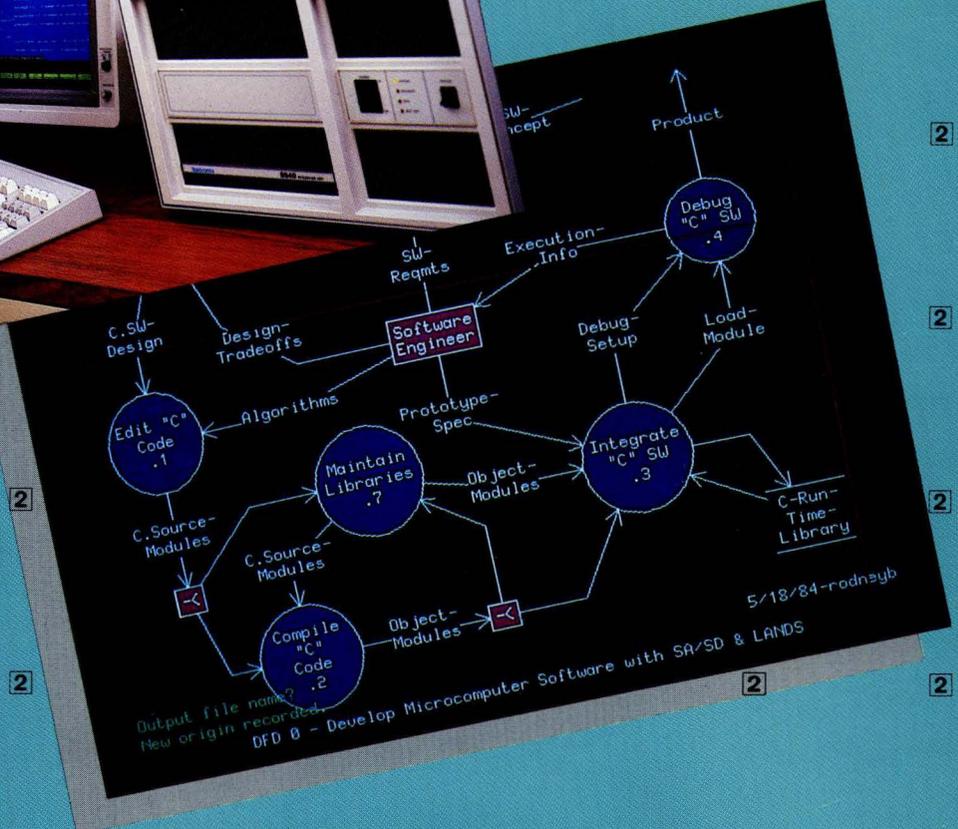
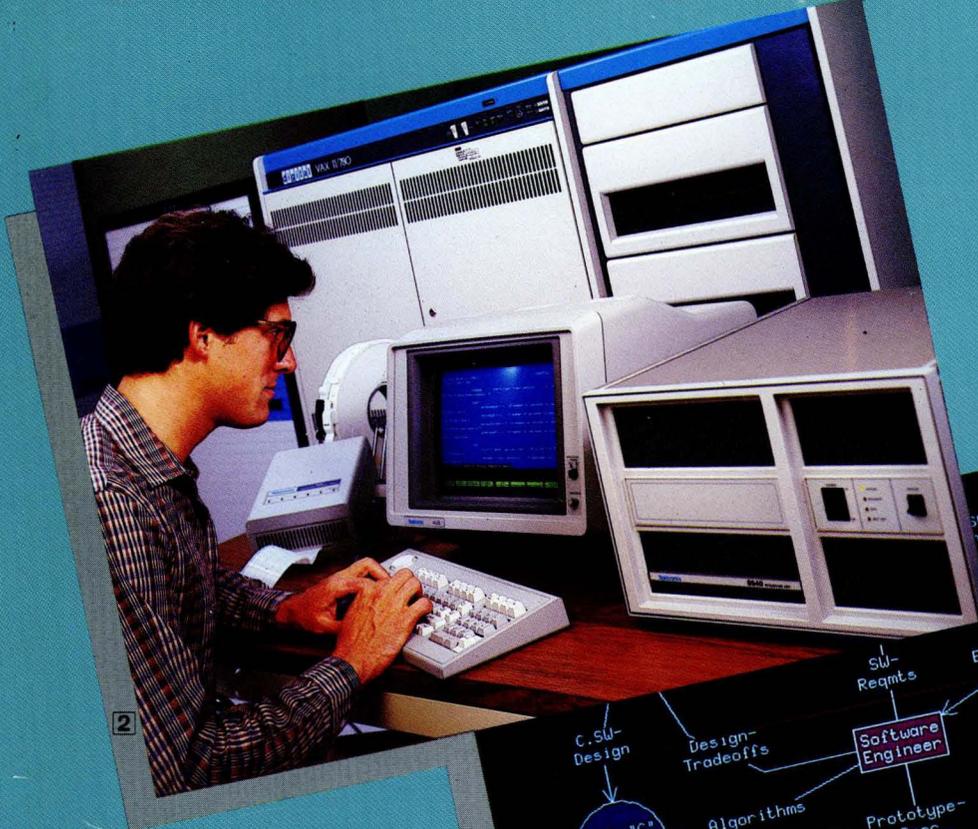


# USER GROUP NEWS



## FOREWORD

### A NEW INDEX FOR RANDOM ACCESS AND USER GROUP NEWS

The pages of "USER GROUP NEWS" are now sequentially numbered from one issue to the next. Each index page we produce will contain the index for all issues of the current volume. This will allow a faster search for information contained in the issues to date in the volume.

The new index with each issue will contain all the information of the previous index plus the added index items for the current issue.

For clarity, under each index item both the title and page number of the articles will be shown.

All articles appearing in the "APPLICATIONS", "PRODUCT PERFORMANCE", and "USER GROUP LIBRARY ABSTRACTS" sections are additionally listed under index items of the same name. Thus if you want to find a particular abstract item that was printed in the volume, but the issue is not known, simply look for the appropriate title under the index item "abstracts".

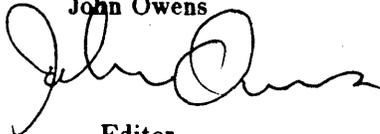
### CALL FOR ARTICLES

If you have application articles or just some good ideas, we would like to print them in the *Applications Section* and/or place them in the *User Group Library Section* of "USER GROUP NEWS". A submittal form is located at the end of the *Abstracts Section* for your convenience.

### ABOUT THE "USER GROUP LIBRARY SECTION"

We have collected a number of application programs since the last issue and we will make the programs reported in the User Group Library available through your Tektronix Applications Engineer. Each issue will report updates to the library and a separate total listing will be produced annually.

John Owens

A handwritten signature in cursive script, appearing to read "John Owens".

Editor

## CONTENTS

<b>PRODUCT INFORMATION SECTION</b>	49
V80186 AND V80188 SUPPORT SYSTEMS	49
8562 - LSI-11/73 BASED SOFTWARE DEVELOPMENT SYSTEM	50
CMOS 8086 & OTHER CMOS CHIP SUPPORT	51
NEW VERSION OF ICOM40 FOR VMS	51
NEW MANUALS	52
MDP SUPPORT OF THE TI SPR9000 SERIES	53
NEW SOFTWARE RELEASES	53
MIL-STD 1750A SUPPORT FROM TEKTRONIX	54
<b>APPLICATIONS SECTION</b>	57
68000/68XXX UPGRADES AND CONVERSION PACKAGES	57
8550 PROGRAM MEMORY IN AN 8002	58
8540/8560 CONFIG TERM I/O REDIRECTION	59
SA TOOLS TERMINAL CONFIG FILES	60
STRUCTURED ANALYSIS TOOLS CONFIGURATIONS	61
MULTI-DISK FILE BACKUP IS IMPROVED	63
CVT, THE TNIX KERNAL CONFIGURATOR, REVISITED	64
AN OFTEN OVERLOOKED RESOURCE IN THE TNIX SHELL	68
SPELLING ERRORS CORRECTED SEMI-AUTOMATICALLY	70
LINE PRINTERS SUPPORTING X-ON/X-OFF ON THE 8560	71
MDLu WITH THE NSC800 EMULATOR	71
EMULATING THE OMNIBYTE OB68KA1 WITH 68XXX	72
OBJXREF AVAILABLE FOR THE 8560	72
8560 PERFORMANCE ISSUES	72
68000 PROTOTYPE CONTROL PROBE ENHANCEMENT.	73
EPROM SUPPORT	74
RECOVERING ORIGINAL SOURCE FROM ASSEMBLER LISTING	75
HELP USING VMS	75
Z80 EMULATION RE-VISITED	78
Z80 PERIPHERALS AND THE DATA BUS	80
<b>PRODUCT PERFORMANCE SECTION</b>	81
SA TOOLS & FLOATING POINT PROBLEM	81
TNIX 2.1 SETKSH ERROR FIX	82
MDP BUG BASE	82
UNICOM FILE SIZE LIMITATION	82
ICOM40 AND PIPED OUTPUT	83
8086 CALL INSTRUCTION ERROR	83
8086 ASM I/O INST. ERROR	83
NO ERROR ON INPAGE JUMPS TO OUTOF PAGE CODE	84
68000 C COMPILER NAMED MODULES	84
ASM FAILS IN MAKEFILE	84
8550 ASSEMBLER FILE ATTRIBUTES	84
SETKSH CREATED FILE ERROR	85
SA EDIT OF DFD FILES	85
NESTED INCLUDES WITH VMS ASSEMBLERS	85
SYMBOL TRUNCATION WITH PDB	86
NO ESCAPE FROM PBD MODIFY	86
WRITELN FUNCTION REQUIRES 8540 TERMINAL	86
PRODUCT PERFORMANCE REPORT	87

<i>THIRD PARTY SOFTWARE</i> .....	89
JOINT MARKETING AGREEMENT WITH HUNTER & READY .....	89
<i>USER GROUP LIBRARY ABSTRACTS</i> .....	91
MUGL - MDP USER GROUP LIBRARY .....	91
CALEN - APPOINTMENT CALENDAR MAINTAINER .....	92
FORTH - UNIX FORTH .....	92
KERMIT - FILE TRANSFER UTILITY .....	93
OBJXREF - OBJECT CROSS REFERENCER .....	93
PROGRAMNAME-SUMMARY .....	93
UNHEX - TEKHEX TO TEKLAS OBJECT CONVERTER .....	94
UNHEX - TEKHEX TO TEKLAS OBJECT CONVERTER .....	94
Z8 ASSEMBLER .....	94
MDP USER GROUP SOFTWARE LIBRARY/ARTICLE SUBMITTAL FORM .....	94

## *PRODUCT INFORMATION SECTION*

---

---

### **V80186 AND V80188 SUPPORT SYSTEMS**

Tektronix MDP announces full in-circuit emulation for the 80186 and 80188. MDP's support for INTEL's 80186 and 80188 processors now includes V-Systems for both the 80186 and 80188 and individual probes for both processors.

The 80186/80188 emulator emulates the operation of the target processor device and allows debugging in three different emulation modes. It provides real-time support with no wait states. The input frequency to the emulator clock (used in emulation mode 0) is 8Mhz or 16MHz. The maximum input frequency from the prototype's clock (used in modes 1 and 2) is 16 MHz.

The V-Systems are systems designed to provide complete hardware and software support for design engineers needing the highest quality design tools available. The V-Systems are configured to integrate with an existing host computer, either one of the 856X family units or a VAX<sup>1</sup> Computer and include all the hardware and optionally the software to do so. Included with the V-Systems are Tektronix's 8540 Integration Unit, 64KBytes of memory, 80186/188 Emulator and Probe, and Integrated Logic Analysis (TTA).

### **SOFTWARE SUPPORT**

ASM8086 supports the 80186/80188 as well as the 8086/8088 and is available for the 856X family and VAX, both UNIX 4.1 and VMS 3.X. The Pascal LANDS, which includes the compiler for full memory configurations of the 8086/88 and 80186/88, is now available for the 856X family. The PDB in the LANDS package supports only the small code/small data option of the compiler.

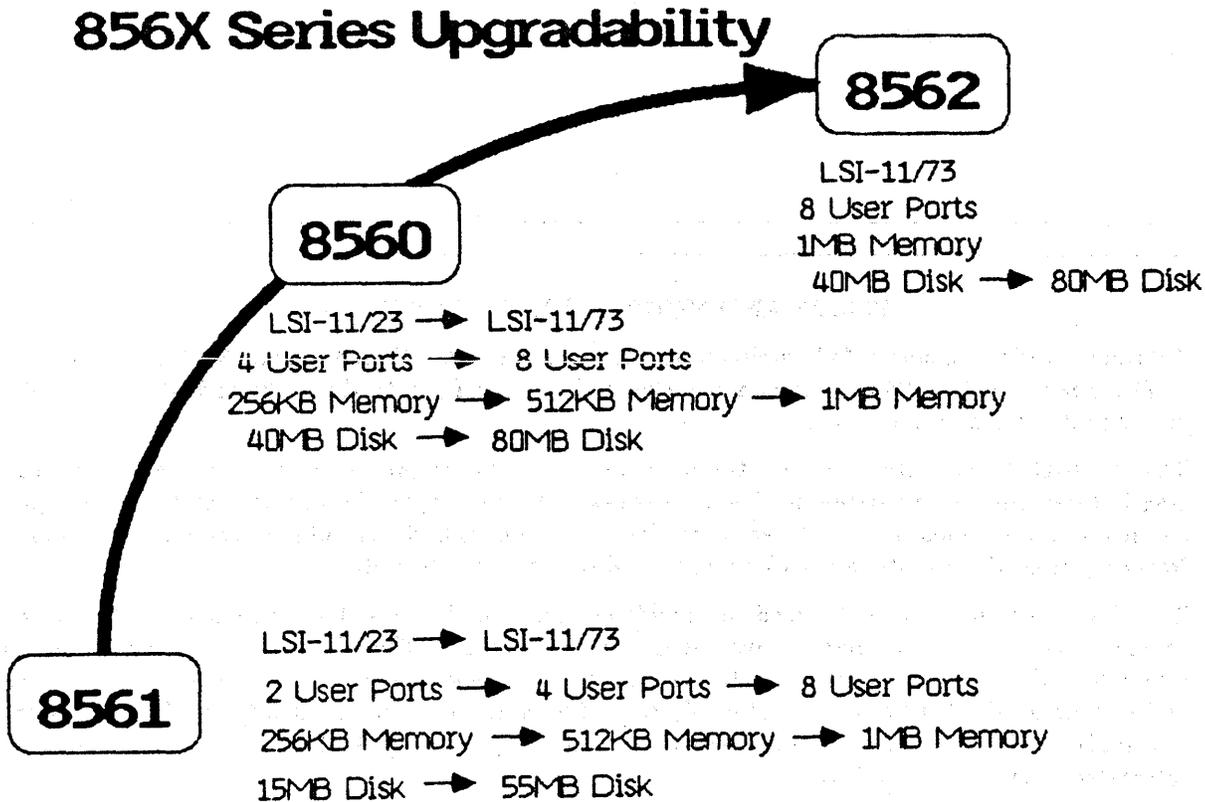
Marilyn Hanson    MDP Marketing

---

<sup>i</sup> Trademark of Digital Equipment Corp.

**8562 - LSI-11/73 BASED SOFTWARE DEVELOPMENT SYSTEM**

Tektronix MDP announces the newest member of the 856X Series of Software Development Systems - the 8562. The 8562 is a high performance software development system which supports up to eight users. Compatible with the 8561 and 8560, the 8562 offers an LSI-11/73 CPU, 40Mbyte hard disk, a 1 Mbyte flexible disk, 1 Mbyte of RAM memory, eight user ports and two line printer ports. The 8562 can be upgraded with a GPIB interface for enhanced disk backup capability, or 80 Mbytes of disk storage within the same mainframe.



The 856X Series is designed to be 100% software compatible and hardware upgradable. In addition to the 8562, the 8561 is a powerful, low cost entry-level system supporting one or two users and the 8560 is medium cost/performance system.

The 8561 provides a complete set of microcomputer design tools to the smaller design team while accommodating future expansion through a simple, cost-effective upgrade path. The 8561 consists of an LSI-11/23 CPU, a 15 Mbyte hard disk, a 1 Mbyte flexible disk, 256Kbyte of RAM memory, two user ports and two lineprinter ports.

The 8560 offers an LSI-11/23 CPU, 40 Mbyte hard disk, 1 Mbyte flexible disk, 256 Kbytes of RAM memory, four user ports and two line printer ports. The 8560 can be easily upgraded within the same mainframe to an LSI-11/73 CPU, 80 Mbyte of hard disk, 512K or 1 Mbyte of RAM, a GPIB interface for disk backup and 8 user ports.

All three systems in the 856X family are fully 100% software compatible and hardware upgradable - buy what you need, upgrade later.

Marilyn Hanson MDP Marketing

---

---

~ ~ ~ ~ ~

---

### CMOS 8086 & OTHER CMOS CHIP SUPPORT

CMOS microprocessors are becoming more available and popular each day. What are the limitations to supporting a CMOS 8086 with our probe? Our present emulator and probe can be used in a CMOS environment, if the following restrictions are heeded:

- 1) Vcc is limited to 5V.
- 2) The clock must not be slower than the microprocessor in the probe allows. In case of the 8086 that would be no less than 2MHZ. Keep in mind that most CMOS chips, including the 80C86, can go down to DC.
- 3) If the prototype uses 4000 series CMOS or 74HC CMOS which are not TTL compatible then problems may arise. Both drive and level incompatibility are of concern. The use of pullup resistors may help but this or any other solution would have to be custom designed and CAN NOT BE DETERMINED without the prototype. On the other hand, if the prototype uses 74HCT type buffers the emulator and prototype will be compatible.
- 4) The "prototype power" LED on our 8086 probe draws its power from the user. This could be undesirable in "flea power" designs.
- 5) Timing. If the CMOS microprocessor differs in its timing from the one used in the probe problems could arise. This is not likely and can be determined from the spec sheet.

Wolfgang Takatsch Hardware Product Support Manager

---

---

~ ~ ~ ~ ~

---

### NEW VERSION OF ICOM40 FOR VMS

A new version of ICOM40 for VMS has been released! This version, V02.00-00, has a number of enhancements, which this article will explain in brief. The new manual for the product has been carefully reviewed for its accuracy, and should be studied. In particular, read the technical notes in section four.

#### ENHANCEMENTS:

The single line configuration now has been added to ICOM40 ( terminal to 8540 to host ). This is referred to as the "Alternate Configuration" in the user's manual. The terminal is connected to the 8540, and VAX/VMS login port connected to J101 or J102 on the 8540. The separate "dual line" configuration is still supported( terminal to host and 8540 to host ).

ALL of the SVCs for the 8540 are now implemented. Besides the usual logical assignments ICOMPOR, ICOMSPEED, ICOMFILE, AND ICOMRETRY, there is a new logical which is

the ICOMFTYPE. This logical allows the user to specify which file type (ASCII or BINARY) the SVCs will create with SVCs 10 (assign file), and 90 (create file). If a file already exists, which is being assigned, the logical is ignored, and ICOM knows the type at the time of opening the file. The ASCII file type is described as a variable length, carriage return controlled, as produced by a VMS editor. The BINARY file type is 512 fixed length, as produced by a compiler or assembler. The default of ICOMFTYPE is ASCII.

Control-C works for all commands except, LO, DISP, and SAV.

There are now two different MODES a user has available when using ICOM40.

They are:

"8540/VMS command mode", which allows the user to inter-mix VMS DCL and 8540 commands. This is the same as version 01.00-00 of ICOM40 VMS.

"8540 command mode", in which all commands are sent to the 8540 directly, and VMS DCL commands will be flagged as errors.

The advantage of the 8540 command mode (entered by the command "ICOM", and indicated by the prompt "8540>") is that there is no chance of 8540/VMS DCL name conflicts. You must remember that there are 5 8540/VMS name conflicts, which must be prefixed by "ICOM", when not using the "8540 Command Mode". One of the more noticeable is "LO", which logs the user off the system. Also if an emulator has additional commands specific to the emulator and not generic to the 8540, they must be prefixed each time with "ICOM", unless you are in the "8540 command mode". A little bit of experimenting with this mode will bring out a lot of it's advantages.

Marty Kennedy      Software Project Leader

---

### NEW MANUALS

Following are new manuals published since the last User Group News issue.

8560 Series TNIX 2.1 Install Info Instruc Sheet	070-5057-00
TNIX 8560 Series System Reference Manual	070-4729-01
Pascal Compiler 8086/8088 Users Manual 8560 Series	070-3878-01
8500 Series 7807/7809 & 7810/7811 Users & Install Instruction Sheet	070-4997-00
8500 Series 7807/7809 & 7810/7811 Probe Service	070-4996-00
SA Tools Reference Card for 8560 Series Host	070-5177-00
SA Tools Users Manual for 8560 Series & VAX/UNIX	070-5098-00
Assembler 1750A Users Manual	061-3023-00
Assembler 1750A IEEE Standard Reference Booklet	070-5132-00
Assembler 1750A Mil-Std Reference Booklet	070-5133-00
Assembler 7810/7811/7816 Specifics Users	070-5148-00
8500 MDL Series Z80 Emulator Specifics	070-5159-00
Microlab I 80186/80188 Pers Card Supplement	070-5068-00
8560 Series 7810/7811/7816 Assembler Ref Booklet	070-5245-00
856X LSI-11/73 Upgrade Install Info Instruc Sheet	070-5095-00
ICOM40 Integrated Comm Sys Users Manual for VAX/VMS V2	070-4742-01

Charlene Eason      Sales Support Specialist

---

---

### MDP SUPPORT OF THE TI SPR9000 SERIES

Information obtained from TI indicates that the new series is the same as the SBP9989, TMS9900 and the SBP9900 except that the new chips run at 8.8MHZ instead of 4.4MHZ. In order to obtain the higher speed the new chips can only drive 7ma instead of 16ma, and sink 8ma instead of 16ma.

We may be able to support the chips in EM2. However, our customers can (if the TI info is correct in its 'total compatibility' claim) design and debug using our current emulators up to the specified clock speed of the emulator.

Harriette Lilly      Applications Engineer

---

---

### NEW SOFTWARE RELEASES

Following is software which has been released since the last issue of User Group News.

- NEW PRODUCT!! ASM7811 opt 1A, 7811 Assembler for 856X, version 01.00-16.
- NEW PRODUCT!! ASM1750 opt 1A, 1750 Assembler for 856X, version 01.00-16. There are two disks with this product.
- NEW PRODUCT!! ASM7809 opt 1A, 7809 Assembler for 8560/61, version 01.00-16.
- PAS8086 opt 1A, 8086/87/88 Pascal Compiler & ICS for 856X, version 02.10-00, which replaces version 01.1-05. There are two disks with this product. This updated version adds full memory support, requires 512K byte memory and TNIX version 2.1.
- TNIX Revision disk, version 2.1a. Installation procedures are enclosed with the disk.
- PDB68K opt 1A, 68000/68010 Pascal Debug, version 02.03.XX. This is an "unofficial" update which fixes the control-c problem in version 02.03-01.
- ASM68K opt 1F, VAX/VMS mag tape.
- ASMZ80 opt 1F, VAX/VMS mag tape.
- ICOM40 opt 1F, VAX/VMS mag tape.
- 856X System Diagnostics, version 01.03-01, which replaces version 01.03-00. This is a bug fix and includes an enhancement for the 8561/62.

If you are currently subscribing to our Software Subscription Service for any of the above products and have not received a new copy, please contact your Tektronix Sales Engineer.

Charlene Eason      Sales Support Specialist

---

---

## MIL-STD 1750A SUPPORT FROM TEKTRONIX

Tektronix offers a complete set of development tools to support MIL-STD 1750A processor-based systems. These tools are available TODAY so you can start your program NOW! Here are some of the highlights of the MIL-STD 1750A support from Tektronix.

### BACKGROUND

MIL-STD 1750A was developed by the U.S. Air Force to standardize on a 16 bit processor for use in embedded avionics weapons systems. The standard is unique in that it specifies the "instruction set architecture" of the processor, but not the physical or electrical implementation. For this reason, MIL-STD 1750A processors are available in many different shapes, sizes and performance ranges.

Tektronix made a commitment to support MIL-STD 1750A in 1982 as a natural evolution of their current design tool product line. We currently support over 35 different microprocessors including those that are popular in military applications, like the SBP 9900/9989, Z8001, Z8002, and 1802. So far, Tektronix is the only development system vendor to support the general MIL-STD 1750A environment.

Here are some highlights of Tektronix support:

### STRUCTURED ANALYSIS - A Tool for the Early Design Phase

The Structured Analysis System is used in the first design stage to specify the overall system requirements. This approach gives designers a better understanding earlier in the development cycle, before the actual design begins.

The designer uses a color graphics notation, displayed on a Tektronix 410x Color Display Terminal, to build the system requirements diagram. Graphics is the key in making the system specification easy to understand and communicate, and, it eliminates any time consuming drafting process. All of the Structured Analysis algorithms are handled by the computer so errors and omissions are uncovered automatically, before any design implementation.

### SOFTWARE DEVELOPMENT - A Choice of Host Computers

Several different host environments are supported for 1750A software development. Tektronix' own host computers, the 8560-Series, are multi-user systems featuring the TNIX (ie UNIX) operating system. These systems are ideal for small-to-medium projects using assembly language. A complete assembler package is available for the 8560-Series that supports both the MIL-STD and IEEE versions of instruction mnemonics. 8560 systems can also be linked together into local area networks, or connected to a large mainframe like a VAX to operate as a workstation.

The same Tektronix assembler package is available for VAX hosts plus, a special VAX interface package called ICOM40 for hi-speed downloading of run-time modules to the emulation system.

Since many 1750A design projects require an HOL (High Order Language) support such as Jovial, the same interface package can be used to transfer HOL run-time modules to the emulation system. Other host systems, like the IBM 370 or PDP 11/70, are also supported by using a high-speed RS-232 line to download executable modules to the emulation system.

### HARDWARE DEVELOPMENT - Hi-Performance Logic Analysis

Besides the traditional hi-performance oscilloscopes that are a Tektronix trademark, "specialty" tools like Logic Analyzers are available to support 1750A hardware development.

The Logic Analyzer is a "digital" oscilloscope with lots of channels to acquire and display timing information from a 1750A system. The circuit information can be displayed in the traditional timing, or waveform, display, or reformatted into the instruction mnemonics of the 1750A.

The DAS 9100 is a hi-performance logic analyzer that offers a color display, 72 to 96 channels and 10 MHZ performance. (Note: A 20 MHZ F9450 would only require 5 MHZ performance because the timing of the bus signals is 1/4 of the clock rate.) The DAS 9100 can also be configured to support up to 660 MHZ performance for ultra-high speed applications.

The same basic capability is available in a portable package with the 1240 Logic Analyzer. The 1240 features push-button "screen" keys with 72 channels at 50 MHZ performance. The 1240 Logic Analyzer also has a unique master/slave communications package that allows circuit information to be transmitted from a remote site (the slave), to another 1240 (the master), via a phone line. This means technical personnel can analyze circuit conditions without having to visit the site.

**SOFTWARE EXECUTION AND INTEGRATION - A Unique Emulation System**

The emulator is the most important tool used in the development of 1750A-based systems. The emulators basic function is to trace the flow of software as it executes, either on the actual target processor, or on a software execution unit. The software execution unit is used for testing the software when the target hardware is not available. Testing the software with the target (or a surrogate) processor system is the most important stage in the design verification process.

Tektronix has developed a unique emulator system to support the 1750A development environment. This emulator has two different probing techniques, or ways of connecting the emulation system to the 1750A hardware system.

The first probe, the 1750A Probe, was developed to support different implementations of MIL-STD 1750A processors. It connects to the electrical bus of the processor. The second probe is an in-circuit probe for the Fairchild F9450. It connects, or plugs into the socket for the F9450. Both probes support full function emulation but there are advantages to each technique.

	1750S Bus Probe	F9450 In-Circuit Probe
Supports the F9450	yes	yes
Performance	20 MHZ F9450 or 5 MHZ bus rate with no wait states	20 MHZ 9450 with no wait states
Supports other 1750A processors	yes	no
Can be connected externally	yes	no, chip replacement required
Executes software without target	yes, with Software Execution Unit	yes

**In-Circuit Probe vs Bus Probe.  
Which one is best for the application?**

Some customers may benefit from using both probing techniques. For instance, the prototype may be developed using the in-circuit F9450 Probe but the Bus Probe may be used in the final design because chip replacement is not practical. ie chip soldered in or not accessible.

**TEKTRONIX - A total solution for MIL-STD 1750A development**

Tektronix is firmly committed to supporting MIL-STD 1750A. We will continue to build and strengthen our support tools and some exiting new products are in the horizon.

**Tek has 1750A support....NOW and in the future!**

Bill Bevan

MDP Product Marketing

---

## *APPLICATIONS SECTION*

---

---

### **68000/68XXX UPGRADES AND CONVERSION PACKAGES**

The following table summarizes the upgrade and conversion packages available for the 68000 and 68XXX Emulator and Prototype Control Probes.

**Note:**

- The 020-xxxx-xx numbered kits listed in the above table require modifications and installation by the Customer or Tektronix Service Personnel. Assembled and tested conversion packages are available through the Tektronix Board Exchange Program. It is recommended that the Board Exchange conversion packages be used where possible.
- The 68000 Emulator 8MHZ to 10MHZ Board Exchange conversion package part number is 020-1242-00. The conversion package 020-1242-00 replaces 020-1076-00.
- The 68000 Emulator 8MHZ or 10MHZ to 68XXX Emulator 12.5MHZ Board Exchange conversion package part number is 020-1255-00. The conversion package 020-1255-00 replaces both 020-1077-00 and 020-1235-00.

#### **Additional Conversion/upgrade Notes**

- The 68000 Prototype Control Probe (8300P26) can be used with the 68XXX Emulator if the 68000 Prototype Control Probe (8300P26) has been upgraded for 12.5 MHZ operation. The 68000 to 68XXX Emulator Processor Conversion Kit 020-1077-00 or the Board Exchange Conversion package 020-1235-00 can be used to obtain 12.5 MHZ operation. It will also be necessary to configure jumpers P3161 and 3167 on 68XXX Emulator Board 1 for the 68000-F.
- The 68000-A Prototype Control Probe (8300P39) and the 68010 Prototype Control Probe (8300P40) are essentially the same, with the exception of the Microprocessor and jumper changes. 020-1078-00 and 020-1079-00 provide the 68000-12 or 68010 Microprocessor respectively. A description of the jumper configurations can be found in the 68XXX Emulator Processor User and Installation Manual part number 070-4691-00 in section 5 (this is a standard accessory with the 68XXX Emulator Processor).

68000/68XXX Upgrades and Conversions					
FROM\TO	and Probe 10 MHZ	and Probe 12.5 MHZ	and 68000-A Probe	and 68008 Probe	and 68010 Probe
68000 EMULATOR and Probe 8 MHZ	order 020-1076-00 *	order 020-1235-00 *	order 020-1235-00 * and 8300P39	order 020-1235-00 * and 8300P38	order 020-1235-00 * and 8300P40
68000 EMULATOR and Probe 10 MHZ		order 020-1077-00 *	order 020-1077-00 * and 8300P39	order 020-1077-00 * and 8300P38	order 020-1077-00 * and 8300P40
68XXX EMULATOR and 68000-A Probe				order 8300P38	order 020-1079-00
68XXX EMULATOR and 68008 Probe			order 8300P39		order 8300P40
68XXX EMULATOR and 68010 Probe			order 020-1078-00	order 8300P38	

Kevin King

Applications Engineer II

### 8550 PROGRAM MEMORY IN AN 8002

It is possible to use 8550F02 (Static Program Memory 670-6542-XX) in an 8001, 8002 or 8002A. Several advantages exist in using the 670-65432-XX in the 800X series systems. A few of these advantages are as follows.

- The 670-6542-XX Static Memory is faster than the 800X program memory.
- Parity checking is not done on the 670-6542-XX. This will have the advantage of never seeing another DOS ERROR 66, Program Memory Parity Error. In most cases where DOS ERROR 66's are seen, the error is generated by noise and doesn't represent an actual failure.

One strap change to the 8550F02 (670-6542-XX) Program Memory is required to allow it to function in an 800X System. W7080 on the 8550F02 must be cut. In the 8550 and 8540 systems Pin 56 of the system bus is a ground. This is one of several additions to the 85XX Series systems to reduce noise. In the 800X Series Systems Pin 56 of the system bus is the HOLD(L) control line. If W7080 were not cut Pin 56 the 800X HOLD(L) line would be asserted low and the 800X System Bus will lock up. Memory Board jumpers and straps should be configured as shown in the following table.

Program Memory Board 800X Operating Configuration		
J5175	Memory Relocation Jumper	Pins 2 and 3
J6175	Low/High Board Jumper	2 and 3 Low Memory (0-32K) 1 and 2 High Memory (32-64K)
J6179	Program/System Memory Jumper	Pins 1 and 2
J7171	Extended Bank Jumper	Pins 2 and 3
W5011	Delayed Read Strap	No Change (Pins 1 and 2)
W7080	Line Grounding Strap	Must Be Cut (Open)
S7170	Extended Memory DIP Switch	ON or Closed position

Kevin King      Applications Engineer

---

-----

---

### 8540/8560 CONFIG TERM I/O REDIRECTION

A feature of the 8540/8560 config term interface is I/O redirection. The intent of this article is to provide a few examples of the use of 8540 I/O device's. Three basic functions are described. Redirecting the output of an 8540 command to a 8540 I/O device , copying a 8560 file to an 8540 I/O device and redirecting the input from an 8540 I/O device to an 8560 file.

When redirecting the output of an 8540 command to an 8540 I/O device the following command syntax can be used.

```
(8540 command) \>(I/O Device Name)
```

Note a "\ " placed in front of the redirect. This removes the special meaning of the ">" to the 8560 and allows it to be passed to the 8540. An example of redirecting the output of the 8540 dump command to the 8540 device LPT follows.

```
d 0 100 \>LPT
```

It is also possible to copy a file on the 8560 to an 8540 I/O device. The following command syntax can be used.

```
cop (8560 file name) "(8540 I/O device name)"
```

Note in this instance the undocumented 8540 command cop is used. The file referenced by the cop command is a file on the 8560. Also the 8540 I/O device name LPT is enclosed in double quotes. The double quotes allow the 8540 I/O device to be referenced. An example of copying a 8560 file called junk to the 8540 I/O device LPT follows.

cop junk "LPT"

Information can also be input from an 8540 I/O device to be placed into a file on the 8560. The following command syntax can be used.

cop "(8540 I/O device name)" (8560 file name)

The following is an example of reading information from the 8540 I/O device CONI into a file on the 8560 called junk. In this instance the 8540 I/O device CONI is the terminal keyboard. Note: a <CTRL> <D> is required to indicate the end-of-file.

cop "CONI" junk

There may be other methods of redirecting 8540 I/O while in the config term mode. These are just a few examples to aid in getting started.

Redirection to ports on the 8540/50 can take place only when an 8540/50 command is used.

Kevin King

Application Engineer

## SA TOOLS TERMINAL CONFIG FILES

SA Tools work only with Tek graphic terminals (see **SA TOOLS CONFIGURATIONS**). SA Tools (856x version) includes an **sacap** file that has entries for 4010, 4014, 4025, 4105, 4107, 4113B, 4115B, and 4112. The **sacap** file provides information that governs the use of the terminal's graphic capabilities and the appearance of DFD's. It's like termcap, only for graphics instead of text. Users can modify **sacap**, create entries for other Tek terminals, and reference local copies of **sacap**. Edit **sacap** with **ed** or **lde** to avoid splitting the long lines.

### DFD Appearance

With **sacap**, users can determine the use of color, size of text, location of title and author lines, smoothness of circles and (on VAX only) data-flow lines, angle/size of arrows, boundary and route point markers, and other characteristics. An example of modifying the default characteristics would be to increase arrow size/angle and circle/line smoothness and change the fill patterns for processes and externals.

### Graphics Editing

With **sacap**, users can specify the range sensitivity of cross-hairs to an object, number of dialog lines, use of graphic segments, 401x or 410x style coordinates, and the set-up and restoration of terminal parameters. An example of terminal set-up would be to change the dialog text to green on transparent cell and background so the DFD is visible beneath (remembering to change it back in the termination string). The reference booklets for 410x provide the control sequences for setting the terminal initialization and termination strings in the **sacap** entry.

Rodney Bell      Product Line Manager

---

-----

---

## STRUCTURED ANALYSIS TOOLS CONFIGURATIONS

The configuration of an SA Tools system has different requirements than  $\mu$ P support and involves a variety of components. Contact your Tektronix Sales Engineer for assistance in configuring an SA tools system.

### Host

SA Tools is orderable now for 856x and VAX-UNIX. The 8562 increases performance by about a factor of 2; however, this may not be noticed in the interactive DFD edit session. The 856x version requires TNIX 2.1. The 856x must also have the floating point processor; this is a separate option for the 8560/1, but is included in the 11/73 upgrade and the 8562. The VAX-UNIX version requires UNIX BSD4.2.

### Terminals

SA Tools work with Tek graphic terminals only. If you already have a Tektronix graphic terminal you can use SA Tools with it. If not, the Tek 410x line is recommended for its cost, color, combination of graphics and text support. Only the 410x line supports ColorKey+ and screen editors effectively. The 4105 is attractive as the lowest cost terminal in the line. A new member, the 4106, is like the 4107 without much segment memory. However, it appears not to have enough to support SA Tools. The 4107 has better screen resolution (copies are same resolution as 4105 only larger size) and segment memory, which enhances the move, delete, and redraw commands of the dfd editor. Its 30% increase in resolution provides a clearer screen for viewing DFD's. It produces larger copies on the 4695 and will produce transparencies on the 4695 at half size. The 4109 has a larger screen size for more complex diagrams and a video output for connecting to monitors and 4634 grey-scale imaging copier. The 4109 and 4107 zoom/pan functions can be used to some extent, but these are terminal functions only - not integrated with SA Tools software.

SA Tools work with 405x, 401x, and 402x terminals. Contact your Tektronix sales engineer for information about how SA tools work with these terminals.

The high-performance 411x line works with SA Tools also. The 4115B offers the most powerful graphics (excellent resolution, easier zoom/pan, more colors, largest screen), but no screen text editing and a higher price.

### Hard Copy Devices

A variety of hard copy devices can be used - some for DFD's, some for DD and MS's, and some for both. For immediate hard copy of DFD's or MS's, Tek copiers are recommended. The 4695 attaches to the 410x line of terminals and produces a ink-jet color copy of the screen in 1-3 minutes. The 4634 (and older 4632) with a calibration option will attach to the 4109 to produce high-resolution, grey-scale screen copy in seconds. For higher-quality, but slower, copies of DFD's use Tek plotters (eg 4662). Plotters attach directly to 4107/4109 terminals for screen copies or to the 856x (tty port) for all users to plot DFD's.

The Tek 4643 line printer can be used to print the textual parts of an SA Spec, the DD and MS's. Third party printers can also be used - some only with a VAX. Imagen laser printers and Printronix line printers with graphics option can produce mixed text and graphics. Tektronix will consult users on configuring them. Imagen printers cost between \$10,000 and \$30,000 and produce fast, high-quality, grey-scale copies.

### Text Editing

The user can specify any text editor for use with SA Tools by setting an environment variable (on UNIX or TNIX). SA Tools do not include a text editor. Either ACE or LDE are available on the 856x. If these are to be used with terminals besides the 4105, configuration files will have to be obtained or developed. Users will already have an editor (eg vi) on VAX. LDE can be used to enter pseudo-Pascal for the MS's. For screen editor use, select 410x terminals; other Tek graphics terminals don't support screen editing.

### Text Processing

Users will probably want the ability to format their documentation. On the 856x, the Text Processing Package is available. On the VAX a variety of packages are available from other vendors.

### ColorKey+

SA Tools includes a ColorKey+ interface script for use with 410x terminals. The 856x comes with the basic ColorKey+ support. Colorkey+ support is available separately for the VAX-UNIX.

### 856x Configuration

A 856x-based SA Tools system might include:

Name	Option	Description	Use
STRUCTA 8560	Opt 1A	SA Tools for 856x SW Development Unit	
	Opt 2	Floating Point	Required to run SA Tools
	Opt 3	5-8 Ports	Support for 7 users
	Opt 7	1M Byte Total Memory	for performance
ACEDIT 8560U01	Opt 1A	Ace Screen Editor	editing DD, MS's, specs
8560U05		Text Processing Package	produce documentation
4105		UNICOM	send SA Spec to other system
4106		Color graphics terminal	DFD and text editing
4695		" " "	better graphics support
4643		Color ink-jet copier	copy DFD from 4105/6 screen
4662		Line Printer	print MS's, DD, other specs
		Plotter	high quality DFD copies

### VAX-UNIX Configuration

A VAX-UNIX-based SA Tools system might include:

Name	Option	Description	Use
STRUCTA	Opt 1C	SA Tools for VAX-UNIX	
COLORKY 4106	Opt 1C	ColorKey+ for VAX-UNIX	supports SA Tools ColorKey+ DFD & text editing
4695		Color ink-jet copier	copy DFD from 4106 screen
4109		" " "	Video out, fast screen copy
4634		Grey-scale image copier	copy DFD fast from 4109

Rodney Bell

Product Line Manager

### MULTI-DISK FILE BACKUP IS IMPROVED

The following scripts are an update to the "bkup" command described earlier. The changes were made to improve reliability. If you are currently using the command, the following changes should be implemented.

Rather than reference the directory selected in all the commands, a "cd" to that directory simplifies subsequent commands. The "bkup" command text follows with changes highlighted:

```
cd ${1-.}
echo "Gathering data"
if test -f .UPDATE
then
  find . -newer .UPDATE -exec ls -dsl {} \; | \
  sed -n -f /usr/lib/bkup/sedf | \
  awk -f /usr/lib/bkup/awkf >/usr/tmp/bkup$$
else
  lr -lsa | \
  sed -n -f /usr/lib/bkup/sedf | \
  awk -f /usr/lib/bkup/awkf >/usr/tmp/bkup$$
fi
chmod 777 /usr/tmp/bkup$$
/usr/tmp/bkup$$
rm /usr/tmp/bkup$$
touch $1${1+}/.UPDATE
echo "DONE"
```

The sed script requires a change in line seven.

```
s/^\/\//
/\.\.\/{
s:/\//
s/^\/\// # /
p
}
/^.....dr/{
s/\(.....\) .* \(.*)\/\& \1 \2/
p
}
/^.....-/{
s/\(.....\) .* \(.*)\/% \1 \2/
p
}
```

The awkscript command parameters for the number of blocks per disk allowed is reduced to account for directories that are split between disks. The text follows:

```
BEGIN {SIZ = 0
CNT = 0
AA=""
TRAP = "0"
LLEN = 11
print "echo -n \"Enter return when first disk is ready!\""}
print "read ready"
LAST = "fbr -c"
{LEN = length($3)
/# /{AA = $2
PLEN = length(AA)}
{if ($1 != "#") {
  if ( SIZ + $2 <= 1970 && CNT + 1 <= 250 ) {
    if ( $1 == "%" ) {
      if ( LEN + PLEN + LLEN + 2 >= 1024 ) {
        print LAST
```

```

    print "fbr -u \\"
    LAST = AA$3
    LLEN = PLEN + LEN + 11
  }
  else {
    LLEN = LLEN + PLEN + LEN + 2
    print LAST" \\"
    LAST = AA$3
  }
  SIZ = SIZ + $2
  CNT = CNT + "1"
}
else{
  TRAP = "1"
}
}
if (TRAP == 1) {
  FIL = FIL + 1
  CNT = 1
  SIZ = $2
  if ( $1 == "%") {
    print LAST
    print "echo -n \\"Enter return when next disk is ready!\\"
    print "read ready"
    print "fbr -c \\"
    LAST = AA$3
    LLEN = LEN + PLEN + 11
  }
  TRAP = "0"
}
}
END { print LAST }

```

Very large files will still cause some problems which will be reported by the `"/tmp/bkup$$"` command which is invoked by the `"bkup"` command. If such errors are encountered, kill the command (`^C`) and edit the `"/tmp/bkup$$"` command to remove the problem file reference. re execute the `"/tmp/bkup$$"` command and enter:

```
$ touch .UPDATE
```

The touch command will update the date attribute for subsequent command usage.

Muggsie Nixon      Technical Support Specialist

### CVT, THE TNIX KERNAL CONFIGURATOR, REVISITED

System performance can be effected by the operating systems configuration table. The CVT program allows you to examine or alter the Configuration Variable Table. This table, part of the kernel process, represents the allocation of the operating system's resources. CVT allows changes to the configuration of this table without having to recompile the kernel. The following expands on the changes to the size of these structures and its effect on system performance. Its important to note that any changes you make affect the free byte count which is redisplayed with each change. The free byte count represents the available bytes left in the kernel process. You must not exceed the free byte count. The counting mechanism may be 'bugged' (we are looking into this now) so its best to keep track of the byte count yourself or reinvoke CVT to get a new count of the free bytes

available. CVT will not update the configuration table unless there is sufficient memory available.

- MOUNT

MOUNT selects the number of mountable file systems allowed in the system. The minimum is one mount, but 4 have been allocated in the default configuration. With the default of 4 mounts the number of bytes remaining for the kernel process is so small that reconfiguration with 4 mounts is not feasible. Note that this is not the case with 11/73 based systems( 8562 ). There is a larger amount of free bytes on the 11/73 kernel process than on the 11/23 because the 11/73 uses separate instruction and data space. So with an 11/23 based systems, if you do not need the other 3 mountable (nonroot) file systems you can free up the memory allocated for mounts for use in one of the other data structures. For each mount you free you will gain an about 500 bytes.

- BUFFER

Most system performance can be gained by allocation of most of the memory just released from the mounts to buffers, especially in applications that are disk intensive. The buffers number can be increased into the 50's or 60's. Further increases in buffers may reduce performance because the overall search time for all programs will increase.

- PROC

The number of process table entries. Any program or command running on the 8560 is considered a process. See the man page for the command 'ps' for more insight into Processes. This data structure contains one entry for each process. Each entry contains all information the system needs about a process while the process is swapped out of primary memory (ram). The size of this table determines how many processes may exist within the system at one time. The default value for this is 50. More important, for each user, the maximum processes allowed are 15. This number is not configurable. Memory upgrades to the 8560's will show increased performance by allowing more processes to exist in primary memory(ram) at one time. Note that the value for process must always be greater than or equal to the value for the text parameter, and less than or equal to the value for the swapmap parameter.

- INODES, FILES, and TEXT

INODES represent the number of active inode table entries. Each table entry identifies a unique, active (open) file. The default value for inode is 62. If you increase inodes, you must increase the file parameter proportionally.

FILE represent the number of open file table entries. This parameter is similar to the inode parameter, except that table entries occur for each instance that a file is opened. For example, if three independent processes opened the same file, there would be one entry in the active inode table and three entries in the active file entry. (However, if the processes were forked from a common parent, they will share a common open file table entry.) The default value for file is 80; the value must always be greater than or equal to the inode parameter value.

TEXT represents the number of text table entries. This table contains one entry for each current read-only text (code) segment. Each entry contains a segment's address in primary and secondary memory, and a reference count that indicates how many processes share the segment. The default value for proc is 25; the value must always be less than or equal to the value for the proc parameter.

Allocating more kernel process memory above the default values on these data variables will have less of an effect on system performance than do buffers and is application dependent. Running assemblers, linkers, and text processing programs like nroff which are FILE and TEXT intensive will require higher limits. If your system reports "file table overflow" or "inode table overflow" errors, you should modify CVT. Increase these values if you have kernel process memory available or can

afford to reduce buffers table size.

- SWAPMAP and COREMAP

Both SWAPMAP AND COREMAP indirectly effect performance.

SWAPMAP represents the maximum number of swap map entries. Each entry identifies a discontinuous section of the swap space (located on the root device); therefore, the parameter sets the maximum fragmentation of the swap space. The default value for swapmap is 60; the value must always be greater than or equal to the value for PROCESSES.

COREMAP represents the maximum number of core (primary memory) map entries. Each entry identifies a discontinuous section of available user memory; therefore, the parameter sets the maximum fragmentation of the memory. The default value for coremap is 60. The limit on this variable should be increased with memory upgrades.

The following table shows a default set of CVT values and two optimized sets for 8560s with 1 megabyte memories and single mount systems. Remember that this is application dependent and can still be 'fined tuned'. Note that on 11/23 systems there is not a lot of free bytes available in the kernel process whereas in the 11/73 counting free kernel bytes will not be necessary.

Parameter	Default	Optimized	
		11/23	11/73
MOUNT	4	1	1
BUFFER	20	30	60
PROCESS	50	50	50
INODE	62	80	90
FILE	80	90	90
TEXT	25	35	35
SWAPMAP	60	90	90
COREMAP	60	120	120

The Structures for CVT is as follows:

```
#define CVTLOC 050

struct cvtptr {
    unsigned cpcvt; /* this is at location CVTLOC */
    unsigned cpkend; /* pointer to real CVT */
    unsigned cpend; /* end of kernel instruction/data memory */
    /* end of used memory */
};

struct cvt {
    unsigned cnbuf; /* this is at location CVTLOC>cpcvt */
    unsigned csbuf; /* number of buffers */
    struct buf *cbuf; /* size of a buffer entry */
    struct buf *cebuf; /* start of buf table */
    /* end of buf table */

    char *cbstart; /* start of actual buffers */
    unsigned ciocache; /* number of i/o page buffers */

    unsigned cninode; /* number of inodes */
    unsigned csinode; /* size of an inode entry */
    struct inode *cinode; /* start of inode table */
    struct inode *ceinode; /* end of inode table */
};
```

```

unsigned      cnfile;          /* number of file table entries */
unsigned      csfile;          /* size of a file table entry */
struct file   *cfile;          /* start of file table */
struct file   *cefile;         /* end of file table */

unsigned      cnproc;          /* number of process table entries */
unsigned      csproc;          /* size of a process table entry */
struct proc   *cproc;          /* start of proc table */
struct proc   *ceproc;         /* end of proc table */

unsigned      cnmount;          /* number of mountable filesystems */
unsigned      csmount;          /* size of a mount table entry */
struct mount  *cmount;          /* start of mount table */
struct mount  *cemount;        /* end of mount table */
unsigned      cntext;           /* number of text table entries */
unsigned      cstext;           /* size of a text table entry */
struct text   *ctext;           /* start of text table */
struct text   *cetext;          /* end of text table */

unsigned      cnsmmap;          /* number of swapmap table entries */
unsigned      cssmap;           /* size of a swapmap table entry */
struct map    *csmmap;          /* start of swapmap table */
struct map    *cesmmap;         /* end of swapmap table */

unsigned      cncmap;           /* number of coremap table entries */
unsigned      cscmap;           /* size of a coremap table entry */
struct map    *ccmap;           /* start of coremap table */
struct map    *cecmap;          /* end of coremap table */

devt          rootdev;          /* root device (major/minor) */
devt          swapdev;          /* swap device (major/minor) */
devt          pipedev;          /* pipe device (major/minor) */

unsigned      nswap;            /* amount of swap space */
daddrt        swplo;            /* location of swap space */

timet         ctime;            /* system time of day */
char          tzname[4];        /* time zone name (3 char plus null) */
char          tsdayname[4];     /* daylight time zone name */
short         tzoffgmt;         /* minutes west of GMT */
short         tzflag;           /* time zone flags */

unsigned      maxmem;           /* max user process memory (clicks) */
unsigned      memstart;         /* start of user memory (clicks) */
unsigned      memend;           /* end of user memory (clicks) */

char          *cpmsg;           /* pointer into buffer */
char          *cmsgbuf;         /* start of msg buffer */
char          *cemsgbuf;        /* end of msg buffer */

struct IOInfo {                 /* buffer cache profiling (bio.c) */
    long      nread;            /* # blocks read */
    long      nreada;           /* # blocks read ahead */
    long      ncache;           /* # blocks found in cache */
    long      nwrite;           /* # blocks written */
    long      *bufcount;        /* # free cache blocks (histogram) */
} ioinfo;
};

```

Joe Morabito      Application Engineer

---

---

## AN OFTEN OVERLOOKED RESOURCE IN THE TNIX SHELL

When you login, a shell program is started for you. In addition to being a command interpreter with control constructs and local variables, the shell makes available a local environment that can be used for more than storing the few default variables that the system establishes for the user.

1. Commands that are needed for a short time can be stored in your environment. The following example demonstrates how "command files" can exist in your environment as well as in files. A file collection command (Shell Script) which might be invoked many times in a session and then not needed again is simply stored as a shell variable. When the command is needed it can be executed as shown below. The advantage is that the command does not have to be put in a file that is referenced by your PATH variable, is independent of your current location, and saves having to enter the command manually each time it is needed. Another benefit is that your operating directories and command directories are not littered with once-needed and long-forgotten commands.

```

$ : Enter the string

$ gather='for i in *.c;do cp $i $HOME/c.src;done'

$ : And use it.

...
$ eval "$gather"
...
$ cd elsewhere
$ eval "$gather"
...

```

When you are finished gathering source files, your c.src directory in your home directory will contain the selected files.

Another example of the use of the environment to contain commands is to execute a text reduction filter commands such as:

```

$ : Enter the string
$ compress="sed '{;
> s/ //g;
> /`$/d;
> }'"

$ : Verify that the command is entered correctly

$ echo "$compress"
sed '{;
s/ //g;
/`$/d;
}'

$ : And use it

```

```

$ eval "$compress <sourcefile >destfile"
...
$ cd elsewhere
...
$ eval "$compress <otherfile >shortfile"
...

```

Note that special characters must be quoted to avoid early interpretation as demonstrated above. In the first example, a "" quote was used around the whole command to avoid the interpretation of "\*" and "\$". In the second example, a "\ " was used to escape interpretation of the "\$" and the entire multiline command was quoted with "" to prevent immediate interpretation of the embedded return characters in the command string.

For those wishing to simplify the process to the maximum, a command can be written to execute the above examples with less typing. The command file would contain.

```
eval eval '$$@
```

If the command is called "x" and is placed in your command "PATH" the above examples would be reduced to:

```

$ export compress gather
$ x compress <thisfile >thatfile
...
$ x gather
...

```

Which is indeed easier to use. The use of shorter string names would further reduce the typing required to enter a command.

- Users frequently need to access information in several directories that require a long command line if included in a command. A better technique is to develop alias variables in your environment for frequently used directories. The shell provides "\$HOME" which is your defined login directory. Either in your ".profile" file or dynamically the user can add additional aliases such as:

```

$ tmac="/usr/lib/tmac"
$ uus="/usr/spool/uucp"
$ uup="/usr/spool/uucppublic"
$ specs="/usr/boss/project1/bmodule/modulespecs"

```

The aliases, when used in commands, reduce the time required to enter a command; thus providing a productivity gain.

for example:

```
$ ls $specs
```

John Owens      Marketing Information Manager

---

---

## SPELLING ERRORS CORRECTED SEMI-AUTOMATICALLY

Here's a handy shell script which takes the drudgery out of manually fixing the spelling errors found by the spell command.

```
: fixspell - fixes spelling errors according to file.err
: create file.err by editing the spell output file and deleting
: words that are correct and inserting the correct spelling next to each
: incorrectly spelled word.
: usage: fixspell fname
for i
do
infile=$i
cat $infile.err |
while true
do
read aa||exit
set 'echo $aa'
echo s/$1/$2/g >>sed$$
done
sed -f sed$$ <$infile >tmp$$
mv tmp$$ $infile
/bin/rm -f sed$$
done
```

First, run spell on your input source file(s). I use the following interactive shell script if I have more than a couple of files to process:

```
$ for i in *.ms
> do
> spell <$i >$i.err
> done&
```

This creates a .err file for each .ms source file in the current directory.

Next, use your favorite screen editor and edit each .err file. Delete any words that spell thinks are wrong. For each word that really is wrong, simply type the correct spelling next to it on the same line. For example, if spell's output .err file looked like this:

```
errormessage
ie
makefile
recieve
MDP
Mgr
seperate
TNIX
workaround
```

After editing, it should look like this:

```
recieve receive
seperate separate
```

Now run the fixspell command:

§ fixspell foo.ms

Fixspell will read your edited .err file(s), construct a sed script to make the corrections, and finally run sed to fix your file.

Gordon Glather    Software Support

---



---

### LINE PRINTERS SUPPORTING X-ON/X-OFF ON THE 8560

Line printer ports do not support x-on/x-off protocol on the 8560. If you have a tty port available it can be configured as a line printer port with x-on/x-off support. Perform the following procedure as the superuser to move the line printer link from an aux device to a tty:

1. Choose what /dev/tty?? is going to be the new line printer. For example, choose /dev/tty03 as the new /dev/lp1.
2. Change the entry for the terminal in the file /etc/ttys so that there will be no login for the terminal tty03. (See ttys)
3. Change the link to the new /dev/tty?? from the old /dev/aux. For example:

```
rm /dev/lp1
ln /dev/tty03 /dev/lp1
```

4. Move your line printer connector from the "line printer 1" port to the "HSI I/O 3" port.

To move the line printer back to the /dev/aux port, reverse the above process.

Joe Morabito    Application Engineer

---



---

### MDLu WITH THE NSC800 EMULATOR

Any object code created for the 8080/85 uP will run on a Z80, and anything that runs on a Z80 will also run on an NSC800. So, 8080/85 or Z80 MDLu can be used to create code for the NSC800, right? Well, right with a minor qualification. When using service calls with the NSC800, the service request block (srb) must be defined with the large address space (las) format. The 8080,8085, and Z80, on the other hand, require the small address space (sas) format.

MDLu generated code produces the older sas format which is not compatible with the NSC800 emulator. Therefore, MDLu application programs that use service calls won't work directly on an NSC800 emulator. However, there is a simple patch to 8540/8550 system memory that will fix everything. After selecting the emulator with the sel command, patch system memory location 40B1H to 00 with the following command:

```
p *40B1 00 <cr>
```

This system location defines which srb format to use. Sas format is a 00, while las format is FF. Once patched, it will remain so until you re-select the emulator. It's a simple matter to create your own shell script or DOS-50 command file to automatically make this patch whenever you re-select the emulator.

Gordon Glather    Software Support Manager

---

-----

---

### EMULATING THE OMNIBYTE OB68KA1 WITH 68XXX

The Omnibyte dynamic memory refresh is done completely in hardware and requires AS and the various Chip Selects as inputs. When another bank of memory is accessed, for instance ROM, the refresh circuitry will cycle the dynamic memory. If dynamic memory is accessed for more than 15usec, the refresh circuitry will request the bus from the 68000 and expects a bus grant to be returned as spec'ed for the 68000. This, of course, requires that the emulator allows this action during non-emulation operations like ex, f, and lo which can hold the dynamic memory for more than 15usec. The refresh circuitry is 90% on a PAL with little or no documentation.

The user-DMA capability must be enabled at all times. Therefore, jumpers P2075 and P2121 must be in the [2-3] position.

The DTACK signal on the Omnibyte board is generated on a PAL, and a timing problem is created. The solution is to delay DTACK. Set P7123, P7124, and P2121 to the [7-8], [3-4], and [2-3] positions respectively. Also, set P2123 which controls the driving of AS to the prototype to position [2-3] ( Adds a wait state ).

Since the DTACK delay is only effective in mode 1, mode 2 can not be used without restrictions. The MAC board has been added to this configuration with no jumper changes required.

Ed Marks Application Engineer

---

-----

---

### OBJXREF AVAILABLE FOR THE 8560

Objxref was originally created only for use on an 8550 which didn't have the data reduction tools like grep, awk, etc. Later, versions were made for VAX/UNIX and VAX/VMS, but never for TNIX. Because of numerous requests, we have now compiled an 856X TNIX version. It will be available on the soon-to-be-released MUGL Volume 5. Check for availability with your local Tektronix Sales or Applications Engineer

Gordon Glather    Software Support Manager

---

-----

---

### 8560 PERFORMANCE ISSUES

Several factors should be considered when 8560's performance is at issue.

- Primary Memory (RAM)

Since all processes run in primary memory (ram), performance is effected by the number and size of processes running. If the size of processes running exceeds the available primary memory, process are temporarily placed in the swap area of the hard disk. A swapper routine, executed by the operating system, handles the scheduling of process swapping between memory and the hard disk. By adding

more memory you will reduce the incidents of swapping and increase system performance.

- **Hard Disk Capacity**

System performance will degrade when the file system exceeds about 80 percent utilization of the hard disk capacity. The system will have to spend more time searching for free blocks over discontinuous memory on the hard disk. System performance can be returned by either removing files or adding additional disk storage.

- **Disk Space Distribution**

System performance can be improved by the addition of a second hard disk. On a single disk system Swap Space is located on the high end of disk memory while the file system starts on the low end of disk memory. If a second hard disk is installed and the Swap Space is put onto it, you will reduce the average seek time the hard disk requires. In this case the first drive is used to access users files and commands and the second drive will be used primarily to access the Swap Space.

- **CPU Power**

The 8560's can be purchased with an 11/23 or an 11/73 CPU. Performance will vary depending on applications. Raw compute power for the 11/73 is about 4 or 5 times that of the 11/23. Overall 11/73 system performance gain will be at least 2 times that of the 11/23 systems. (Some small programs running on the 11/73 will show performance gains approaching 4 times that of the 11/23 systems.)

- **Operating System Configuration (CVT)**

CVT is a program that allows the user to examine or change the resource allocation tables of the kernel process. Changing the CVT table will effect system performance. See the article in this issue titled " CVT revisited " for more insight into CVT's operation.

Joe Morabito      Application Engineer

---

-----

---

### **68000 PROTOTYPE CONTROL PROBE ENHANCEMENT.**

MDP has a solution for customers with system integration problems due to noisy address strobe lines in 68000 user applications. A Prototype Impedance Matching (PIM) module (010-0439-00) will be orderable by October for 68000-A, 68000, and 68010 probes. The same module will fit in all three probes. Cost is \$150. The module fits piggy back on the buffer board. The board has circuitry which reduces the likelihood of address strobe glitches occurring.

Joe Morabito      Application Engineer

---

---

**EPROM SUPPORT**

The following should help determine what EPROMS we support with what module.

8550F30 PROM Programmer Controller

8550F31 2716/2732 PROM Module  
Module 1 supports:

TI	TMS-2508	2508/TI	1K x 8 bit EPROM
Intel	2758	2758/I	1K x 8 bit EPROM
Intel	2758-S1865	2758S1865/I	1K x 8 bit EPROM
TI	TMS-2516	2516/TI	2K x 8 bit EPROM
Intel	2716	2716/I	2K x 8 bit EPROM
Intel	2816	2816/I	2K x 8 bit EEPROM
TI	TMS-2532	2532/TI	4K x 8 bit EPROM
Intel	2732	2732/I	4K x 8 bit EPROM
Intel	2732A	2732A/I	4K x 8 bit EPROM
Motorola	MCM-68764	68764/M	8K x 8 bit EPROM

8550F32 8748/8741A/8755 PROM Module  
Module 2 supports:

Intel	8755A	8755A/I	2K x 8 bit Peripheral (EPROM with I/O)
Intel	8748	8748/I	1K x 8 bit Microcomputer
Intel	8749	8749/I	2K x 8 bit Microcomputer
Intel	8741A	8741A/I	1K x 8 bit Peripheral Interface Microcomputer

8550F33 2764/27128 PROM Module  
Module 3 supports:

TI	TMS-2564	2564/TI	8K x 8 bit EPROM
Intel	2764	2764/I	8K x 8 bit EPROM
Intel	27128	27128/I	16K x 8 bit EPROM
Intel	2817	2817/I	2K x 8 bit EEPROM

8550F34 68701 PROM Module  
Module 4 supports:

Motorola	68701	68701/M	2K x 8 bit Microcomputer
----------	-------	---------	--------------------------

8550F35 8751 PROM Module  
Module 5 supports:

Intel	8751	8751/I	4K x 8 bit Microcomputer
-------	------	--------	--------------------------

Mike Mihalik MDP Engineering

---

---

## RECOVERING ORIGINAL SOURCE FROM ASSEMBLER LISTING

The following is an easy way to recover source from a listing file. This procedure shows one way to do the job.

Create the following file (minus the explanatory comments in parenthesis) and call it "script":

```
g/asm/d      (delete banner/heading lines)
g/8560/d     (delete 2nd line of all banners)
g/`$/d      (strip out any blank lines)
1,$s/^[0-9 A-FR]*// (delete all the line numbers, hex address, obj junk)
1           (go back to the beginning)
/end/       (find your "end" statement, may need to modify)
.+1,$d      (delete all from the next line to the end)
w           (write it out)
q           (exit the editor)
```

Edit (a copy) of your listing file with:

```
ed listfile <script
```

This was tested on a Z80 asm listing and the output was identical to the original source. You may need to modify some of the search strings, but this should provide a good model to work from. If you have assembler macros, you will need to handle them separately.

Gordon Glather    Software Support

---

## HELP USING VMS

### VMS Output redirection and background

You are a UNIX user and you need to work with a VMS system. You may want to use multiple commands on a line, output redirection and background processing. Although VMS doesn't explicitly provide the user with these exact UNIX like facilities, they can be implemented. This article presents three VMS command files that can provide the user with these functions.

Initially, the command files must be incorporated into the user's account setup. In the top directory, create a sub-directory called BIN.DIR. BIN.DIR holds commands being discussed. Here's how the routines are named, however they can be called anything.

```
MULTIDCL.COM        Executes multiple command lines
REDRCT.COM         Redirects command output
BACKGND.COM        Generates a "background like"
                    capability
```

In the LOGIN.COM, the following definitions are made.

```
DEF BIN PUBLIC:[TSSSYRC.BIN]
MC:==@BIN:MULTIDCL.COM ""
RO:==@BIN:REDRCT.COM ""
```

```
BGR:==@BIN:BACKGND.COM ""
```

The three quotation marks cause a singular double quote to be passed as a parameter. This is essential for the routines to be able to parse the command line. The routines themselves are listed at the end of this article. Here's a cross reference of how these utilities compare in operation to UNIX.

UNIX command	VMS command
<code>cd newdir;ls</code>	<code>MC SET DEF NEWDIR DIR</code>
<code>ls -l &gt; dirlist</code>	<code>RO DIR/FULL &gt; DIRLIST.DAT</code>
<code>(cd newdir;ls -l&gt;dirlist)&amp;</code>	<code>BGR SET DEF NEWDIR RO DIR &gt; DIRLIST.DAT</code>

Lastly, the routines follow. They are written in DCL, the VMS command language.

```
$ LV='F$VERIFY(0)'
!
!           MULTIDCL.COM
!           -----
!
! Executes multiple DCL commands separated by backslashes. Cant use
! semi-colons because their significant in VMS filenames.
!
! For example:      MC SHOW DEF DIR
!
! Where 'MC' is defined as @MULTIDCL "". The quote passed as a
! parameter is essential to command parsing.
!
$ LL=0
$ LOOP:
$   LH = 'F$LOCATE("
$   IF ""F$LOGICAL("SYS$INPUT")" .NES. "SYS$COMMAND" THEN -
$     ASSIGN/USER SYS$COMMAND SYS$INPUT
$   IF LV THEN SET VERIFY
$   'F$EXTRACT(LL,LH,P1)'
$   LV='F$VERIFY(0)'
$   LL=LL+LH
$   P1[LL,1]:=" "
$   LL=LL+1
$   IF LL .LT. 'F$LENGTH(P1)' THEN GOTO LOOP
$ IF LV THEN SET VERIFY
```

```
$ LV='F$VERIFY(0)'
!           REDRCT.COM
!           -----
!
! This routine generates a capability for single line
! output redirection. Example:
!
!   RO DIR > FILE.TMP
```

```

!
! Where 'RO' is defined as @REDRCT """. One double quote passed as
! a parameter is essential to the parsing of the line.
!
! LL=0
! LH = 'F$LOCATE(">",P1)-LL
! IF ""F$LOGICAL("SYS$INPUT")" .NES. "SYS$COMMAND" THEN -
!     ASSIGN/USER SYS$COMMAND SYS$INPUT
!     CMD:='F$EXTRACT(LL,LH,P1)'
!     EOL='F$LENGTH(P1)-LH+1
!     OUT:='F$EXTRACT(LH+1,EOL,P1)
!     IF LV THEN SET VERIFY
!     ASSIGN/USER 'OUT' SYS$OUTPUT
!     'CMD'
! IF LV THEN SET VERIFY

```

```

!
!           BACKGND.COM
!
!

```

```

! This routine will generate a VMS command file based upon the commands
! entered on the invocation line. Each command must be delimited by
! a backslash. The resulting command file will be entered in SYS$BATCH
! to execute. The command file will be left in the default directory
! labeled (user).BGR. This file can subsequently be resubmitted into
! SYS$BATCH to re-execute.
!

```

```

! For example:

```

```

!     BGR SET DEFAULT [USER.LISTS] DIR TYPE *.LIS
!

```

```

! Where 'BGR' is defined as @BACKGND """. The quote passed as a
! parameter is essential to line parsing.
!

```

```

! Any output generated by the batch process will be deposited in
! the user's login directory titled (user).LOG
! The user is notified upon completion of the batch process.

```

```

$ OPEN/WRITE COMFIL 'F$DIRECTORY()'F$PROCESS()'.BGR
$ WRITE COMFIL "$ SET VERIFY"
$ WRITE COMFIL "$ ON ERROR THEN GOTO DONE"
$ WRITE COMFIL "$ SET DEF 'F$DIRECTORY()'"
$ LL=0
$ LOOP:
$   LH = 'F$LOCATE("
$   WRITE COMFIL "$ 'F$EXTRACT(LL,LH,P1)'"
$   LL=LL+LH
$   P1[LL,1]:=" "
$   LL=LL+1
$   IF LL .LT. 'F$LENGTH(P1)' THEN GOTO LOOP
$ SUB:
$ WRITE COMFIL "$ EXIT"
$ WRITE COMFIL "$ DONE:"
$ CLOSE COMFIL
$ SUBMIT/NOTIFY/NOPRINT 'F$DIRECTORY()'F$PROCESS()'.BGR
$ EXIT

```

\$ ERROR:  
\$ WRITE SYS\$OUTPUT "No command to execute"

Mark Malinoski Application Engineer, Syracuse

---

---

### Z80 EMULATION RE-VISITED

The Z80 emulator has experienced many refinements over the years as well as an enhancement in speed from 4MHZ to 6MHZ. Soon a new driver/receiver board (inside the Z80 probe) will be in production. To understand what some of the enhancements mean to the user a brief historic review follows.

The old Z80A emulator and probe were designed with the idea that a cpu must be stopped dead in its tracks when a break point is encountered or in between instructions in trace all. This required extra circuitry to keep things like refresh and DMA available to the user. (Hence the old probe had a "auxiliary refresh counter" on it as well as a circuit that returned BUSACK whenever it saw BUSREQ and gating to tri-state appropriate buffers to allow user DMA while the Z80 was halted).

Next came the 6MHZ Z80 and with it the need for a Z80B emulator and probe. This time we had a better idea in that we kept the Z80 running in a JMP to self loop while "stopped" at a break. The advantage is that now REFRESH and the response to user BUSREQ could come directly from the Z80 without the need for auxiliary circuitry. This worked quite well, but 6MHZ operation brought on some critical timing considerations. The following paragraphs should help you understand emulation in general and some Z80 concerns in particular.

With any of our emulators we are faced with essentially the same task, namely routing "information" between the CPU and

- 1) Program memory in the development system
- 2) Memory associated with internal routines
- 3) User memory on the prototype

All of these may, at least in part, occupy the same identical address space. "Information" means

- 1) Address
- 2) Data
- 3) Control signals

The emulation mode, the memory map and whether or not user code is executing, determine the rules for routing these signals. Let's take each of these signal groups and consider them in turn as seen at the point where the probe plugs into the prototype.

Address:

This is the easiest group to route because it only goes in one direction from the CPU: out. The only exception is during processor hold (DMA) when the address bus must be tri-stated. So, the rule our emulator follows is simply:

- 1) Addresses are always driven to the prototype even when we are executing internal routines.
- 2) During processor hold (DMA) the address bus is tri-stated.

Data follows the map

- 1) In emulation mode 2 data buffers point in the direction dictated by READ and WRITE .
- 2) They are tri-stated during processor hold.
- 3) In emulation mode 1 the map is followed and the data buffers are tri-stated unless the address accessed is mapped to user.

Control signals: the new approach.

The routing of control signals depends on

- 1) whether the signal is an input or an output
- 2) mapping
- 3) strapping

It is beyond the scope of this article to cover all control signals in detail. Emulator specifics and the service manual serve that purpose. What is important here is to point out the difference in approach taken between the present Z80B probe and the new one. The difference is simply this: in the present Z80B probe, control signals followed the map. In the new one, no attempt is made to block control signals from the user. Writes that occur during internal routines are converted to reads on the prototype. Why this change?

Enter stage left: the wicked glitch of MREQ.

You see, some control signals come out almost at the same time as the address. Like MREQ for example. If you try to inhibit them after the mapping decision has been made one of two things will happen: either a short glitch is allowed through or a shortened (delayed) control signal results. Clearly not acceptable.

So what is the answer?

The answer.

Finally we settled on a scheme where most problems are overcome. We simply tri-state the data bus to the prototype when the map indicates that a fetch or read or write should be done from development system memory. Thats all. All control signals appear on the prototype all the time (including when we are running internal routines). Also, when we are "stopped" at a break point for example, the Z80 is kept running (jump to self routine) so that refresh etc. continues available to the user. This approach has the following side effects:

- 1) If memory is being written to inside the development system and the same address exists on the prototype it also will be written to (although the data bus will be tri-stated so that WHAT is being written will differ).
- 2) If memory is being read, both prototype and emulation memory will respond and put data on the data bus. No problem because the data bus to the prototype is tri-stated thus avoiding bus contention.
- 3) Internal routines perform IO reads from port "F0" which will also be seen on the prototype. Again, no problem UNLESS the prototype has port "F0" implemented with a device that is cleared by a read (such as an ACIA for example).

All in all things look very good. There are also some straps on the emulator and probe that should help accommodate those customers who have special needs.

Speaking of special needs, Z80 peripheral chips that monitor the data bus present a challenge. For a discussion see the article: "Z80 PERIPHERALS AND THE DATA BUS".

Wolfgang Takatsch Product Support Manager

---

---

### Z80 PERIPHERALS AND THE DATA BUS

Certain Z80 peripheral chips monitor the data bus to detect when the Z80 fetches a RETI instruction. This is done so the peripherals in question (CTC, PIO, and others) can tell when the Z80 is done with an interrupt service routine and is ready for the next maskable interrupt.

Care must be taken in emulation mode 1 so that RETI is seen on the users bus. (For a discussion of our approach to emulation see the previous article, "Z80 EMULATION RE-VISITED".) Basically there are two ways this can be done.

- 1) Arrange mode 1 mapping in such a way that the RETI instruction will be fetched from user memory on the prototype.
- 2) Remove all memory from the prototype and set strap J3051 and J1041 on the old, or 2051 on the new driver/receiver board to drive the data bus without regard for the mapping. The reason for removing all memory from the prototype is to avoid bus contention. Incidentally this would include ANY devices capable of driving the data bus in response to an address and/or read control signals (like buffers for example !). This may not be a trivial task, especially if the chips are not socketed.

Of these two options, number 1 is by far the most desirable. The objection usually raised to this approach is that interrupt service routines do not necessarily all reside in one area in the address space so that they can easily be mapped. A possible work around to this objection may be to end all interrupt service routines with a jump to a single module which contains the RETI instruction. This module can then be located at a convenient place in user prototype memory (CAUTION: our new Z80B probe duplicates all internal reads and writes on the prototype. Thus, an RETI that exists in prototype memory will be seen on the prototype even though that memory address is mapped into the development system !).

Wolfgang Takatsch Product Support Manager

---

## *PRODUCT PERFORMANCE SECTION*

---

---

### **SA TOOLS & FLOATING POINT PROBLEM**

SA Tools require the Floating Point Processor to be installed in the 856x. The FP processor is used to calculate positions of DFD elements for display on the terminal. TNIX 2.1 (and all previous versions) have a "bug" that causes two or more processes sharing FP to get incorrect results. Basically, FP registers are not saved or restored properly when FP is shared by more than one process. One obvious symptom is the appearance of "PACMAN"-like process bubbles when several users are editing or displaying DFD's. Rarely, a "divide by zero" exception occurs which will crash the process and hang-up the port.

Until TNIX is updated, a preliminary fix (a new version of the TNIX kernel) will be installed by a Tektronix representative for users of SA Tools or upon request. Contact your Tektronix field office.

Alternatively, the following will patch tnix (11/23 only) after each time it is booted. This forces the register save/restore.

```
adb -w /dev/mem
0120702?w1
$q
```

Rodney Bell      Product Line Manager

---

**TNIX 2.1 SETKSH ERROR FIX**

TNIX 2.1 and the setksh command have an annoying problem you should know about. When a user runs setksh, telling it he is using an 8540 or 8550 IU, setksh will put a line of the form 'stty IU >/dev/ttyx' into a file called .setTEK in the user's home directory. If the user then attempts to login WITHOUT the IU on that port (or with it powered off), the login process will hang.

The solution, for the time being, is to manually remove the offending line from the .setTEK file. HOWEVER, if the IU in question is an 8550, the command is needed, so the solution in that case is to assure the IU is always powered up before logging in. We are working on a more permanent solution.

Byron Lutz            Customer Support Manager

---

-----

---

**MDP BUG BASE**

The following product performance reports are contained in our data base. If you have encountered additional problems not listed here or in previous issues, please use the product performance report form provided at the end of this section. We will keep you informed about the progress toward the solution of the problem. We will also try to provide a "work-around" immediately.

John Owens        Marketing Applications Manager

---

-----

---

**UNICOM FILE SIZE LIMITATION****Configuration**

856X TNIX 2.0/2.1 with UNICOM V1.01

**Problem**

Uucico, which is the file transfer process of uucp, fails to transfer files to a remote machine when the file /usr/lib/uucp/USERFILE gets too large.

This is due to the fact that uucico is a very large program. USERFILE contains access privilege information for remote machines. This file is read into memory by uucico. When this file reaches a certain size (18 lines 244 characters in our case), uucico will fail because it no longer can allocate enough memory for the process of sending files to the host machine.

The only workaround for 11/23 users is to limit the size of their USERFILE.

---

## ICOM40 AND PIPED OUTPUT

### Configuration

ICOM40 running on a VAX UNIX system. The 8540 is connected to the VAX in the Alternate (single line) configuration.

### Problem

When an 8540 command is issued and the resulting output is "piped" into the "more" command, problems occur when the output of the 8540 command exceeds a page of display.

This occurs because the tty port has been placed in "raw" mode to complete the 8540 command. If the output of the 8540 command exceeds one page, the "more" command will attempt to start displaying the text before the 8540 command has completed. Since the port is still in "raw" mode, the output displayed by the more command is garbled on the screen and the continue feature (space bar) of the "more" command no longer works.

The workaround is to redirect the output of the 8540 command to a file and then use the "more" command on this file.

---

---

## 8086 CALL INSTRUCTION ERROR

### Configuration

asm 8086/87/88/186 V2.04-11 (8560)

### Problem

The assembler incorrectly calculates object code for arguments in a "calls" instruction.

The problem results when the "assume" directive is incorrectly used.

---

---

## 8086 ASM I/O INST. ERROR

### Configuration

8560 8086/87/88/186 Assembler V02.04-11

### Problem

This assembler generates incorrect opcodes for inb, inw, outb, and outw instructions. The problem is a difference in the intel doc iapx 86/88 186/188 Users manual and the asm86 reference manual. The two string operations inb and inw have opcode as 6e and 6f. The outb and outw have opcodes 6c and 6d according to the ref manual. The opcodes are being switched. The assembler needs to be corrected according to the user's manual referenced above.

The next release of the assembler will correct this problem.

---

---

**NO ERROR ON INPAGE JUMPS TO OUTOF PAGE CODE****Configuration**

8560 Tnix V2.0, 8048 ASM V2.04-08 LINKER V2.08-00

**Problem**

The linker does not produce an error when an in-page jump instruction (i.e. jnc) destination address is resolved to an address which is outside the current page.

---

---

**68000 C COMPILER NAMED MODULES****Configuration**

VAX AND UNIX Versions

**Problem**

All separately compiled modules are given the same name, "c.", rather than appending a unique name. This makes it impossible to combine modules into a library file with libgen. A tedious work-around is possible by editing the .s source file from the compiler and adding unique names before assembling.

This will be fixed this summer with Version 2.

---

---

**ASM FAILS IN MAKEFILE****Configuration**

VAX UNIX, version number unknown but recent (2/28/84)

**Problem**

Attempts to use the assembler from a makefile fail, apparently due to an error-code return from the call to the assembler driver. The command

```
asm -b obj lstg src
```

works outside a makefile but fails within it.

The assembler driver doesn't exit correctly. This problem corrected on VMS and will be corrected on unix 4.2 version.

---

---

**8550 ASSEMBLER FILE ATTRIBUTES****Configuration**

8550 DOS-50 version 2.1 with any B series assembler

**Problem**

With B-series assemblers, the assembler does not issue a message if the object and/or listing files are not writable by the current user, and it does not change the contents of these files. This problem requires a correction in the operating system for the language processor. This is a problem for any software that executes on the language processor (A series assemblers, EDIT, ACE, MDL...) This problem is not an assembler bug, it is an acknowledged dos-50 bug.

---

---

**SETKSH CREATED FILE ERROR****Configuration**

8560, 8561, or 8562 with version 2.1 TNIX specifically, problem is in version 1.0 of the setksh command

**Problem**

The setksh command creates several files in the users \$HOME directory which are called and executed via .profile on login. One of these files is '.setTEK'. Within this file is the command 'stty IU >/dev/ttyx'. If the user does not actually have an IU on port x, or if it is turned off when he logs in, his login hangs.

Once in this situation, the user must attach and/or power on the 8540 or reboot the 8560 in order to proceed. Possible workarounds are:

- 1) Manually remove the offending stty command line from the .setTEK file. This will work for an 8540 IU, but not for an 8550 IU. Subsequent setKSH commands, however, will recreate the .setTEK file.
  - 2) Edit out the command which creates the offending line (line #339 in the file /tek/ksh/lib/getiunit). Users using 8550 IUs must then ADD the line to their .setTEK files.
  - 3) Always make sure the 8540 is attached and powered up before logging on.
- 
- 

**SA EDIT OF DFD FILES****Configuration**

LDE V2.02-00(8560), SA tools and Tnix2.1

**Problem**

IF you edit a dfd file using the SA tools it sets up the terminal so that GAMODE is REPLACE. When you then use LDE the help background is overwritten. This is easily fixed by adding a GAMODE command to the lde init file.

---

---

**NESTED INCLUDES WITH VMS ASSEMBLERS****Configuration**

ASM6809 V02-01-09 VAX/VMS V3.4

**Problem**

Nested include directives are not working properly. Given three files: file1, file2, and file3. In file1, there is an "include file2". In file2, there is an "include file3". File1 does call file2, and file2 does call file3. However, after including file3, the assembler returns to file1 instead of file2.

Nested calls may take place in the last line without error. Nested calls should be avoided until a new version is available.

---

---

**SYMBOL TRUNCATION WITH PDB****Configuration**

VAX VMS

**Problem**

The 68000 Pascal Compiler or linker truncates its symbols to 16 characters. PDB does not do any truncation. So if you have a symbol in your listing THISIS\_A\_VERYVERYLONGSYMBOL and you want to reference it in PDB you have to type exactly THISIS\_A\_VERYVER , otherwise you get an unknown symbol error.

---

---

**NO ESCAPE FROM PDB MODIFY****Configuration**

VAX VMS

**Problem**

There is no way to escape from the PDB modify command. This is a real problem if you are modifying a large array. Control-C throws you out of PDB.

---

---

**WRITELN FUNCTION REQUIRES 8540 TERMINAL****Configuration**

VAX VMS

**Problem**

When a Pascal program is using a writeln and being executed through PDB the writeln will not output anything unless a terminal is connected to the 8540. The terminal on the 8540 does not need to be logged on it just needs to be physically connected to the 8540.

---

---

**PRODUCT PERFORMANCE REPORT**

**PRODUCT NOMENCLATURE:** Enter the product description and/or order name, i.e. "ACEDIT OPT. 1A"

**SYSTEM CONFIGURATION AND VERSION NUMBERS:** Include version numbers for all involved products and operating system.

**PROBLEM DESCRIPTION:** Include source, results obtained, and results expected. Please submit the minimum source code required to demonstrate the problem. Complete documentation will enable us to duplicate the problem.

**REPORTED BY:**

Customer Name \_\_\_\_\_ Date \_\_\_\_\_

Company Name \_\_\_\_\_ Title \_\_\_\_\_

Company Address \_\_\_\_\_

Internal Address/Dept \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_

Area code \_\_\_\_\_ Tel. No. \_\_\_\_\_ Ext. \_\_\_\_\_

Send to:  
 MDP Technical Support Manager  
 Tektronix Inc  
 Del. Station 92-635  
 P.O. Box 4600  
 Beaverton, Oregon 97075  
 or if you have access to USENET  
 {uw-beaver,zentel,decvax,...}!tektronix!tekmdp!mdpbug



## *THIRD PARTY SOFTWARE*

---

### **JOINT MARKETING AGREEMENT WITH HUNTER & READY**

Tektronix and Hunter & Ready have signed a cooperative marketing agreement to promote selected products mutually. The Integrated Solutions Agreement identifies three system solutions that Tektronix and H&R will market cooperatively. These are major 16-bit development systems, military micro development systems, and Intel-replacement development systems. MDP and H&R will work together to identify market requirements, plan and develop systems solutions, and market integrated systems in each of these three areas.

#### **16-bit Development Systems**

The primary components of this system are Tek's V8086 and V68000 with PLANDS or CLANDS options and Hunter & Ready's VRTX for 8086 and 68000 families. The initial development effort focuses on the PLANDS 68000 compatibility with VRTX (compiler interface library and run-time library support).

#### **Military Micro Development Systems**

This system supports 1750 development. Principal components are Tek's V1750A, PSS's Jovial 1750, and H&R's VRTX 1750. We are negotiating a cooperative marketing agreement with PSS to complete the system.

#### **Intel-Replacement Systems**

The system provides a complete replacement for Intel 8086 development system. The principal components are Tek's V8086, CFG's PLM 8086, SSI's Asm/Link 186, and H&R's VRTX 8086. We are negotiating cooperative marketing agreements with CFG and SSI to complete the system.

Rodney Bell      Product Line Manager

---



## *USER GROUP LIBRARY ABSTRACTS*

---

---

### **MUGL - MDP USER GROUP LIBRARY**

The MDP User Group Library (MUGL) is provided as a service to MDP users for collecting and distributing user-contributed software for all Tektronix Microcomputer Development Products. Copies are available from your Tektronix Applications Engineer. All software is provided with the understanding that there will be no Tektronix support. Many application engineers and customers have created some useful programs which may benefit others. The program works like this:

- All users are encouraged to submit their creations to MUGL, MDP Marketing, PO Box 4600, 92-635, Beaverton, OR 97075. All submissions will be considered and are made with the understanding that the software may be placed in the public domain. Please don't send your only copy, as we are unable to return any submissions, whether accepted or not. For your convenience, a software submission form is included in this issue and on each MUGL disk volume. We must have the author's name to consider a submission, but we will withhold it if you prefer not to be contacted by anyone.
- We will generally check out the programs, but no guarantees of any kind can be made. We prefer to have the object, source, documentation, and manual page (as applicable) submitted on a floppy disk, but we'll take whatever you have.
- All accepted programs will be archived in MDP Marketing and as soon as we have enough to reasonably fill a disk, a new volume will be released. We will not be able to supply any copies, other than those sent to the local sales offices.
- New submissions will be announced in User Group News in the MUGL abstracts section and a complete library summary will be published periodically.

There are currently five volumes available for release. Contact your local sales or application engineer for details. I'm sure many other useful programs are out there, so let's send 'em in!

Gordon Glather    Software Support Manager

---

**CALEN - APPOINTMENT CALENDAR MAINTAINER**Utility  
856XMUGL TNIX Vol V  
C/Lex/Shell**Abstract**

Another appointment reminder program, offering more versatility than the standard calendar program. The directory contains sources for two programs; "calend" - an appointment calendar maintainer and "remind" - a program that reminds you to log off the system and why.

Calend is a program that takes appointments specified in a .calrc and notifies the user about them when the user wants to be notified. An example .calrc is as follows:

```
: print out a message about bill's party on July 21 the week
: before it; delete when done.
ad *-7 7/21 Party at bill's on %2
: tell me by mail the week before the monthly rent is due.
m *-7 {jan - dec} 1 %m2's Rent is due
: remind me to get off the system for math class every mon wed and fri at 2.
r {mon wed fri} * 1400 Go to math class
: the week before its due, tell me about my insurance payment once
1 *-7 {jan 12, apr 14, sept 15, dec 10} insurance payment by %2
: tell me on the next payday, and then move the message forward two weeks.
: (i.e. tell me about bi-weekly paydays)
a+ sept 23 * +14 Payday today
```

In addition, there is a shell script here called 'addcal' which makes it a little easier to add an entry to your .calrc.

The remind program nags you to get off the system at some time. An example call is:

```
remind 1830 'go home for dinner'
```

which will nag you to get off the system for dinner at 6:30pm. This program is a modification of the 'leave' program.

Author: Bruce Israel, (distributed in net.sources)

**FORTH - UNIX FORTH**Language Compiler  
856XMUGL TNIX Vol V  
PDP-11 asm/Forth**Abstract**

Unix-FORTH is a subset of FORTH-83 adapted for use with UNIX. This adaptation takes advantage of Unix's I/O structure. Includes full PDP-11 assembler source for a minimal bootstrap Forth compiler, which in turn is used to compile the Forth source for the full Forth compiler. Copyright 1984 by the Johns Hopkins University/Applied Physics Lab, Laurel, Maryland. Free non-commercial distribution encouraged, provided that: 1 - This copyright notice is included in any distribution, and 2 - You let JHU/Applied Physics Lab know you're using it.

Author: John R. Hayes, JHU/Applied Physics Lab, MD  
(Distributed via net.sources)

**KERMIT - FILE TRANSFER UTILITY**File Transfer Utility  
Various host O/SMUGL TNIX Vol V  
C**Abstract**

This is another adaptation of the Kermit file transfer program, implemented by MDP Design Engineering for the 8560. This public domain file transfer utility allows transfer of files between any two systems having their own version of Kermit. File transfers can be made via modem or direct connection between the computers. Kermit utilizes error checking with re-try when a checksum error is detected. Versions presently exist for Apple, IBM-PC, Heath/Zenith H-89/Z-100, IBM Mainframes, VAX/UNIX, and other systems. Kermit originated at Columbia University, who encourages distributing and adapting the program for other hosts. Includes fully commented C source, manual page, and additional documentation.

Original Author: Bill Catchings, Columbia University  
TNIX Implementation: Keith Rule, MDP Design Engineering

**OBJXREF - OBJECT CROSS REFERENCER**Cross Reference  
856XMUGL TNIX Vol V  
C**Abstract**

A version of "objxref" for the 8560. For usage instructions, see the Technical Notes section of the B-Series Assembler Core Manual. Executable binary image only.

**PROGRAMNAME-SUMMARY**Conversion Utility  
856XMUGL TNIX Vol V  
C**Abstract**

A program which produces a mapping from arbitrary-length identifiers to identifiers which are unambiguous in the first N (default 7) characters. It produces them as #defines; all it takes is a flexnames version of cpp to compile the original source files with a non-flexname compiler, with no modification beyond including the output of shortc into the sources. If such a cpp is not available or creatable (not hard given any cpp source), the shortc output can be turned into a sed script and the sources can be compiled after being modified. Such modification does not fill the sources with identifiers like X12345 or MuWdldr (for MultiWordIdentifier); rather it converts

```
MultiWordIdentifier      MultiWordIdentifier
MultiWordThingy         into  AMultiWordThingy
MultiWordyProgrammer    BMultiWordyProgrammer
```

Author: Jim Balter, Interactive Systems Corp  
(from net.sources)

**UNHEX - TEKHEX TO TEKLAS OBJECT CONVERTER**Converter utility  
856XMUGL TNIX Vol V  
C**Abstract**

Unhex generates an LAS binary file from an extended Tekhex file. The binary file is written into a file with ".bin" appended to the hexfile name.

Unhex was written to support the IBM PC marketing program. It is intended to be used with the Microtec assemblers. These assemblers generate the uP vendor's binary format. Microtec provides a converter from this format to Tekhex. The use of unhex to then convert to the LAS binary format will allow faster binary download

Note that the LAS binary file which unhex generates is only suitable for downloads with the "lo" and "symlo" 8540 commands. The file CANNOT be used as input to the LAS linker, lstr, libgen, or objxref. The reason for this is that several shortcuts were taken in records and fields of records that "lo" and "symlo" do not use. In many cases, 0's were written into fields. The blocks also come out in the same order as they are seen in the hex file. This restriction should pose no problem for users since the LAS utilities will not be ported to the IBM PC.

Author: Cindy McMeekin, MDP Marketing

---

**UNHEX - TEKHEX TO TEKLAS OBJECT CONVERTER**System Utility  
856XMUGL TNIX Vol V  
C**Abstract**

This utility is a replacement for the default /etc/update which permits a user selected variable cycle period. By specifying a longer period (5 minute cycle recommended) rather than the default 30 second period, system performance can be improved somewhat.

Author: Stefan Stapelberg, Mannesmann Rexroth GmbH, Germany

---

**Z8 ASSEMBLER**Assembler  
856XMUGL TNIX Vol V  
Tek LASASM Macros**Abstract**

A set of Z8 macros configured for use with the user-configurable null assembler which was previously distributed in MUGL. Includes sources, test file, convenient shell script driver and usage instructions.

Author: Steve Beard, TRW, Redondo Beach, CA

---

**MDP USER GROUP SOFTWARE LIBRARY/ARTICLE SUBMITTAL FORM**

The following form may be used to submit software which you feel might be of interest to other MDP users.

The form and the program(s) should be forwarded to:

Technical Support Manager  
Tektronix Inc.  
P.O. 4600  
Del Sta. 92-635  
Beaverton Or.  
97075

or if USENET is available:

{uv-beaver,zentel,decvax,...}!tektronix!tekmdp!mdpbug

**MDP USER'S GROUP SOFTWARE LIBRARY/ARTICLE SUBMITTAL FORM**

1. ABSTRACT. \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

2. Execution CPU \_\_\_\_\_ Primary Language \_\_\_\_\_  
Hardware configuration required \_\_\_\_\_  
\_\_\_\_\_

Software configuration required (include source if non-Tek) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

3. \_\_\_\_\_ Do you want the following to appear in U.G.N.

Author's name \_\_\_\_\_ O yes \_\_\_\_\_ O no

Company Name \_\_\_\_\_ O yes \_\_\_\_\_ O no

Area code \_\_\_\_\_ Tel. No. \_\_\_\_\_ O yes \_\_\_\_\_ O no

Company address \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Program Title \_\_\_\_\_

Program Function \_\_\_\_\_

5. Source. If insufficient room is provided, please submit a disk (containing the information requested) attached to this form.

6. I am submitting the program/article described above for possible placement in the MDP User's Group Library. I understand there is no compensation due to me for an accepted program/article. This program/article is of my own design; the data contained in this submittal is not copyrighted and does not break any obligation to another person or organization relating to proprietary or confidential information. Tektronix, Inc. is authorized to distribute (free of charge on customer supplied media) or publish copies of this program to Tektronix MDP users.

Signature \_\_\_\_\_ Date \_\_\_\_\_

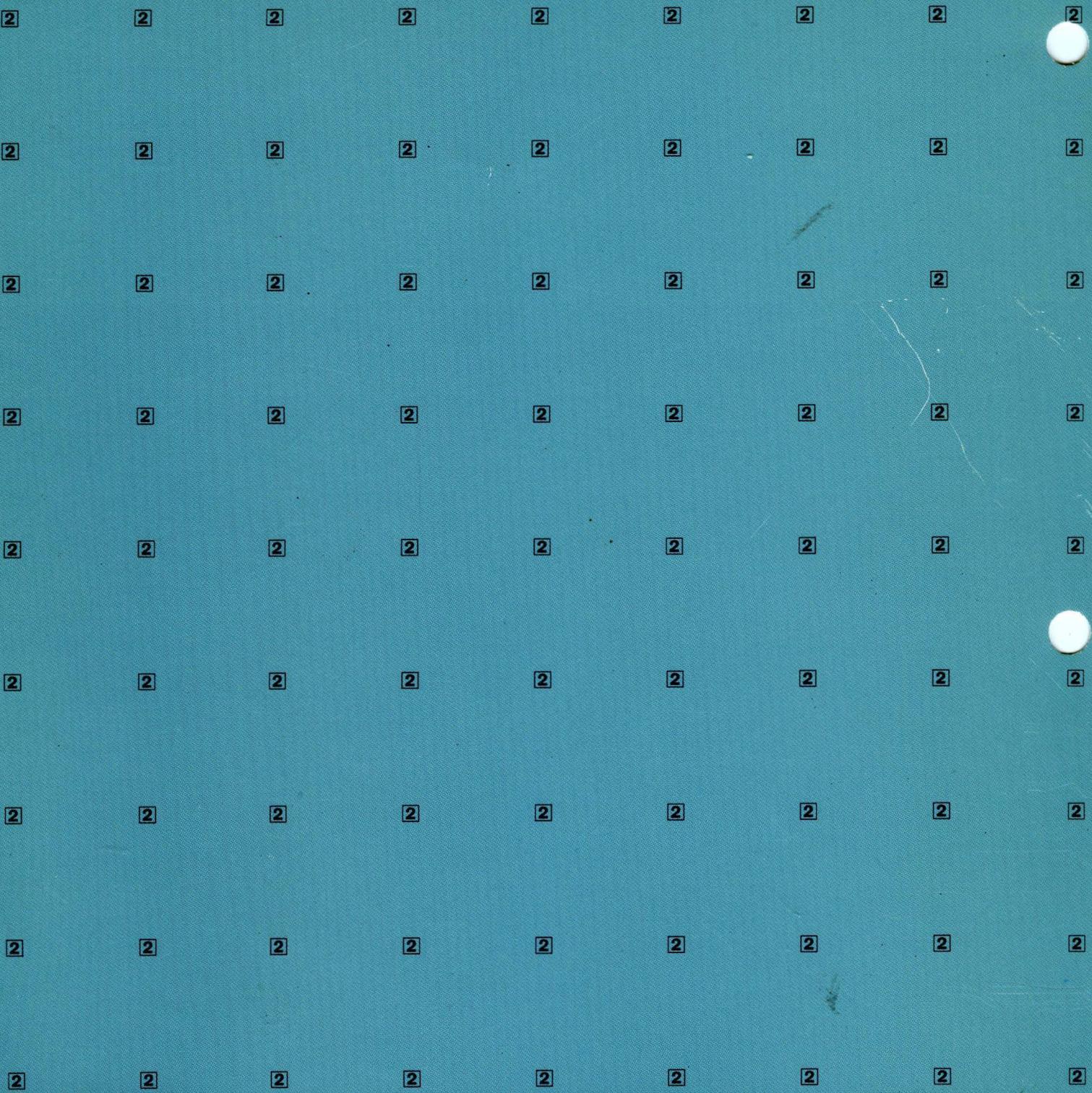
## INDEX OF VOLUME 3

Title	Section	Page
1750A		
MIL-STD 1750A SUPPORT FROM TEKTRONIX.....	ProdInfo	54
4105		
KSH ERROR WITH VER 3.0 4105 .....	ProdPerf	27
CONVKB - DVORAK CONFIGURATION FOR THE 4105 TERMINAL.....	Abstract	40
TABA4105 - 4105 NROFF TABA FILE.....	Abstract	46
68000		
PIN-GRID ADAPTER FOR 68000 PROBES .....	Applications	19
DISPOSE ERROR IN 68000 PASCAL.....	ProdPerf	26
68000 PASCAL AND LIST -L OPTION.....	ProdPerf	27
68000/68XXX UPGRADES AND CONVERSION PACKAGES .....	Applications	57
EMULATING THE OMNIBYTE OB68KA1 WITH 68XXX .....	Applications	72
68000 PROTOTYPE CONTROL PROBE ENHANCEMENT.....	Applications	73
68000 C COMPILER NAMED MODULES .....	ProdPerf	84
6801		
6801 WAI INSTRUCTION.....	Applications	19
6809		
NESTED INCLUDES WITH VMS ASSEMBLERS .....	ProdPerf	85
8002/1		
8550 PROGRAM MEMORY IN AN 8002 .....	Applications	58
80186		
TEKTRONIX SUPPORTS 80186/80188.....	ProdInfo	9
8086 PASCAL VERSION 2 SUPPORTS 80186.....	ProdInfo	9
V80186 AND V80188 SUPPORT SYSTEMS.....	ProdInfo	49
80188		
TEKTRONIX SUPPORTS 80186/80188.....	ProdInfo	9
V80186 AND V80188 SUPPORT SYSTEMS.....	ProdInfo	49
8048		
8048 IN PAGE JUMP INSTRUCTION.....	ProdPerf	80
NO ERROR ON INPAGE JUMPS TO OUTF PAGE CODE.....	ProdPerf	84
8085		
INTEL-COMPATIBLE PLM 8085 DEVELOPMENT SYSTEMS.....	TPSW	33
8086		
8086 PASCAL VERSION 2 SUPPORTS 80186.....	ProdInfo	9
I/O INSTRUCTION ERROR IN 8086 ASSEMBLER.....	ProdPerf	26
8086 PASCAL PORT ADDRESS LIMITATION.....	ProdPerf	28
8086 PASCAL ERROR IN OPTION D.....	ProdPerf	30
8550 8086 ASSEMBLER TRUNCATION ERROR.....	ProdPerf	31
INTEL-COMPATIBLE PLM 8086 DEVELOPMENT SYSTEMS.....	TPSW	35
NEW SOFTWARE REFERRALS.....	TPSW	37
CMOS 8086 & OTHER CMOS CHIP SUPPORT.....	ProdInfo	51
8086 CALL INSTRUCTION ERROR.....	ProdPerf	83
8086 ASM I/O INST. ERROR.....	ProdPerf	83
8540		
8540/8560 CONFIG TERM I/O REDIRECTION .....	Applications	59
8550		
8540/8560 CONFIG TERM I/O REDIRECTION .....	Applications	59
8550 ASSEMBLER FILE ATTRIBUTES .....	ProdPerf	84
8560		
8562 - LSI-11/73 BASED SOFTWARE DEVELOPMENT SYSTEM.....	ProdInfo	50
CVT, THE TNIX KERNAL CONFIGURATOR, REVISITED .....	Applications	64
8560 PERFORMANCE ISSUES.....	Applications	72
TNIX 2.1 SETKSH ERROR FIX.....	ProdPerf	82
8562		
8562 - LSI-11/73 BASED SOFTWARE DEVELOPMENT SYSTEM.....	ProdInfo	50
ACE		
ACECONFIG LIMITATION.....	ProdPerf	28
ACE AND USE OF THE BREAK KEY .....	ProdPerf	30
H10CFG - HEATH/ZENITH H/Z-19 ACECONFIG.....	Abstract	43
ACE/LDE CONFIG FOR ICL6402/KDS7362 TERMINAL.....	Abstract	43
Abstract		
2764PP - 2764 PROM PROGRAMMER SOURCE.....	Abstract	40
CALENDAR - ANOTHER REMINDER SERVICE.....	Abstract	40
CONVKB - DVORAK CONFIGURATION FOR THE 4105 TERMINAL.....	Abstract	40
CUSTOMKEY - CREATE CUSTOM KEY SHELL.....	Abstract	41
DUMPTAB - DE-COMPILE NROFF DRIVE TABLE.....	Abstract	41
ETABS - TAB EXPANSION UTILITY.....	Abstract	41

EXGPB - 8560 GPB INTERACTIVE INTERFACE.....	Abstract.....	42
FDUP - FAST FLOPPY DISK DUPLICATOR FOR 856X.....	Abstract.....	42
GONE - TERMINAL PROTECTOR.....	Abstract.....	42
H1094 - HAZELTINE 1094 ESPRIT LDE CONFIG.....	Abstract.....	43
H10CFG - HEATH/ZENITH H/Z-19 ACECONFIG.....	Abstract.....	43
HEXIFY - CPM.COM TO INTEL HEX CONVERTER.....	Abstract.....	43
ACE/LDE CONFIG FOR ICL6402/KDS7362 TERMINAL.....	Abstract.....	43
KERMIT - FILE TRANSFER UTILITY.....	Abstract.....	44
MACROASM - USER DEFINABLE UTILITY.....	Abstract.....	44
MODEM - CONFIGURE MODEM PORT FOR AUTO-ANSWER/DIALOUT.....	Abstract.....	44
PAGER - PAGINATE INPUT.....	Abstract.....	44
PAMS - PUBLIC ACCESS MESSAGE SYSTEM DATABASE.....	Abstract.....	45
PUSHD - PUSH DIRECTORY.....	Abstract.....	45
SETUID - SET USER ID ON EXECUTION.....	Abstract.....	45
STYLE - STYLISTIC TEXTURAL ANALYSIS.....	Abstract.....	46
TABA4105 - 4105 NROFF TABA FILE.....	Abstract.....	46
VT100X - LDE CONFIG FOR DEC VT-100.....	Abstract.....	46
XBAS - MOTOROLA TO TEK ASM SOURCE CONVERTER.....	Abstract.....	47
XT - TAB EXPANSION FILTER.....	Abstract.....	47
MUGL - MDP USER GROUP LIBRARY.....	Abstract.....	91
CALEN - APPOINTMENT CALENDAR MAINTAINER.....	Abstract.....	92
FORTH - UNIX FORTH.....	Abstract.....	92
KERMIT - FILE TRANSFER UTILITY.....	Abstract.....	93
OBJXREF - OBJECT CROSS REFERENCER.....	Abstract.....	93
PROGRAMNAME-SUMMARY.....	Abstract.....	93
UNHEX - TEKHES TO TEKLAS OBJECT CONVERTER.....	Abstract.....	94
VUPDATE - VARIABLE /ETC/UPDATE.....	Abstract.....	94
Z8 ASSEMBLER.....	Abstract.....	94
<b>Applications</b>		
AN OFTEN-OVERLOOKED RESOURCE IN THE TNX SHELL.....	Applications.....	15
I-NODES ON DUMP DISKETTES.....	Applications.....	18
6801 WAI INSTRUCTION.....	Applications.....	19
PIN-GRID ADAPTER FOR 68000 PROBES.....	Applications.....	19
WHEREIS SCRIPT.....	Applications.....	21
Z8001 EMULATOR OPERATION WITH A 10MHZ CLOCK.....	Applications.....	21
68000/68XXX UPGRADES AND CONVERSION PACKAGES.....	Applications.....	57
8550 PROGRAM MEMORY IN AN 8002.....	Applications.....	58
8540/8560 CONFIG TERM I/O REDIRECTION.....	Applications.....	59
SA TOOLS TERMINAL CONFIG FILES.....	Applications.....	60
STRUCTURED ANALYSIS TOOLS CONFIGURATIONS.....	Applications.....	61
MULTI-DISK FILE BACKUP IS IMPROVED.....	Applications.....	63
CVT, THE TNX KERNAL CONFIGURATOR, REVISITED.....	Applications.....	64
AN OFTEN OVERLOOKED RESOURCE IN THE TNX SHELL.....	Applications.....	68
SPELLING ERRORS CORRECTED SEMI-AUTOMATICALLY.....	Applications.....	70
LINE PRINTERS SUPPORTING X-ON/X-OFF ON THE 8560.....	Applications.....	71
MDL <sub>u</sub> WITH THE NSC800 EMULATOR.....	Applications.....	71
EMULATING THE OMNIBYTE OB68KA1 WITH 68XXX.....	Applications.....	72
OBJXREF AVAILABLE FOR THE 8560.....	Applications.....	72
8560 PERFORMANCE ISSUES.....	Applications.....	72
68000 PROTOTYPE CONTROL PROBE ENHANCEMENT.....	Applications.....	73
EPROM SUPPORT.....	Applications.....	74
RECOVERING ORIGINAL SOURCE FROM ASSEMBLER LISTING.....	Applications.....	75
HELP USING VMS.....	Applications.....	75
Z80 EMULATION RE-VISITED.....	Applications.....	78
Z80 PERIPHERALS AND THE DATA BUS.....	Applications.....	80
<b>Assembler</b>		
I/O INSTRUCTION ERROR IN 8086 ASSEMBLER.....	ProdPerf.....	26
Z80 ASSEMBLER FORWARD REFERENCE PROBLEM.....	ProdPerf.....	28
8048 IN PAGE JUMP INSTRUCTION.....	ProdPerf.....	30
8550 8086 ASSEMBLER TRUNCATION ERROR.....	ProdPerf.....	31
INTEL-COMPATIBLE PLM 8086 DEVELOPMENT SYSTEMS.....	TPSW.....	36
MACROASM - USER DEFINABLE UTILITY.....	Abstract.....	44
RECOVERING ORIGINAL SOURCE FROM ASSEMBLER LISTING.....	Applications.....	75
8086 CALL INSTRUCTION ERROR.....	ProdPerf.....	83
8086 ASM I/O INST. ERROR.....	ProdPerf.....	83
NO ERROR ON INPAGE JUMPS TO OUTOF PAGE CODE.....	ProdPerf.....	84
ASM FAILS IN MAKEFILE.....	ProdPerf.....	84
8550 ASSEMBLER FILE ATTRIBUTES.....	ProdPerf.....	84
OBJXREF - OBJECT CROSS REFERENCER.....	Abstract.....	93
Z8 ASSEMBLER.....	Abstract.....	94
<b>C</b>		
NEW SOFTWARE REFERRALS.....	TPSW.....	36

68000 C COMPILER NAMED MODULES .....	ProdPerf	84
CPM		
HEXFY - CPM.COM TO INTEL HEX CONVERTER .....	Abstract	43
CVT		
CVT, THE TNIX KERNAL CONFIGURATOR, REVISITED .....	Applications	64
Calendar		
CALENDAR - ANOTHER REMINDER SERVICE .....	Abstract	40
CALEN - APPOINTMENT CALENDAR MAINTAINER .....	Abstract	92
F9450		
MIL-STD 1750A SUPPORT FROM TEKTRONIX .....	ProdInfo	64
File Backup		
I-NODES ON DUMP DISKETTES .....	Applications	18
MULTI-DISK FILE BACKUP IS IMPROVED .....	Applications	63
File Transfer		
KERMIT - FILE TRANSFER UTILITY .....	Abstract	44
KERMIT - FILE TRANSFER UTILITY .....	Abstract	93
GPIB		
856X GPIB APPLICATION NOTE AND SUPPORT SOFTWARE .....	ProdInfo	12
EXGPIB - 8560 GPIB INTERACTIVE INTERFACE .....	Abstract	42
Hunter & Ready		
JOINT MARKETING AGREEMENT WITH HUNTER & READY .....	TFSW	89
IBM PC		
NEW SOFTWARE REFERRALS .....	TFSW	36
ICOM40		
NEW VERSION OF ICOM40 FOR VMS .....	ProdInfo	51
ICOM40 AND PIPED OUTPUT .....	ProdPerf	83
KSH		
KSH ERROR WITH VER 3.0 4105 .....	ProdPerf	27
CUSTOMKEY - CREATE CUSTOM KEYSHELL .....	Abstract	41
TNIX 2.1 SETKSH ERROR FIX .....	ProdPerf	82
SETKSH CREATED FILE ERROR .....	ProdPerf	85
LDE		
LDE - UMASK INTERACTION .....	ProdPerf	28
SHELL COMMANDS IN LDE .....	ProdPerf	29
LONG TEXT LINES IN LDE .....	ProdPerf	29
H1094 - HAZELTINE 1094 ESPRIT LDE CONFIG .....	Abstract	43
ACE/LDE CONFIG FOR ICL0402/KDS7362 TERMINAL .....	Abstract	43
VT100X - LDE CONFIG FOR DEC VT-100 .....	Abstract	46
LSI-11/73		
LSI-11/73 NOW AVAILABLE ON 856X SERIES .....	ProdInfo	9
Line Printers		
LINE PRINTERS SUPPORTING X-ON/X-OFF ON THE 8560 .....	Applications	71
Linker		
8560 LINKER SYMBOL TABLE .....	ProdPerf	24
Load		
DASH MODIFIER OF LOAD COMMAND .....	ProdPerf	26
MDLu		
MDLu AND NEGATIVE ADDRESS ORIGIN .....	ProdPerf	25
MDLu WITH THE NSC800 EMULATOR .....	Applications	71
Make		
ASM FAILS IN MAKEFILE .....	ProdPerf	84
Manuals		
NEW MANUALS .....	ProdInfo	13
NEW MANUALS .....	ProdInfo	52
NSC800		
MDLu WITH THE NSC800 EMULATOR .....	Applications	71
Nroff		
DUMPTAB - DE-COMPILE NROFF DRIVE TABLE .....	Abstract	41
PASCAL		
SYMBOL TRUNCATION WITH PDB .....	ProdPerf	86
PDB		
SYMBOL TRUNCATION WITH PDB .....	ProdPerf	86
NO ESCAPE FROM PDB MODIFY .....	ProdPerf	86
WRITELN FUNCTION REQUIRES 8640 TERMINAL .....	ProdPerf	86
PLM		
INTEL-COMPATIBLE PLM 8085 DEVELOPMENT SYSTEMS .....	TFSW	33
PROMS		
EPROM SUPPORT .....	Applications	74
Pascal		
8086 PASCAL VERSION 2 SUPPORTS 80186 .....	ProdInfo	9
CASE STATEMENT ERROR IN Z8000 PASCAL .....	ProdPerf	25
DISPOCE ERROR IN 68000 PASCAL .....	ProdPerf	26

68000 PASCAL AND LIST -L OPTION.....	ProdPerf	27
8086 PASCAL PORT ADDRESS LIMITATION.....	ProdPerf	28
8086 PASCAL ERROR IN OPTION D.....	ProdPerf	30
PASCAL LSTR ERROR ON MULTI-DIMENSIONED ARRAYS.....	ProdPerf	30
Performance		
8560 PERFORMANCE ISSUES.....	Applications	72
From Programmer		
2764PP - 2764 FROM PROGRAMMER SOURCE.....	Abstract	40
SA Tools		
STRUCTURED ANALYSIS TOOLS ANNOUNCEMENT.....	ProdInfo	1
SA TOOLS TERMINAL CONFIG FILES.....	Applications	60
STRUCTURED ANALYSIS TOOLS CONFIGURATIONS.....	Applications	61
SA TOOLS & FLOATING POINT PROBLEM.....	ProdPerf	81
SA EDIT OF DFD FILES.....	ProdPerf	85
SPR9000		
MDP SUPPORT OF THE TI SPR9000 SERIES.....	ProdInfo	63
Software		
SOFTWARE REGISTRATION CARDS NEEDED.....	ProdInfo	12
NEW SOFTWARE RELEASES.....	ProdInfo	13
NEW SOFTWARE RELEASES.....	ProdInfo	63
Spell		
SPELLING ERRORS CORRECTED SEMI-AUTOMATICALLY.....	Applications	70
Symlo		
INVALID SYMLO FILE LOAD.....	ProdPerf	29
System Calls		
USE OF SYSTEM ACCOUNTING CALLS IN TNIX.....	ProdPerf	23
TNIX		
TNIX TIME SYSTEM CALL.....	ProdPerf	27
AN OFTEN OVERLOOKED RESOURCE IN THE TNIX SHELL.....	Applications	68
TNIX 2.1 SETKSH ERROR FIX.....	ProdPerf	82
TTA		
NOT OF ADDRESS FAILS WITH TTA HLP.....	ProdPerf	26
Tekhex		
UNHEX - TEKHEX TO TEKLAS OBJECT CONVERTER.....	Abstract	94
UUCP		
UNICOM FILE SIZE LIMITATION.....	ProdPerf	82
User Group Library		
MUGL - MDP USER GROUP LIBRARY.....	Abstract	91
VMS		
NEW VERSION OF ICOM40 FOR VMS.....	ProdInfo	51
HELP USING VMS.....	Applications	75
VRTX		
JOINT MARKETING AGREEMENT WITH HUNTER & READY.....	TPSW	89
Whex		
WHEX OMITTS CHECKSUM IN TERM BLOCK.....	ProdPerf	29
Z8		
Z8 ASSEMBLER.....	Abstract	94
Z80		
Z80 ASSEMBLER FORWARD REFERENCE PROBLEM.....	ProdPerf	28
NEW SOFTWARE REFERRALS.....	TPSW	37
Z80 EMULATION RE-VISITED.....	Applications	78
Z80 PERIPHERALS AND THE DATA BUS.....	Applications	80
Z8000		
CASE STATEMENT ERROR IN Z8000 PASCAL.....	ProdPerf	25
Z8001		
Z8001 EMULATOR OPERATION WITH A 10MHZ CLOCK.....	Applications	21



**For further information, contact:**

**U.S.A., Asia, Australia, Central & South America, Japan**

Tektronix, Inc.  
P.O. Box 1700  
Beaverton, Oregon 97075  
For additional literature, or the address and phone number of the Tektronix Sales Office nearest you, contact:  
Phone: (800) 547-1512  
Oregon only: (800) 452-1877  
TWX: (910) 467-8708  
TLX: 151754  
Cable: TEKWSGT

**Europe, Africa, Middle East**

Tektronix Europe B.V.  
European Headquarters  
Postbox 827  
1180 AV Amstelveen  
The Netherlands  
Phone: (20) 471146  
Telex: 18312 - 18328

**Canada**

Tektronix Canada Inc.  
P.O. Box 6500  
Barrie, Ontario L4M 4V3  
Phone: (705) 737-2700

**Tektronix**<sup>®</sup>  
COMMITTED TO EXCELLENCE