Time: 00:14.0
3. 8192 records as above. Time: 02:20.5
4. As a more realistic test, 8192 records were constructed with a randomly gener-

ated 5-digit zip code in the first 5 positions. There were 15 zips, in the range 23201 through 23215; so the first three characters in every record were identical, and there were only 15 possible variations of the next two characters, forcing the sort to make many more character compares. The remaining bytes were randomly generated alpha characters as above. Time: 02:44.6

The above times include disk I/O. I made several runs on identical copies of the file in test (3) above, and broke the times down as follows:

1. Load the program and read all data. Time: 00:18.5

2. Sort 8192 records: Time: 00:52.8

3. Write sorted data back to disk: Time: 01:09.3

Remember, these are stopwatch times; so there is bound to be some reaction-time inaccuracy.

Not too shabby for a 6 MHz, 8-bit machine! Now I'm waiting for Richard King to perfect the 512K upgrade for the gate array version so I can sort 14,336 records at a time! That should cut the time to sort 65,535 records down to under an hour.

## Using XLR8er RAM as a GRAPHICS VIDEO RAM

J.F.R. "Frank" Slinkman  
4108-C Fairlake Lane  
Glen Allen, VA 23060  
(804) 273-0937  
CompuServe [72411,650]

One of the most powerful, yet least used features of the HD64180 processor on the XLR8er board are the two Direct Memory Access (DMA) channels.

```

GRAM/ASM
; routine to establish a disk file to function as a hi-res graphics
; board video ram as an ERAMDISK disk file, clear it, and establish
; a high-priority task to use this ram to continuously refresh the
; video display.
;
; written 04 Jan 90, J.F.R. "Frank" Slinkman
;
; this code is placed in the public domain, and all are welcome to
; use it, modify it, and/or adapt it to suit your own purposes and
; uses. However, proper credit to the original author in the pro
; gram or in the program docs would be appreciated.
;
*GET EQUATES ;SVC and HD64180 definitions
;
; ORG 3000H
;
ABORTMS DM LF,LF,'Job aborted',CR
BILBORD DM 28,31,LF,TAB+37,'GRAM',LF
DM TAB+24,'by J.F.R. Slinkman 04 Jan 90',LF,TAB+14
DM 'Establishes graphics video ram at 0xB000 in bank 10',LF
DM TAB+21,'and GRAM/IMG disk file on the ERAMDISK',LF
DM TAB+17,'and enables task slot 11 graphics refresh
task',LF,LF
DM TAB+22,'Type number of desired refresh rate:',LF,LF
DM TAB+22,'[1] 60 lines/sec (4 seconds/screen)',LF
DM TAB+22,'[2] 120 lines/sec (2 seconds/screen)',LF
DM TAB+22,'[3] 180 lines/sec (1.333 sec/screen)',LF
DM TAB+22,'[4] 240 lines/sec (1 second/screen)',LF
DM TAB+22,'[5] 300 lines/sec (0.8 sec/screen)',LF
DM TAB+22,'[6] 360 lines/sec (0.667 sec/screen)',LF
DM TAB+22,'[8] 480 lines/sec (0.500 sec/
screen)',LF,LF,CR
CRE8 DM 'create gram/img:'
DRIVNO DM '0 (lrl=80,rec=256)',CR
ERD DM 'ERD',CR
INACTMS DM 'ERAMDISK inactive!',ETX
NOBNKMS DM 'ERAMDISK does not include Bank 10!',ETX
NOERDMS DM 'ERAMDISK not installed!',ETX
NOSPACE DM 'Memory needed for GRAM/IMG already in use!',ETX
NOT2MSG DM 'ERAMDISK must be Type 2!',ETX
PERMSG DM 'Parameter error!',ETX
STRTCYL NOP
TASKMSG DM 'Task slot 11 already in use!',ETX
;
BEGIN LD HL,BILBORD
SVC @DSPLY
BGN002 SVC @KEY ;get response to menu prompt
CP '1'
JR C,BGN002 ;nfg if < '1'
CP '7'
JR Z,BGN002 ;240 won't divide by 7
CP '9'
JR NC,BGN002 ;nfg if > '8'
XOR 30H ;convert to binary
LD (RFSH010-1),A ;result is number of lines per
iteration
LD C,A ;# of lines/itr is divisor
LD E,240 ;screen height in lines is dividend
SVC @DIV8
LD (CODE+5),A ;initiate iteration counter in task code
LD (RFSH030+1),A ; and it's replacement value
LD DE,ERD
SVC @GTMOD ;is ERAMDISK installed?
JR Z,BGN010 ;go if so
LD HL,NOERDMS
ABORT SVC @DSPLY
LD HL,ABORTMS
SVC @DSPLY
SVC @EXIT
;

```

```

BGN010 LD      C,7      ;here, HL -> ERAMDISK module
BGN020 SVC      @GTDCT  ;test each drive slot in turn (7 to 0)
      LD      A,(IY)    ;p/u 1st byte of DCT
      CP      0C9H      ;drive inactive?
      JR      Z,BGN030  ;go if so
      LD      A,(IY+1)
      CP      L         ;do lsb match?
      JR      NZ,BGN030
      LD      A,(IY+2)
      CP      H         ;do msb match?
      JR      Z,BGN040  ;go if ERAMDISK DCT found
BGN030 DEC      C       ; else try next lower slot number
      JP      P,BGN020  ; while slot number >= 0
      LD      HL,INACTMS ;here if ERAMDISK inactive
      JR      ABORT
;
BGN040 LD      A,C       ;C is ERAMDISK drive slot number
      OR      30H       ;convert to ASCII
      LD      (DRIVNO),A ;put drive number in CREATE
command
      LD      A,(IY+8)
      AND     1FH       ;isolate sectors per granule
      CP      1         ;is it a type 2 drive?
      JR      Z,BGN050  ;go if so
      LD      HL,NOT2MSG
      JR      ABORT
;
BGN050 LD      BC,20AH  ;B is function code, C specifies bank 10
      SVC      @BANK     ;is bank 10 in use?
      JR      NZ,BGN060  ;go if so
      LD      HL,NOBNKMS
      JR      ABORT
;
BGN060 CP      2BH      ;was there a param error?
      JR      NZ,BGN080  ;go if not
BGN070 LD      HL,PERRMSG
      JR      ABORT
;
BGN080 LD      A,(IY+6) ;p/u highest cyl number (offset 0)
      SUB     4         ;we need five x 4K cyls to hold 19,200 bytes
      LD      (STRTCYL),A ;store result
      ADD     A,5        ;A = # of ERAMDISK cylinders
      SRL     A
      SRL     A
      SRL     A          ;div by 8 to get # of eramdisk banks
      SUB     11
      NEG
      LD      C,A
      LD      B,0
      SVC      @BANK     ;select 1st ERAMDISK bank
      LD      HL,9000H  ;-> GAT
      LD      A,(STRTCYL)
      LD      L,A        ;-> GAT entry for 5th from last track
      LD      B,5        ;count 5 tracks we need
      XOR     A          ;test value
BGN090 CP      (HL)
      JR      NZ,BGN100  ;go if track in use
      INC     HL
      DJNZ    BGN090
      JR      BGN110    ;go if all 5 top tracks available
BGN100 LD      HL,NOSPACE
      JP      ABORT
;
BGN110 SVC      @FLAGS
      LD      A,(IY)     ;preserve AFLAG$ value
      PUSH    AF
      LD      A,(STRTCYL)
      LD      (IY),A     ;force system to put file in top 5 tracks
      LD      HL,CRES
      SVC      @CMNDR    ;create GRAM/IMG file
      POP     AF
      LD      (IY),A     ;restore original AFLAG$ value

```

The chip provides two DMA channels: Channel 0 (DMAC0), which is primarily useful for transferring large blocks of memory from place to place within the machine, and Channel 1 (DMAC1), which is useful for data transfers between internal memory and external I/O ports.

The transfer rates are very impressive. DMAC0 memory-to-memory transfers can be as fast as 1 megabyte per second (i.e., 6T per byte @ 6MHz) if memory waits are set to 0. DMAC1 memory-to-I/O or I/O-to-memory transfers can be as fast as 857K per second (i.e., 7T per byte), assuming 0 memory and the one mandatory I/O wait state.

At a setting of 0,1,80, DMAC0 transfers are more than twice as fast (14/6) as LDIR and LDDR instructions, and DMAC1 transfers are more than twice as fast (15/7) as OTIR or INIR instructions.

Both commands also have the advantage that memory addresses can be up to 19 bits, which means they can address up to 512K of RAM. This means you can move memory to and from any place in either the "bottom 64K" or in XLR8er RAM to any place else in those areas of RAM, without having to do a lot of bank switching and double buffering.

For example, let's say you wanted to copy 32K of data in bank 0 to bank 10. Using only the @BANK SVC, the only way to accomplish this without resorting to an external storage device would be to copy the data in portions to addresses below 0x8000, switch banks, and copy it back up to address above 0x7FFF.

Using DMAC0, you need only write the source address (0x08000), the destination address (0x78000), the number of bytes to copy (0x8000) and one control byte to the proper internal registers, and the job is done — fast! — more than four times as fast, since the bank-switching method requires each byte to be moved twice.

As some of you know, I have developed and market some hi-res graphics programs. One of these products (since removed from the market for hardware, not software, reasons) was an XLR8er version of SLOTMOD4, which uses DMAC1 to send



animation data to the graphics board. The animation consists of spinning slot machine reels, and all takes place in a 320 pixel (40 byte) x 60 pixel area — 2400 bytes of data per animation “frame,” with a display rate of 60 animation frames per second.

The DMAC1 transfer rate is so fast that I actually had to use task slot 11 as a “brake” to keep the program from displaying the data faster than the monitor’s 60 times per second refresh rate could display it.

Then, a few weeks ago, I saw Mike Harrow’s program which loads graphics images in /HR format and sends them to the graphics board via DMAC1. I think, judging from a message he posted in the LDOS/TRSDOS SIG in CompuServe, that he took a “cue” for that program from SLOTLX.

In turn, I took a cue from his program. Why couldn’t a background task be set up to use DMAC1 to continuously refresh the graphics board from a dedicated area of RAM? Thinking it out a little further, if this dedicated area of RAM exists in XLR8er RAM, it could also be treated as an ERAMDISK disk file!

This means that any program, instead of bothering with the irritating, time consuming and sometimes complicated overhead of doing all the port twiddling required to transfer data to and from the graphics board, could simply read and write data to and from this disk file as an easy, “transparent” way of incorporating hi-res graphics into any program. Further, establishing the file with a logical record length of 80 makes each record correspond to one 80-byte line of the graphics display.

The program, GRAM/CMD, does this. It is not completely developed, by any means, but the “guts” are there for anyone who wants to incorporate this capability into their own software. GRAM/CMD is intended mainly as a demonstration of the concept.

It has a lot of restrictions. First, there must be a Type 2 ERAMDISK installed, and it must include Bank 10, the top bank. There is no reason it couldn’t be done

```

LD      A,H
OR      L
JR      Z,BGN120 ;go if no error (from create command)
SVC     @EXIT
;
BGN120  LD      BC,10          ;code to select bank 10
SVC     @BANK
JP      NZ,ERROR
LD      HL,0B000H           ;-> start of graphics image area
LD      DE,0B001H
LD      BC,19199
LD      (HL),0
LDIR    ;null out graphics ram area
LD      HL,CODE
LD      BC,CODELEN
LDIR    ;put guts of task code at top of bank 10
LD      A,0FEH
OUT     (CONTROL),A         ;put both RS and uL boards
OUT     (YREG),A            ; in normal mode
LD      A,80                ;set gfx board to 80,254 (these coordinates
OUT     (XREG),A            ; exist only on RS board)
LD      A,55H               ;graphics board test value
LD      E,A
OUT     (GFXDAT),A         ;send test value to board
LD      BC,1200
SVC     @PAUSE              ;wait 1/60th of a second
IN      A,(GFXDAT)          ;read value back
CP      E                   ;does value read = value written?
JR      NZ,BGN130           ;if not, it's a uL board
LD      A,0BBH              ;RS board code for auto inc X on
OUT     (CONTROL),A        ; write, waits on, gfx on
LD      A,1
OUT     (8EH),A            ;enable RS board text overlay
JR      BGN140
BGN130  LD      A,0B9H        ;uL board code for auto inc X on
OUT     (CONTROL),A        ; write, text on, gfx on
BGN140  LD      BC,0
SVC     @BANK               ;select bank 0
JP      NZ,ERROR
LD      C,11
SVC     @CKTSK              ;is task slot 0 free?
JR      Z,BGN150 ;go if so
LD      HL,TASKMSG
JP      ABORT
;
BGN150  LD      HL,(206H);get address of 1st available lmem byte
PUSH    HL                  ;save old value
EX      DE,HL              ;lmem ptr in DE
LD      HL,TASKLEN
ADD     HL,DE               ;possible new lmem ptr
LD      BC,1301H           ;top of lmem + 1
SBC     HL,BC               ;set CY if room in lmem
JR      NC,BGN160 ;go if task must be put in high memory
LD      HL,2                ;here if task will fit in lmem
ADD     HL,DE               ;calculate value of header word
LD      (GFXTASK),HL        ;and write it to task header
LD      HL,GFXTASK
LD      BC,TASKLEN
LDIR    ;put task in low memory
LD      (206H),DE          ;update lmem address
POP     DE                  ;old value is task address
JR      BGN180              ;go to initiate task
BGN160  EQU     $            ;here if task must go to high memory
POP     HL                  ;clear stack
LD      B,0
LD      HL,0
SVC     @HIGH$              ;get high memory pointer
JR      Z,BGN170
ERROR   OR      40H
LD      C,A
SVC     @ERROR
;

```

```

BGN170 EX    DE,HL    ;old hi-mem pointer becomes dest address
      LD    HL,ENDTASK ;-> end of task code
      LD    BC,TASKLEN
      LDDR                      ;put task in high memory
      EX    DE,HL
      SVC   @HIGH$             ;update HIGH$ (B=0 from LDDR)
      JR    NZ,ERROR
      INC   HL                  ;-> start of task
      EX    DE,HL              ;DE is now task address
      LD    HL,2
      ADD   HL,DE               ;HL -> start of actual task code
      EX    DE,HL
      LD    (HL),E
      INC   HL
      LD    (HL),D              ;write actual start to task header
      DEC   HL
      EX    DE,HL              ;and restore to DE
BGN180 LD    HL,VRAMADD-ADJUST
      LD    BC,528H
      OTIMR                      ;initiate DMAC0 source and dest addresses
      LD    C,11                ;specify task slot 11
      SVC   @ADTSK              ;initiate the task
      LD    HL,0
      SVC   @EXIT
;
GFXTASK DW    DO_IT
DO_IT   LD    HL,0FB00H ;address of actual task code in bank 10
      LD    BC,8AH           ;bank 10 with bit 7 set
      SVC   @BANK            ;this amounts to a jump to 7FB00H
ENDTASK RET                      ;difficult to error check AND turn off task
TASKLEN EQU    $-GFXTASK ; within the task, so to heck with it.
; Besides, all requirements for success already
; checked and verified.
;
ADJUST EQU    0FB00H-$ ;offset for hard addresses after code is
moved
;
CODE    PUSH   HL              ;return address
      PUSH   BC                ;previous bank info
      JR    REFRESH
;
ROW     EQU    $+ADJUST
NOP
ITCTR   EQU    $+ADJUST
DB      40                ;iteration counter (set from menu response)
VRAMADD EQU    $+ADJUST
DB      0                  ;data for MAR1L (28H) \
DB      0B0H              ;data for MAR1H (29H) > source address
DB      7                  ;data for MAR1B (2AH) /
DW      GFXDAT             ;data for IAR1 (2BH & 2CH) dest port
LENGTH EQU    $+ADJUST
DW      80                 ;data for BCR (2EH & 2FH) bytes to send
DB      80H                ;data for DSTAT (30H) DMAC0 enable
;
REFRESH IN0    E,DCNTL ;store DCNTL value
      XOR    A              ;set DCNTL for 0,1,80 and level sense (for
      OUT0   DCNTL,A ; some reason, edge sense won't work)
      LD    D,6 ;count lines of data to send (set from menu)
RFSH010 XOR    A
      OUT    (XREG),A        ;set gfx board to column 0
      LD    A,(ROW)
      OUT    (YREG),A        ; and to current row number
      INC   A
      LD    (ROW),A          ;update ROW
      LD    HL,LENGTH
      LD    BC,32EH ;3 bytes for internal ports 2EH to 30H
      OTIMR                      ;this sends one 80-byte line
      DEC   D
      JR    NZ,RFSH010      ;while line counter > 0
      LD    HL,ITCTR
      DEC   HL
      JR    Z,RFSH030        ;has entire screen been filled?
      JR    Z,RFSH030        ;go if so

```

with a Type 4 ERAMDISK—I just didn't bother to do it, since this is only a demonstration.

There is a reason, however, for not doing it on a Type 5 ERAMDISK, that being that the tracks are 4.5K, not 4K or 8K or some other factor of 32K. Thus the physical addresses of the top tracks will vary depending on how many banks are being used. This makes it difficult to correlate record numbers in the disk file to graphics board line numbers.

GRAM/CMD requires the top five tracks of the ERAMDISK to be free, as the XLR8er RAM from 0x7B000 to 0x7FFFF is used by the program for both the graphics data (0x7B000 to 0x7FAFF) and most of the program code used by the background task which continually refreshes the contents of the graphics board RAM.

Since it uses task slot 11, it requires this task slot to be free.

Because the data in the graphics RAM is treated as being in /HR file order, the graphics board must constantly be kept in an "auto increment X on write" mode. The only options you have are whether the board is set for "text only," "graphics only," or "text overlay" mode.

Since any application making use of the concept would be writing graphics data to the disk file, and not directly to be board itself, the restrictions in the above paragraph should not be overly confining. In fact, it would be pointless to write to the graphics board itself since whatever you wrote there would quickly be overwritten by the data in the graphics RAM.

Therefore, I would suggest you set up three macros or subroutines, one to establish each of the three permissible graphics board states, and rely solely on those to switch between text display, graphics display, or text and graphics overlaid.

However, if you do need to change the graphics board from the "auto inc X on write" mode, you MUST disable the interrupts FIRST, and restore the board to one of the three "legal" modes when you've finished, and BEFORE you re-enable the interrupts. Otherwise, the results will not

be very pretty to look at.

If you're writing software intended to be run on either the Radio Shack or Micro-Labs board, you should be aware of the differences between the two. Bit 1 of Port 83H is used to enable/disable video waits on the RS board, and to disable/enable text overlay on the uLabs board.

Text overlay on the RS board is accomplished by writing a value with Bit 0 set to port 8EH. This must be done after graphics display is enabled. Text overlay will remain enabled, however, even if you turn graphics display off, then on; so you must write a value with Bit 0 reset to Port 8EH to disable text overlay.

For more complete information, I recommend the text files PORTS/DOC and PORTS2/DOC, which can be found in the CompuServe TRS80PRO SIG's DL4.

The operation of GRAM/CMD is very simple. From DOS Ready, enter: GRAM.

You will be given a menu of the various available refresh rates, both in terms of both graphics board lines per second, and seconds per full-screen refresh. These range from 60 lines/second, which translates to 4 seconds per full-screen refresh, to 480 lines/second, which translates to two full-screen refreshes per second.

The faster the refresh rate, the more the system is going to be slowed down. You can figure roughly 3/4 of 1% slowdown for each 60 lines/second of refresh. Thus a 480 line/second refresh rate will result in just under a 6% slowdown.

The task module, GFXTASK, is only 12 bytes long. It will be placed in low memory if space is available, and in high memory if not. The reason it's so short is that all it does is set up the registers for a @BANK SVC jump to physical address 0x7FB00. The real work is done by the 76-byte code segment which is copied to that address.

Upon return to DOS Ready, the 19,200-byte graphics video ram will have been established as the ERAMDISK file GRAM/IMG and filled with nuls, and the high priority background task which performs

```

RFSH020 OUT0    DCNTL,E          ;restore original DCNTL value
              POP     BC          ;old bank info
              POP     HL          ;return address
              SVC     @BANK       ;return to task module
RFSH030 LD      (HL),40          ;reset iteration counter (set from menu)
              LD      HL,VRAMADD
              LD      BC,328H
              OTIMR           ;reset video ram address to start of data
              XOR     A
              LD      (ROW),A     ;reset ROW to 0
              JR      RFSH020
;
CODELEN EQU    $-CODE
;
              END      BEGIN

```

continuous graphics board refresh will be installed and functioning. The graphics board will be in auto inc X on write with text overlay mode.

To verify the task is working, simply copy any /HR file to the image file using the following syntax:

```
COPY filename/HR GRAM/IMG
(LRL=80)
```

The (LRL=80) parameter is IMPORTANT! Even though GRAM/IMG was CREATED, and is therefore never supposed to shrink, it **does** shrink if you omit the LRL=80 parameter. Also, do not copy any file longer than 75 sectors, as this will overwrite the task code and cause the system to crash.

Because interrupts are disabled during disk I/O, display cannot begin until the COPY is completed, although, occasionally, if the timing is just right, you'll see a couple of lines be displayed between sector or track reads done by the COPY function, when the interrupts are temporarily enabled.

Unfortunately, there's a chance you won't see anything. The hardware gurus tell me this is because the HD64180's DREQ1 pin is left "floating," which means DMAC1 transfers work on some machines and not on others. This is the hardware problem I mentioned which forced me to pull the XLR8er version of SLOTMOD4 off the market. The solution is to tie this pin to ground with a 150 ohm resistor.

An interesting demonstration can be put together by building a JCL file to display a number of /HR files in succession. Using

your draw-type program, create one /HR file filled with nothing but black (BLANK/HR) and another with nothing but white (WHITE/HR). Then build your JCL file as follows:

```

copy filename/hr gram/img (lrl=80)
cls
//delay 75
copy white/hr gram/img (lrl=80)
cls
//delay 10

```

repeat the first six lines for each /HR file you wish to include in the "slide show," using actual file names instead of "filename," of course:

```

copy blank/hr gram/img (lrl=80)
//exit

```

Then DO the JCL, stand back and watch.

If you have any questions, comments, suggestions, complaints or whatever, I can be contacted via CompuServe [72411,650], by phone [(804) 273-0937] or by mail at 4108-C Fairlake Lane, Glen Allen, VA 23060.



## FellSwoop PRO- WAM Export Utility

Jeff Joseph  
270 8th St #3A  
Wheeling, IL 60090

PRO-WAM's ability to export data to other programs is perhaps its most valuable feature. To be useful, however, the data must be transmitted in a form acceptable to the program receiving it. I suspect most people who make use of this feature, like myself, use it primarily for transporting text to or from a word processor. PRO-WAM's ability to terminate data with or without a carriage return makes it well suited to this use.

However, I often have need to export information to a spreadsheet. At work I keep production records in a Little Brother database, and usually need to massage the numbers (and generate new ones) using Microsoft Multiplan (MP). LB is ideal for storing and retrieving the data, but MP with its comprehensive math capabilities is ideal for manipulating it. Also, I have to submit the finished report to the suits in the front offices, and they all use PS/2s running Excel (diehard that I am, I'm keeping my trusty TRS-80!). MP is the perfect vehicle for moving data to Excel, since Excel, another Microsoft product, can directly import MP files (which are all identical whether created with a Model 4, a Mac, CP/M, or MSDOS).

The problem with performing an export to MP is that MP expects each data item to be terminated with an arrow key to move the cell pointer to the next cell. A carriage return (CR) moves MP to its main menu, and subsequent exported characters are interpreted as MP commands. This locks up the computer for ten minutes or so as MP pitches a fit at the invalid commands PRO-WAM exports to it. Also, data items beginning with numerics but containing alphabetic or spaces (such as part numbers, names, or addresses) will crash the export since MP expects a value when it sees the first numeric character.

```
; FELSWOOP/ASM - final release date 11/18/89
;
; A PRO-WAM app for preparing a text file for export to a
; spreadsheet assembled on the Model 4 using MISOSYS PRO-CREATE
; copyright (c) 1989 Jeff Joseph
; Released into the public domain for the enjoyment of all Model 4
; users! Many thanks to whoever thought of PRO-DD&T !
;
; FellSwoop creates an output file containing embedded arrow key
; codes, which when exported to a spreadsheet by TED, cause the cell
; pointer to move in a manner such that the finished spreadsheet
; accurately depicts the original database printed report format.
; The source file is assumed to be an ASCII text file created by a
; database program printing to disk. Additionally, the source file
; is screened for invalid character sequences that would crash
; the export process. These are either eliminated or are surrounded
; with quotes to force the spreadsheet to accept them as labels.
;
; ATTENTION : This source file assembles into the Multiplan version!
; Changes required for the Visicalc version are commented
;
@WKEYIN EQU      0
ETX      EQU      3
@WPOS     EQU      3
@GET      EQU      3
@PUT      EQU      4
@WCREAT EQU      7
@WCLOSE EQU      8
LFT_ARW EQU      8
RT_ARW  EQU      9
@WDSP     EQU      9
DWN_ARW EQU     10
@WDSP$    EQU     10
CR        EQU     13
SPC       EQU     32
@INIT     EQU     58
@OPEN     EQU     59
@CLOSE    EQU     60
@FSPEC    EQU     78
@HEX8     EQU     98
@SOUND    EQU    104
@WINDOW   EQU    124
SVC       MACRO    #NUM
            LD      A,#NUM
            RST     28H
            ENDM
            OPTION  CI
            ORG     2700H
            DB      'PROWAM'
            DB      'FellSwoop ',3
            DC      .HIGH$.SHL.8-$.+256,0
            IFNE    $.2800H
            ERR      'Something is wrong in the header'
            ENDIF
START      LD      B,@WCREAT
            LD      HL,0E04H ;Origin at (14,4)
            LD      DE,073BH ;to have 7 rows 59 cols
            SVC     @WINDOW
            JR      Z,OK
            LD      B,84H          ; if not OK
            SVC     @SOUND          ; then BEEP and exit
            RET
OK          LD      HL,OPNMSG        ; FellSwoop banner
            LD      B,@WDSP$
            SVC     @WINDOW
            LD      B,@WDSP$        ; display border
            LD      HL,BORDER
            SVC     @WINDOW
SRCPMT     LD      B,@WPOS
            LD      HL,0201H        ; source prompt at (2,1)
            SVC     @WINDOW
            LD      HL,SRCMSG
```

So, I have been forced to print my LB data to the printer and enter the data into MP by hand. Needless to say, this was tedious and error prone, and went on for some time until I decided there had to be a better way. Some peeking around inside PRO-WAM with MED proved fruitful when I discovered that the CR (x'0D') added during an export resides at memory address x'FDCA' (assuming PRO-WAM was loaded at the top of memory). If this byte is zapped to an x'0A', PRO-WAM will terminate each line of exported data with a down-arrow key code. This allowed me to print my LB report to a disk file, enter MP, invoke Mr. Ed's TED application, and export a column. The x'0A' moves the cell pointer down to the next cell and then PRO-WAM exports the next data item.

While this is certainly better than spoon-feeding the data to MP, it still has serious drawbacks. Numeric columns must be padded with leading zeros so all items are the same width, because MP doesn't like spaces preceding numerics. Items beginning with numerics and containing alphabets or spaces must be surrounded with quotes to be accepted as labels. This means a lot of tedious editing is necessary before it will export properly. Only one column at a time can be exported since MP will choke on the spaces between columns, and right- and left-arrow key codes are needed to move the cell pointer between columns. If only there were a way to do it all in one fell swoop...

After a little thought I figured a way to do it, and several gallons of Pepsi later my idea became the program FellSwoop. The solution is to take the text file created by the database (printed to disk) and build a new text file with embedded arrow-key control codes to move the cell pointer as required. The data can also be screened for invalid character sequences that would crash the export process. The invalid data is surrounded with quotes so the spreadsheet accepts them as labels. For the ultimate in convenience, the program is coded as a PRO-WAM application, FELSWOOP/APP. I have placed this program into the public domain.

FellSwoop opens a window near the lower left corner of the Model 4 screen and

```

LD      B,@WDSP$      ; display source prompt
SVC     @WINDOW
LD      B,@WDSP$      ; display target prompt
LD      HL,TARMSG
SVC     @WINDOW
LD      B,@WPOS
LD      HL,0213H      ; (2,19)
SVC     @WINDOW      ; pos cursor back to source
LD      HL,SRCPIL ; point at buffer for source filename
LD      B,@WKEYIN
LD      C,22          ; KEYIN 22 characters maximum
SVC     @WINDOW
JR      NC,TSTSRC     ; go if no BREAK or EXPORT
APPEXIT LD      B,@WCLOSE ; else close window and return
SVC     @WINDOW
RET
TSTSRC  LD      DE,FCBSRC ; point to FCB buffer for source
SVC     @FSPEC        ; test source filespec
JR      Z,TARPMT      ; get target filespec if OK
CALL    BADFILE ; else display bad filespec message
JR      SRCPMT        ; and try again
TARPMT  LD      B,@WPOS
LD      HL,0313H      ; pos cursor to (3,19)
SVC     @WINDOW      ; for target file input
LD      HL,TARFIL
LD      B,@WKEYIN
LD      C,22          ; 22 characters maximum
SVC     @WINDOW
JR      C,APPEXIT     ; if <BREAK> then quit
TSTTAR  LD      DE,FCBTAR ; else test target filespec
SVC     @FSPEC
JR      Z,OPNSRC ; if target filespec OK then get started
CALL    BADFILE ; else display bad filespec message
JR      SRCPMT        ; and try again
OPNSRC  LD      B,@WPOS
LD      HL,0501H
SVC     @WINDOW      ; display working message
LD      B,@WDSP$
LD      HL,WORKMSG
SVC     @WINDOW
EX      AF,AF'
PUSH    AF' ; preserve A' - we will be using it
XOR     A ; and then zero it
EX      AF,AF'
LD      DE,FCBSRC ; now open source file
LD      HL,BUFR1
LD      B,0 ; use LRL 256
SVC     @OPEN
JP      NZ,ERROR
LD      DE,FCBTAR
LD      HL,BUFR2
SVC     @INIT
JP      NZ,ERROR

```

```

;
; this is the start of the main body of the program, at this point
; both source and target files are opened. The main body consists
; of two major sections. The first section, starting here, reads the
; source file one line at a time and loads it into LINBUFR.
; Control is passed to the second section when the end of a line
; (a CR) is reached. As the line is read to memory, consecutive
; spaces are compressed and replaced with right arrow codes, which
; the second section will use to delimit cells. Consecutive carriage
; returns (blank lines in the source file) are also
; compressed and replaced with down arrow codes to
; conserve screen lines in the target file.

```

```

; Register usage in this section:

```

```

; DE -> points to FCB for source file
; HL -> points to line buffer at current character position
; C -> tracks the number of characters read into the buffer
; B -> used as flag register

```



```

;      Bit 0 = previous character was a space
;      Bit 1 = character before last was a space
;      Bit 7 = pointer is at Beginning Of Line
;
NEXTLIN LD      DE,FCBSRC      ; point to source file and
LD          HL,LINBUFR        ; start of line buffer
LD          BC,8101H ; set BOL and space flags, clear char
counter
GETLINE SVC     @GET          ; get character from source file
JP          NZ,ERROR
CP          CR                ; have we reached the end of the line ?
JR          Z,SCREEN ; yes, begin the data screening process
CP          SPC                ; was character a space ?
JR          NZ,CHRBUFFR ; no, continue getting characters
BIT        0,B                ; yes, was previous char also a space ?
JR          NZ,SETSPC2        ; yes, set 2nd space flag (don't
put in buffer)
SET        0,B                ; no, set the first space flag
LD          (HL),A            ; put the first space into the buffer
INC        HL
INC        C
JR          GETLINE
CHRBUFFR BIT    1,B            ; is second space flag set ?
JR          Z,SKPARW ; if not don't insert the right arrow
BIT        7,B                ; if at BOL also don't insert right arw
JR          NZ,SKPARW
DEC        HL                ; put a right arrow into
LD          (HL),RT_ARW        ; the buffer over space
INC        HL                ; and then the character
SKPARW LD       (HL),A          ; into the buffer
INC        HL
INC        C
LD          B,0                ; clear flags
JR          GETLINE ; and get next character
SETSPC2 SET     1,B            ; if more than one space set 2nd space flag
JR          GETLINE ; but don't put the space into the buffer
SCREEN BIT     7,B            ; is <CR> at BOL ?
JR          Z,TERMLIN ; if not terminate line and begin
screening
LD          (HL),DWN_ARW        ; put a down arrow in the buffer
INC        HL
INC        C                ; bump line counters up
JR          GETLINE ; go back and continue getting the
(next) line
TERMLIN DEC     HL            ; look at char just before the end of line
LD          A,(HL)
CP          SPC
JR          NZ,WRTCR ; if a space put the <CR> over it,
DEC        C                ; otherwise a value at end of line
LD          (HL),CR ; would become a label
WRTCR INC       HL
LD          (HL),CR ; put the <CR> into buffer
LD          HL,LINBUFR ; go back to beginning of buffer
LD          DE,0 ; zero screen counters
LD          B,1 ; set BOC flag
;
; This is the start of the second major portion of the program.
; We are done reading a line into the buffer and are going to read
; the line buffer one cell at a time, identifying cells that
; wouldn't export properly as values and inserting quotes so the
; spreadsheet will handle them as labels instead. Also, dollar
; signs and commas preceding numeric values will be eliminated.
;
; Register usage in this section:
;
; HL -> pointer to current character position in line buffer
; D -> current pointer position relative to start of line
; E -> current pointer position relative to start of cell
; (D and E are needed to reposition HL back to Beginning Of Cell)
; C -> still keeps number of characters in the line buffer
; B -> used as flag register
; Bit 0 = indicates the line pointer is at a Beginning Of Cell

```

prompts for source and target filespecs. The source file is presumed to be an ASCII text file created by a database program printing to disk, although you might find other kinds of text files suitable or desirable for export to a spreadsheet. The program begins after both filenames are entered and validated. The program supplies no defaults. When the conversion is complete an appropriate message is displayed and you may perform another. If an error is encountered the program supplies the DOS error number and you may begin again. On a stock Model 4 running on floppies the program might take a minute or two to convert a large file.

FellSwoop loads the source file into a buffer one line at a time. Characters with single spaces between them are entered as one cell (e.g. Mr. Smith). Characters separated by two or more spaces (such as the gutters between columns of a database report) are entered as two separate cells, with an embedded right-arrow key code between them to move the MP cell pointer between cells. Then the line is scanned for invalid character sequences. If found, the buffer pointer is backed up to the beginning of the cell and a quote is inserted. At the end of cell a trailing quote is inserted. This trailing quote is the only difference between the Multiplan and Visicalc versions of this program. Visicalc doesn't require a trailing quote, and if entered it becomes part of the label. Visicalc users should eliminate the two routines that add the trailing quote (as commented in the source code). After the entire line has been validated, it is padded out with spaces to put the CRs in video column 79 of the Model 4 display. This is necessary because the CRs are not to be exported with the data, and to allow for lines of differing lengths. A down arrow code is then added and then an appropriate number of left arrow codes for moving the MP cell pointer back to the beginning column. Finally, the line is written to the target file and the process repeats. Blank lines in the source file (CR only) are compressed to conserve screen space in the export file, maximizing the amount of data that can be exported at one time. The compression is achieved by inserting down arrow codes at the beginning of the first line found to contain data following the



blank lines.

Before the export is performed, be sure to activate MP's Alpha or Value command (depending on whether the first item to export is a label or value). Then invoke TED and perform the export. The export file may appear strange at first, but the arrow-key codes and quotes won't appear in the finished spreadsheet. The beginning block marker must be in the north-west screen corner, and the ending block marker must be in column 78 of the south-east screen corner so all the arrow codes along the right edge of the screen are exported. The CRs in column 79 must not be exported, and the export request must be terminated with <SHIFT><ENTER>. The data will be placed starting where you last left the cell pointer. The cell pointer will magically move by itself from cell to cell, replicating the report format just as the database printed it.

FellSwoop supports reports up to 240 characters across. Lines longer than 80 characters will occupy two or three video lines. I've found that such long lines won't export properly, however. The MP cell pointer always moves five columns short of the starting column, even though I've verified that FellSwoop is inserting the correct number of left-arrow codes. I can't understand why this is. I've even tried putting in extra left-arrows, but it is still always five columns short. For best results I recommend reports no wider (after processing by FellSwoop) than 80 characters.

The program is quite adept at identifying what should be exported as a value and what should be exported as a label. An item like -5070 is exported as a value, but 50-70 is exported as a label. Dates or part numbers in the form 11/16/89 or 11-16-89 become labels. 6.2 is handled as a value but something like 6.02.01 as a label. A number with a leading zero like 02789 is treated like a label, but a zeros alone or zeros with a decimal are entered as the value zero. Numbers preceded with dollar signs like \$7000 are entered as values; the dollar sign is eliminated. Commas preceding numbers such as 7,000 are handled in a like manner. (some databases like PFS:File cannot suppress the insertion of commas into numeric values) FellSwoop

```

; Bit 1 = indicates this cell begins with a numeric character
; Bit 2 = indicates cell crosses video column 79 and that a
;         space has been inserted to keep export chars out of
;         that column
; Bit 3 = indicates this cell contains a decimal point (or
period)
; Bit 4 = indicates first char in cell is a zero. If any chars
;         follow in the cell EXCEPT for zeros or a decimal
;         point, the cell is treated as a label. Examples:
;         02789-> "02789", 06:30-> "06:30", 0.00-> 0.00,
;         $00.00-> 00.00 precede any non-zero numeric with a
;         dollar sign to be handled correctly as a value,
;         or suppress leading zero in report
;         0.79 -> "0.79" but $0.79 -> 0.79 , .09 -> .09
; Bit 5 = indicates this cell is being handled as a label
;         which begins with a numeric character, and thus has
;         to begin and end with quotes
;         (for Visicalc, only must begin with quote)
; Bit 7 = indicates previous char was comma or dollar sign. If
;         next char is a numeric, the comma/dollar sign is covered up.
; A' -> keeps track of the number of right arrows in the line
buffer
RDBUFR LD      A,79      ; buffer positions 79 and 159 must be blank
CP      D              ; or the characters occupying those posi-
tions
JR      NZ,COL159 ; will not export because they will be in
same
WAS159 PUSH    BC        ; video column as the <CR>s.
CALL    MVRIGHT ; open buffer up by one char (must keep
flags
POP      BC          ; in case label handler bit was in effect)
INC      C
LD      (HL),SPC ; and put a space in video column 79
INC      HL
INC      D          ; (must maintain line counters)
INC      E
BIT      2,B        ; 1st or 2nd vidshift for this cell? (2nd
JR      Z,VIDSHFT ; vidshift occurs if label handled after
1st)
INC      HL          ; 2nd vidshift in effect; must eliminate
space
INC      HL          ; HL points to char after 2nd vidshift
space
INC      D
INC      D
CALL    MOVLEFT ; close up the space from the first vid
shift
VIDSHFT SET     2,B        ; indicate video shift in effect for this
cell
LOOKBFR LD      A,(HL) ; here we begin examining the line buffer
CP      RT ARW ; have we encountered the end of cell ?
JR      Z,ENDCELL ; yes, process next cell
CP      CR        ; we're done reading cells, put the stuff
at
JP      Z,ENDLINE ; end of the line now (spaces and left
arws)
CP      DWN_ARW ; if line is preceded by bunch of <CR>s
(blank
JR      Z,NXTCELL ; lines), keep reading until we see cells
CP      '0' ; by now we have a character of some sort
JP      M,H002F ; char is in range 00 - 2F
CP      ' '
JP      P,H3AFF ; char is in range 3A - FF
BIT      7,B ; does numeric follow comma or dollar sign?
CALL    NZ,MOVLEFT ; if so then bury it
CP      '0'
JR      Z,ZERO
BIT      4,B ; does a zero start the cell ?
JR      Z,SETNUM ; if not then process as normal
CALL    HLTBOC ; if it does then handle cell as a
label even

```

```

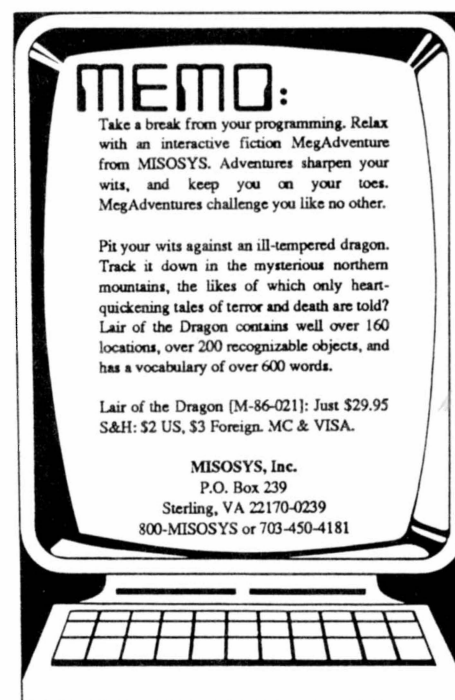
JR      RDNEXT      ; if following chars are numeric
SETNUM BIT 0,B      ; is numeric char at BOC ?
JR      Z,RDNEXT ; if not continue reading buffer
SETNUM1 SET 1,B      ; if numeric @ BOC set numeric-BOC flag
RDNEXT INC E        ;
RES     0,B        ; not at beginning of cell anymore
NXTCELL INC HL      ; point to next char
INC     D          ; inc cell character counter
JR      RDBUFR
COL159 LD A,159
CP      D          ; pointer was not at video column 79, is it
JR      Z,WAS159 ; at buffer pos 159 (also vid col 79)
JR      LOOKBFR ; if not examine the char at this position
H002F CP ','
JR      Z,COMADLR
CP      '$'      ; if comma or dollar set the comadlr flag
JR      Z,COMADLR
BIT     0,B        ; if not at BOC and numeric at BOC
JR      Z,INVALID ; invalid char triggers HLTBOC/MVRIGHT
CP      ','
JR      Z,SETNUM1 ; if at BOC then char must be tested
CP      '('      ; for . + ( - which trigger value input
JR      Z,SETNUM1 ; so numeric-BOC flag must be set
CP      '+'
JR      Z,SETNUM1
CP      '-'
JR      Z,SETNUM1
INVALID BIT 1,B      ; if first char in cell not numeric go
JR      Z,RDNEXT
CP      '.'      ; decimal point not invalid even if <> BOC
JR      Z,DECIMAL
CALL    HLTBOC ; all other chars invalid as a value, so
JR      RDNEXT ; open up buffer and insert quote
H3AFF CP '='
JR      Z,SETNUM ; = at BOC also triggers value input
JR      INVALID ; all other chars in this range are invalid
ZERO BIT 5,B        ; if cell already handled as a label
JR      NZ,RDNEXT ; skip the zero handling routine
BIT     0,B        ; does the zero start the cell ?
JR      Z,RDNEXT ; if not then go to the next character
SET     4,B        ; but if it does set zero flag
SET     1,B        ; set numeric flag in case alpha chars
JR      RDNEXT ; follow and then continue
COMADLR SET 7,B
JR      RDNEXT
ENDCELL LD E,0      ; clear cell char counter
EX      AF,AF'
INC     A          ; inc arrow counter
EX      AF,AF'
BIT     5,B        ; has numeric cell been handled as a label?
JR      NZ,GORIGHT ; if so insert ending quote (for M/plan)
;
; Note: the GORIGHT routine is necessary for Multiplan to insert the
; closing quote after a 'forced' label. Visicalc users may eliminate
; all the code under the label GORIGHT, and the above two instruc
; tions. The Visicalc version should on down to CELAFMV after the
; last EX AF,AF'. FELSWOOP/VC will be a few bytes shorter, run a
; tad faster, and each line in the export text file will usually
; have a few more characters available for data to export.
; Also don't forget to remove ENDLINE!
;
CELAFMV LD B,1      ; clear flags but set BOC for next
JR      NXTCELL ; cell and read next cell
DECIMAL BIT 5,B      ; if cell already handled
JR      NZ,RDNEXT ; as label keep going
BIT     3,B        ; if flag reset - this is
JR      Z,FRSTDEC ; first decimal in cell
CALL    HLTBOC ; second decimal in cell,
JR      RDNEXT ; so handle as a label
FRSTDEC SET 3,B      ; if first decimal,
JR      RDNEXT ; set flag and keep going
ENDLINE BIT 5,B

```

will even export an item like \$7,000,000 as the value 7000000. If you have non-zero numerics that you want entered as values, use a dollar sign to make FellSwoop properly evaluate them: 0.76 becomes "0.76", but \$0.76 becomes 0.76. 95% is a label. Most databases print lines of dashes across their reports; FellSwoop treats such lines as one label since spaces delimit cells. After the export it's a small matter to copy the dashes across the spreadsheet. (MP will issue the error message 'Line too long' while exporting dashes, but the export will continue)

FellSwoop can correctly handle nearly anything you might throw at it. There is one case where editing is necessary. Every field in the database report MUST have something printed in it, or all fields following it will be shifted one cell left of where they should be. It's impossible to discriminate between a column gutter and a blank field. As a fix, you can insert dummy values into the source file before FellSwoop processes it. The dummy values have to be blanked out of the finished spreadsheet. The best bet is to ensure all database fields have SOMETHING printed in them. FellSwoop can handle everything else.

I hope someone else finds this program as useful as I have.





# WANTED

## Scripts or SuperScripts Users ...

Now you can use that fabulous word processor from Radio Shack with your non-Tandy printer! No point in missing out! Your printer CAN work great with SuperScripts+!

## PowerDRIVER

These printers are fully supported

EPSON MX-80/100 Series  
EPSON FX/RX 80/100 Series  
Most all EPSON "compatibles."  
C.Itoh 8510 Prowriter  
C.Itoh Starwriter Daisy Wheel  
C.Itoh A10-20 Daisy Wheel  
Okidata 92/93 Dot Matrix

Model 4 Now Supported!

All features of SuperScripts are supported to the fullest capabilities of the printer involved. Drivers are easy to install, easy to use, and no extra commands to learn!

**PowerDRIVER** Only \$17.95 each  
PLEASE SPECIFY PRINTER TYPE

## PowerSCRIPT

A HIGH-POWER modification  
for Scripts from Radio Shack.

Give Scripts Full Power! PowerSCRIPT is a modification for the original Scripts/LC which allows it to work on Model I or III and gives it the power of many of the newer expensive word processors. Define printer codes for ANY printer. Embed printer codes in the middle of a line! Alphabetized directories! User definable printer filters! User-definable HELP-FILE! Chain files together at print time from any part of the text. Logical EXIT to DOS, optional automatic linefeed after C/R, FETCH, CHAIN, and KILL command and more! Compatible with most major DOS's. Simply use the enclosed INSTALL program, and you will be up and running in NO TIME! Includes full documentation. Get lots of new features for a very small price!

**PowerSCRIPT** Only \$24.95 each

US/Canada Please add \$3 Shipping/Handling Foreign \$10

**POWER SOFT**  
PRODUCTS FROM MISOSYS, INC.

MISOSYS, Inc

P. O. Box 239

Sterling, VA 22170-0239

703-450-4181

```

JR      Z, SPACES ; if last cell not handled as label don't
quote   LD      (HL), '"' ; insert 2nd quote for last cell on the
line    INC      HL
        INC      C

; Note: Visicalc users should delete the above 5 instructions for
the same
; reason as documented under ENDCELL. Here you must change
the label
; SPACES to ENDLINE so that when the <CR> is detected the
code jumps
; directly to the EX AF,AF' instruction below.
;
SPACES  EX      AF,AF'
        PUSH     AF ; save A' (contains number of right arrows)
        ADD      A,C ; line chars + left arrows = total
        LD       C,A ; line length into C
        LD       A,79
        SUB      C ; number of spaces needed now in A'
JR      C,LINES2 ; if carry we need more than one video line
CMPSPCS LD      B,A ; spaces into B for loop counter
        ADD      A,C ; spaces + chars + arrows
        LD       C,A ; = final line length (79, 159, or 239)
WRTSPCS LD      (HL),SPC
        INC      HL ; put spaces into the line buffer to pad
        DJNZ     WRTSPCS ; out the video line
        LD       (HL),DWN_ARW ; then a down arrow
        POP      AF ; number of left arrows to put in buffer
JR      Z,PUTLINE ; if zero don't write any left arrows
        LD       B,A ; put into B for loop counter
LFTARW  INC      HL
        LD       (HL),LFT_ARW ; left arrow codes into line buffer
        DJNZ     LFTARW
PUTLINE XOR     A ; zero arrow counter for next line
        EX      AF,AF' ; and put it back
        LD       B,C ; total char count into B for loop counter
        LD       DE,FCBTAR ; prepare to @PUT line buffer
        LD       HL,LINBUFR ; to target file
PUTCHAR LD      C,(HL)
        SVC      @PUT ; write line to target file
JR      NZ,ERROR
        INC      HL
        DJNZ     PUTCHAR
        LD       C,CR ; terminate line with a <CR>
        SVC      @PUT
JR      NZ,ERROR
JP      NEXTLIN ; and process next line in source file
GORIGHT CALL    MVRIGHT
        INC      HL ; inc HL twice to get past arrow
        INC      D
        JR      CELAFMV ; and go get the next char
LINES2  LD       A,159
        SUB      C
        JP      M,LINES3 ; if negative, line takes up 3 video lines
        CMPSPCS
LINES3  LD       A,239
        SUB      C
        CMPSPCS
ERROR   LD       C,A ; save error code in C for @HEX8 processing
        LD       DE,FCBSRC
        SVC      @CLOSE ; close source file
        LD       DE,FCBTAR
        SVC      @CLOSE ; close target file
        EX      AF,AF'
        POP      AF ; restore A' to original condition
        EX      AF,AF'
        LD       A,C ; error code into A
        CP       1CH ; did we reach end of source file?
        JR      NZ,FATAL ; any other code means an error occurred
        LD       B,@WPOS

```



```

LD      HL,0501H
SVC     @WINDOW          ; display success msg at (5,1)
LD      B,@WDSP$
LD      HL,PRPMSG
SVC     @WINDOW
CALL    RESUME           ; display resume prompt and wait
JP      SRCPMT           ; go back to source prompt
FATAL   LD      HL,ERRCODE ; point to buffer for @HEX8
SVC     @HEX8           ; convert error code to ASCII for display
LD      B,@WPOS
LD      HL,0501H ; display error msg @ (5,1)
SVC     @WINDOW
LD      B,@WDSP$
LD      HL,ERR1MSG ; display first half of DOS error message
SVC     @WINDOW
LD      B,@WDSP
LD      HL,ERRCODE
LD      C,(HL)
SVC     @WINDOW
LD      B,@WDSP          ; then the DOS error code
INC     HL
LD      C,(HL)
SVC     @WINDOW
LD      B,@WDSP$
LD      HL,ERR2MSG ; and the last half of error message
SVC     @WINDOW
CALL    RESUME           ; display prompt to resume and wait
JP      SRCPMT
HLTOBOC LD      B,E ; if char invalid and cell starts with numeric
GOBACK  DEC     D
DEC     E
DEC     HL
DJNZ    GOBACK          ; pos HL back to BOC (HL - E = BOC)
MVRIGHT PUSH    BC
PUSH    HL              ; save counters and pointers
PUSH    DE
LD      A,C
SUB     D
LD      HL,LINBUFR
LD      B,0
ADD     HL,BC          ; Destination = LINBUFR + line length
LD      E,L
LD      D,H
DEC     HL              ; HL one less than destination
LD      C,A              ; Byte Counter = C - D
LDDR    ; (line length - current line position)
POP     DE
POP     HL              ; restore counters and pointers
POP     BC
LD      (HL),' ' ; insert quote and inc line char counter
INC     C              ; (buffer extended by 1)
SET     5,B            ; set label handler flag
RES     1,B            ; reset numeric flag so invalid chars won't
RET     ; trigger HLTOBOC/MVRIGHT again for cell
MOVLEFT PUSH    HL
PUSH    DE              ; this routine moves everything left by one
PUSH    BC              ; to cover up the dollar sign or comma or
LD      A,C              ; to cover up the second space left after a
SUB     D              ; video shift occurs twice in a label cell
LD      C,A
LD      B,0              ; Byte Counter set up (C-D = bytes to move)
LD      D,H              ; (line length - current pointer position)
LD      E,L              ; HL already points to source
DEC     DE              ; Destination one less than source
LDIR
POP     BC
POP     DE
POP     HL              ; restore counters and pointers
RES     7,B              ; then reset the comma/dollar flag
DEC     HL
DEC     D
DEC     C              ; we shrunk buffer by one

```

## Using Double Duty with PRO-WAM

Here's a little hint for you folks with extended memory - how to use Double Duty and PRO-WAM together, especially when running with a hard drive.

First, you can't use the XLR8er RAM-DISK - but then, its not needed that much when running from a fast hard drive as the system disk. In low memory is the standard XLR8er Michel Houdé patched system for bank control, then the hard disk driver, then PRO-WAM is installed before DoubleDuty. That's necessary because Double Duty installs about five distinct modules of code - each not large but separately relocated. A few have to fit in low memory, not all of them. Actually, DDUTY loads the most amount of code into one of the extra banks of memory.

When you load PROWAM, specify a bank other than 1 or 2. Use something like PRO-WAM (BANK=3). This keeps PRO-WAM from loading into bank 1 which is needed for DDUTY. Then install DDUTY. It will put about two modules into low memory and the rest into high memory. But it will then need no extra high memory reservation for the library partition since it can overlap that with the PRO-WAM memory module in high memory.

With this arrangement of PRO-WAM and DoubleDuty, you can pop up a PRO-WAM window into either partition 1 or 2. You can't export across partitions directly, but with the DOLOAD and DOSAVE APPs, you can! Just pop up DOSAVE from partition 2, for instance; save the screen; exit PRO-WAM; switch to partition 1; pop up DOLOAD; reload the screen; then export. All kinds of possibilities open up such as exporting a multiplan screen into scripsit while both are active!

## HPB VECTOR CO.



**CHROMALUX:** Are full spectrum light bulbs developed at the request of the Finish government for those people who must work in the north, where there is nearly total darkness half the year. Chromalux light bulbs contain a rare earth material known as neodymium. Scientific evidence confirms:

Improved response with people inflicted SAD - Seasonal Affective Disorder, typical symptoms include: Depression, Fatigue, Lack of motivation, and irritability. Clearer vision, Better contrast, and Productivity increases in work tasks, Decor enhanced by true colors, i.e. more eye-appealing true color contrast.

**Sizes:** 100, 150W, 120V, Frosted and Clear.

<b>Qts:</b> 1-5	\$19.95 each
6-25	\$12.95 each
26-50	\$10.95 each
51-up	call

Other sizes available, including tubes

Postage and handling included in USA only.

Call/Send check or money order to:

**HPB VECTOR CO WSO**  
607 DUNSMUIR AVE., #104  
LOS ANGELES, CA 90036  
213-938-0498

**ALLOW TWO WEEKS  
FOR DELIVERY**

```

DEC      E
RET
BADFILE LD      B,@WPOS
LD      HL,0501H ; put bad filespec msg at (6,1)
SVC     @WINDOW
LD      B,@WDSP$
LD      HL,BADMSG
SVC     @WINDOW ; display it
RESUME  LD      B,@WDSP$
LD      HL,RSMMSG
SVC     @WINDOW ; display the resumption message
LD      B,@KEYIN
LD      C,0 ; single-key input request
SVC     @WINDOW ; wait for input
JR      NC,RESCNT ; cont if no <BREAK> or <EXPORT>
LD      B,@WCLOSE ; else close window
SVC     @WINDOW
POP      AF ; uncover real return address frm stack
RET      ; and exit from PRO-WAM

RESCNT LD      B,@WPOS
LD      HL,0501H ; continuation - clear msgs and ret
SVC     @WINDOW
LD      B,@WDSP$ ; position back up to first message
LD      HL,NULL$ ; erase it
SVC     @WINDOW
LD      B,@WDSP$
SVC     @WINDOW ; and erase the second message line
RET      ; and go back to subroutine caller

OPNMSG  DEFM    'Fellswoop - Prepare text file for export to
spreadsheet',CR
BORDER  DEFM    '

SRCMSG  DEFM    'Source filespec: ',CR
TARMSG  DEFM    'Target filespec: ',CR
BADMSG  DEFM    'Invalid file specification!',CR
PRPMSG  DEFM    'Target file prepared for export.',CR
ESMMSG  DEFM    'Any key to resume or <BREAK> to exit.',ETX
ERR1MSG DEFM    'DOS error x',27H,ETX
ERR2MSG DEFM    '27H,' encountered! Consult DOS manual page A-69',CR
WORKMSG DEFM    'Working...',ETX
NULL$   DEFM    '
',ETX
FCBTAR  DEFS    32
FCBSRC  DEFS    32
SRCFIL  DEFS    22
TARFIL  DEFS    22
BUFFR1  DEFS    256
BUFFR2  DEFS    256
LINBUFR DEFS    256
ERRCODE DEFS    2
IF      $.GT.3000H
ERR      'MEMORY OVERFLOW !!!
ENDIF
IFLT     $,3000H
DC      .HIGH$.SHL.8-$(+256,0
ENDIF
END      START

```



# MISOSYS Hardware and Software

## XLR8er™

### Memory expansion & Speedup

- Faster: Hitachi 64180; up to 8 MHz Z80 equivalent
  - Expanded RAM - 256KB additional memory
  - Software - LS-DOS 6.x, LDOS 5.x, or CP/M
  - Simple plug-in installation • **New lower price!**
- XLR8er 256K Ram [R-MB-004] **\$182+\$5S&H**  
Specify Computer Model & Number (26-) and if Hires graphics board installed.

### Floppy Drives and Accessories

5.25" 360K 1/2-height	\$75+\$4 S&H
3.5" 720K in 5.25" frame	\$85+\$3 S&H
3.5" 1.44M in 5.25" frame (PC-AT)	\$99+\$3 S&H

Our Model 2SV5 dual half-height vertical external floppy disk drive enclosure will hold two 5.25" half-height disk drives or one standard height drive.

2SV5 drive case/PS	\$60+\$5 S&H
8" Dual floppy extender cable	\$15
four-foot connecting cable	\$10

Note: S&H prices are UPS ground to continental U.S.

### Hard Drive Pre-assembled Kits

Our kits come assembled in a cabinet holding up to two half-height drives, 60 watt p/s, fan, SCSI controller, and host adaptor. Software includes: driver, formatter, archive, restore, and sub-partitioning - all for one DOS: LS-DOS or LDOS. Give us a call to discuss building one for you! Individual parts available.

20 Megabyte kit: M3 or M4	\$495+S&H
40 Megabyte kit: M3 or M4	\$645+S&H
Hardware clock option	\$30
Joystick option with joystick	\$20
Host interface cable	\$20
Additional software interface	\$30

### Miscellaneous parts

Ribbon cables: We custom manufacture and test using Cirris Systems cable tester: DB25, Printer 36, edgcard 34 & 50, header 34, etc. Call/write for pricing with your specifications.

DRAM: 64K-150ns (\$2/chip); 256K-150ns (\$4/chip); U72 PAL (\$8); PAL/PLD/PROM programming (call/write)

Kel-Am connectors: 34-pin male (\$8); 34-pin female (\$5) [pair needed for 4P external floppy mod: see 12/87 80MICRO]

## Double Duty

DoubleDuty, published previously by Radio Shack (cat 26-2231), is now available from MISOSYS. DoubleDuty divides your 128K TRS-80 Model 4 computer's memory into three complete and independent partitions. Two partitions each operate as if they were their own 64K Model 4. The third can be used to execute DOS library commands. If you thought you needed another computer, think again. With DoubleDuty, you can now have two for the price of one! **Just \$49.95 (+\$2S&H)!**

**MISOSYS, Inc**

**P. O. Box 239**

**Sterling, VA 22170-0239**

**703-450-4181 or 800-MISOSYS**

## PRO-WAM™ Version 2

### Window & Application Manager

Our applications turn your 128K Model 4 into a sophisticated business or personal machine rivaling the best of them. Because easily installed PRO-WAM comes with many useful and powerful menu-driven time savers and work organizers. PRO-WAM is accessed with a single key-stroke; its export and import functions allow you to move data across windows between programs.

- Address CARDS, LABELS, and HEAD display & export
- BRINGUP tickler file; new PRINTING and sorting
- CALENDAR flags BRINGUP items visually on screen
- Ten 3 x 5 CARD files with FORMS and FIELDS
- Virtual PHRASE access for export
- New TODO list manager with "who does it"
- Plus many other vital applications!

PRO-WAM [M-51-025]

**\$74.95 + \$5S&H**

## LB Data Manager

### A flexible data manager

LB is easily used by anyone for managing their data. It's menu driven for ease of use; absolutely no programming needed. Requires a Model 4 with 128K or a hard drive. LB86, an MS-DOS version is also available. Now activate PRO-WAM from newly compiled LB beta release (hardware restrictions apply)

- Store up to 65534 records per data base
- Up to 1024 characters (64 fields) per record
- Nine field types for flexibility
- Select and sort on up to 8 fields (multiple indexes)
- 10 input/update screens per data base
- 10 printout formats per data base
- Extensive on-line help available

LB [L-50-510]

**\$74.95 + \$5S&H**

## TRS-80 Model I/III/4 Language Software

**MRAS (\$59.95+\$4S&H):** An advanced Z80 assembly package for the programmer who wants a powerful and flexible development system. Includes a macro assembler which generates either relocatable object code modules or CMD files directly, a linker, a cross reference tool for directly generated CMD files **MLIB**, our REL module librarian, and our SAID advanced full screen text editor.

**EDAS (\$44.95+\$4S&H):** Powerful disk-based line editor and Z80 macro assembler assembles from nested source files or memory buffer; nested conditionals with ten pseudo-ops, nested MACROS with parameters both positional and by keyword, cross reference listings; and a separate full screen text editor.

**MC (\$79.95+\$5S&H):** a complete C compiler which adheres to the standards established by Kernighan and Ritchie. The package is supplied with the compiler, pre-processor, an optimizer, assembler macro files, C libraries, a Job Control Language file, the header files, and a 400+ page user manual. MC requires the use of either M-80 or MRAS (available separately), 2 disk drives, and upper/lower case.

**EnhComp (\$59.95+\$4S&H):** handles most of Microsoft BASIC; floating point single and double precision functions; random file access ("X" mode recls to 32767), turtle graphics, pixel graphics, keyed array sort, multi-lined functions, user commands, REPEAT-UNTIL, line labels, and more. Built-in Z80 assembler to easily create hybrid programs of BASIC and in-line assembly code.

**DSMBLR (\$24.95+\$2S&H):** Direct disassembly from CMD disk files, automatic partitioning of output disk files, data screening and full label generation. It even generates the ORGs and END statement.

**HartFORTH (\$49.95+\$3S&H):** a full 79-STANDARD FORTH; is designed to run under an operating system. The virtual Memory that it accesses for storage and retrieval purposes is a normal DOS file supports double length integers, string handling, cursor manipulation, graphics, random numbers, and floating point.



# The Cornsoft Group's arcade-type games Now available exclusively from MISOSYS

## Great Bouncing Boulders, It's ...



### BOUNCEOIDS

Where do they come from? What do they mean? You'll leave those questions to the scientists. Your duty is clear: DESTROY! Utilizing your newly issued class 4 annihilator, you must reduce the massive BOUNCEOID BOULDERS to cosmic dust! BEWARE! They bounce off of walls like giant pool balls, and the smallest touch means instant death! And as an added complication, terrible shaking bugs come slithering out of null space to crunch you into bug lunch! If you can prove yourself worthy, you will be challenged by the mysterious alien invaders in a nerve grinding test of incredible coordination and targeting skills. Will you survive?

## Crazy Painter



Become a "Crazy Painter" and create a masterpiece

...if you can

A mischievous puppy, snakes and poisonous turpentine buckets force changes in your painting. In higher skill levels, "paint eaters" start chewing up your work. Suddenly, everyone's a critic. Can you overcome them? And on top of it all, in the Exclusive Challenge Mode, you have to catch an army of runaway puppies. Crazy Painter is joystick compatible.

## The Official FROGGER™

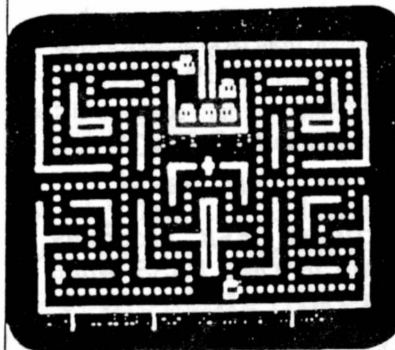


by SEGA

The popular arcade game for TRS-80 Models I & III.

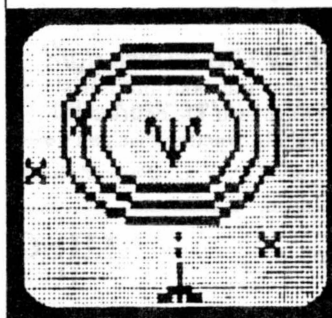
## SCARFMAN

Will you SCARF your way to victory?  
Can you avoid the MONSTERS that guard the maze?



Answers to these questions await the brave, who dare to enter the maze where the monsters live. SCARFMAN, written by the author of MISSILE ATTACK, features quality GRAPHICS, SOUND, REAL TIME MONSTERS, and hours of ENJOYMENT. Nine progressive skill levels make the game more difficult as you get better. Do you have what it takes to reach skill level 9?

## SPACE CASTLE



Will you save the Andromeda Galaxy by destroying the Space Castle or will the evil warlord Yugdab continue to rule, uncontested? Locked in battle with Yugdab, your main defense is your ability to skillfully handle your ship and its projectiles. If you are not careful, Yugdab's intelligent mines will hunt you down and blow you into space dust. Space Castle has fast action and exciting sound.

MISOSYS has licensed the powerful action games previously published by The Cornsoft Group: Frogger™, Scarfman, Bounceoids, Crazy Painter, and Space Castle are exceptional action games with great video and Alpha joystick support (even the new MISOSYS joystick). All games are for Model III/I (or in III mode). Each is priced at \$14.95; Buy any two and take 5% off. Buy all five and take 30% off. Add \$2 for S&H in U.S.; Canada = U.S. + \$1; Foreign U.S. \*3.

MISOSYS, Inc.

P.O. Box 239

Sterling, VA 22170-0239

703-450-4181 or 800-MISOSYS

# TRSCROSS™

(Pronounced TRISS-CROSS)

***TRSCROSS runs on your PC or compatible, yet reads your TRS-80 diskettes! Copy files in either direction!***

*The FASTEST and EASIEST file transfer and conversion program for moving files off the TRS-80™ and over to MS-DOS (or PC-DOS) or back*

## TRSCROSS™

Copyright 1986 by Breeze/QSD, Inc.  
All rights reserved

- 1 - Copy from TRS-80 diskette
- 2 - Copy to TRS-80 diskette
- 3 - Format TRS-80 diskette
- 4 - Purge TRS-80 diskette
- 5 - Display directory (PC or TRS-80)
- 6 - Exit

*Shown above is the Main Menu displayed when running TRSCROSS on your PC or compatible.*

*TRSCROSS is as easy to use as it looks to be!* The program is very straight forward, well thought out, and simple to operate. TRSCROSS has several "help" features built into the program to keep operation as easy as possible. Just pop in your TRS-80 disk to your PC and copy the files right to your PC data disk or hard disk. *It couldn't be any faster or easier!* Packed in the PowerSoft binder is a typeset instruction manual with Index. All steps are detailed. Advanced features, for those that desire to use them include executing menu options right from DOS or from a .BAT file or macro. This can really speed up transfers when similar operations are performed frequently.

***TRSCROSS allows you to "TAG" all files to be moved in ONE pass!***

***TRSCROSS converts TRS-80 BASIC programs and SuperSCRIPsIT files in ONE PASS while COPYING to MS-DOS!***

No need to save your programs or files in ASCII or run a separate conversion program first before transferring. TRSCROSS reads your tokenized BASIC program or SuperSCRIPsIT file directly off your TRS-80 disk and performs the conversion all in ONE pass while being transferred directly to your PC or compatible computer. ***Automatically converts most BASIC syntax***, and lines that need special attention can be listed to a printer. (Does not convert PEEKs, POKEs, graphics, machine language calls or sub-routines.)

***TRSCROSS will even FORMAT a TRS-80 disk right on your PC!*** (Handy for those who use both machines!) Former TRS-80 users who no longer have their TRS-80, but still have diskettes with valuable data... this is exactly what you've been waiting for!

TRSCROSS will READ FROM and COPY to the following

TRS-80 double-density formats:

TRSDOS 1.2/1.3, TRSDOS 6.2\*, LDOS 5.3\*,  
DOSPLUS, NEWDOS/80\*, & MultiDOS.

DOS formats listed above flagged with \* signify that earlier versions of these DOS's are readable as well, but one or more sectors may be skipped due to a format problem in that version of the DOS. (Disks that were formatted with SUPER UTILITY™ or SUW4/4P™ do not have this problem.) TRSDOS 6.02.01, or higher should not have this problem. Disks formatted in any 80 track format, or single density are not supported.

TRSCROSS Requires: PC or compatible computer, 128K and a normal 360KB (40 track) PC or 1.2MB (80 track) AT drive. Double-sided operation is fully supported. If you have more than one disk drive, fixed drive, or RAM disk, operation will be much smoother. TANDY 1000 requires more than 128KB memory (DMA). TANDY 2000 is not supported at this time due to a difference in disk controller and floppy drives. "Special" data files (like PROFILE+) would need to be converted to ASCII on a TRS-80 first before they would be of use on a PC or compatible.

If you use both types of computers, or you plan to retire your TRS-80, this is for you. TRSCROSS will allow access to your TRS-80 diskettes for years to come. Copy your TRS-80 word processor data files as well as your Visicalc data files over to MS-DOS and continue using them with your new application.

**Only \$89.95**

Plus \$3 S&H (U.S.) or \$5 Canadian or \$12 Foreign  
Virginia Residents must add appropriate sales tax.

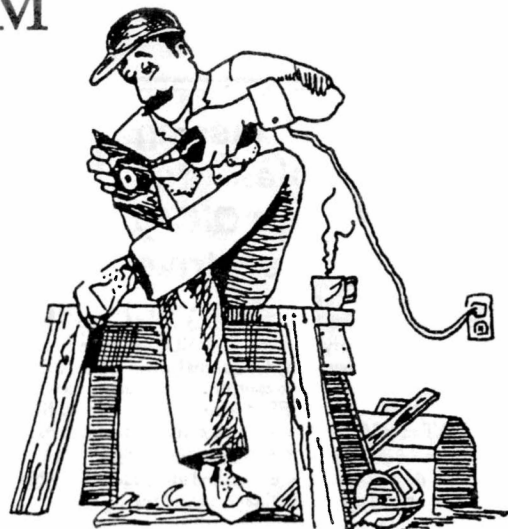
**MISOSYS, Inc.**  
P.O. Box 239

Sterling, VA 22170

Phone: 703-450-4181 (Orders only: 800-MISOSYS)

# Super Utility™

## for PC/MS-DOS™



A powerful new *file recovery* and disk media utility for floppies, internal fixed, and many external fixed and cartridge storage devices.

For IBM PC™, XT™, AT™ and many other PC compatibles including COMPAQ™, AT&T™, and TANDY 1000, 1200, 2000, and 3000.

PC/MS-DOS users... your wait is over for a *GREAT* new disk utility! Accidental deletions and disk directory problems *can happen to anyone at anytime...* a power spike, fingerprint, speck of dust, hardware problem, or simply typing DEL \*.\* in the wrong sub-directory can destroy critical data in a moment *without warning*. It's gone. Therefore we introduce *Super Utility™* for the PC - an easy to use disk utility containing many functions sorely needed in today's PC/MS-DOS computing environment.

With *Super Utility™* you can **restore damaged or deleted files** using two different methods (one easy, the other a little tougher). Even clusters of an erased file assigned to another file can still be restored, unless the user has physically written over every byte of the original data (especially useful in word processing files).

In addition to file repair and recovery, *Super Utility™* provides sector verify, sector editing, modification of sectors in Hex or ASCII, ease of renaming of files and setting their attributes, string search, copying sectors to a file, diagnostic sector checking, mapping of the FAT table of a file or an entire drive, visual graphics pertaining to your system, and full directory and sub-directory editing without endless menu-hopping - all in one program. The sector display mode displays all 512 bytes on-screen at one time and allows you to fully explore your disks. SEARCH and CHANGE are nice here! Compatible with DOS versions 2.0 - 3.1 on most systems. Some computers may require the use of PC-DOS. Color, composite, or monochrome video are supported. A great tool for fixed disk users as well as floppy. Also compatible with IOMEGA's Bernoulli Box™ storage device (also sold by Tandy as the Disk Cartridge System). *Super Utility™* is a totally new program that fills the gaps that PC users have most need of and have asked us for. It's aimed at the beginner, the "office user", hobbyists, students of the PC, and programmers alike.

**So, why not be certain about the safety of your data (and your peace of mind) when you can have *Super Utility* disk insurance right on hand at an unusually low price? Make your new computing life easier, more fun, and knowledgeable all at the same time. *Super Utility* is easy to use, *unprotected*, and great insurance against disaster *if and when* it strikes!**

**MISOSYS Acquisition Special**  
**Only \$29.95**

**MISOSYS, Inc.**  
P. O. Box 239  
Sterling, VA 22170-0239  
Phone: 703-450-4181 Phone Orders Only: 800-MISOSYS (800-647-6797)

Add \$3 S&H in U.S.; Add \$4 S&H in Canada & Mexico; Add \$9 S&H all others. Virginia residents add 4.5% Sales Tax. VISA and MasterCard accepted. Checks not drawn on an American bank are not accepted.

Requirements: IBM PC or compatible running PC/MS-DOS 2.x-3.x, min. 128K memory, one disk drive. PC-DOS may be required for use.

IBM, PC, XT, and AT are registered trademarks of International Business Machines Corp. MS is a registered trademark of Microsoft.



## Are you still fussing with floppies for BACKUP? Our QIC40 JUMBO tape drive from MISOSYS is your solution!



The Colorado Memory Systems' JUMBO tape drive is one drive that fits all computers. It comes ready for internal use in AT's, XT's, and PC's, and connects to your floppy disk controller. Our adaptor board can be used to connect JUMBO when all FDC connectors are in use. Kits are available for XT/AT or PS/2 machines which convert Jumbo to external use.

- In about 5.5 minutes, JUMBO backs up 10MB's file-by-file - the fastest in the industry! 40MB's gets backed up in about 18 to 20 minutes. Uses industry-standard DC2000 tape cartridges.
- JUMBO plugs into your floppy disk controller to save cost, power, and a slot. Needs 5-1/4" (or 3.5" with faceplate) mounting slot.
- Our tape adapter board mounts in your host computer to provide an additional tape port — especially useful in 286 and 386 computers. When used with our external JUMBO, it lets you share your drive between computers. Note: external adaptor includes "Tape Adaptor"
- JUMBO has custom chips, high speed brushless motors, automatic circuits, no pots to vibrate out of calibration, and fewest parts to assure long life and technical leadership for years to come.

DJ10 Jumbo	\$295 (\$5S&H)
Tape Adaptor	\$75 (\$3S&H)
External Adaptor	\$110 (\$5S&H)
DC2000 tape	\$22.50

M/C & VISA. S&H are U.S. only.

**MISOSYS, Inc.**  
P.O. Box 239  
Sterling, VA 22170-0239  
800-MISOSYS  
or 703-450-4181

## Now get the clearest view of your disk drive and memory contents with our DED86™



### Powerful features in Version 2!

When you need to travel through your disk drive, why settle for a tool that isolates you from the road? DED86 gives you the direct controls you need to explore your disk. It's a full-screen sector-oriented disk/file editor and a page-oriented memory editor. When you want to "unerase" erased files, DED86's flexible KEEP facility does the job without you fussing over FATs.

### Check out this list of features

- ✓ Look by cylinder/head/sector, sector or cluster; even reserved/hidden sectors
- ✓ Scan free clusters to search for erased data
- ✓ Jump about subdirectories
- ✓ Keep sectors & clusters for writing to a file; Great for partial file recovery
- ✓ Edit bytes in hexadecimal or ASCII, zap in 0s; Undo last edit!
- ✓ Search your disk drive, a file, or memory for ASCII or hexadecimal strings
- ✓ Touch a directory file entry with your date and/or time
- ✓ Obtain complete disk statistics in one screen; includes usage data
- ✓ Alter file attributes: archive, system, hidden, read
- ✓ Save/Restore sectors to/from auxiliary buffers; for moving data around
- ✓ DOS subshell available while using DED86 lets you shell to other programs
- ✓ Handles 5.25" & 3.5" drives & RAM disks; even large partition drives

**DED86 reduced by more than 50%; Now only \$29.95 + \$2S&H**

<b>Fujitsu Floppy</b>	5.25" 360K 1/2 height .....	\$75+\$4S&H
<b>Disk Drives:</b>	3.5" 720K in 5.25" frame .....	\$85+\$3S&H
	3.5" 1.44M in 5.25" frame .....	\$99+\$3S&H

# MISOSYS

**Programmer's Journal** downloads are now available from MISOSYS' forum on CompuServe: GO PCS49

## HartFORTH-86™

HartFORTH is a full 79-STANDARD FORTH. It is a Direct Threaded Code implementation which provides an execution speed between 10% and 40% faster than the classical Indirect Threaded Code implementation. HartFORTH is designed to run under DOS so the Virtual Memory that it accesses for storage and retrieval purposes is a file created and controlled by the operating system. Enhancements to the 79-STANDARD have been built into the HartFORTH kernel in the form of functions to call the DOS file handling routines so that other files may be created and accessed if required. A library of standard screens is supplied with HartFORTH to provide often used extensions to the language, such as double length and floating point math, editing of source screens, string manipulation, arrays, etc.

### Here's the FINE PRINT

*Any MISOSYS software product purchased directly from MISOSYS may be returned within 30 days of the invoice date for a full refund of the product purchase cost. Returns must be in a "like-new" saleable condition and be accompanied by a copy of the original invoice.*

**HartFORTH-86**  
\$59.95 + \$5S&H (US)

**MISOSYS, Inc.**  
P.O. Box 239  
Sterling, VA 22170-0239  
800-MISOSYS or 703-450-4181

- HartFORTH programs can invoke other programs via EXEC and EXEC.PROG.
- Functions create new files from within HartFORTH, and allow the current Virtual Memory file to be changed for another and manipulated at the individual block level.
- Provides the recommended 79-STANDARD DOUBLE NUMBER STANDARD EXTENSION word set that implements 32-bit operations.
- CASE: and SWITCH: functions allow multi-way branching decisions to be taken with execution continuing in-line once the word branched to completes.
- String manipulators include: "VARIABLE, "CONSTANT, "!, "LEFT, "RIGHT, "MID, "+, "COMPARE, and ">.
- DOS software and hardware interrupt vector access support via: GET.VECTOR, PUT.VECTOR, THIS.SEG, DI, and EI.
- V24 program input, output, and interrupt input support.
- Overlay management words: FORGET.OVLY, OVLYNAME, PUT.DATA, OVLY.ENTRY, SAVE.OVLY, CORRECT?, LOAD.OVLY, NEW.OVLY, RUN.OVLY, and LEAVE.OVLY.
- Screens provide trigonometric functions: SIN., COS., TAN., SIN, COS, and TAN.

# PowerSoft Products from MISOSYS, Inc

**SUPER UTILITY PLUS:** *The greatest utility ever written for the TRS-80. Every TRS-80 magazine has said so!*

Five-Star Excellent Reviews in 80-MICRO, 80-US, INFOWORLD, POPULAR ELECTRONICS, FAMILY COMPUTING, Creative Computing & more!

If you use a TRS-80 with disk drives, then this is a must-have program that you will wonder how you did without for so long! Super Utility has won numerous awards, has received many 5-Star reviews and this could be your last chance to purchase a copy at this unheard of price. Super Utility does so many things, you will never use its full potential, but it isn't that hard to use since it is completely menu-driven with the most common defaults built right in. It is configurable for all the popular TRS-80 operating systems and will even allow you to set one drive for one system and another drive for a different operating system and copy files easily between the two. Even between Model I and III or 4, regardless of density, track number, number of sides, or system used. We have thousands of letters in our files over the years about how Super Utility has saved the user from various problems. Super Utility removes or decodes passwords (strips them right off a disk in one pass), reformats a disk without erasing the data, fixes problems, backs up most protected disks, etc. This was the very best utility ever written for the TRS-80 and now is the time to get your own copy. Super Utility has over 65 functions and features. Too many to describe! A fantastic buy. Does not work on hard disks. Our ToolBox or ToolBelt has similar features for hard drive use, as well as floppy. SU+ does not support Newdos/80 double-sided disks.

Specify Model I/III or Model 4/4D/4P: On sale until April 30th for just \$24.95 + \$4 S&H

**\$34.95**

**LDOS ToolBox (Hard Disk Check, Repair, Modify, much more! Like a "SU+" for hard disk)** **\$24.95**

If you own a hard disk and use LDOS, this is the perfect insurance policy for your data. The LDOS TOOLBOX is like a Super Utility+ for hard disks. Features Disk Check and Disk Repair, Sector Modification, plus many, many other useful utilities that makes using a hard drive even easier. Each program contains a builtin help command, so many times you don't even need to look things up in the manual - just press <Enter> for help! A very wise buy for hard disk users.

**Model 4 ToolBelt (same for Model 4 DOS 6 use. OK for 6.3. Like a "SU+" for hard disk)** **\$24.95**

This is similar to the LDOS TOOLBOX, except it is for the Model 4 TRSDOS 6 operating system (all versions).

**Back/Rest - Super Fast Hard Disk Backup and Restore. Saves hours of time! For I, III or 4.** **\$34.95**

BACK/REST has proven to be a great time-saver for thousands of TRS-80 hard drive users. When reviewed by 80-MICRO, they gave it FIVE STARS - perfect! It saves hours of time and is very easy to use. BACK/REST can back up 10 megabytes in about 10 minutes and 20 meg in about 30-40 minutes. It also tells you how many disks to have ready. Works under LDOS or TRSDOS 6 (both versions on same disk). Great utility for hard disk users!

**Superior Hard Disk Drivers for Tandy disk systems. Mix Model III and 4 easily.** **\$49.95**

These hard disk drivers out-perform the Tandy drivers in many ways. Our drivers allow you to combine LDOS and TRSDOS 6 on the same drive and boot from either system (with floppy disk). They run faster and take much less memory from the system. Only for use with Tandy Hard Drives.

**PowerMail Plus (Please specify Model 4, III/L) 5 Star mailing list-data system!** **\$34.95**

This program was because all the other mailing list/data base systems couldn't keep track of all the types of data most folks wanted to keep track of. You needed speed, you needed hard drive support, and you needed a crash-proof data structure. PowerMail+ was top-rated (5 stars) in several publications and has never been topped. Works on floppies or hard disk under all popular TRS-80 operating systems. Allows importing of data from several other once popular mailing systems to avoid re-typing. Written in machine language by the author of Super Utility, this program is FAST and sorts up to 10 levels very quickly. If you keep track of names and addresses along with associated data for any situation, this is the one to use. Many churches, organizations and businesses use PowerMail+ for all the different kinds of lists they need to pull from. Each record has 24 user-definable "flags" to allow total customization for your exact needs.

**Text-Merge Form Letter Module - Create customized "form letters" and Labels with PowerMAIL+!** **\$15.00**

This optional module for PowerMail allows you to create customized "form letters" or custom labels, lists, etc. with PowerMail Plus and any word processor that saves text in ASCII format. Very easy to use and really gets the effect you want. Allows completely definable report generating from your PowerMail+ data.

**PowerDraw (animated TRS-80 screen graphics! Easy to use. Great for kids or adults!)** **\$19.95**

INFOWORLD, 80-MICRO and 80-US magazines really loved this program when they reviewed it. It does many things and is fun to use as well. First, PowerDRAW allows you to create graphics (mixed with text if desired) and save them to disk. It also allows you to create up to 33 "frames" of animation and "play" them like a movie. It also allows you to save the graphics in several modes, including BASIC listings, CMD file format, and others. These can then be merged into your own programs, etc., either in BASIC or machine language! Many of PowerSoft's opening screens were created with PowerDraw. In fact, it even creates animated opening screens (like we use in Super Utility, PowerTool, etc.) to really pep up the program. It also allows you to print the screens on Epson-type and several other type of printers. Lastly, PowerDraw has the ability to load in many types of TRS-80 graphic's and convert them to BASIC listings like a BASIC program generator!

**PowerDOT 2.0 for printers. Mix text with graphics - no problem. Build new fonts.** **\$19.95**

This program is similar to PowerDraw, but quite different. It allows you to create "hi-res" type screen graphics combined with text, and allows you to create drawings much larger than your screen. The screen is a "window" to a much larger drawing area and you use the arrow keys to move about the drawing. In a way, it is similar to Macpaint for the Macintosh computer. It also allows you to create custom fonts for ads, etc. Many of our early ads were created with PowerDot. It creates the hi-res effect due to each TRS-80 block pixel being printed as a single dot. Please specify if EPSON, Okidata, Prowriter, or Radio Shack printer.

**Note: Technical support of these products is not included at these prices.**

**MISOSYS, Inc.**

**P. O. Box 239**

**Sterling, VA 22170-0239**

**703-450-4181 (Orders only to 800-MISOSYS [647-6797])**

Please include \$3S&H per program package in US; Canada is US+\$1; Foreign is US\*3. COD (cash/money order) is \$3.50 additional.

Virginia residents add 4.5% sales tax, VISA/MC accepted.

## Model 4 / 4D / 4P Products From M. A. D. Software - October 1, 1989

**HBUILD6 - Boot your Model 4P, 4 or 4D directly from your Hard Disk**      Specify 4GHB600      \$10.00

This utility allows the Model 4, 4D or 4P owner with a hard disk to boot directly into LS-DOS 6.3 or TRSDOS 6.2.x from the hard disk without using a start-up floppy. Most systems can install HBUILD6 in less than five minutes. Completely transparent to TRSDOS 6.2.x and LS-DOS 6.3 and uses no program memory. Detailed instructions assist you during the installation if you have a non-standard system configuration.

### Requirements:

CPU	Model 4P systems with a standard (Z80A) or any superset processor, including the XLR8er adapter. Model 4 or 4D systems must also have XROM/XDROM firmware. (See the ordering information below.)
Floppy Drive	A 5.25" SSDD or DSDD floppy drive is used during installation.
Drive Controller	A Radio Shack hard disk controller or a programmable equivalent
Hard Drive(s)	Any combination of 5, 10, 12, 15, 20, or 35 Megabyte drives that are supported by the Radio Shack hard disk drivers ("TRSHD6"). Any combination of drives that are supported by the MISOSYS "RSHARD" hard disk drivers. TRSDOS 6.2.x or LS-DOS 6.3 must be installed and a system drive must start on head 0 of the primary drive. (Other operating systems that are stored on other parts of the drive or on secondary drives will not be disturbed.)

Other disk controllers and drivers may also work with HBUILD6. Write to us with specifics of your system along with a SASE.

<b>XROM - Extended ROM</b> for Model 4 systems	Specify 4BJXROOO	\$10.00
(Model 4 owners with an XLR8er adapter *)	Specify 4BXR005	\$10.00
<b>XDROM - Extended ROM</b> for Model 4D systems	Specify 4DXR010	\$20.00
(Model 4D owners with an XLR8er adapter *)	Specify 4DXR015	\$20.00

The XROM or XDROM improve performance when using any Model III operating system or ROM BASIC. They also allow you to boot directly from a hard disk or Networks III or 4. These ROMs replace the "C" ROM or "D" ROM in your Model 4 or 4D. Detailed instructions assist you in removing the cover and installing the ROM in your system. Installation takes less than 30 minutes. The new Model III-mode features include:

- The keyboard produces all 128 ASCII characters, handles key-rollover, and performs "debouncing" correctly at 2 and 4 MHz.
- The keyboard repeat rate is adjustable.
- The time-of-day clock keeps correct time even when the system is running at 4 MHz. The cursor blinks at the same rate regardless of the current CPU speed.
- A fully programmable 6845 CRT controller can be used in place of the 68045.
- All documented calls into the "C" ROM, including those documented by LDOS and Radio Shack produce the same results as in the original part.
- Interrupt-driven software performs more reliably than in the original "C" ROM.
- Includes built-in tests for memory, video, floppy drive, and printer (Only available in the XDROM version, via a factory-test hardware strap.)

The XROM/XDROM comes with documentation on the new Model III-mode features.

### Requirements:

CPU	Model 4 system. Order "X" ROM. Model 4D system, Order "XD" ROM.
Processor	Z80A or a superset processor, including the XLR8er accelerator. (Use correct order number.)
Floppy Drive	Any Radio Shack-compatible configuration including no drives.
Hard Drive	Any Radio Shack-compatible configuration including no drive.
Software	Operating Systems tested: LDOS 5, TRSDOS 6, LS-DOS 6, CP/M+, NEWDOS, TRSDOS 1.3/1.4.
Firmware	Model 4 owners must make sure they have the latest "A" ROM, Radio Shack part number 8048364A. ** This part is available elsewhere in the catalog. Model 4s built after 1983 and Model 4Ds should have the correct part.
Hardware	The XROM/XDROM may be used with a SmartWatch(TM) clock chip.

\* The XLR8er(TM) accelerator is supported provided that it worked with the system prior to the installation of the XROM or XDROM. If you have an XLR8er, specify 4BXR005 for the XROM (Model 4) and 4DXR015 for the XDROM (Model 4D). The standard XROM/XDROM will detect a 64180 CPU and will program it to run at speeds greater than what a XLR8er adapter may handle.

\*\* Model 4's built in 1983 may also require a new "A" ROM to be able to use the "X" ROM. To test your machine, boot your system into ROM BASIC (hold down BREAK key and press RESET.) Once you are in BASIC, type the following statement and a single character will be displayed:

```
PRINT CHR$(PEEK(1490))
```

If the character was 'O', you have the old "A" ROM and must replace it to use the "D" ROM.

### IMPORTANT

**When ordering the XROM or XDROM, be sure to specify the correct part for your system. The XROM will not work in a Model 4D system, and the XDROM will not work in a Model 4.**



**Radio Shack ROMs and PALs**

These ROMs and PALs were obtained from Radio Shack when they reduced their Model 4, 4D and 4P part inventories. We carry them for people who desire to make their own repairs or upgrades.

**Universal Character Generator - used in all Model 4 systems** Specify 4GCG000 \$4.00

If your Model 4 was built before 1984, it may contain an obsolete character generator. About that time, Radio Shack switched to a "universal" character generator which was used in all subsequent U.S. and international versions of the Model 4, 4P & 4D. To check to see if you have the older part, go into BASIC and type:

?CHR\$(0);CHR\$(26)<ENTER>

If a vertical line is displayed, then you have the newer part. If you see the letter "a" or some other character, you should consider upgrading your machine. Instructions for installation of this genuine Radio Shack part are included. Documentation on the universal character set begins on page A-58 of the TRSDOS 6.2 Disk System Owners Manual.

**Model 4/4D "A" ROM - 8048364A - Checksum 8F46** Specify 4GR64A0 \$8.00

The latest "A" ROM, used in all Model 4 and 4D systems. Replaces the obsolete 8048364, which was used in early Model 4 production.

**Model 4 "B" ROM - 8040332 - Checksum 407C** Specify 4BR3200 \$8.00

For three-ROM Model 4 and Model III systems.

**Model 4 U4 PAL - Video clock generator - 8075268** Specify 4BPU400 \$6.00

For non-gate array Model 4 systems.

**Model 4 U59 PAL - ROM/Video/Keyboard address decoder - 8075368** Specify 4BPU590 \$8.00

For non-gate array Model 4 systems.

Ordering information  
M.A.D. Software  
P.O. Box 331323  
Ft Worth Texas 76163

Prices are subject to change without notice. Texas residents must add state sales tax (6%). (The rate for Texas residents of Tarrant County is 7.5%.)

Shipping for one or two items is \$2 U.S.A., \$3 Canada, \$6 Foreign. For each additional item, add \$.50. We may ship your order via U.P.S. unless you specify otherwise or if your address is a P.O. Box.

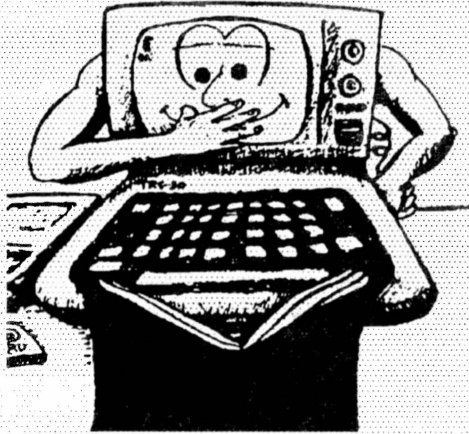
Please include a daytime phone number that we can use to reach you if there is a question about your order. Most orders will be shipped within one week. If the order is delayed by more than three weeks, we will contact you and give you the option to cancel.

Under development for Model 4/4D/4P systems

- New Boot ROM for Model 4P systems - Boot Model III operating systems from a floppy in less than three seconds. Includes video, disk and memory diagnostics as well as other features. The ROM can be personalized with your name or license number appearing on the screen while booting. Comes with a new Model III ROM image (MODEL4/III) containing all the features of the XROM.
- Networking Software for LS-DOS 6.3 using OMNINET/Network 4 hardware.
- New version of Model 4 BASIC with new functions and faster operation.

If you are interested send a SASE and we will alert you when the item becomes available.

# TRSTimes magazine



TRSTimes is the bi-monthly magazine devoted exclusively to the TRS-80 Models I, III & 4/4P/4D.

We are in our third year of publication and each issue typically features:  
"Type-in" programs in Basic and Assembly language,  
Hands-on tutorials, Hints & Tips, Reviews,  
CP/M coverage, Questions & Answers, Letters,  
Nationwide ads, Humor and more.

1990 calendar year subscription rates:  
U.S & Canada: \$18.00  
Other Countries: \$23.00 surface mail or \$29.00 air mail  
(all payments in U.S currency, please)

**TRSTimes magazine**  
**20311 Sherman Way. Suite 221**  
**Canoga Park, CA. 91306**

## Model 4/4P Users!

**T/Maker™**  
**\$49**  
ONCE \$450! NOW

"I was astonished at the power T/Maker put into my Model 4P computer. It was compact and versatile, a sort of digital Swiss Army knife, and I kept finding more ways I could use it." - Howard Graves

PC USERS - Buy T/Master, the next generation, for \$295 and we'll give you a free T/Maker for your Model 4/4P. Get the best of both worlds.

Add \$4.00 for ground shipment or \$7.00 for 2nd day air shipment. California residents add 7&1/4% sales tax.

T/Maker	\$49
Tandy 2000 or IBM Compatible disks	+\$20
PCDOS T/Master & Free T/Maker	\$295

T/Maker Integrated Software includes:

WORD PROCESSOR	SPELLING CHECKER
DATABASE	SPREADSHEET
BAR CHARTS	UTILITIES
APPLICATIONS LANGUAGE	

T/Maker Research Company  
812 Pollard Road (Suite 8)  
Los Gatos, CA 95030



(408) 866-0127



TRS 80 SALE

TRS 80 SALE

# RADIO SHACK TANDY OWNERS

Find The Computer Equipment That Tandy No Longer Sells

## Computers

Mod 3 2 Drive	\$245
Mod 4 2 Drive	\$345
Mod 4 2 Drive 128K	\$365
Mod 4P 2 Drive	\$335
Mod 100 24K	\$235
Mod 6000 15 Meg H.D. Computer.	\$895

## Printers & Hard Drives

Tandy 5 Meg H.D.	\$335
Tandy 12 Meg H.D.	\$395
Tandy 15 Meg H.D.	\$465
Tandy 35 Meg H.D.	\$875

*All hard disks include cable & software*

DMP 105	\$105
DMP 120	\$145

DMP 130	\$175
DMP 200	\$155
DMP 420	325
DMP 430	\$365
DMP 2100 24 Pin	\$395
DWP II	\$325
DWP 410	\$245
DWP 210	\$215
DWP 510	\$395
Line Printer 5	\$195
Line Printer 6	\$135
Line Printer III	\$125
410 Tractor (New)	\$ 95
Tractor for 2100	\$115
DW II Tractor	\$125
DW II Sheet Feeder (New)	\$245

## Software and Miscellaneous

Mod 3 TRS DOS & Manual	\$22
Mod 3 General Ledger	\$29
Mod 3 Accounts Payable	\$29
Mod 3 Profile 3 Plus	\$45
Mod 3 Investment Portfolio	\$10
Mod 4 P.F.S. File	\$45
Mod 4 Disk Scripsit	\$39
Mod 4 Deskmate	\$69
Mod 4 Payroll	\$45
Mod 4 Profile 4 Plus	\$79
Mod 4 TRS DOS & Manual	\$24
Mod 4 TRS 80 Pascal	\$49
Mod 4 Videotex Plus	\$34
Mod 3 Visicalc	\$29
Mod 3/4 Formation	\$18
Mod 4 Target Planner Calc	\$35
Mod 4 Multi Plan	\$79
Mod 4 Accounts Receivable	\$45
Mod 3 Dow Jones	\$15

Mod 3 Personnel Manager	\$15
Mod 4 General Ledger	\$55
Mod 4/4P Technical Ref	\$29
Mod 4 CBASIC	\$45
Mod 4 Superscripts	\$55
Mod 4 PFS Report	\$34
Mod 4 Visicalc	\$35
Mod 4 TRS-80 C	\$55
Mod 4/4P Teleterm	\$40
Mod 3/4 External disk drive	\$85
Mod 3/4 Keyboards	\$55
Modem 1B	\$29
Modem II	\$34
Printer Cables Mod 3-4	\$14
Mod 3/4 Disk Drives	\$55
Modem 4P	\$45
Printer Controller	\$65
Mod 4 Cobol	\$69
Mod 4 Cobol Query	\$49

- All equipment is guaranteed to be in good working order.
- Equipment is cleaned and tested.
- Drives are cleaned and timed as needed.

We accept VISA & MasterCard or C.O.D. The above prices do not reflect shipping cost. Inventory changes daily; please call for availability. If you don't see what you need, please call and we will do our best to locate it for you.

**Pacific Computer Exchange** The One Source For Used Tandy Equipment!

**(503) 236-2949**

**PACIFIC COMPUTER EXCHANGE**

1031 S.E. Mill, Suite B • Portland, Oregon 97214



# Our new LS-DOS 6.3.1 release has a little something for everyone



- ☆ The DATE command, "Date?" prompt on boot, and the @DATE SVC now support a date range of 32 years; from **January 1, 1980 through December 31, 2011.**
- ☆ **Enable or disable the printer time-out** and error generation with SYSTEM (PRTIME=ON|OFF)
- ☆ Customize the display of the time field in the DIR command to display **12-hr or 24-hr clock time** with SYSTEM (AMPM=ON|OFF).
- ☆ Both ASCII and hexadecimal display output from the LIST command is **paged a screen at a time.** Or run it non-stop under your control.
- ☆ MEMORY displays (or prints) the status of switchable memory banks known to the DOS, as well as a **map of modules** resident in I/O driver system memory and high memory.
- ☆ Specify SYSTEM (DRIVE=d1,SWAP=d2) to **switch drive d1 for d2.** Either may be the system drive, and a Job Control Language file may be active on either of the swapped drives.
- ☆ The TED text editor now has commands to **print the entire text buffer**, or the contents of the first block encountered. Obtain directories from TED, too!
- ☆ Have extended memory? The SPOOL command now permits the BANK parameter entry to range from 0-30 instead of 0-7.
- ☆ **Alter the logical record length** of a file with "RESET filespec (LRL=n)"
- ☆ Specify "RESET filespec (DATE=OFF)" to restore a file's directory entry to the old-style dating of pre-6.3 release. Specify "RESET filespec (DATE=ON)" to establish a file's directory date as that of the **current system date and time.**
- ☆ Felt uncomfortable with the *alleged* protection scheme of 6.3? **LS-DOS 6.3.1 has no anti-piracy protection!** MISOSYS trusts its customers to honor our copyrights.
- ☆ Best of all, **an LS-DOS 6.3.1 diskette is available as a replacement disk for \$15** (plus \$2 S&H in US). There's no need to return your current master.
- ☆ The 6.3.1 diskette comes with a 30-day warranty; written customer support is available for 30 days from the purchase date. Versions for the Model 4 and Model II/12 are available. **If you do not already have an LS-DOS 6.3.0, order the 6.3.1 Upgrade Kit with 90 days of customer support for \$39.95 (+\$2 S&H).**

MISOSYS, Inc.  
P. O. Box 239  
Sterling, VA 22170-0239  
703-450-4181 [orders to 800-MISOSYS (647-6797)]

## MISOSYS SPECIALS OF THE MONTH

### PRO-WAM Mister ED Application Pack half-off until March

31st



Mister ED is loaded with editor applications. All are full screen which make your editing jobs easy. Best of all, these are PRO-WAM applications so they can pop up even when you are using other Model 4 programs.

Mister ED includes: DED to edit disk sectors; FED, to edit file records; MED, to edit memory (even banked); VED, to edit the video screen; and TED, similar to TED/CMD to edit text files.

Only \$19.98 + \$3 S&H until April 30th

### SUPER UTILITY PLUS - The greatest floppy disk utility ever written for the TRS-80, now even greater at \$10 off - Only \$24.95 + \$4 S&H!

SU+ is completely menu-driven and is configurable for all the popular TRS-80 operating systems. SU+ removes or decodes passwords, reformats a disk without erasing the data, fixes problems, backs up most protected disks, etc. SU+ has over 65 functions and features. Too many to describe! Does not work on hard disks. SU+ does not support Newdos/80 double-sided disks.

Specify Model I/III or 4. \$24.95 + \$4 S&H until April 30th.