TEK MICROCOMPUTER
DEVELOPMENT PRODUCTS

USER NEWSLETTER
VOLUME II, ISSUE NO. 1, JUNE, 1983

USER GROUP NEWS

```
walklength = 30;(number of seconds to wait during the walk sig.)
dontwalklength = 10;(number of secs to wait during the dontwalk)
greenlength = 45;(number of secs to force green for Maple St.)

YPE
  color = (green, yellow, red);(color for a Maple Street signal)
  message = (walk, dontwalk);(message for crosswalk signal )

ROCEDURE wait( seconds : integer ); EXTERN;
        { waits for "seconds" seconds, then returns }

UNCTION pedestrian : boolean; EXTERN;
        { returns "true" if pedestrian pushed the walk button }

ROCEDURE maple( newcolor : color ); PUBLIC;
        { changes Maple Street light to "newcolor" }

  BEGIN
    write('Changing Maple Street signal to ');
    CASE newcolor OF
      green :
        write('green');
      yellow :
        write('yellow');
      red :
-More--(63%)[Hit space to continue, Rubout to abort]
ee the file to be shown: demo.ps
```

Tektronix®
COMMITTED TO EXCELLENCE

CONTENTS

PRODUCT INFORMATION SECTION

SOFTWARE SUBSCRIPTION SERVICE

Tektronix is offering a new program for supporting our software products, "THE SOFTWARE SUBSCRIPTION SERVICE". With this new service you will always be assured of having the latest release of a software product, be guaranteed a prompt written response to any problems reported, and never be bothered again by unplanned purchases of software updates.

This service will be provided free of charge for the warranty period with any new product purchased, and to any customers currently under warranty for the remainder of their warranty period. Thereafter, for a minimal charge you will be supported for an entire year with free updates of software and other support services.

Some of the additional services provided with the Software Service will be: a guaranteed written response to any comment or problem submitted via a Software Performance Report, continuing copies of User Group News (the first several issues will be sent to all customers free of charge), and preferential scheduling of local Application Engineering time.

Altogether, this adds up to a support package that you won't want to do without. For more information on Tektronix's Software Subscription Service, contact your local Tektronix sales or applications person.

Doug Johnson MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~

A NEW USER INTERFACE

There has been a lot of attention on user interfacing in the development systems market lately . This is because of the user's situation - you want a development system that is easy to use. Ease of use translates to less learning time and more productivity. However, a simple user interface that does not recognize how people work can become an impediment to user productivity after it has served its initial purpose of immediate operation of the development system.

TEKTRONIX IS INTRODUCING A NEW USER INTERFACE

Listed below are some of the features that make the new interface convenient and flexible.

- **Push button access to commonly used system commands**

  The interface is a way to select areas of operation and to execute commands with single keystrokes using eight preprogrammable function keys. The new interface is like soft keys. However, it is a significant improvement on a good idea. **TEK OFFERS YOU AN EASY WAY TO USE YOUR DEVELOPMENT SYSTEM.** You have the convenience of push-button operation of your development system without compromising direct access to TNIX commands.

- **Tree-structured menus with task-oriented branches**

  The interface is a multi-level interface. Each level represents a set of choices either for selecting a task to do or a command to execute. The first level is organized to support the design cycle (setup, design, code and integrate) and common user tasks (files, system info and mail). Selection of one of these branches leads the user to single keystroke-selected TNIX commands that will help him complete tasks in that area. **THE INTERFACE IS ORGANIZED TO HELP YOU COMPLETE YOUR TASKS.**

- **Color display with Tek's 4105M Computer Display Terminal**

  With the 4105 the new interface comes to you in color. The intelligent use of color in an interface enhances user productivity and reduces user fatigue and boredom. The Interface colors were selected for optimum visual performance. **YOU HAVE THE ADVANTAGE OF COLOR TO ENHANCE USER PRODUCTIVITY.**

- **Color**

  A First in the development systems market! **COLOR HAS BEEN PROVEN TO IMPROVE HUMAN UNDERSTANDING OF DATA. THE NEW INTERFACE COLORS WERE CHOSEN BASED ON SOUND PHYSIOLOGICAL DATA IMPROVE PRODUCTIVITY AND REDUCE USER FATIGUE.** The colors were selected with the help of a perceptual psychologist.

White on Blue	User Area
Black on Yellow	Current Key Labels
White on Rust	Interface Message Area
White on Red	TNIX Messages

- **Intelligent Keys that learn your optional parameters**

  The interface learns optional parameters that are entered by the user and puts them on a function key for the next use. **THE NEW INTERFACE REDUCES THE NUMBER OF KEYSTROKES REQUIRED, WHICH REDUCES YOUR ERRORS AND INCREASES YOUR PRODUCTIVITY. THE INTERFACE ADAPTS TO YOU. AS YOU WORK IT LEARNS WHAT YOU DO AND GROWS TO INCREASE YOUR PRODUCTIVITY.** The interface contains a capability to save sessions from one time to the next. You can invoke the interface using the setup keys available in the interface to recall a particular session. When you do this the keys will be just as you left them the last time you used the interface.

- **Editable History**

  Using the static keys you can scroll previously executed commands through the command window. Command history gives you quick access to all recent commands. When you reach a command that you want to use again with different parameters you can edit the command line and re-execute it. **MOST COMMANDS ARE REPEATED MANY TIMES WITH SMALL VARIATIONS.** Use of the history scrolling keys brings up a command line editor on the function keys for convenience in editing the command line. **RECALL AND EDIT OF PREVIOUSLY USED COMMAND SEQUENCES SAVES TIME AND NEEDLESS ERRORS.**

- **Instant Mapping**

  Instant Mapping shows a user where he is in the interface. **NEW USERS CAN BE INTIMIDATED BY MULTI-LEVEL INTERFACES. THE WHEREAMI KEY GIVES YOU AN INSTANT MAP SO YOU ARE NEVER LOST IN THE INTERFACE.** Often you can get deep into a selection sequence and either forget what you are doing or, more likely, be interrupted and lose your place. The interface can give the user an *Instant Map* of the most recent the levels that you have gone through to get where you are and your selections at each level. **The whereami static key gives the user an instant context of what he is doing.**

- **Totally Transparent**

  As the user pushes the function buttons to execute system commands, the interface displays the command on the command line. Over time, this will have the effect of educating the user in the operation of the system. Eventually the user will learn very well the commands that he most commonly uses. **THE INTERFACE TEACHES YOU TO OPERATE THE SYSTEM.** At a certain point the user will want to directly enter commands and by-pass the interface. This is totally supported by the interface at any time the user can directly enter commands to TNIX and execute them. **YOU ARE NOT RESTRICTED TO USING JUST THE NEW INTERFACE.** Initially you will execute commands that are frequently used. Eventually you will interface directly and use the interface only as a convenience or to guide yourself in areas of the operating system that you have not yet learned. **THE INTERFACE IS ALWAYS AVAILABLE AS A CONVENIENCE. YOU FIND THE NATURAL BALANCE BETWEEN THE INTERFACE AND DIRECT ENTRY.** This allows the user to find the optimum balance between structured operation via the interface and free form operation that will develop.

- **The interface is available for 8560 series systems.**

John Owens MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

UNICOM

Intersystems Communications with 8560s and UNIX * Systems

* Expand/Unify Your Development Systems
* Communicate with 8560 or UNIX Systems
* Transfer Software and Documents
* Execute Commands on Other Systems
> Send Mail to Other Systems' Users
> Access Remote Systems Transparently
> Intersystems Communications Network

```
 -------------------------------------
|  ________                ________   |
| |A       |              |D       |  |
| |  8560  |--------------|  UNIX  |  |
| |________|     RS232    |________|  |
|     | RS232                         |
|  ___|____                ________   |
| |B       |              |E       |  |
| |  8560  |>>>>>>>>>>>>>>|  UNIX  |  |
| |________| > phone      |________|  |
|     || HSI  >  lines                |
|  ___||___     >          ________   |
| |C       |      >       |F       |  |
| |  8560  |       >>>>>>>|  8560  |  |
| |________|              |________|  |
|-------------------------------------|
```

Figure 1. UNICOM Intersystems Communications Network

* UNIX and UNIX System III are trademarks of Bell Labs
~~~~~~~~~~~~~~~~~~~~

UNICOM (UNIx COMmunications) is a software option to the Tektronix 8560. It enables 8560s and UNIX systems to be linked in an intersystem communication network. UNICOM provides enhanced versions of the UNIX Systems III * commands, *uucp*, *uux*, and *cu*. Bell Labs developed *uucp* and *uux* to distribute and exchange UNIX software over a network of 80 systems connected mainly by phone. Now, UNICOM allows 8560 customers to expand their development systems facility.

UNICOM offers an inexpensive, reliable, and convenient communication facility for situations like these. You're adding 8560s or UNIX systems to expand your design facility. You want your 8560 to communicate with the remote system of a client, supplier, or field group. You need to unify your design environment by communicating among existing systems. With UNICOM on your 8560s, you can organize a multi-system design facility for large or related projects.

Consider UNICOM compared to other network solutions and to unconnected systems. UNICOM is much less expensive than networks (e.g., Local Area Networks) that use special transmission facilities. Compared to unconnected systems, UNICOM offers benefits in productivity, equipment utilization, and project control. You can increase productivity through rapid communications, distribution of software tools, or by partitioning design tasks among connected systems. You can improve equipment utilization by sharing expensive peripherals (e.g., like a tape drive) from connected systems. You can improve project control by using intersystems mail, maintaining a central project library, and providing rapid access to specifications. If you're expanding or unifying your 8560 development facility, UNICOM is a valuable package.

UNICOM provides basic intersystem communication functions and uses common transmission facilities. UNICOM provides for transfering files between systems, sending commands to execute on another system, sending mail to users on other systems, and accessing remote systems transparently. Intersystem links can be RS232 cable or modem/telephone lines.

- Communications Between 8560s.

  With UNICOM installed on each 8560, connected 8560s can communicate. UNICOM communications works best with directly connected 8560s (eg, A to B in Fig 1). Between indirectly connected 8560s (eg, A to C in Fig 1) communication requires intermediate transfers.

- Communicate with UNIX Systems.

  UNICOM lets an 8560 communicate with any UNIX host that has UUCP. Communication functions with UNIX are the same as with an 8560.

- UUCP Configuration

  UNICOM provides UUCP-CONFIG, an enhancement to UNIX versions of UUCP that simplifies UUCP setup. This interactive program prompts the system administrator for information and then automatically creates directories, edits files, and installs commands.

- Send Files, Commands, and Mail.

  UUCP transfers files between connected systems. Transfers are checked for transmission errors. UUX executes commands sent from a connected system. With UUX a command's input can come from or output can go to other systems. MAIL allows users to send mail to other users on any system in the network. UNICOM's MAIL replaces the one that comes with the 8560.

- Login to Remote Systems.

  CU connects an 8560 user to a remote system. The 8560 becomes transparent to the user's dialog with the remote system. CU can transfer ASCII files immediately between the 8560 and the remote system, but without error checking.

- Create an Intersystems Communications Network.

  Various configurations can be implemented to suit the application. Systems can be connected by RS232 cable or telephone. Some functions work over HSI between 8560s, but RS232 connection is recommended. For telephone access, UNICOM works with two auto-dialers, Racal-Vadic 3451 with Autodial Option and Hayes Smart Modem (300 baud version). Figure 1 shows an example of a UNICOM network.

UNICOM REQUIREMENTS.

To configure a UNICOM intersystems communications network, please note:

- UNICOM works with any version of TNIX on any 8560.
- UNICOM communicates with
  (1) another 8560 that has UNICOM installed,
  (2) a UNIX system that has UUCP and UUX installed, or
  (3) possibly other systems (via CU only).
- Each 8560 in the network must have UNICOM installed.

Contact your Tektronix salesperson about

- interconnection hardware like cables, modems, and 8560 ports
- network configuration alternatives and guidelines

ORDERING INFORMATION

The order number for UNICOM is 8560U05. Extra manuals are available (070-4636-00). Contact Tektronix sales for price and availability.

WARRANTY, SUPPORT, AND LICENSING

UNICOM is a Category C product. There is no warranty; the software subscription service is not offered. UNICOM is furnished under the Tektronix Software License Agreement. A SEPARATE SOFTWARE LICENSE FOR UNICOM IS REQUIRED FOR EACH 8560 ON WHICH UNICOM IS INSTALLED.

RELATED ARTICLES (in USER NOTES)

- "Files Used by UUCP ... An Oveview"
- "Using cu with Systems Other Than UNIX Systems"

Rodney Bell, MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

NEW 8561 MULTI-USER SYSTEM AVAILABLE

Tektronix is introducing a new development system, the 8561. Its features include:

8561 Key Features

- Powerful, low-cost entry-level microcomputer development system supporting one or two user ports with 13.6 Mb disk storage, 1 Mb floppy and 265K of RAM memory.

- Easily expanded into a full Tektronix 8560 development system supporting up to eight user ports, 140 Mb disk storage and 1 Megabyte memory.
~~~~~~~~~~~~~~~~~~~~

- Full software design and integration support for over twenty 8-bit and 16-bit microprocessors.

- TNIX operating system based on the powerful, widely-accepted UNIX operating system from Bell Laboratories including the new Color User Interface.

- Large selection of software design tools, including full Pascal support for 16-bit processors that covers the entire design cycle from source code entry to debug operations.

- Complete compatibility with Tektronix 8540 Integration Unit for hardware/ software integration through real-time emulation.

The Tektronix 8561 Software Development Unit provides a powerful and complete set of microcomputer design tools to the smaller design team while accommodating future expansion through a simple cost-effective upgrade path. The basic 8561 fully supports two workstations, which may be either standard CRT terminals or Tektronix 8540 Integration Units designed specifically to handle hardware/software integration tasks through real-time emulation. Through a series of upgrade options, this basic package can be expanded to accommodate up to eight workstations.

The basic version of the 8561 includes an LSI 11/23 16-bit processor, 256Kb of RAM, 13.6 Mb hard disk storage, 1 Mb of flexible disk storage, 2 user ports and 2 line printer ports. This basic system can be easily upgraded within the same mainframe to up to 8 user ports and 35.6 Mb of hard disk storage and 1 Megabyte of main memory. All versions of the 8561 run under TNIX.

The 8561 can fully utilize a wide range of Tektronix microcomputer software design tools covering over 20 8-bit and 16-bit microprocessors. These include compilers(PASCAL LANDS), assemblers, editors and text processors plus the symbolic debug tools and high-level debug tools used in conjuction with the 8540 Integration Unit.

Charlene Eason MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

DIGITAL DESIGN LAB SUPPORT NOW AVAILABLE

Support for the Digital Design Lab is now available. The Digital Design Lab is a combination of software and hardware which can assist the design engineer in debugging single or multiple microprocessor prototypes. DDL consists of: 1) software which executes on the 8560, 2) a state stamp probe which connects a DAS 9100 system with a Trigger Trace Analyzer housed in the 8540 to provide time correlation, 3) an RS-232 cable to connect the DAS with the 8560, and 4) documentation.

Digital Design Lab allows the designer to understand how events being monitored on the prototype circuitry relate to what is occurring within the emulated microprocessor, i.e. what are the cause and effect relationships being observed. The process of relating these events to each other is called Time Correlation, and this is the major feature of the Digital Design Lab.

Besides Time Correlation, DDL has several other important features:

1) 8540 and DAS can be controlled from the 8560.
2) Commands and data can be exchanged between the 8560, 8540 and DAS.
3) Data acquired from 8540 and DAS can be stored and analyzed on 8560.
4) Emulation and Logic Analysis tools can be used separately until necessary to configure as DDL.
5) User can enter DDL commands directly or use a menu-driven interface.

Product Requirements

1) TNIX Version 1.3 or a later version must be installed on the 8560 Multi-User Software Development Unit
~~~~~~~~~~~~~~~~~~~~

2) To more fully analyze acquisition data, the 8560 Auxiliary Utilities (8560U03) should be ordered.
3) To use the DDL mnemonics feature (disassembly on the 8560), the DAS Tape Option must be installed and the DAS firmware must be Version 1.09 or later.
4) To perform Time Correlation, the DAS 91A32 card is required. DDL does not support the 91A08 or 91A24 cards for Time Correlation. These cards can be used for data acquisition, however.
5) The DAS must be ordered with the I/O interface, option 2, to allow communications with the 8560.
6) If the DAS Personality Module is used, two 91A32 cards are required by DDL.
7) DDL also requires the new version of the TTA. The new TTA (with ID numbers B03XXXX and above) supports DDL and is now being shipped. Customers having earlier versions of the TTA must order DDL with the TTA Compatibility Kit - 8560L02.

Nomenclature

8560L01 Digital Design Lab Support
Includes:
DDL Software on flexible disk
State Stamp Probe
20' RS-232 Cable
Documentation

8560L02 Digital Design Lab Support
Includes:
Same items as 8560L01 and TTA Compatibility
Kit for upgrading previously purchased TTAs

State Stamp Probe	(021-0366-01)
DDL Software Disk	(062-6721-00)
DDL Users Manual	(070-4550-00)
RS-232 Cable	(012-0757-00)

Chuck Smith MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

68000 PASCAL FLOATING POINT LIBRARIES

Tek's floating point support conforms to IEEE standard!

A fast floating point library package for our 68000/08 Pascal Compiler will be released later this summer. Users who subscribe to the Software Subscription Service will receive this package automatically. The package will replace the floating point library that was distributed with the Pascal Compiler for the 8560.

Marilyn Hanson MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

ASSEMBLER SUPPORT FOR THE 68010

Tektronix is now offering assembler support for the 68010 microprocessor. The 68K Assembler will be available on the 8560/8561 and on the Vax* 730/750/780/782 mainframes with UNIX** 4.1bsd and VMS* 3.X operating systems.

The 68K Assembler includes support for the 68000, 68008, and the 68010 microprocessors. The 68000 and 68008 processors accept the same instruction set; the 68010 supports additional instructions and registers. An assembler directive allows you to select between the instruction sets.

The current 68000 Assembler on the 8560, 8560B17, will be deleted from the product line. For a limited time, users who have 8560B17 will be able to receive the 68K Assembler as a software update under the Software Subscription Service. Those who have 8560B17 and have already subscribed to the Software Subscription Service will receive the 68K assembler, with support for the 68010 instructions, automatically. Since the SSS has been available for only a short time, some users may not have had the opportunity to subscribe. If users subscribe to the SSS prior to August 15, they will receive the ASM68K as an update to their 68000 Assembler.

Marilyn Hanson MDP Product Marketing

UPDATE TO 8560 ASSEMBLERS

The following assemblers available on the 8560 Multi-User Software Development System have recently been updated to Version 2.0:

PRODUCT	DESCRIPTION
ASMZ80 opt 1A	Assembler for the Z80/NSC800
ASM8086 opt 1A	Assembler for the 8086/88
ASM6809 opt 1A	Assembler for the 6809
ASM8048 opt 1A	Assembler for the 8048

Version 2 of the Assemblers includes the capability of a "virtual" symbol table for user-defined symbols. The choice of using the minimum symbol table or the "virtual" symbol table is made by the user at assembler invocation. If the user chooses to use the minimum symbol table, the assembler is the same as the existing version 1. However, if the user has large numbers of user-defined symbols within an assembler module, he may select the "virtual" symbol table with a -b on the invocation line. This will write the symbols out to disk when the symbol table is exceeded. Other than some speed degradation when the -b is used, there are no changes in the user interface or features of the assembler. The release of these versions also resolves any reported bugs.

The Assembler for the 8086/8088 also includes an updated 8087 macro-library. The library has been grouped into functionally similar sets of 8087 instructions for inclusion at assembly time.

Version 2 of the 8048 Assembler is a bug fix only and does not include the "virtual" symbol table since the current version allows an adequate number of user-defined symbols.

In addition, the current 68000 Assembler will be replaced with ASM68K which includes support for the 68000, 68008 and 68010 processors. ASM68K will include the "virtual" symbol table capability.

CUSTOMER UPDATES

Users who are under warranty or have subscribed to the Software Subscription Service will receive the new version automatically.

* VAX and VMS are trademarks of Digital Equipment Corporation

** UNIX is a trademark of Bell Laboratories

Marilyn Hanson MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

THE 8560 GPIB INTERFACE

The 8560 GPIB Interface is now orderable.

The 8560 GPIB Interface is intended for backup to 9-track magtape only. The operating system supports this capability only for the Dylon Corporation's Model 1015B Controller. This controller can be used with Dylon's Series 3 and Series 9 Magnetic Tape Systems. Average price of these systems is $6.5K to $16K. The user can use either Kennedy or Cypher Data tape drives.

The GPIB hardware fully supports the IEEE 488 Interface standard, BUT... the TNIX operating system doesn't. High level language support is **not** provided to support full GPIB functionality. Native programming tool are available should users decide to develop their own drivers to support other GPIB instruments. However, instrument applications will not be supported by Tektronix. The only support for the above use is provided in the GPIB user manuals.

The GPIB interface requires TNIX Version 1.4.

Chuck Smith MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

SPEED OF 8560 DISK BACKUP USING GPIB OPTION

There are approximately 64,000 blocks on the 8560's 35.6MB disk. Using the dump command to backup the entire disk on tape takes about 1 hour(35 mbytes), or an average of .5 megabytes per minute. This assumes a blocking factor of 5, which is the default blocking factor used by dump. Larger blocking factors will, of course, provide faster backup. Dump supports up to a blocking factor of 20; however, we have only tested up to a blocking factor of 8.

Testing a larger blocking factor requires more memory on the 9 track magnetic tape unit than we originally purchased. We are in the process of acquiring more memory, and will report the test results when finished. Hopefully, we can provide backup timing using a blocking factor of 20 at that time.

Chuck Smith MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

TEKTRONIX NOW SUPPORTS THE VAX

You can now combine Tektronix superior microprocessor development tools with the power of a VAX minicomputer. Tek's VAX*
software products are now available for VAX 730/750/780/782 mainframes, and are completely compatible with 8560 series software products. The following products are now orderable for UNIX** 4.1bsd and VMS* 3.X operating systems:

Assemblers:

Z80/NSC800 Assembler
8085 Assembler
6809 Assembler
68000/08/10 Assembler

* VAX and VMS are trademarks of Digital Equipment Corporation

** UNIX is a trademark of Bell Laboratories
~~~~~~~~~~~~~~~~~~~~

8086/88 Assembler

Compilers:

68000/08 C Compiler with ICS
68000/08 Pascal LANDS which includes:
68000/08 Assembler
Pascal Language Directed Editor
68000/08 Pascal Compiler with ICS
68000/08 Pascal Debug
(Orderable separate or as package)

Communication package:

ICOM40 (A communication package for
the VAX to 8540 interface --
includes 'TERM mode' and 'mload')

Tek Assemblers offer a full feature set, including conditional assembly, macros, includes, and relocatable code, allowing unrestricted use of the microprocessor.

The 68000/08 "C" Compiler translates statements written in "C" (as defined by Kernighan and Ritchie) into executable object code for the 68000 microprocessor. Included with the "C" Compiler is the Integration Control System, a unique Tek tool that automates the software hardware integration process.

The Pascal LANDS package is a unique language development system that supports the entire software design cycle from code entry to debug. P-LANDS includes: Language Directed Editor that knows Pascal and can check Pascal syntax during the edit cycle; Compiler that accepts ISO standard Pascal plus allows microprocessor extensions for interrupts and bit control; Integration Control System that automates the software hardware integration process; and Pascal Debug that allows debugging in Pascal while your program executes in real time on the emulator.

Emulation/debug resources necessary for microprocessor development are available on the Tektronix 8540 Integration Unit. The ICOM40 communication package for the VAX/8540 interface allows you to remotely access your 8540(s) while on a terminal connected to the VAX. By connecting 8540's to your VAX and running Tektronix software, you can have an integrated hardware/software development environment.

System Expansion

Since the VAX software products are functionally and operationally compatible with our 8560 software products, system expansion is easy.

A VAX can be added to an 8560/8540 configuration to increase computing power. An 8560 can be added to a VAX/8540 configuration to offload the emulation/debug tasks from the VAX. Since the 8560 and VAX software tools are source and object code compatible, any existing software programs developed on the 8560 (or VAX), can be used on the VAX (or 8560).

Please contact your local Tektronix representative for more information.

Diane Wortsmann MDP Product Marketing

8560 1MB MEMORY PERFORMANCE BENCHMARK

If you are considering buying one megabyte of memory for your 8560, the following benchmark gives one indication of how much extra performance to expect. A benchmark was performed in Europe that compared two systems. System A was a standard 8560 with 256KB of memory, and system B was an 8560 with 1MB of memory. The benchmark program used was a PASCAL compilation that required 60 seconds to execute on a single user 256KB memory 8560.

Three tests were performed:

Test 1: Foreground compilation of one job.
Test 2: Background compilation of 6 jobs.
Test 3: Background compilation of 10 jobs.

The results were as follows:

	Mem.	Test 1	Test 2	Test 3
	Mbytes			
Execution Time	1/4	60 secs.	2280 secs.	2 hours +
	1	60 secs.	400 secs.	680 secs.

As you can see, the performance increase is quite dramatic when there are a number of ”large memory utilization” programs.

Chuck Smith MDP Product Marketing

LDE WITH TELEVIDEO 950 TERMINAL

A copy of the termcap for a Televideo terminal is available. A termcap is encoded information representing a terminal. LDE and some other tools use termcap entries. If you want to use LDE with Televideo 950 terminals, contact your Tektronix Sales or Applications Engineer.

Rodney Bell MDP Product Marketing

NATIVE PROGRAMMING PACKAGE

Installation. TNIX 1.4 provides some 'include' (suffix: .h) files that are more recent versions than the versions being currently distributed with the Native Programming Package. The installation of Native Programming tools overwrites these files. If you are installing the Native Programming tools on a TNIX version 1.4 system, save the /usr/include files first. Version 2 of Native Programming, coming soon, will not have this problem.

Rodney Bell MDP Product Marketing

8500 CONFIGURATION for RT11/50

RT 11/50 is an optional operating system for the 8550. RT 11/50 is based on the popular RT-11* operating system.Here is one means to configure an 8550 system for RT11/50: convert the 8550 into an 8501GPS and 8540. The 8501GPS is 8501 with RT11/50 installed, resulting in an 8501 General Purpose Computing System with RT-11. The 8301 is converted to an 8540 using the 8301 to 8540 conversion kit (020-0953-00). The 8501GPS and the 8540 are connected via RS-232 and COMM mode, with the 8501GPS serving as a general purpose host. This configuration offers several benefits when compared to switching between RT11/50 and DOS50:

- constant operating environment
- convenience and speed in downloading object to emulate
- 8540 advantages over 8550

The configuration looks like this:

```
 --------            ------            ---------           -------
|        |          |      |          |   8501    |        |       |
|Terminal|-------   | 8540 |-------   |(RT11/50)  |------- |Printer|
|________| RS232    |______| RS232    |___________| RS232  |_______|
```

Rodney Bell MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

UPDATED 8540 ROMPATCHES - LEVEL 46

The latest version of 8540 Rompatches through level 46 are listed below. It's a good idea to update all 8540s with the appropriate patches, even if there are no apparent operational problems.

Bill Bevan MDP Product Marketing

```
rompatch -i 02fd0 000000 0 0
rompatch -i 02fd1 000000 0 1
rompatch 09a3f 1 1a5 /140100/2 68
rompatch 052a 2 41d /DEFLT/EX[] 02
rompatch 0e523 3 429 /DEFLT/EX[] 00
rompatch 04a03 4 18b /KERNL/INIT 0418
rompatch 0a6f2 5 28 /KERNL/INIT 3f2b4fc0
rompatch 02066 6 34f /KERNL/INIT 060107cf860117
rompatch 05240 7 190 /KERNL/GO 3bba
rompatch 09251 8 1ab /KERNL/GO 3bba
rompatch 0c145 9 0c /KERNL/DEVDB 9595b5b5
rompatch 0e33f 0a 39 /KERNL/DEVDB 80
rompatch 02688 0b 0d /KERNL/PCB.NMLO 3e
rompatch 02771 0c 06 /KERNL/PCB.TYPE 00
rompatch 0a84f 0d 30 /KERNL/START 95
rompatch 0cdb9 0e 3c /KERNL/RCMINT c7
rompatch 0d29a 0f 98 /KERNL/INIT 1a
rompatch 09126 10 24c /DEFLT/CONFIG[] 3f2490
rompatch 0a6bc 11 490 /DEFLT/CONFIG[] 20cc84980c8451175795
rompatch 0cef5 12 79 /KERNL/RESPTR 7f
rompatch 0102c 13 0 /DEFLT/DI[] 2306
rompatch 0334e 14 306 /DEFLT/DI[] 0623070b170006230711171 81f2203
rompatch 0d641 15 0 /138200/3 2800
rompatch 05028 16 800 /138200/3 062807051700062807 0b17181f2330
rompatch 05761 17 0A4A /DEFLT/ROMPATCH[] 3f2dab
rompatch 09a4d 18 0DAB /DEFLT/ROMPATCH[] cc0b83cc0b840c0d8617
d 0
rompatch 0630a 19 21 /KERNL/DEVDB 03
rompatch 0d342 1a 5f0 /KERNL/EMUSVC 3635ff
rompatch 0c6d3 1b 11e2 /137800/1 42
rompatch 02346 1c 2d /KERNL/DEVDB 40
```

* RT-11 is a trademark of Digital Equipment Co.
~~~~~~~~~~~~~~~~~~~~

```
rompatch 0c80b 1d 51 /KERNL/DEVINT 1b
rompatch 026cf 1e 1748 /137800/1 30
rompatch 0f6cd 1f 1086 /137800/1 05
rompatch 09e2b 20 1747 /137800/1 c0c0c0c0c0c0
rompatch 08add 21 0e5 /KERNL/FH85B 38e8
rompatch 08996 22 166 /KERNL/INIT c0c0c00daa05cc0a08c0c0c0
rompatch 092ee 23 18d /KERNL/INIT 0e
rompatch 08e79 24 195 /KERNL/INIT 0a062a07fa1b04062b07023f29e90471cc0a0704a3
rompatch 05fa5 25 1aa /KERNL/INIT cc0a081f2b56c0c0c01824
rompatch 0be60 26 1de /KERNL/INIT 01017540
rompatch 0be85 27 356 /KERNL/INIT 04290504cdea0504000511cdea050406098945f061
rompatch 0ea55 28 36b /KERNL/INIT c884d4ee1b021fbf3f29f854cef4081f29b3
rompatch 0c5fc 29 5 /KERNL/CMDINT 31
rompatch 0ff51 2a 3d /KERNL/DEVHAN e6001808
rompatch 0fa01 2b 51 /KERNL/DEVHAN e6011808
rompatch 052e5 2c 100 /KERNL/QUESUB 0a
rompatch 03728 2d 3c /KERNL/RCMINT 2dc7
rompatch 04ae9 2e 280 /KERNL/GO 1f3fe9
rompatch 0ee30 2f 6ca /KERNL/GO cc9c14063f07f117070100001f3ba2
rompatch 04936 30 0b6 /DEFLT/CONFIG[] 10
rompatch 0145b 31 3bb /DEFLT/CONFIG[] 1f249ac01f249ec0
rompatch 0da57 32 49a /DEFLT/CONFIG[] 041e1b02042bc877
rompatch 04d76 33 4a2 /DEFLT/CONFIG[] 0c844318053f2286
rompatch 03217 34 4aa /DEFLT/CONFIG[] 1b033f22b208681f23e4
rompatch 02740 35 1c4 /KERNL/DEVINT 3fdc
rompatch 0d928 36 1fdc /KERNL/DEVINT 3f234f1f2227
rompatch 08cad 37 1adb /137800/1 030dfa12ec1a109c1b350605
rompatch 0aec0 38 1b05 /137800/1 07
rompatch 0c6c1 39 1b1f /137800/1 06
rompatch 0bde1 3a 1b36 /137800/1 060dfa12c12006000707
rompatch 07454 3b 0658 /137800/1 1f1bee
rompatch 09cf4 3c 1bee /137800/1 04c0d4d004ffd4f3d4f820d498d4d2d4a21f065b
rompatch 0d590 3d 0f10 /136500/0 05
rompatch 0e85a 3e 1004 /173601/0 05
rompatch 0101d 3f 102e /173601/0 05
rompatch 08a6e 40 040 /KERNL/FH85A 35d6
rompatch 0a79d 41 314 /KERNL/INTSRV 87201f3f39
rompatch 0dcb3 42 1f39 /KERNL/INTSRV 1a0404011b030c8738cc873a17
rompatch 06eb9 43 292 /138600/0 3f381e
rompatch 0f190 44 1816 /138600/0 6508cd018c17
rompatch 0cebe 45 5fa /138600/0 4f
rompatch 046b2 46 184 /138600/0 0f
```

~~~~~~~~~~~~~~~~~~~~~~

MDP MANUALS LIST

This list contains ordering information for all MDP manuals. Be sure to check subsequent issues of *USER GROUP NEWS* for updates to this list. Manuals are listed in the following categories:

Charlene Eason MDP Customer Support

8560 Users Manuals
8550 Users Manuals: DOS/50 V.2
8540 Users Manuals
8500 MDL Series B Assembler Users Manuals
8500 MDL Series Emulator Specifics Manuals
Other 8500 Series Users Manuals
8550 Users Manuals: DOS/50 V.1
8500 MDL Series A Assembler Users Manuals
8500 Series Service Manuals
8500 Series Installation Manuals
MicroLab I Manuals
8002A Users Documentation
8001 Users Documentation
8001/8002A Service Manuals
~~~~~~~~~~~~~~~~~~~~~~

Other Peripheral Service & Users Manuals

8560 USERS MANUALS	PART NUMBER
8560 MUSDU TNIX V1 Users Manual	070-3940-00
8560 MUSDU TNIX V1 Reference Manual	070-3941-00
8560 MUSDU TNIX V1 Reference Booklet	070-3942-00
8560 MUSDU TNIX V1.3 System Users Supplement	070-4496-00
8560 MUSDU TNIX V1.3 System Reference Supplement	070-3211-00
8560 MUSDU Processing Package Users Mnl.	070-4272-00
8560 MUSDU Native Programming Pkg Users Mnl.	070-4271-00
8560 MUSDU Auxiliary Utilities Pkg.Users Mnl.	070-4270-00
8560 MUSDU ACE Reference Card	070-4190-00
8560 MUSDU ACE Users Booklet (version 2)	070-4468-00
8560 MUSDU Language-Directed Editor Users Manual	070-4252-00
8560 MUSDU Language-Directed Editor CT8500-Edition Reference Card	070-4249-00
8560 MUSDU Language-Director Editor Template for CT8500 Keyboard (package of 4 templates)	070-4622-00
8560 MUSDU Pascal Debug 8086/8088 Reference Card	070-4283-00
8560 MUSDU Pascal Debug Z8001/Z8002 Reference Card	070-4464-00
8560 MUSDU Pascal Debug 68000 Reference Card	070-4465-00
8560 MUSDU 8086/8088 Pascal Language Ref. Manual	070-4378-00
8560 MUSDU 8086/8088 Pascal Compiler Users Manual	070-3878-00
8560 MUSDU Z8001/Z8002 Pascal Compiler Users Manual	070-3876-00
8560 MUSDU 60000 Pascal Compiler Users Manual	070-3875-00
8503 Disk Expansion Unit Users Manual	070-4463-00
8560 MUSDU Intel COMM Users Manual	070-4481-00
8560 MUSDU User Information Instruction Sheet	070-4679-00
8560 MUSDU Digital Design Lab Users Manual	070-4550-00
8560 MUSDU UNICOM Users Manual	070-4536-00
8560 MUSDU Magnetic Tape Interface Users Manual	070-4586-00

8550 USERS MANUALS: DOS/50 V.2

8550 MDL Users Manual: DOS/50 V2	070-3936-00
8550 MDL System Reference Booklet: DOS/50 V2	070-3937-00
8550 MDL System Users Manual DOS/50 Version 2.1A Supplemental Information	070-4553-00
8550 MDL GUIDE Installation Manual	070-4402-00
8550 MDL Editor V4.X Manual	070-3571-00
8550 MDL Editor V4.X Reference Card	070-3572-00
Real-Time Prototype Analyzer Users Mnl: DOS/50 V2	070-3922-00
8550 MDL ACE Users Booklet (version 2)	070-4363-00
8550 MDL Intel COMM Users Manual	070-4480-00
8550 MDL Pascal 8086/8088 Compiler Users Manual	070-3877-00
8550 MDL Pascal 8080/85 Compiler Users Manual V4.0	070-4336-00
8550 MDL Pascal 8080/8085 Compiler Version 4.02 Update Information	070-4591-00
8300H01/02 MDL/u Compiler Users Manual	070-3601-00
8300H01/02 MDL/u Compiler Reference Booklet	070-3602-00
-- 8300H01 8080A MDL/u Compiler Specifics	070-3598-00
-- 8300H02 6800/02 MDL/u Compiler Specifics	070-3599-00
8300D15 8086 Prototype Debug Specifics	070-3603-00
8300D15 8086 Prototype Debug Reference Card	070-3604-00
8550 MDL RT11/50 Users Manual: Volume 1, System	070-4409-00
8550 MDL RT11/50 Users Manual: Volume 2, System	070-4410-00
8550 MDL RT11/50 Users Manual: Volume 3, System	070-4411-00
8550 MDL RT11/50 Users Manual: Volume 4, FORTRAN IV	070-4412-00
8550 MDL RT/11 Installation Sheet	070-4404-00

8540 USERS MANUALS

8540 Integration Unit System Users Manual OS/40	070-3939-00
8540 Integration Unit Reference Booklet OS/40	070-3992-00
8540 Integration Unit System Users Manual OS/40 Version 1 Supplemental Information for COM Software Version 4.1	070-4552-00
8540 Integration Unit Intel COMM Users Manual	070-4479-00

8500 MDL SERIES B ASSEMBLER USERS MANUALS

8500 MDL Series B Assembler Core Users Manual	070-3856-01

-- 8550 Host Specifics	070-3943-01
-- 8560 Host Specifics	070-3944-01
-- Z80A Assembler Specifics	070-3949-00
-- Z80A Assembler Reference Card (8560)	070-3950-00
-- Z8001/Z8002 Assembler Specifics	070-3854-00
-- Z8001/Z8002 Assembler Reference Card (8550)	070-3973-00
-- Z8001/Z8002 Assembler Reference Booklet (8560)	070-3958-00
-- 1802 Assembler Specifics	070-4507-00
-- 1802 Assembler Reference Booklet (8560)	070-4506-00
-- 6800/01/02 Assembler Specifics	070-3947-00
-- 6800/01/02 Assembler Reference Card (8560)	070-3948-00
-- 6809 Assembler Specifics	070-3960-00
-- 6809 Assembler Reference Card (8550)	070-4369-00
-- 6809 Assembler Reference Card (8560)	070-3961-00
-- 68000 Assembler Specifics	070-3855-00
-- 68000 Assembler Reference Booklet (8550)	070-3974-00
-- 68000 Assembler Reference Booklet (8560)	070-3959-00
-- 8048/8021/8041A/8022 Assembler Specifics	070-3955-00
-- 8048/8021/8041A/8022 Assembler Ref Card (8560)	070-3956-00
-- 8051 Assembler Specifics	070-4321-00
-- 8051 Assembler Reference Card (8550)	070-4364-00
-- 8051 Assembler Reference Card (8560)	070-4320-00
-- 8080A/8085A Assembler Specifics	070-3945-00
-- 8080A/8085A Assembler Reference Card (8560)	070-3946-00
-- 8086/8088 Assembler Specifics	070-3853-00
-- 8086/8088 Assembler Reference Booklet (8550)	070-3852-00
-- 8086/8088 Assembler Reference Booklet (8560)	070-3957-00
-- 9900/9989 Assembler Specifics	070-4373-00
-- 9900/9989 Assembler Reference Card (8560)	070-4368-00
-- 9900/9989 Assembler Reference Card (8550)	070-4367-00

8500 MDL SERIES EMULATOR SPECIFICS USERS MANUALS

-- Z80A Emulator Specifics	070-3964-01
-- Z8001/Z8002 Emulator Specifics	070-3969-00
-- 6800/6802 Emulator Specifics	070-3963-00
-- 6801/68120 Emulator Specifics	070-3991-00
-- 6809 Emulator Specifics	070-3971-00
-- 68000 Emulator Specifics	070-3970-01
-- 8048/8021/8041A/8022 Emulator Specifics	070-3967-01
-- 8080A Emulator Specifics	070-3962-00
-- 8085A Emulator Specifics	070-3966-00
-- 8086/8087/8088 Emulator Specifics	070-3968-01
-- 9900/9989 Emulator Specifics	070-3965-00
-- TMS9900 Emulator Specifics	070-4397-00
-- 3870/3872/F8 Emulator Specifics	070-4438-00

OTHER 8500 SERIES USERS MANUALS

8500 MDL Series ACE Users Manual (Version 1)	070-3573-01
8500 MDL Series ACE Reference Manual (Version 2)	070-4361-00
8500 MDL Series ACE Users Reference Card (Version 1)	070-3574-00
8500 MDL Series Pascal Debug Users Manual	070-4281-00
8500 MDL Series Pascal Language Reference Manual	070-3880-00

8500 MDL Series 2716/2732 PROM Programmer Specifics	070-3868-00
8500 MDL Series 2764 PROM Programmer Specifics Users	070-4375-00
8500 MDL Series 8748/etc. PROM Programmer Specifics	070-3869-00
8500 MDL Series 8751 PROM Programmer Specifics Users	070-4414-00
8500 MDL Series 68701 PROM Programmer Specifics Users	070-4413-00
8500 MDL Series Trigger Trace Analyzer Users Manual	070-3760-01
8500 MDL Series Extended Hex Interface Instruction Sheet	070-4478-00

Simplifying Microcomputer-Based Product Design	062-5812-00
CT8500 Video Display Terminal Operator's Manual	070-3737-00

8550 USERS MANUALS: DOS/50 V.1

8550 MDL System Users Manual	070-3457-00
-- 8300E01 8080A Emulator Specifics	070-3562-00
-- 8300E02 6800/02 Emulator Specifics	070-3563-00
-- 8300E04 Z80A Emulator Specifics	070-3564-00
-- 8300E05 TMS9900 Emulator Specifics	070-3565-00
-- 8300E06 8085A Emulator Specifics	070-3566-00
-- 8300E07 3870/3872/F8 Emulator Specifics	070-3567-00
-- 8300E09 1802 Emulator Specifics	070-3568-00
-- 8300E10 8048/8021/8041A/8022 Emulator Specifics	070-3569-00
-- 8300P28 6809 Emulator Specifics	070-3851-00

8550 MDL System Reference Booklet: DOS/50 V1	070-3458-00
Real-Time Prototype Analyzer Users Manual: DOS/50 V1	070-2785-01

8500 MDL SERIES A ASSEMBLER USERS MANUALS

8300AXX Assembler Users Manual	070-3575-01
-- 8300A01 8080A/8085A Assembler Specifics	070-3576-00
-- 8300A01 8080A/8085A Assembler Reference Card	070-3577-00
-- 8300A02 6800/01/02 Assembler Specifics	070-3578-00
-- 8300A02 6800/01/02 Assembler Reference Card	070-3579-00
-- 8300A04 Z80A Assembler Specifics	070-3580-01
-- 8300A04 Z80A Assembler Reference Card	070-3581-00
-- 8300A05 TMS9900 Assembler Specifics	070-3582-00
-- 8300A05 TMS9900 Assembler Reference Card	070-3583-00
-- 8300A07 3870/3872/F8 Assembler Specifics	070-3584-00
-- 8300A07 3870/3872/F8 Assembler Reference Card	070-3585-00
-- 8300A09 1802 Assembler Specifics	070-3586-00
-- 8300A09 1802 Assembler Reference Card	070-3587-00
-- 8300A10 8048/etc. Assembler Specifics	070-3588-00
-- 8300A10 8048/etc. Assembler Reference Card	070-3589-00
-- 8300A15 8086/8088 Assembler Specifics	070-3592-00
-- 8300A15 8086/8088 Assembler Reference Card	070-3593-00
-- 8300A20 Z8000 Assembler Specifics	070-3594-00
-- 8300A20 Z8000 Assembler Reference Card	070-3595-00
-- 8300A26 68000 Assembler Specifics	070-3596-00
-- 8300A26 68000 Assembler Reference Card	070-3597-00
-- 8300A28 6809 Assembler Specifics	070-3692-00
-- 8300A28 6809 Assembler Reference Card	070-3693-00

8500 SERIES SERVICE MANUALS

8301 Microprocessor Development Unit Service Manual	070-2976-01
8301/8540 Conversion Instruction Sheet	070-4447-00
8501 Data Management Unit Service Manual	070-2975-00
8540 Integration Unit Service Manual	070-3920-00
8560 MUSDU Service Manual	070-3900-00

8503 Disk Expansion Unit Service Manual	070-4356-00
8560 MUSDU GPIB Interface Service Manual	061-2768-00
CT8500 Video Display Terminal Service Manual	061-2431-00
DataTrak 8" Flexible Disc Drive Service Manual	070-4253-00

Real-Time Prototype Analyzer Service Manual	070-2724-01
Trigger Trace Analyzer Service Manual	070-3762-00
PROM Programmer Controller Service Manual	070-3757-00
-- 2716/2732 PROM Programmer Module Service Manual	070-3758-00
-- 2764 PROM Programmer Module Service Manual	070-4350-00
-- 8751 PROM Programmer Module Service Manual	070-4352-00
-- 8748/etc. PROM Programmer Module Service Manual	070-3759-00
-- 68701 PROM Programmer Service Manual	070-4351-00
64K/128K Program Memory Service Manual	070-3924-00
Memory Allocation Controller Service Manual	070-3926-00
8500 Modular MDL Series 8086-to-8086/8087 020-0959-00 Technical Instruction Sheet	070-4561-00
8500 Modular MDL Series 8088-to-8088/8087 020-0960-00 Technical Instruction Sheet	070-4562-00

Z80A Emulator Processor Service Manual	070-2715-01
Z8001/Z8002 Emulator Processor Service Manual	070-3772-00
1802 Emulator Processor Service Manual	070-2631-01
3870/3872/F8 Emulator Processor Service Manual	070-2634-01
6500/1 Emulator Processor Service Manual	070-2887-00
68xx Emulator Processor Service Manual	070-3768-00
68xx Emulator Processor Field Modification Sheet	070-4458-00
6800/6802 Emulator Processor Service Manual	070-2354-03
6801/68120 Prototype Control Probe Service Manual	070-3864-00
6809 Prototype Control Probe Service Manual	070-3867-00
6809E Prototype Control Probe Service Supplement	070-4461-00
68000 Emulator Processor Service Manual	070-3770-00
8048/8021/8041A/8022 Emulator Processor Service Manl	070-2632-01
8080A Emulator Processor Service Manual	070-2353-03
8085A Emulator Processor Service Manual	070-2716-01
8086/8088 Emulator Processor Service Manual	070-3774-00
9900 Emulator Processor Service Manual	070-2712-01
9900/9989 Emulator Processor Service Manual	070-4157-00

8500 SERIES INSTALLATION MANUALS

8540 Integration Unit Installation Guide	070-3921-00
8550 Microcomputer Development Lab Installation Guide	070-2974-01
8560 MUSDU Installation Guide	070-3899-00
8560/8561 MUSDU Installation Guide	070-4627-00
8503 Disk Expansion Unit Installation Manual	070-4355-00
8560 GPIB Interface Installation Service Manual	070-4476-00

Z80A Emulator Processor/PCP Installation Manual	070-3665-01
Z8001/Z8002 Emulator Processor/PCP Installation Mnl.	070-3773-00
1802 Emulator Processor/PCP Installation Manual	070-3667-00
3870/3872/F8 Emulator Processor/PCP Installn. Manual	070-3669-00
68xx Emulator Processor Installation Manual	070-3769-00
6800/02 Emulator Processor/PCP Installation Manual	070-3663-00
6801/68120 Prototype Control Probe Installation Manl	070-3865-00
6809 Prototype Control Probe Installation Manual	070-3866-00
6809E Prototype Control Probe Instl Serv Supplement	070-4462-00
68000 Emulator Processor/PCP Installation Manual	070-3771-01
8048/8021/8041A/8022 Emul. Proc./PCP Installn. Manual	070-3671-00
8080A Emulator Processor/PCP Installation Manual	070-3664-00
8085A Emulator Processor/PCP Installation Manual	070-3666-00
8086/8088 Emulator Processor/PCP Installation Manual	070-3775-00
9900/9989 Emulator Processor/PCP Installation Manual	070-4158-00
TMS9900 Emulator Processor/PCP Installation Manual	070-3670-00

Trigger Trace Analyzer Installation Manual	070-3761-00
PROM Programmer Controller Installation Manual	070-3903-00
64K/128K Program Memory Installation Manual	070-3923-00
Memory Allocation Controller Installation Manual	070-3925-00

MICROLAB 1 MANUALS

MicroLab I Instruction Manual	070-2827-01
---F8 Personality Card Supplement	070-2864-01
---MCS-48 Personality Card Supplement	070-2937-01
---Z80A Personality Card Supplement	070-2861-00
---Z8000 Personality Card Supplement	070-2863-00
---1802 Personality Card Supplement	070-2866-01
---3870 Personality Card Supplement	070-2862-01
---6500/1 Personality Card Supplement	070-2941-01
---6801/68120 Personality Card Supplement	070-3983-00
---6802 Personality Card Supplement	070-2939-01
---6809 Personality Card Supplement	070-3984-00
---6809E Personality Card Supplement	070-4460-00
---68000 Personality Card Supplement	070-3982-00
---8085A Personality Card Supplement	070-2860-00
---8086 Personality Card Supplement	070-2865-00
---9900/9989 Personality Card Supplement	070-4362-00

8002A USERS DOCUMENTATION

TEKDOS System Users Manual	070-2701-02
---Supplement for 6800/6802 Emulator Processor	070-2714-00
---Supplement for 8048/etc. Emulator Processor	070-2856-00
---Supplement for 6500/1 Emulator Processor	070-3433-00
8002A System Reference Booklet (TEKDOS Version 3)	070-2706-01
8002A TEKDOS Editor Version 3.X Reference Card	070-3442-00
8002A TEKDOS Editor Version 3.X	070-3441-00
8002A MDL/u Compiler Users Manual	070-2584-01
8002A MDL/u Compiler Reference Booklet	070-2629-01
8002A Pascal Compiler Users Manual	061-2417-00
Simplifying Microproc-Based Product Design (8000 Ser)	062-3771-00

8002A 8086 Prototype Debug Users Manual	070-3455-00
8002A 8086 Prototype Debug Reference Card	070-3504-00
8002A Z8000 Prototype Debug Users Manual	070-3507-00
8002A Z8000 Prototype Debug Reference Card	070-3508-00
8002A 68000 Prototype Debug Users Manual	070-3511-00
8002A 68000 Prototype Debug Reference Card	070-3512-00

8002A F8/3870/3872 Assembler & Emulator Users Manual	070-2615-00
8002A F8/3870/3872 Assembler & Emulator Reference Crd	070-2616-00
8002A Z80 Assembler & Emulator Users Manual	070-2705-01
8002A Z80 Assembler & Emulator Reference Card	070-2710-01
8002A 1802 Assembler & Emulator Users Manual	070-2627-00
8002A 1802 Assembler & Emulator Reference Card	070-2628-00
8002A 6800/6802 Assembler & Emulator Users Manual	070-2703-02
8002A 6800/6802 Assembler & Emulator Reference Card	070-2708-02
8002A 8048/8021/8041A/8022 Assembler & Emulator Users Manual	070-2611-00

8002A 8048/8021/8041A/8022 Assembler Reference Card	070-2612-01
8002A 8080A/8085A Assembler & Emulator Users Manual	070-2702-01
8002A 8080A/8085A Assembler & Emulator Reference Card	070-2707-01
8002A 9900 Assembler & Emulator Users Manual	070-2704-01
8002A 9900 Assembler & Emulator Reference Card	070-2709-01

8002A Assembler Users Manual	070-3454-00
---Supplement for 6500/1 Assembler	070-2790-00
8002A 6500/1 Assembler Reference Card	070-2784-00
---Supplement for 8086 Assembler	070-3505-00
8086 Assembler Reference Card	070-3506-00
---Supplement for Z8000 Assembler	070-3509-00
Z8000 Assembler Reference Card	070-3510-00
---Supplement for 68000 Assembler	070-3513-00
68000 Assembler Reference Card	070-3514-00

8001 USERS DOCUMENTATION

8001 System Users Manual	070-2464-00
---Supplement for F8/3870/3872 Emulator Processor	070-2822-00
---Supplement for 1802 Emulator Processor	070-2854-00
---Supplement for 6800/6802 Emulator Processor	070-2713-00
---Supplement for 8048/etc. Emulator Processor	070-2855-00
---Supplement for 6500/1 Emulator Processor	070-3432-00
8001 System Reference Card (TEKOPS version 2)	070-2471-01

8001/8002A SERVICE MANUALS

8001/8002A Microprocessor Lab Service Manual	070-2711-00

8001/8002/8002A Microprocessor Lab Installation Guide	070-2717-01
---Supplement for 8080A Emulator Processor	070-3380-00
---Supplement for 6800/6802 Emulator Processor	070-2951-00
---Supplement for Z80 Emulator Processor	070-3382-00
---Supplement for 9900 Emulator Processor	070-3381-01
---Supplement for 8085 Emulator Processor	070-2871-01
---Supplement for F8/3870/3872 Emulator Processor	070-2872-01
---Supplement for 1802 Emulator Processor	070-2882-01
---Supplement for 8048/etc. Emulator Processor	070-2473-00
---Supplement for 6500/1 Emulator Processor	070-3475-00

1702A PROM Programmer Service Manual	070-2722-00
2704/2708 PROM Programmer Service Manual	070-2723-00
Maintenance Front Panel Service Manual	070-2725-00
Flexible Disc Unit Service Manual	070-2587-00
118-0195-00 Flexible Disc Drive Service Manual	070-2786-00

OTHER PERIPHERAL SERVICE & USERS MANUALS

CT8100 CRT Terminal Service Manual (4023)	070-2362-00
CT8100 CRT Terminal Users Manual (4023)	070-2359-00
CT8101 Printing Terminal Service Manual	070-2363-00
CT8101 Printing Terminal Users Manual	070-2360-00
LP8200 Line Printer Service Manual	070-2364-00
LP8200 Line Printer Users Manual	070-2361-00
4025 Computer Display Terminal Operator's Manual	070-2401-02
4025 Computer Display Terminal Programmer's Reference Manual	070-2402-00
4025 Computer Display Terminal Programmer's Reference Card	070-2437-03
4024/4025 Computer Display Terminal Service Manual	
.....volume 1	070-2830-00
.....volume 2	070-2831-00

APPLICATIONS SECTION

8086/8087 EMULATION CONCERNS

Intel sells an adapter circuit board (with an 8087 on it) so that single board computer users can add 8087 support. The question has come up, how well does our previous 8086 probe (8300P15 not 8300P17) work in an environment where the 8087 exists on the prototype.

The 8087 is a "closely coupled" co-processor to the 8086/88 and when you separate them they quickly get out of step with each other, which usually results in a bus hang. Consider, for example, what happens at an emulation break. We tri-state the bus but the addresses and control signals continue to go out to the prototype which, of course, confuses the 8087.

If you are not executing any 8087 code with the 8087 in the prototype, you will still run into problems; particularly when using trace. Random data coupled with certain control signals (like a fetch that the 8087 sees) may look like an escape instruction to the 8087, which in turn, causes it to wake up and start negotiating for the bus ... again the result is a hung bus.

If you wish to use and/or have an 8087 present on your prototype, you will need to purchase an 8086/87 or 8088/87 probe for proper support; or if you already have the older style probe it can be upgraded to provide full 8087 support.

Wolfgang Takatsch MDP Customer Support

~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~

8086 CLOCK DESKEW AND OTHER STRAPS

The main purpose of the deskewer is to solve a problem with the Intel 8288 bus controller and excessive delay of the processor status output. The clock deskewer fixes the problem by making the prototype and emulator clock rising edge have zero phase error. It is this leading edge of the clock that the 8288 and the processor key off of. The falling edge is assumed to be at the proper location with respect to the deskewed edge (a 33.33% duty cycle is assumed) and in most cases this will be true. Note, however, that this falling edge is not actively deskewed and therefore the resulting phase error will vary for different probes and prototypes. Note: *If the clock has a non standard duty cycle you may have to turn the clock deskewer off.*

The 8086 processor uses the falling edge of the clock to strobe the ready line. The 8086 is VERY sensitive to ready with respect to setup and hold timing. If done incorrectly, the processor internally hangs. That is why Intel "suggests" that their 8284A clock chip be used. The 8284A assures that READY is properly timed so that the 8086 will work. With the clock deskewer OFF, the falling clock edge and READY relationship is maintained which keeps the 8284A/8086 working together. At 8MHZ however, the deskewer is needed to keep the 8288 bus controller properly timed. But what if the user wants both, 8MHZ and wait states (READY) ?

If the deskewer is turned on, then the factory set straps can be re-adjusted DOWNWARD to a lower number (first note the factory setting so you can put things back the way they were if need be). There are two sets of jumpers on the control board inside the probe (next to the delay lines). Each set is appropriately labeled, the numbers represent nano seconds. One steps in 6, the other in 2 nanosecond steps. The idea is to reach an acceptable compromise: enough skew to make the 8288 function while at the same time keeping the 8284A functional. Incidentally, MICROLAB (a Tektronix MDP service instrument) uses both the 8288 bus controller and the 8284A clock chip and wait states as well with no problems.

The following chart summarizes what has been stated above.

	BYPASS	DESKEW	
Minimum mode at 8 Mhz	X		
Minimum mode 5 Mhz	X	X	
No Wait States		X	
Max Mode 8 Mhz		X	Adjust to eliminate wait state problems
Max Mode 5 Mhz		X	

Wolfgang Takatsch MDP Customer Support

NOTES ON THE USE OF ASSEMBLER SECTION NAMES

The name of a section is equated to the value of the assembler program counter. The program counter for all sections is initialized to zero. Thus the section name value is always zero for any section. If the section is subsequently relocated the linked value will reflect the location address of the section as expected. If an "org" statement is encountered, the assembler program counter is then set to the value specified to evaluate subsequent instructions. Even if the first instruction is an "ORG" assembler directive, **the section address remains at zero.** The "ORG" directive does not affect the address associated with the section name, and this will be reflected in the reported size of the section as well as references to the section name. This does not create a problem for relocatable sections, but can yield unexpected results when used in absolute sections if our conventions are not understood.

for example:

```
        SECTION JUNK,ABSOLUTE
```

```
            ORG      1000H
DATA        BLOCK    256H
            SECTION  ANOTHER
            LDA A    JUNK          ; **** JUNK HAS THE VALUE 0000 ****
            LDA B    DATA          ; **** THIS WILL BE CORRECT    ****

JUNK  =  0000
DATA  =  1000
```

- Generally avoid the use of the "org" statement within sections and locate the section with the linker.
- Absolute sections are generally not needed except where the micro can take advantage of short(base page) memory addressing. All other uses of the absolute directive can be accomplished with relocatable sections and linker location of code or data.

John Owens MDP Customer Support

REGISTER NAME USE WITH 8080, 8085, Z80 ASSEMBLERS

Several problems have been reported regarding invalid assembler commands generating code without error. We chose to adopt Intel's convention of using a value to select a register when we set the specifications for our assembler.

When a register name is used where an immediate value is expected, the assembler assumes you want to use the value associated with the name. Likewise if a value is specified where a register is expected, the register selected is the one associated with that value. This allows the user to use symbolic names for the registers. For example, a programmer might find that equating the name "STACK" to the value of "SP" or "BUFF_PTR" to the value of "X" would improve the readability of his code.

There is no distinction between values that you assign arbitrary labels and predefined micro specific values (labels). Thus no error would be generated for the following "MVI" instruction:

```
                    ;It could be that the programmer had
                    ;in mind:
      MOV    A,M    ;But assume that
                    ;Due to a typo the programmer entered:
      MVI    A,M    ;The actual assembler value for M would be used
                    ;to load the A register on execution
                    ;and the obvious error would not be caught.
                    ;But then it just could be that he did
                    ;want this to happen. Thus we should not generate
                    ;an error.
```

In adopting Intel's convention, we made the portability between Intel assembler source and Tektronix assembler source more manageable. At the same time this implies that possibly insane or bizarre instructions will be understood by the assembler without error. The 8080, 8085, and Z80 assemblers are the only ones with this idiosyncrasy.

John Owens MDP Customer Support

HINTS ON USING CONFIG TERM ON THE 8550

- Only 8550 commands will transfer files to/from memory or disk! Do not use 8560 commands for transfers to or from the 8550!
- 8560 commands will work to and from the screen-keyboard only! These will NOT work for file transfers!
- Local files are 8550 files and must be enclosed in quotes for all transfer commands such as "8550.file" and if a re-direct is also needed for that command, it must be included in the quotes i.e. "<8550.file"
- The 8560 utilizes two software transfer protocols (Terminal and HSI) and is configurable to either of two hardware configurations on any port (RS232 and RS422). For 'CONFIG TERM' HSI software protocol must be selected. And the hardware configuration must be RS232 in order to hook up to the 8550 J101 port.
- When working with binary files use the -b modifier to the appropriate command.
- The command modifier "t" was left out of most 'config term' command examples also. It is needed if you are running at RS232 speeds (up to 9600 baud) to give the system enough time to load the command.
- With some 8550 commands, it is best to reconfigure to "local" rather than pass the command and results over the cable twice ('config local'). The time-out may come into play, and you could hang your 8560 port.
- When an 8560 port is in IU mode and the attached device is powered down, the 8560 will not recognize a "hang up"; thus, processes that attempt to write to that port will hang.

SETUP: Connect the 8560 port to the 8550 J101 (Remote DTE Port). Match the Baud Rates. Configure that 8560 port for HSI protocol by typing the following:

```
> COM [cr]
```

(now log on to the 8560, which will prompt with the '$' sign) This could have been pre-established for that port, and as such this would not be necessary. You could then immediately log on under 'config term'.

```
$ stty IU [cr]
```

(a framing error message will appear and the DOS-50 prompt '>') This configures that port to be HSI, if not done so already.

```
> config term t=7 [cr]
```

(the 8560 will now respond to any keyboard input)

OPERATION: The "t" modifier sets a time-out limit, and will not affect the speed of operation for the transfer commands. Without the "t" parameter, transfers will hang. You now may issue either 8550 or 8560 commands to view, create, manipulate, or otherwise mutilate files. They generally will all work with the exceptions noted above. Let's say you now have a good object file on the 8560 called '8560file.obj'.

TRANSFERS: Following are some examples of file transfers. There may be more ways of doing this, so go ahead and experiment. But this at least will get you started.

8560 BINARY FILE TO 8550 MEMORY

To move a file called '8560file.obj' into 8550 memory, type:

```
$ lo -b <8560file.obj [cr]
```

(The system will prompt again waiting for the next command) You may now issue various debug commands, or execute, or...

8550 BINARY FILE TO 8550 MEMORY

To load a local file called '8550file.obj' type:

```
$ lo -b "<8550file.obj"
```

8550 BINARY FILE TO 8560 BINARY FILE

To send the file called ‘8550file.obj‘ to the 8560, type:

```
$ con -b "<8550file.obj" >8560file.obj
```

8550 ASCII FILE TO 8560 ASCII FILE

To send a file called ‘8550ASCII.FILE‘ to the 8560, type:

```
$ con "<8550ASCII.FILE" >8560ascii.file
```

8560 ASCII FILE TO 8550 ASCII FILE

To send a file called 8560ascii.file to the 8550, type:

```
$ con <8560ascii.file ">8550ASCII.FILE"
```

I hope this helps.

John Owens MDP Customer Support

CRON HAS ITS OWN CONCEPT OF TIME

If you have entries in /usr/lib/crontab that don't appear to be executing at the correct time, it may be because "date" and "cron" do not always agree on the present date and time.

The only time that "cron" consults "date" for the correct date and time is when "cron" is started. From then on, "cron" keeps track of time from it's beginning. This usually occurs at system restart time as a command in the startup file "/etc/rc". If you change the date after entering multiuser mode, cron has already been started, and cron will not receive the new date.

Therefore, if you have to change the date and/or time, after boot-up, you should also kill and restart "cron". This is done as follows:

1. Login as "root" on console port (tty0).
2. Execute the command "ps -ax"
3. Execute a "kill -9" using the process id of the process "/etc/cron".
4. Re-invoke cron by entering the command "/etc/cron".

Re-booting the system will also synchronize "cron" and "date" since the date is asked for before cron is started.

Gordon Glathar, MDP Customer Support

USING CU WITH A HOST

The cu (call UNIX) program which is part of the Unicom package for the 8560, is a program which allows a user on one 8560 to "call up" and login on another Unix system. The program also allows the transfer of ascii files between the systems. Although cu was intended to be used between two Unix systems, it is possible to establish communications with systems using other operating systems as well. The following describes some of the requirements of the host computer to allow communication using cu.

1. Data Format

   To avoid parity problems when using cu, the host should be configured to ignore parity (the 8th bit). The parity bit is ignored on input to cu.

   The data format on an 8560 is always 1 start bit, 8 data bits, and 2 stop bits. Parity, when specified, replaces the 8th data bit. One of the requirements of cu is that the non-login port which is interfaced to the host, be placed in "raw" mode. This is done to allow each character, which is sent by the host, to be passed directly to the cu program (the IOP normally waits for a <CR> or <LF> before sending the entire line to the requesting process). The primary characteristic of "raw" mode is that all input and output is passed unchanged. (see STTY(1), page 1-119 of the 8560 System Reference Manual) This means that all character translation and parity generation must be done by the process sending or receiving the data. For the most part, cu sends "mark" parity (8th bit always set). However, when using the "~$cmd" syntax, the resulting output will have the 8th bit cleared. This is because the "cmd" which is issued using this syntax is a child process of cu, and cu is unable add the parity bit.

2. Handshaking

   The host must have the ability to control data flow through the use of ctrl-S ctrl-Q (XON XOFF) software protocol, or CTS, DTR hardware protocol. These are selected in cu through the use of the -x (XON, XOFF) or -d (CTS DTR) options to cu.

3. Character Echo

   When attempting to perform file transfers between 8560 and host, it is recommended that character echo on the host be turned off. This is because the cu program may not be able to keep up with the echo. The actual file transfer will take place without data loss. Only the echoed characters will be lost. Suppressing the echo is desirable to prevent this misleading apparent loss of data.

4. File Transfer; 8560 to Host

   The main requirement of the host for performing file transfers is the ability to concatenate commands either in a single command line or a command file. The following is an example of the commands that would have to be sent to an 8560, acting as a host, to receive a file from an 8560 using cu. The equivalent commands for your host will need to be found.

   ```
   stty -echo; cat >filename; stty echo<CR>
   ```

   This command (actually three), turns off port echo, then copies from "standard input" to a file, and finally, restores port echo.

   At this point, the command issued to cu would be either "~<filename" or "~$cat filename". Either command results in the "filename" on the local system being transfered to the host system. In this particular example, a ctrl-D would need to be entered after the file transfer to terminate the "cat" command on the host 8560.

5. File Transfer; Host to 8560

   The procedure for transfering a file from the host to the 8560 is described on page 2-4 of the Unicom Users Manual. The procedure described is as follows:

   ```
   ~>[>][:]file
   0 or more lines of text entered to file from the terminal.
   ~>
   ```

The description implies that the commands are to be issued from the 8560 terminal. This is not the case; all lines must come from the host. If the host were another 8560, the commands which the host 8560 would execute would be as follows:

a. echo "~>:filename"

Causes cu to start redirecting input from the host to the file "filename".

b. cat hostfile

Copy the file to be transfered from the host.

c. echo "~>"

Send termination string to cu to terminate file transfer.

Again, the equivalent commands will need to be determined for your particular host.

An alternate method which could be used would be to edit the file on the host and place the "~>:filename" at the beginning of the file, and append the "~>" at the end of the file. Then when the file is copied, the transfer would take place without the need for multiple commands.

It should be noted that when cu sees the "~>:filename", keyboard input to cu is ignored until the transfer is complete. Because of this, creating commands on the host which sends the "~>:filename" sequence, or editing them in the file to be transfered, should not be done through cu, since the echo of the string will initiate a transfer.

The information presented here, in combination with Unicom Users Manual should provide you with enough information to establish communications with your host computer. If you need more assistance contact your local Applications Engineer.

Gordon Glathar MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

SYSTEM FLOATING POINT CHIP ROLE

1) AWK will not run without the floating point chip.

2) NROFF enjoys a 2 to 1 performance boost when the F.P. chip is present (That can be a real blessing !).

3) Programs written for the native C compiler as well as programs written in 11/23 assembly language utilize the mathchip.

The floating point chip option is a requirement for the purchase of the Auxiliary Utillities, Text Processing, and Native Programming tools.

Wolfgang Takatsch MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

USE OF 8540/HSI PROTOCOL/RS232 THROUGH MODEMS

It is possible for the 8540 to communicate with the 8560 in term mode over RS232 using modems. However, there are some limitations:

> *Use COM* This is the recommended mode of operation. COM in the 8540 and mload in the 8560 support this technique. Optional handshaking assures an error-free download of memory contents. The 8540 can communicate with the 8560 in transparent-terminal mode.
>
> *Term Mode* This involves using RS232 with HSI protocol between the 8540 and 8560. If you experiment, you may find it "generally" works, especially if you use a "t=20" on the "config term" command line. There is still a problem because the CTS and RTS signals used between the 8560 and 8540 mean different things to modems. In other words, with modems between the 8560 and 8540, the 8540 cannot tell the 8560 to suspend transmission and vice versa.
>
> If you choose to use this configuration, you should be aware that it is not currently a supported mode of operation.

NOTE: We do not recommend the use of modems between the 8540 and 8560 unless they are used in COM mode.

Byron Lunz MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

LINKER LISTINGS

The following shell scripts provide a very complete linker listing.

```
: "
:       This program takes the named object files (arguments)
:       and prepares five reports concerning the distribution
:       of symbols in the object modules. The reports
:       are
:               'Sections by Module Name'
:               'Sections Alphabetically'
:               'Data by Defining Module'
:               'Defined Data Alphabetically'
:               'Undefines by Calling Module'
:
:       Typical usage would be:
:
:               getsyms *.o | lpr
:
: "
lstr -o $* > SYMS
grep " [ASDCRl] " SYMS > DSYMS
grep " [uU] " SYMS > USYMS
: sed -e "s/:/ /" DSYMS > tmp$$
: mv tmp$$ DSYMS
: sed -e "s/:   / xxx /" USYMS > tmp$$
: mv tmp$$ USYMS
awk '/ [ASCR] / {printf("%-16s\t%-16s\t%10s\n", $1, $4,$2)}' \
     DSYMS | sort > DTSYMS
pr -h "Sections by File Name" DTSYMS
awk '/ [ASCR] / {printf("%-16s\t%10s\t%-16s\n", $4,$2, $1)}' \
     DSYMS | sort > DTSYMS
pr -h "Sections Alphabetically" DTSYMS
awk '/ D / {printf("%-16s\t%-16s\t%10s\n", $1, $4,$2)}' \
```
~~~~~~~~~~~~~~~~~~~~

```
      DSYMS | sort > DDSYMS
pr -h "Data by Defining File" DDSYMS
awk '/ D / {printf("%-16s\t%10s\t%-16s\n", $4,$2, $1)}' \
      DSYMS | sort > DDSYMS
pr -h "Defined Data Alphabetically" DDSYMS
awk '/^/ {printf("%-16s\t\t\t%-16s\n", $1, $4)}' \
      USYMS | sort > USYMS1
pr -h "Undefines by Calling File" USYMS1
awk '/^/ {printf("%-16s\t\t\t%-16s\n", $4, $1)}' \
      USYMS | sort  > USYMS2
pr -h "Undefines Alphabetically" USYMS2
rm -f USYMS1 DDSYMS DTSYMS USYMS DSYMS SYMS USYMS2
```

```
    : "
    :         This program takes a loadfile and uses the lstr program
    :         and  TNIX tools to produce a traditional link
    :         map.  Typical usage would be:
    :
    :                  linkmap loadfile | lpr
    : "
    lstr -sn $1 >sym
    grep " [ASDCRl] " sym >dsym
    awk '/ [ASCR] / {sec = $3; \
                    printf("%s\t%10s %s %-16s\t%10s\n",sec,$1,$2,$3,$4)} \
       / [Dl] /   {printf("%s\t%10s %s %-16s\n",sec,$1,$2,$3)} \
       ' dsym | sort +0 -2 > secsym
    awk '/ [ASCR] / {printf("\n\nSection %-16s\tLocation \
                  = %10s\tSize = %10s\n", \
                       $1,$2,$5)} \
            / [Dl] / {printf("\t%-16s\t%10s\n",$4,$2)} \
            ' secsym  >map
    pr -h "Linker Map for $1" map
    rm sym dsym secsym map
```

The Auxiliary Utilities Package must be installed for the above to be implemented.

John Owens   MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

PRINTING ON AN 8540 BASED PRINTER

1 To print an 8560 file on the printer attached to the 8540, enter "config term" on the 8540 terminal and then enter:

$ cop filename LPT (CR)

The "$" is the prompt.The above command line works no matter if you have the terminal connected to the 8540 or the 8560.

2 When printing a file downloaded from a HOST you will be in "COM" mode communication and the "redirect" symbol ">" can be invoked to route what comes down the line to the line printer. HOWEVER, under those circumstances there will be absolutely no handshaking between the printer and the host computer. Result? well, your printer must keep up with what the computer is sending to it (i.e. it must be buffered; a slow baud rate used all the way down the chain) or else data will be lost.

3 What about printing an 8540 generated display ? All that needs to be done is cause the 8560 to ignore a re-direct, like this:

$ ts \>LPT (CR)
~~~~~~~~~~~~~~~~~~~~

The backslash flags the 8560 to send the next item to the 8540. In that way then ">" ends up as part of the 8540 command.

Wolfgang Takatsch MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

NOTIFICATION OF INCOMING MAIL

If you would like to be notified as soon as someone sends you mail on an 8560, a shell variable causes this to occur. The variable is "MAIL". To set it up, simply enter the following in your .profile:

```
MAIL=/usr/spool/mail/username ;export MAIL
```

Substitute your login name for "username" above. Once this is done, and someone sends you mail, the shell will issue the message:

```
you have mail
$
```

The notification will not occur until after the completion of a command. For example, if you are in the middle of an editing session, and you receive mail, you will not be notified until you exit the editor.

Gordon Glathar MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

CREATE YOUR OWN MANUAL PAGES

One of the most powerful features of TNIX is the ability of users to create tools. This power is diminished if others cannot easily learn to use the tools. Tools may consist of simple or complex shell scripts, or programs created with the native programming tools. Such tools can simplify and personalize the use of TNIX based systems.

Locally created tools can provide capabilities such as simplified command entry to complex database management. For example, if you use consistent file naming rules, your assembler command could consist of:

```
/bin/asm $1.obj $1.lst $1.src
```

Now your assembler command requires only one parameter, not three.

Once the tools are created, other users can benefit from their use. Locating the new commands in /usr/bin does make them available; but if others don't know how to use them, they will not be used.

The first step in aiding others in the use or your command is to provide a manual page. The second step is to include the manual page in the man command. The following is an nroff template for creating a manual page.

```
.de PT
.tl <^G>\\*(LH<^G>\\*(CH<^G>\\*(RH<^G>
 \"  Where <^G> in the above line is replaced with the control G character.
.HL
..
.ds LH 8560 LOCAL SYSTEM COMMANDS
.ds CH
.ds RH [ command ](9)
.SH
NAME
.EH
.IP
[ name of command and function ]
.SH
SYNTAX
.EH
```

```
.IP
[ actual form of the command and parameters ]
.SH
DESCRIPTION
.EH
.IP
[ description of command function and any options ]
.SH
Also See
.EH
.IP
[ alternate references (if none remove this and the above 4 lines) ]
.SH
FILES
.EH
.IP
[ files required/expected/accessed by the command ]
.SH
Author
.EH
.IP
[ your name and any applicable references ]
.SH
Notes
.EH
.IP
[ Warnings here  (if none remove this and the above 4 lines) ]
```

The output from nroff of the above template for your command can then be located in /usr/man/cat9/<mycommand>.9, where <mycommand> is the name of the command created. Cat1 through cat8 correlate to the sections of the "System Reference Manual," thus a new directory for local commands is advised. The man command will have to be changed to find entries in the /usr/man/cat9 directory. The only changes required are on line 5 and 18. Change [1-8] to [1-9] as shown below.

```
error=0
cd /usr/man
sec='?'
case "$1" in
   [1-9]): 'section number to examine'
        sec="$1"
        shift;;
esac
if test -z "$1"
then
        echo "man: missing string argument"
        echo "Usage:"
        echo "   man [chapter] title ..."
        exit 1
fi
for i
do
        for z in cat$sec/$i.[1-9]*
        do
        if test -r "$z"
        then cat $z
        else
           echo "man: can't find information for \"$i\"" 1>&2
           error=1
        fi
        done
```

```
done
exit $error
```

Note that the man command itself is a short shell script. With the above suggestions implemented, all users of the system can benefit from the tools created by others.

John Owens MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

MULTIPLE EMULATOR CONTROL

The 8560 is an excellent multiple emulator controller which can control up to seven emulators from one terminal. The software to perform this control function consists of a configuration program that allows the user to define logical units with numbers or names and assign each to a specific 8540 Integration Unit. The configuration program creates an executable program called "mltem" which prompts the user for the start address and a output filename for each Integration Unit in the system. Once this information is entered, "mltem" issues a prompt "em:". The user may then enter any TNIX command for direct execution by the 856x. Commands may be sent to Each emulation unit by prefacing the command with "lu<unit number or name >" (assigned during configuration).

For example:

```
em: lumain d 200 2ff
```

will send the dump command to the emulation unit named "main".

Using this procedure, trace may turned on or off, breakpoints can be set, memory dumped, code disassembled etc. In fact all 8540 commands can be executed for each emulator in the multiple emulation system. Once the Integration Units are set up as desired the command "goall" starts all the emulators, "halt" stops all the emulators, and "cont" continues all the emulators from wherever they were previously halted. Since emulators are started sequentially in order of assignment (determined during configuration), synchronizing the emulators requires a routine executed by the emulators themselves, or the use of prototype control signals such as halt or reset. This should be no problem, as long as the user is aware of the limitation.

The following is the content of the "setup" shell script that is used to create a multiple emulation shell script.

```
pa=‘pwd‘
echo This program configures the multiple processor control program
echo
kil=""
echo '     halt) : ' >$pa/halt1
echo 'echo "Program to control multiple emulators and gather data" '>$pa/ports
echo '     goall) : ' >$pa/goall
echo '     cont) : ' >$pa/cont
N=‘expr 0‘
if test $1
then
  for i
  do
          ID=true
          while $ID  ;do
                    echo 'enter physical port number for Logical Unit '$i
                    read U$i
                    eval ID1=$"U$i"
                    case "$ID1" in
                              [1-7]) ID=false ;;
                              *) echo 'use physical port numbers in range 1 to 7 only' ;;
                    esac
          N=‘expr $N + 1‘
          done
```
~~~~~~~~~~~~~~~~~~~~

```
        kil="$kil ?K$i"
        echo '              IU='$ID1' g $A'$i' $D'$i' &' >>$pa/goall
        echo '              IU='$ID1' g $D'$i' &' >>$pa/cont
        echo '              K'$i'=$!;export K'$i >>$pa/goall
        echo '              K'$i'=$!;export K'$i >>$pa/cont
        echo '      lu'$i') eval IU='$ID1' "$EB"  ;;' >>$pa/lu
        echo 'echo Enter start address and filename for output for Logical Unit '$1 >>$pa/ports
        shift
        echo 'echo Include redirect ">" preceding filename:' >>$pa/ports
        echo 'read A'$i' D'$i >>$pa/ports
 done
else
        echo Error:  Invoke with desired Logical Unit numbers | names) as parameters
        exit
fi
echo 'echo you have configured '$N' emulators'>>$pa/ports
echo '          ;;'         >>$pa/goall
echo '          ;;'         >>$pa/cont
echo '          kill -2 '$kil >>$pa/halt1
echo '          ;;'         >>$pa/halt1
tr '?' '$' <$pa/halt1 >$pa/halt
rm $pa/halt1
cat $pa/ports $pa/emul1 $pa/goall $pa/cont $pa/halt $pa/lu $pa/emul2 >$pa/mltem
rm $pa/ports $pa/goall $pa/cont $pa/halt $pa/lu
chmod 777 mltem
```

The following shell script files are used by "setup" to create the "multem" shell script that actually controls the emulators.

- emul1

  ```
  while true ; do

    echo -n "em:"
    read EA EB || exit
    case "$EA" in
  ```

- emul2

  ```
    *) eval $EA $EB ;;
    esac
  done
  ```

When "setup" is executed the following file is created.

```
echo "Program to control multiple emulators and gather data"
echo Enter start address and filename for output for Logical Unit 1
echo Include redirect ">" preceding filename:
read A1 D1
echo Enter start address and filename for output for Logical Unit 2
echo Include redirect ">" preceding filename:
read A2 D2
echo you have configured 2 emulators
while true ; do

        echo -n "em:"
        read EA EB || exit
        case "$EA" in
        goall) :
                IU=2 g $A1 $D1 &
                K1=$!;export K1
```

```
                        IU=3 g $A2 $D2 &
                        K2=$!;export K2
                        ;;
                cont) :
                        IU=2 g $D1 &
                        K1=$!;export K1
                        IU=3 g $D2 &
                        K2=$!;export K2
                        ;;
                halt) :
                        kill -2  $K1 $K2
                        ;;
                lu1) eval IU=2 "$EB"  ;;
                lu2) eval IU=3 "$EB"  ;;
                *) eval $EA $EB ;;
                esac
done
```

When "mltem" is executed, you can control several emulators at one time and direct commands to any of the emulators as needed; i.e., full access and control over multiple emulators and all the normal TNIX support as well. The shell script adds three new emulation control commands.

- goall - sends a go to all emulators.
- cont - sends a go from current address to all emulators.
- halt - halts the emulators.
- lu(name) <8540 command> - sends 8540 command to the named integration unit.

One note of caution. *The startup and stopping of emulators is not simultaneous.*

Lee Dilley Sales Engineer, Philadelphia Field Office

NEW EMULATOR SPECIFIC COMMANDS

Just a brief reminder that your 8560 needs to be informed of any NEW COMMANDS that it should direct to the 8540.

For example: sndp. To support the 8087 a new set of emulator control software/firmware must be installed on your 8540 (or 8550). This new software allows the use of a new, 8087 specific command called "sndp" (Set Numeric Data Processor). This command allows the user to set values into various 8087 registers, to format the 8087 specific display and so forth. But, if you are running your 8540 from an 8560, you have to make sure that your 8560 knows that "sndp" is a valid command. All that is needed is to link "sndp" to an existing, valid 8540 command. To do this you must be logged in as root and be in "/bin". Type "ln 8540 sndp" followed by a carriage return and the job is done.

The above will hold true for all NEW COMMANDS that come along for the 8540. (See 6801/68120 Emulator Specifics manual, page 7M-36 for example.)

Wolfgang Takatsch MDP Customer Support

SIZE OF 8540 SYMBOL TABLE

Users who develop their software on a host computer need to know how many symbols they can download into their 8540. Of course no exact number can be given because it depends on the size of the symbols. However, here are a few specifics that should be helpful.

Essentially there are 11.5K bytes available in all. The overhead for each symbol is 8 bytes for symbols and 12 bytes for section names.If N is the number of bytes in the symbol then the following formula holds true:

- N + 8 = number of bytes for each symbol.
- N + 8 + 4 = number of bytes for each section name.

So, if your symbols are 5 bytes long and you have only symbols then you could have as many as 880(est.) symbols.

Wolfgang Takatsch MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

PASCAL DEBUG/TERMINAL CONNECTION

Since Pascal Debug requires I/O to/from the console, the best configuration is to connect the terminal to the 8540 or 8550 which is connected to the 8560. If you want to connect the terminal to the 8560, you must connect another terminal (null connector) to the console port on the 8540.

There have been some reported problems with 8086 PDB used with the 8550. Hardware breakpoints, which include "STEP", "TRACE", breakpoints on variables, can cause reboot problems. The next version of 8086 PDB (a mod is planned this summer) should alleviate the problem. 68000 PDB should work with the 8540 or the 8550, however, we recommend the following to change the timeout in the 8550:

```
> config term t=7
```

Marilyn Hanson MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

PASCAL RUNTIME ERROR MESSAGE

The 8086 Pascal Users Manual omits the following runtime error message:

02	Case index is not specified in any case constant list and no otherwise clause is specified

Marilyn Hanson MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~

THE GO COMMAND AND LP1R

QUESTION: When using my 8560/8540 I enter trace all and then "g|lp1r" nothing goes to the printer. I entered ^C to stop the emulator. When I enter "g >file" it does work.

ANSWER: When you send a ^C to the emulator you also send it to all foreground processes. That includes the lp1r process. Lp1r does not send anything to the printer until the input stream terminates; i.e., lp1r puts the entire output stream in a file and when it gets a EOT signal it then starts the print process. That can't happen when you stop emulation (and all other processes) with a ^C. The output redirection works because the control C simply causes the process to close the file (send control D).

John Owens MDP Customer Support

SOME USEFUL C PROGRAMS

Mdisk is a Motorola disk reader. This version of mdisk operates correctly only on text files for Exordisk II*. The basic syntax is mdisk option [filename]. The available options are :

d - directory
t - suffix type
a - all files
f - one file

Mdisk was developed by Eric Osborne and Jeff Meyers.

The following is the "C" language source for the mdisk command.

```
/* 6800.c    Jan. 19, 1983    jpm
             By which ASCII-format files from the EXORciser can be
             read in to the Tek. and (after conversion from space-
             compression to tabbing) written to disk                    */

#include <stdio.h>
#define TRUE 1
#define FALSE 0

FILE  *fp,*fo, *fopen();
char         buffer[128], fixname[11],cluster[512];
long     offset;
int          ribpsn;

main(argc,argv)
int argc;
char *argv[];
{
             int sector, row, letter;
             int c, spot;
             char *argptr;

             fp = fopen("/dev/fd0","r");
             if ((argc < 2) || (argc > 3) || (strlen(argv[1]) != 1))
                          error();
             argptr = argv[1];
             switch (*argptr)
             {            case 'a':
                                       if (argc != 2)
                                                    error();
                                       all();
                                       return;
                          case 't':
                                       if (argc != 3)
                                                    error();
                                       type(argv[2]);
                                       return;
```

* Trademark of Motorola Corp.

```
                case 'f':
                        if (argc != 3)
                                error();
                        fixfilename(argv[2]);
                        onefile(fixname,argv[2]);
                        return;
                case 'd':
                        if (argc != 2)
                                error();
                        directory();
                        return;
                default:
                        error();
                }


}

error()
{       printf("ERROR....the options are:\n");
        printf("    d                               (print directory)\n");
        printf("    a                       (whole directory)\n");
        printf("    t  followed by a suffix    (directory files of one type)\n");
        printf("    f  followed by a file name  (one file)\n");
        exit(1);
}

onefile(filename,name2)
char *filename,*name2;
{
        int count, c,i,j,k, found, row, sector;
        found = FALSE;
        for (sector=3;sector<23;sector++)
        {       getsector(sector);
                for (row=0;row<8;row++)
                {       if (strncmp(filename,&buffer[16*row],10) == 0)
                        {   found = TRUE;
                            ribpsn=buffer[16*row+10];
                            ribpsn *= 256;
                            ribpsn += (buffer[16*row+11] & 0x00ff);
                            break;
                        }
                }
                if (found == TRUE)
                        break;
        }
        if (found == FALSE)
        {       printf("\n%11s not found on this diskette.\n",name2);
                exit(1);
        }
        fo=fopen(name2,"w");
        getsector(ribpsn);
        i=0;
        while   ((buffer[2*i] & 0x00ff) <128)
        {
                sector=(buffer[2*i+1] & 0x00ff);
                sector +=((buffer[2*i] & 0x0003)* 256);
                count=(buffer[2*i] & 0x00fc)/4;
                ++count;
                for     (;count>0;–count)
                {
                        if      (ribpsn==(4*sector))
                                k=128;
                        else
                                k=0;
                        getcluster(sector);
                        ++sector;
                        for     (;k!=512;++k)
                        {
                                c=cluster[k] & 0x00ff;
                                if      (c < 128)
                                {
                                        if      (c==13)
                                                c=10;
                                        if      (c==0)
                                                break;
                                        putc(c,fo);
                                }
                                else
                                {
                                        for     (c -=128;c >0; –c)
```

```
                                                putc(' ',fo);
                                }
                        }
                }
                ++i;
        }
        fclose(fo);
}

getsector(i)
int i;
{
        int count;
        offset = i*128l;
        fseek(fp,offset,0);
        for (count=0;count<128;count++)
                buffer[count] = getc(fp);
}

getcluster(i)
int i;
{
        int count;
        offset = i*512l;
        fseek(fp,offset,0);
        for (count=0;count<512;count++)
                cluster[count] = getc(fp);
}

type(name)
char *name;
{
        int sector, row, count, i, j, periodin;
        char c, nameformat[12], *temp;
        char fixname[12], *temp2;
        temp = nameformat;
        temp2 = fixname;

        for (i=0;i<2;i++)
                name[i] = name[i] & '\337';
        for (sector=3;sector<23;sector++)
        {       getsector(sector);
                for (row=0;row<8;row++)
                {       c = buffer[16*row];
                        if ((c == '\0') || (c == '\377'))
                                continue;
                        if (strncmp(name,&buffer[16*row+8],2) != 0)
                                continue;
                        fixname[0] = c;
                        for (i=1;i<10;i++)
                                fixname[i] = buffer[16*row + i];
                        fixname[10] = '\0';
                        j=0;
                        periodin = FALSE;
                        for (i=0;i<11;i++)
                        {       if ((i==8) && (periodin == FALSE))
                                {       periodin = TRUE;
                                        nameformat[j++] = '.';
                                }
                                if (fixname[i]==' ')
                                {   if (periodin)
                                        continue;
                                    else
                                    {   periodin = TRUE;
                                        nameformat[j++] = '.';
                                        continue;
                                    }
                                }
                                nameformat[j++] = fixname[i];
                        }
                        for (i=0;i<11;i++)
                        {       c = nameformat[i];
                                if ((c>='A')&&(c<='Z'))
                                        nameformat[i] = c | 0x20;
                        }
                        onefile(fixname, nameformat);
                }
        }

}
all()
```

```
{
        int sector, row, count, i, j, periodin;
        char c, nameformat[12], *temp;
        char fixname[12], *temp2;
        temp = nameformat;
        temp2 = fixname;

        for (sector=3;sector<23;sector++)
        {       getsector(sector);
                for (row=0;row<8;row++)
                {       c = buffer[16*row];
                        if ((c == '\0') || (c == '\377'))
                                continue;
                        fixname[0] = c;
                        for (i=1;i<10;i++)
                                fixname[i] = buffer[16*row + i];
                        fixname[10] = '\0';
                        j=0;
                        periodin = FALSE;
                        for (i=0;i<11;i++)
                        {       if ((i==8) && (periodin == FALSE))
                                {       periodin = TRUE;
                                        nameformat[j++] = '.';
                                }
                                if (fixname[i]==' ')
                                {   if (periodin)
                                        continue;
                                    else
                                    {   periodin = TRUE;
                                        nameformat[j++] = '.';
                                        continue;
                                    }
                                }
                                nameformat[j++] = fixname[i];
                        }
                        for (i=0;i<11;i++)
                        {       c = nameformat[i];
                                if ((c>='A')&&(c<='Z'))
                                        nameformat[i] = c | 0x20;
                        }
                        onefile(fixname, nameformat);
                }
        }

}
fixfilename(namein)
char *namein;
{       int count, i,j;
        char nameout[11],*temp;
        temp = nameout;
        while (*namein)
        {
                if ((*namein > 96) && (*namein < 123))
                        *temp++ = (*namein++ & 0xdf);
                else
                        *temp++ = *namein++;
        }
        for (i=0;i<11;i++)
                fixname[i] = ' ';
        i=0;
        while (nameout[i] != '.')
        {       fixname[i] = nameout[i];
                i++;
        }
        fixname[8] = nameout[++i];
        fixname[9] = nameout[++i];
        fixname[10] = '\0';
}
        /* Read and present the Directory...   */


directory()
{
        int sector,row;
        char name[9],suffix[3];
        for (sector=3;sector<23;sector++)
        {       for (row=0;row<8;row++)
                {       offset = 128l * sector + 16 * row;
                        fseek(fp, offset, 0);
                        fscanf(fp,"%8s%2s",name,suffix);
                        if ((name[0] != 0xffff)&&(name[0] != 0))
```

```
                                                printf("%-8s.%2s\n",name,suffix);
                                }
                }
}
```

Hexdump was written by Eric Osborne of Thurman Scale Co. Hexdump provides the same function as the TNIX command "od" but provides a more readable output. The command uses standard input.

```
#define EOF -1
#define LF 10

main()
{
        int c,i,j,temp[16];
        long address
           printf("\n");
        c = address = 0;
        while (c != EOF)
        {
                printf("%06lx  ",address);
                for (i=0;i<16;++i)
                {
                        c=getchar();
                        if (c == EOF)
                                break;
                        printf("%02x ",c);
                        temp[i]=c;
                }
                address=address+16;
                if (i<16)
                {
                        for (j=16-i;j>0;--j)
                        {
                                putchar(' ');
                                putchar(' ');
                                putchar(' ');
                        }
                }
                printf("    ");
                for (j=0;j<16;++j)
                {
                        if (j>i)
                                putchar('.');
                        else
                        {
                                if ((temp[j] & 127 ) > 31)
                                        putchar((temp[j] & 127));
                                else
                                        putchar('.');
                        }
                }
                printf("\n");
        }
}
```

The following "C" program also written by Eric Osborne converts an Intel Hexfile to Rockwell hex format.

```
#include <stdio.h>

#define LF 10
#define CR 13

FILE *fopen(),*infile;
int        records=1;
main(argc,argv)
char       *argv[];
int        argc;
{
        int address,offset,i,j,c;

        if      (argc>2)
        {
                fprintf(stderr,"\nusage: i2r [file]\n");
          exit();
        }
        if      (argc==2)
        {
                if      ((infile=fopen(argv[1],"r")) == NULL)
                {
                        fprintf(stderr,"\ni2r: cannot open %s\n",argv[1]);
                        exit();
                }
                fprintf(stderr,"\nenter load address offset in hex : ");
                scanf("%x%*1c",&offset);
        }
        else
        {
                infile=stdin;
                offset=0;
        }
        c=0;
        while (c != EOF)
        {
                c=getc(infile);
                if (c != EOF)
                {
                        fscanf(infile,"%2x %4x %*2c",&i,&address);
                        if (i != 0)
                        {
                                if (i > 16)
                                {
                                        j=(i);
                                        i=16;
                                        rec(address+offset,i);
                                        address+=16;
                                        i=(j-16);
                                        rec(address+offset,i);
                                }
                                else
                                {
                                        rec(address+offset,i);
                                }
                                fscanf(infile,"%*3c");
                        }
                        else
                        {
                                c=(EOF);
                        }
                }
        }
        printf(";00%04x%04x\n\032\032\n",records,records);
        printf(";00%04x%04x\n\032\032\n",records,records);
}
rec(address,i)
int address,i;
{
        int d,sum;
        sum=(i+((address & 0xff00)/256) + (address & 255));
        printf(";%02x%04x",i,address);
        for (;i!=0;–i)
        {
                fscanf(infile,"%2x",&d);
                sum += d;
                printf("%02x",d);
        }
        printf("%04x\n",sum);
        ++records;
}
```

Eric Osborne and Jeff Meyers   Thurman Scale Co

CONSTANTS LOCATION WITH 8086 PASCAL

Version 1 of the 8086 Pascal Compiler is limited to 64K of instructions (accessed by the Code Segment) and 64K of data (accessed by the Stack Segment, the Data Segment, and the Extra Segment).

The instructions are typically placed in ROM in the final application. This presents no problem because it is not unusual to have a large area of ROM for this purpose designed into an application.

However, the 64K of data is a different story. The data segment is made up of HEAP_STACK_RAM, GLOBA_VAR_RAM, and CONSTANTS_ROM. The heap-stack and global variables must be in RAM but the constants must be in ROM. In some design situations it may not be possible (for whatever reasons) to have the RAM and ROM in the same 64K segment. It is probable that applications will have 64K of RAM and the ROM will be located in some other part of the address space, as in the following example:

```
+--------------+   0FFFFFH          ICS SOURCE
|     64K      |
|     ROM      |               INSTRUCTIONS_ROM   [0F0000H,0FFFFFH]
+--------------+   0F0000H CONSTANTS_ROM          [00F000H,00FFFFH]
|              |           GLOBAL_VAR_RAM         [000000H,007FFFH]
|              |           HEAP_STACK_RAM         [008000H,00EFFFH]
+--------------+
    .
    .
    .
+--------------+   00FFFFH
|     64K      |
|     RAM      |
+--------------+   000000H
```

In this case Instructions will be placed in the ROM located at 0F0000H - 0FFFFFH and the Data (including heap/stack, global variables, and constants) will be placed in the RAM at 000000H - 00FFFFH. The problem now clearly presents itself. The constants are NOT in ROM as is desired. To place them in a ROM somewhere else in the memory space would not fulfill the run-time requirement that all Data must reside in ONE 64K Segment.

SOLUTION:

The "solution" is to store the CONSTANTS_ROM in a ROM (somewhere in memory, even in the code segment, if there is room) and copy them at ICS initialization time to the RAM location specified in the ICS directive CONSTANTS_ROM. We now use the example as stated above with the addition of available ROM "somewhere" in the address space.

```
+--------------+   0FFFFFH                    ICS SOURCE
|     64K      |
|     ROM      |              INSTRUCTIONS_ROM   [0F0000H,0FFFFFH]
+--------------+   0F0000H    CONSTANTS_ROM      [00F000H,00FFFFH]
|              |              GLOBAL_VAR_RAM     [000000H,007FFFH]
|              |              HEAP_STACK_RAM     [008000H,00EFFFH]
+--------------+
  .
  .
+--------------+   0A0FFFH
|      4K      |         <--Available Rom (to be used to hold constants
|      ROM     |               that will be moved to CONSTANTS_ROM section
+--------------+   0A0000H     which must be mapped into RAM space)
  .
  .
+--------------+   00FFFFH
|     64K      |
|     RAM      |
+--------------+   000000H
```

1. Step 1

Change the ICS Directive RESTART_LABEL (See Compiler Users Manual, Section 5) to call a user supplied assembly language routine which will copy the CONSTANTS from the Available ROM to the RAM location specified in the ICS Directive CONSTANTS_ROM.

```
RESTART_LABEL        CNST_CPY
```

2. Step 2

Write the assembly program CNST_CPY. For this example, we will put the assembly program in a file called "copy.asm".

```
         SECTION CNST_CPY_SEC,CLASS=INSTRQQ
         GLOBAL  CNST_CPY,PASCAL_BEGIN,CODEBASEQQ
CNST_CPY XORW    SI,SI              ; Clear the SI register
         MOVW    DS,#0A000H         ; Location of NEW ROM / 16
         MOVW    DI,SI              ; Clear the DI register
         MOVW    ES,#00F00H         ; Destination of CONSTANTS_ROM /16
         MOVW    CX,#01000H         ; Size of CONSTANTS_ROM (0FFFFH-0F000H+1)
         CLD
         REPZ    MOVB               ; Block transfer
         JMPS     PASCAL_BEGIN,CODEBASEQQ  ; Transfer to ICS initialization
```

3. STEP 3

Assemble the program CNST_CPY and produce an object module.

```
asm copy.obj copy.lst copy.asm
```

4. STEP 4

Add another MODULE directive to your ICS file.

```
MODULE          copy.obj
```

5. STEP 5

Rebuild your load file. Burn ROM's as before, except now place the CONSTANTS in ROM at the Available ROM Location.

ALTERNATE SOLUTION:

In the previous example the Available Rom was located at an arbitrary address. In many cases it may be necessary to reserve some of the ROM used for the code space for the constants. This alternative requires several small changes from the previous example. The assembly program CNST_CPY will have the appropriate changes for the new addresses involved. The ICS source will also have to be changed to prevent INSTRUCTIONS_ROM from using the space now reserved for constants. For example:

```
+--------------+   0FFFFFH              ICS SOURCE
|     60K      |
|     ROM      |              INSTRUCTIONS_ROM    [0F1000H,0FFFFFH]
+--------------+   0F1000H    CONSTANTS_ROM       [00F000H,00FFFFH]
|     4K       |              GLOBAL_VAR_RAM      [000000H,007FFFH]
|     ROM      |              HEAP_STACK_RAM      [008000H,00EFFFH]
+--------------+   0F0000H
       .
       .
       .
+--------------+   00FFFFH
|     4K       |
|     RAM      |
+--------------+   00F000H
|     60K      |
|     RAM      |
+--------------+   000000H
```

Marilyn Hanson MDP Product Marketing

CHANGING THE NAME OF MULTIPLE FILES IN A DIRECTORY

It may become necessary at times to change a large number of files to have a different suffix or some other common portion; e.g., change all files ending in .px to end in .ps. The mv command does not accept wildcards in its filenames. Thus "mv *.px *.ps " does not work.

Following are two methods of accomplishing this task.

```
for i in *.px
do
        variable=‘basename $i.px‘
        mv ${variable}.px ${variable}.ps
done
```

or

```
ls *.px|sed 's/\(.*\)./mv & \1s/'|sh
```

Doug Johnson MDP Product Marketing

SETTING LINE PRINTER EOL CHARACTERISTICS

The "slp" command of TNIX is a command which allows you to set the output characteristics of the line printer port. There is, however, an error in the documentation on this command. In the "commands" section of the System Reference Manual an example is given which changes the new line (nl) string to a null form feed. The syntax of this example should be nl=" 00 14". Note the quotes.

There is a problem in the usage of the nl=string parameter of slp. I was unable to send a true null as part of the EOL string. The 8th bit was always set when a null was sent. If the printer you are using uses 7-bit ascii, there won't be a problem. However, if you are using a Tektronix 4643 printer, the 80H will be printed as some form of Egyptian hieroglyphics. The 4643, however, doesn't require a null as part of the EOL string.

The default settings are -nl and -tabs. This means that the EOL is <CR><LF> and tabs are expanded to spaces. If you are using the 4643 printer, you may wish to change from -nl to nl (from <CR><LF> to <LF> only). This appears to speed up printing somewhat. If you wish to make this change permanent, put the command slp /dev/lp1 nl<CR> in the file /etc/rc.

Gordon Glathar MDP Customer Support

MODIFY THE SPELL DICTIONARY ON 8560

The following shell procedure files provide a user friendly interface to add local words (add.words) to the spelling dictionary and to add local stop words (words you wish to fail) (sub.words).

```
     /usr/bin/add.words
cd /usr/dict
echo 'Input new words one per line.  Terminate list with ^d.
  if test -f local.words
  then cat >> local.words
  else echo '' > local.words ; chmod a+w local.words
  fi
echo -n 'Adding words takes some time.  Run in background? (y/n)'
read AA
  if test $AA = 'y'
  then echo New words will be added in background
  /usr/bin/add.words.sub &
  else /usr/bin/add.words.sub
  fi

     /usr/bin/add.words.sub
nice -20 cat local.words american | spellin hlist > hlista.$$
mv hlista.$$ hlista

               /usr/bin/sub.words
          cd /usr/dict
          echo 'Input new words to be removed one per line.  Terminate list with ^d.'
            if test -f local.stop
            then cat >> local.stop
            else cat > local.stop ; chmod a+w local.stop
            fi
          echo -n 'Removing words takes some time.  Run in background? (y/n)'
          read AA
            if test $AA = 'y'
            then echo New words will be removed in background
            /usr/bin/sub.words.sub &
            else /usr/bin/sub.words.sub
            fi

     /usr/bin/sub.words.sub
cat local.stop stop > stop.$$
nice -20 spellin < stop.$$ > hstop.$$
mv hstop.$$ hstop
nice -20 rm stop.$$
```

Sam Crow MDP user (904) 882-3840

~~~~~~~~~~~~~~~~~~~~

FILES USED BY UUCP... AN OVERVIEW

The installation of Unicom and the configuration procedure for uucp is documented in the Unicom manual., The intent of this article is to describe the main files and directories which are created or modified by the installation procedure or the uucp-config program. Where appropriate, manual page references are included for further information.

The installation procedure creates three major directories which are used by uucp. They are:

1. /usr/lib/uucp

   This directory contains all the commands required by uucp. It also contains files which contain information about the network environment (who, what, when, where, how ..to call). The main configuration files in this directory are:

   a. L.sys (3-15, 3-24, 6-29)

      This file provides information about the remote systems. If the remote system is a slave, the entries will contain information such as when to call, on what port, baud rate, login information, etc. If the remote system is a master the entries will contain the remote masters that will be calling, on what ports, etc.

   b. L-devices (6-27)

      This file contains the ports used by uucp and the device types which are connected to them (autodial modems, direct connect RS232 or HSI).

   c. USERFILE (3-15, 3-24, 6-28)

      This file is used by uucp to determine if a user (using uucp) has permission to access the directories he is trying to access.

   d. L-xcmds (2-13)

      This file contains a list of commands which may be executed by a uux command.

2. /usr/spool/uucp

   This directory contains work and command files used by uucp. It also contains files which contain history information on uucp activity. This is also where status files are placed when uucico (the actual file transfer process) is running. These files normally go away when uucico completes. They remain if problems were encountered in the communication. Refer to section 6 of the unicom manual for more information on these status files. (6-31, 6-32)

3. /usr/spool/uucppublic

   This is the login directory for uucp. This directory is a public (read/write by anyone) directory which can be used by calling systems to place files when an alternate path is not known or not permitted. Refer to page 2-8 in the Unicom manual for details.

Other files which are created or modified by the installation procedure are /usr/include/whoami.h, /bin/mail, /usr/lib/crontab, and /etc/ttys.

- /usr/include/whoami.h

  Whoami.h is a "c" include file which contains the name of the local system. As a side issue, although the manual states that a system name may be eight or fewer characters, I encountered problems in using the uux command (remote command execution) with a system name eight characters long. Pick a system name which is seven or fewer characters in length.
~~~~~~~~~~~~~~~~~~~~

- /bin/mail

  The "mail" command is replaced by a version which has remote mailing capabilities. The old version of mail is not destroyed, it is saved in the directory /usr/lib/uucp/DISTBIN. If unicom is de-installed, the old version of mail is re-installed.

- /usr/lib/crontab

  Crontab is a file which is used by cron for execution of commands at periodic intervals. Crontab is modified both by the installation procedure and the uucp-config program. For more information on cron refer to 3-11 in the Unicom manual for a sample crontab entry. For more information on cron in general refer to the 8560 System Reference Manual page 8-4.

- /etc/ttys, /dev/ttyX

  The system file /etc/ttys is modified by the uucp-config program to configure ports which will be used by uucp. If the local system is a master, the ports through which uucp communicates will be non-login. In addition, the ownership of the tty devices in the /dev directory which are used by uucp, are changed to uucp.

The manual part number is 070-4536-00.

Gordon Glathar MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

NOTES ON USING UUCP USING HSI PROTOCOL

If you are planning to use HSI protocol between 8560's in your uucp network, you may want to be aware of a few precautions. Provided these precautions are observed, the HSI-RS422 link is ideal for uucp communications.

Please note that these restrictions apply to all HSI protocol connections whether they are RS-232 or RS-422. However, HSI-RS232 is not a mode which can be set up using uucp-config and is therefore discouraged.

The first thing you must do before attempting uucp over HSI-RS422, is to be sure the latest IOP firmware has been installed in the 8560's. The latest numbers are 160-1406-02 and 160-1407-02. They may be ordered by ordering the kit 040-1107-00.

If a port is set up to use uucp over HSI-RS422, you will need to be user root or uucpa in order to use cu on the port. This is because the permission on the /dev/ttyX device is owner read write only and the owner is uucp. If you change these permissions they will be changed back again the next time uucico is invoked on that port.

If you have a HSI-RS422 uucp port set up as a "master", you must be sure the "slave" connected to the port is always powered up and ready. Otherwise you may crash your "master" system. This is documented on page 3-3 of the UNICOM manual. If it is necessary to remove the "slave" from the network (for maintenance or updating), it is necessary to prevent the slave from being called. This can be done by removing the entries in L.sys and L-devices in the directory /usr/lib/uucp, on the master system, that pertain to the HSI link. If the slave is only going to be off line for a short time, creating a file in the directory /usr/spool/uucp called "LCK..ttyX" where X is the port to which the slave is connected.

Gordon Glathar MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

ARTICLE SUBMITTAL FORM

The following form may be used to submit articles which you feel might be of interest to other readers.

TEKTRONIX MDP USER GROUP NEWSLETTER 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.

Authors name ○ yes ○ no

Company Name ○ yes ○ no

Area code Tel. No. ○ yes ○ 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. To my knowledge 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.

Signature Date

THIRD PARTY SOFTWARE

MDP SOFTWARE REFERRAL SERVICE

INTRODUCTION AND BENEFITS

Tektronix offers you a Referral Service for software products from independent vendors. Published here is information about these vendors and products. Customers locate products using this information and obtain the product from the software vendor.

Benefits to Tektronix customers of this service are:

- Our commitment to increase the value of your MDP systems
- Access to many software vendors for MDP related products
- Promotes independent development of MDP related products
- Quickly find current information on third party software for MDP systems
- Many useful products to enhance your MDP system

Tektronix' Microcomputer Development Products meet your micro development needs.

TEKTRONIX REFERRAL POSITION

Tektronix acts only as a referral source for these products and vendors. Tektronix does not necessarily evaluate, analyze, or approve of these products. Referral does not imply fitness for a particular purpose and it is not a recommendation to buy. Evaluation, purchase, and support of these products are between the customer and the vendor. Although Tektronix believes this information is accurate, we do not guarantee its accuracy. Tektronix suggests its customers obtain current product information and assess compatibility with MDP products before purchase.

PRODUCT INFORMATION OFFERED

Product information is presented in tables following this introduction:

Trademarks:	Vendors' trademarks referenced here
Language Software Cross-Reference:	Select language & micro, find vendor
Software Products for 8550 & 8002A:	Products running on the 8550 or 8002A
Language Software for the 8560:	Products running on 8560s and others by that vendor
Language Software for DEC Minis:	Products for Host-based software development
Real-Time Operating System Kernels:	With some compatibility with MDP systems
UNIX Software Vendors:	For applications other than lanuages
RT-11 Software Vendors:	Products that may run on 8550 with RT11/50
UNIX Services:	Other UNIX software information sources
Vendor Contact Information:	US Vendors of MDP-related language products
International Distributors	For the US vendors in previous table

The principal source of information for these referrals is vendors' product literature. Tektronix updates this information regularly. Any prices shown are approximate; contact vendor for current prices.

PRODUCT LISTING CRITERIA

Inclusion of a software product or vendor in these tables means it has met these criteria:

1. The product is useable on or with Tektronix development systems. Or the vendor markets for other computer products that may, without change, operate on Tektronix development systems.
2. The product is useful for microcomputer application development. Or the product serves an application that MDP customers may want to use their system for.
3. The product is available unbundled and product information is available from the vendor.

Neither price nor vendor is a criteria.

PRODUCT COMPATIBILITY

Compatibility of these products with MDP systems varies. Factors to consider are distribution media, executablity, download formats/routines, interface to MDP debugging tools, and other software interfaces. Some tables indicate the compatibility Tektronix believes exists. Customers can assess compatibility by a demonstration, evaluation version, return policy, contacting other users, etc.

USING THE LISTS

Locating Products. To find vendors of a particular language for a particular micro, consult the Cross Reference table. Then see the product tables for more information about the product. See Vendor Contact Information or International Distributors to learn how to contact the vendor.

Obtaining Products. Locate alternative products using these lists. Obtain further information from the vendor, other users, and your Tektronix sales engineer. Choose desired product and arrange purchase and support with software vendor.

GETTING MORE INFORMATION

Contact the vendor or distributor for more information about their product. Some customers who have used third party software with MDP systems offer an appraisal of that software. Your Tektronix sales engineer may have such customer references. Sources in the UNIX Services table offer additional information about UNIX software.

USER AND VENDOR FEEDBACK

Tektronix solicits from software vendors information about new products and corrections or additions to the information presented here. Tektronix solicits from its customers information about a purchased product's compatibility, quality, value, etc. Tektronix also seeks customers who are willing to be a reference for such information to other customers. If customers develop MDL related software products, they may wish to have it listed herein. Send any of this information to

MDP Third Party Software
Tektronix, Inc.
Walker Road Industrial Park
P.O. Box 4600, M.S. 92-635
Beaverton OR 97075

~~~~~~~~~~~~~~~~~~~~

TRADEMARKS

These trademarks of the indicated companies are used in this catalog.

COMPANY	TRADEMARKS
3Com	UNET, 3Com
Alcyon	REGULUS
Bell Laboratories	UNIX
bytek	COGEN, bytek
Caine, Farber, & Gordon	PDL
Computer Method	XED
Computer SW Des	Data Ace
Computer Sys Co	CALC-11
D.A.T.A., Inc	D.A.T.A. Book
Digital Equipment Corp	PDP-11, VAX, VMS, RT11, 11/23
Digital Research	CP/M
Human Computing Resources	HCR, HCR/EDIT, HCR/PASCAL, RT/EMT, HCR/BASIC
Hunter & Ready	VRTX
Industrial Programming	MTOS
Information Nexus	NEX
InfoPro Systems	UNIQUE
Intel	PL/M
Interactive Systems Corp	INed, INword, INcompose, INmail, INnet, IS/1
Logical Software	LOGIX, Softshell
Mark of Unicorn	The FinalWord
Measurement Concepts	CAST
Micro Focus	CIS COBOL, FORMS-2
Microsystems	proFORTH, RTOS, RTOS-80
Redwood Bureau Services	UNIPLEX
Relational Database Systems	informix, c-isam, performix, ace
Rhodinus	Mistress
Ryan-McFarland	RM/COBOL
Software Components	pSOS
Syscon	PLMX
Systems and Software	REX, MPX
Tektronix	TNIX, TEK, TEKTRONIX
US Software	MICRO, MTK
Unicorp Software	Viewcomp, Unicorp
VenturCom, Inc	Matrix, TEQ, Proforms, SigPak
Virtual Microsystems	The Bridge
Whitesmiths	Idris
~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~

LANGUAGE SOFTWARE CROSS REFERENCE

Vendors of compilers and assemblers on DEC Minis and Tek MDLs.

MICRO	ASSEMBL	PASCAL	C	PLM	FORTRAN	OTHER
----	----	----	----	----	----	----
8086/8	BSO	BSO				
	Cymric	Cymric				
	Enertec	Enertec				
	IDS	I’metrics	I’metrics			
	LangResrc	LangResrc		LangResrc		
	Lantech	1stSys	Lantech		1stSys	IIS
	MicroTec		MarkWm			
	SCO					
	Sys&SW	Sys&SW	Telcon			
	VirtualSys	VirtualSys	VenturCom		VirtualSys	
	Whitesmith	Whitesmith	Whitesmith			
----	----	----	----	----	----	----
68000	BSO		Alcyon			IIS
	Cymric	Cymric	GrHills	GrHills		
	Enertec	Enertec				
	IntDev	I’metrics	I’metrics			
	IDS	LangResrc	I’activeSys			
	MicroTec		MarkWm			
	Oasys	OreSW				
	SCO					
	VirtualSys					
	Whitesmith	Whitesmith	Whitesmith			
----	----	----	----	----	----	----
z8000	BSO					
	Cymric	Cymric				
	Enertec	Enertec				
	IDS		MarkWm		1stSys	
	MicroTec					
	SCO		SCO			
	VirtualSys					
----	----	----	----	----	----	----
8080/5	BSO					
	CFG			CFG		
	Cymric	Cymric				
	IDS	Enertec				
	MicroTec	MicroTec				MicroSys
	NUVTEC					
	SCO	PasDev	Telcon	Syscon		
	VirtualSys	US.SW				US.SW
	Whitesmith	Whitesmith	Whitesmith			
----	----	----	----	----	----	----
680x	BSO	BSO				
	Cymric	Cymric				
	IntDev					
	IDS					
	MicroTec	PasDev		Syscon		
	NUVTEC					
	VirtualSys					
	Wintek					Wintek
----	----	----	----	----	----	----
6809	BSO		Introl			IIS
	Cymric	Cymric				
	IntDev					
	IDS	Enertec				
	MicroTec	Telcon		Syscon		
	VirtualSys					
	Wintek					Wintek
----	----	----	----	----	----	----
z80	BSO					
	Cymric	Cymric				
	IDS		I’activeSys			
~~~~~~~~~~~~~~~~~~~~

MICRO	ASSEMBL	PASCAL	C	PLM	FORTRAN	OTHER
----	----	----	----	----	----	----
	MicroTec	MicroTec				MicroSys
	NUVTEC					
	SCO	PasDev	VanDat	Syscon		
	VirtualSys		VenturCom			
----	----	----	----	----	----	----
6805	BSO					
	IntDev					
	IDS					
	MicroTec					
	VirtualSys					
	Wintek					
----	----	----	----	----	----	----
8048	BSO					
	Cymric	Cymric				
	IDS					
	MicroTec					
	SCO					
	VirtualSys					
----	----	----	----	----	----	----
8051	BSO					
	Cymric					
	IDS					
	MicroTec					
	NUVTEC					
	SCO					
	VirtualSys					
----	----	----	----	----	----	----
1802	BSO	Enertec				
	IDS					
	MicroTec			Syscon		
----	----	----	----	----	----	----
9900	BSO					
	Cymric	Cymric				
	IDS					
	MicroTec			Syscon		
----	----	----	----	----	----	----
650x	BSO		Lantech			
	Cymric	Cymric				
	IDS					
	MicroTec					
	NUVTEC					
----	----	----	----	----	----	----
z8	BSO					
	MicroTec					
----	----	----	----	----	----	----
3870	BSO					
	IDS					
	MicroTec					
----	----	----	----	----	----	----
2900	BSO					
	MicroTec					

LANGUAGE SOFTWARE ON 8560

This lists MDP-related products from vendors that have SOME software known to run on the 8560. Some PDP11-UNIX products also run on the 8560; see "Language SW on DEC Minis" and "Running UNIX Software on the 8560". Cross-Reference Table specifies the micros supported.

VENDOR	LANGUAGE	MICROS	COMPATIBLE	PRICE
----	----	----	----	----
Alycon	C	68000	Tek Asm	$950
----	----	----		----
Boston Sys Off	Assembler	all 8-bit	Tekhex+symb	$2100
	Assembler	all 16-bit	Tekhex+symb	$2900
	Simulator	most 8-bit		$1900
	Simulator	16-bit		$2900

Cymric	Pascal-macros	many	Tekhex	$3000
	Assemblers	many	Tekhex	$1500
	Simulators	many		$3000
Enertec	Pascal-Interp	8080,6809,1802	Tekhex	$2700
	Pascal-Interp	8086,z8000,68000	Tekhex	$3200
	Pascal-CodeGen	8086,z8000,68000	Tekhex	$4450
	Assembler	8086,z8000,68000	Tekhex	unk
Interactive	C	z80		$2000
Real Time Syst	Pascal & C	6809	Tekasm	$1400
Santa Cruz Op	C	z8000	Tekhex	$1800
	Assembler	8085,z80,8048,8051	Tekhex	$1400
	Assembler	z8000,8086	Tekhex	$1600
Virtual Syst	Pascal	8086	Tekhex	$2750
	FORTRAN	8086	Tekhex	$3000
	Assembler	8086,z8000,68000	Tekhex	$3200
	Assembler	8085,8048,8051	Tekhex	$2900
		680x,6809,z80		
Whitesmith	Pascal (& C)	8080,68000,8086		$1050
	C	8080,68000,8086		$900
	Pascal (& C)	Native (for 8560)		$900

LANGUAGE SOFTWARE ON DEC MINIS

Compatibility with MDP systems is shown. Some products are provided in source form (* Pascal, # FORTRAN, % C). Generally, products are available on all hosts listed for that vendor. Cross-Reference Table also specifies the micros supported. Availability on operating systems is indicated by: -U (UNIX only), -D (DEC O.S. only), or -B (both DEC and UNIX O.S.).

VENDOR	HOSTS	LANGUAGES	MICROS	COMPATABLITY
Alcyon	PDP11-U	C	68000	Tekasm
	VAX-U	Assemb/Link	68000	Tekhex
Boston Sys Off	PDP11-B	Assembler	All	Tekhex
	VAX-B	Simulator	Most	w symbols
	DEC10	Pascal	6800,8086	Download
Caine, Farber	PDP11-U	PLM	8085	
& Gordon	Other-U %			
Cymric	PDP11-B	Assembler	most	Tekhex
	VAX-D	Pascal -macros	most	
		Simulator	most	
Enertec	PDP11-B	Pascal	8080,6809,1802	Tekhex
	VAX-B		8086,z8000,68000	
	Other *	Assembler	8086,z8000,68000	
First Systems	VAX-D	Pascal	8086	
		FORTRAN	8086,z8000	
Green Hills SW	VAX-B	C, PLM	68000	
Intelligent	PDP11-D	Assembler	6800,6805,6809	
Devices	VAX-D		68000	
	Other #	Simulators	same as asm	
Intelligent	PDP11-D	RTL/2	68000,6809	
Indust. Syst.	VAX-D			8086
Interactive	PDP11-U	C	z80,68000,8086	

VENDOR	HOSTS	LANGUAGES	MICROS	COMPATABLITY
Systems Co	VAX-D			
Intermetrics	PDP11-B	Pascal	8086,68000	
	VAX-B	C	8086,68000	
International	PDP11-D	Assembler	most	
Data Service		Simulators	many	
Introl	PDP11-U	C	6809	
Language	VAX-D	Pascal	68000,8086	Tekhex
Resources	IBM	PLM & Asm	8086	Download
	Harris			
Lantech Sys	PDP11-B	C	8086,6502	
	VAX-B	Assembler	8086	
		Simulator	8086	
MANX SW Systems	PDP11-U	C	8080,6502	
Mark Williams	PDP11-U	C	8086	
MicroTec	PDP11-D	Assemblers	most	Tekhex
	VAX-D	Simulators	most	Download
	Other #	Pascal	8085	
NUVATEC/INC	PDP11-U	Assemblers	6500,6800,8041	
	VAX-U		8051,8080,z80	
Oasys	PDP11-B	Assembler	68000	
	VAX-B			
Oregon SW	PDP11-D	Pascal	68000	
	VAX-D			
Santa Cruz	PDP11-U	C	z8000	-
Operation	DataGen	Assembler	8085,8048,8051	Tekhex
			8086,z80,z8000	
Systems & SW	PDP11-D	Pascal & Asm	8086	Download
	VAX-D			
Telecon Sys	PDP11-B	C	8080,6809,8086	
Unisoft, Berk.	PDP11-U	C	68000	
	VAX-U			
Van Data	PDP11-U	C	z80	
VenturCom	PDP11-U	C	z80,8086	
	VAX-U			
Virtual Syst	VAX-B	Pascal	8086	Tekhex
	PDP11-B	FORTRAN	8086	Download
		Assemblers	most	
Whitesmiths	VAX-B	Pascal & C	68000,8080,8086	
	PDP11-B	Assemblers	68000,8080,8086	
Wintek	many #	Assembler	6800,6805,6809	
		PL/W	6800,6809	
		Simulator	6800,6801	

REAL-TIME OPERATING SYSTEM KERNELS

These vendors offer real-time multi-tasking O.S. kernels. Many are compatible with MDP systems:
Interface SW: __tables/routines that connect the kernel to application SW
dely on PROM: kernel is delivered in executable form.
<vendor> Asm: kernel is in <vendor> assembly source form
Link <system>: kernel is in relocatable modules linkable by <system>

Other annotations: (*) compatibility under development and (#) multi-processor version.

VENDOR	PRODUCT	MICROS	COMPATIBLE	PRICE
Hunter & Ready	VRTX	8086,68000 z8000	dely on PROM interface SW	royalties
Industrial Programming	MTOS	68000	Tek Asm	$9500
		6800,6809	Moto Asm	$4000
		8080 (MP #)	Intel Asm	$3500
		8086 (MP #)	Intel Asm	$5500
JMI SW Consult	C Exec	68000,6809 8080/5,z80 8086/8,16032	W'smith C	royalties vol discounts $300 for one $20 ea (qty > 500)
Microsystems	RTOS-80	8080/5,Z80	proFORTH source Interface sw	$1750
SW Components	pSOS	68000	dely on PROM interface SW	unknown
Systems & SW	REX	8080	link 8002	$2000
		8086	link 8500 *	$2750
	MPX #	8080,8086	Asm Source	$1000
U S Software	MTK I	8085,z80,6502 6800,6809	Tek Asm	$200
	MTK II	8086	Tek Asm	$250

UNIX SOFTWARE VENDORS

These vendors offer PDP11-UNIX software that may run on the 8560. Products known to run on the 8560 are noted. 11/23 UNIX products may also run on the 8560; see "Running UNIX Software on the 8560" and contact the vendor.

VENDOR	CONTACT	PRODUCT	DESCRIPTION	HOST
Bytek	415-527-1157	COGEN	COBOL generator	
Computer Method	213-998-7979	XED	word processing	
Computer SW Des	714-634-9012	Data Ace	DBMS	
Computer Sys Co	800-428-0714	CALC-11	spreadsheet	
Human Comp Res	416-922-1937	MULTIPLAN	spreadsheet	11/23-UNIX
		RT/EMT	RT-11 Emulator	8560
		HCR/BASIC	ANSI-Stnd BASIC	11/23-UNIX
		HCR/EDIT	editor	11/23-UNIX
		HCR/PASCAL	Pascal compiler	
Info. Nexus	312-637-7995	NEX	screen editor	
Interactive Sys	213-450-8363	INed	screen editor	
		INword,compose	word, text proc	
Logical SW	617-864-0137	LOGIX, Q	DBMS, queries	11/23-UNIX
		Softshell	User Interface	

VENDOR	CONTACT	PRODUCT	DESCRIPTION	HOST
Mark of Unicorn	617-489-1387	The FinalWord	word processing	11/23-UNIX
Measure.Concept	315-337-1000	CAST	CAI language	
Micro Data Base	317-448-1616	MDBS III	DBMS	
Micro Focus	408-496-0176	CIS COBOL	GSA-cert COBOL	11/23-UNIX
		FORMS-2	COBOL generator	11/23-UNIX
Uniq Computer	312-879-1566	Unify	DBMS	11/23-UNIX
Relational DBS	408-746-0982	informix, Ace	DBMS	
		Performix	data entry	11/23-UNIX
		c-isam	indexed files	11/23-UNIX
Rhodinus	416-922-1743	Mistress	DBMS, reports	11/23-UNIX
Ryan-McFarland	213-541-4828	RM/COBOL	ANSI-74 COBOL	
Santa Cruz	408-425-7222	MULTIPLAN	spreadsheet	11/23-UNIX
		UNIPLEX	word processing	11/23-UNIX
		informix, Ace	DBMS, reports	11/23-UNIX
		SCCS	Source Cntl Sys	11/23-UNIX
Softest	201-447-3901	LEX	Word Processing	
Unicorp SW	212-307-6800	Viewcomp	Spreadsheet	8560
UC Berkeley	415-642-4948	basic+	DEC's BASIC	
VenturCom	617-661-1230	Proforms	time & billing	
		Matrix	spreadsheet	
		SigPak	signal process	
		TEQ	math evaluator	
Virtual M'syst	415-841-9594	The Bridge	Run CP/M SW	8560

RT-11 SOFTWARE VENDORS

These products may run on the RT11/50 operating system of the 8550. See RT11/50 data sheet and contact the vendor to assess whether they will.

VENDOR	CONTACT	LANGUAGE SOFTWARE
BSO	617-894-7800	Cross-assemblers, simulators
Bytek	415-527-1157	COBOL code generator
Cymric	617-369-9106	Pascal, Asemblers, Simulators
DISC	916-363-7385	DBL, Business application lang
Intelligent I.	201-865-6550	RTL/2 resident compiler
Lantech Sys	214-340-3900	C & Assemblers (cross)
Loki Engr	617-653-1120	Magic/L programming system
Micro Focus	408-496-0176	CIS COBOL, FORMS-2
MicroTec	408-733-2919	Cross-assemblers, simulators
Pacific SW	415-540-0616	Color graphics
Oregon SW	503-226-7760	Pascal (native and cross)
Ryan-McFarland	408-662-2522	RM/COBOL compiler
Telecon Syst	408-275-1659	C (native and cross)
Virtual Syst	415-935-4944	Pascal, Fortran, & Asm (cross)
Whitesmith	212-799-1200	C & Pascal (native and cross)

VENDOR	CONTACT	APPLICATIONS SOFTWARE
Access Tech.	617-655-9191	spreadsheet
Contel	301-654-9120	FORTRAN Math, DBMS, Debug
Computer Prog	213-794-2857	Programming & system utilities
Computer Sys	317-872-7200	spreadsheet
Discom	213-796-9375	Word Processing
GABA, Inc.	213-907-6622	screen edit, word processing
Geographix	215-925-6690	Graphics chart generator
ITI	503-644-0111	Applications development tools
Interplex	415-969-9050	Format gen & transaction entry
Lachman	312-986-8840	C Libraries, consulting
Lantor	213-821-0642	Graphics SW
MCBA	213-957-2900	COBOL, business applications
MicroTech Exp	415-324-9114	CPM format conversion SW
Midnight Data	617-491-6294	Word Processing, spell
Nyplan	206-822-6074	Financial Modeling
Penn St Univ	814-865-1595	Statistical SW (Minitab)
Precision Vis.	303-449-0806	General purpose graphics
Saturn Syst	612-944-2452	spreadsheet, word processing
SofTest	210-427-4971	Digital elect test
Softpak	213-822-1830	SW Distributor
Struct'l Prog	617-443-5366	Project planning and management
SPSS, Inc	312-329-2400	Statistical analysis SW
Theta Syst	213-245-0917	Business Software
UAP	619-730-1012	Comm & File Transfer SW
Zia Corp.	210-540-9341	Virtual Term, File Transfer

UNIX SERVICES

This lists a variety of UNIX services of interest to 8560 users. Included are newsletters, user groups, software catalogs, UNIX licensing, SW searches, timesharing, research reports, and training courses.

SERVICE	ORGANIZATION	PRODUCT
User Group	/usr/group P.O. Box 8570 Stanford CA 94305-0221	UNIX Catalog commUNIXations (newsletter) UNICOM (conference)
	USENIX Association Box 8, Rockefeller U. 1230 York Ave. New York NY 10021 212-570-8934	UNICOM (conference) newsletter software exchange
	European UNIX User Grp c/o Alan Mason Dept. of EE Heriot Watt University Edinburgh, Scotland	

SERVICE	ORGANIZATION	PRODUCT	
~~	~~~~	~~~~	~~~~
	Canadian UNIX SIG c/o Human Computing Res. 10 Saint Mary St. Toronto, Ontario Canada M4Y 1P9 ph: 416-922-1937		
	Australian UNIX Users Grp c/0 Peter Ivanov Computer Sci, Elect Engr Univ of New South Wales P.O. Box 1 Kensington 2033 Australia		
~~	~~~~	~~~~	~~~~
Newsletter	/usr/group	commUNIXations	
	Uni-Ops P.O. Box 5182 Walnut Creek CA 94596 415-933-8564	Pipes and Filters	
	InfoPro Systems P.O. Box 33 East Hanover NJ 07936 201-625-2925	UNIQUE	
	Southwater Corp 30 Mowry St. Mt. Carmel CT 06518 203-288-0283	UNIX/C Market News	
	Yates Ventures Suite 111 4962 El Camino Real Los Altos CA 94022 415-964-0130	Yates Perspective	
~~	~~~~	~~~~	~~~~
Research	Yates Ventures	marketing research	
~~	~~~~	~~~~	~~~~
SW Catalog	/usr/group	UNIX Catalog	
	International Computer Programs, Inc. 9000 Keystone Crossing PO Box 40946 Indianapolis IN 46240 800-428-6179 317-844-7461 Telex 27-6116	ICP Software Reference Series - DEC Small Computers	
	Intelligent Decisions 6424 Myrtlewood Dr Cupertino CA 95014 408-996-2399	Software Tools Catalog	

SERVICE	ORGANIZATION	PRODUCT	
~~	~~~~	~~~~	~~~~
	Elsevier Intl SW DBase Box TSC-1 52 Vanderbilt Ave. New York NY 10017	The Software Catalog - MicroComputers	
	D.A.T.A., Inc. A Cordura Company PO Box 103 8660 Mirimar Road San Diego, CA 92126	D.A.T.A.Book Microprocessor Software	
	Digital Equipment Corp Attn: SRC Manager Engineering Systems Grp MR 1-1/M75 200 Forest Street Marlboro MA 01752	Engineering Systems Software Referral Catalog	
~~	~~~~	~~~~	~~~~
UNIX License	U of Calif at Berkeley Dept of Computing Svcs 215 Evans Hall Berkeley CA 94720 ph: 415-642-4948	license UCB UNIX (source) & sw tools newsletter courses timesharing	
	Western Electric Co. Patent Licensing Mgr. A.T. & T. Co. Guilford Center PO Box 25000 Greensboro, NC 27420 919-697-2078	license UNIX source	
	International Data Services, Inc.	4.1bsd with Sys III features (Binary license)	
	Human Computing Resources, Inc.	HCR/UNITY: Sys III with 4.1bsd features (Binary license)	
~~	~~~~	~~~~	~~~~
SW Search	USENIX Association (see User Groups)		
	Software Tools User Grp (see User Groups)		
~~	~~~~	~~~~	~~~~
Timesharing	International Data Services, Inc. Sunnyvale CA 408-738-3368	11/70 UNIX v 7 (UCB mod)	
	Marketing Info. Inst. San Diego, CA 619-231-8939	11/45 UNIX v 7	

SERVICE	ORGANIZATION	PRODUCT	
~~	~~~~	~~~~	~~~~
	RLG Corporation 1760 Reston Ave Reston, VA 703-471-6860	11/34s UNIX v 7	
	FENIX Computer Timesharing FARGO Electronic Services 7150 Shady Oak Road Eden Prairie, MN 55344 (612) 941-9470		
~~	~~~~	~~~~	~~~~
Training	U of Calif at Berkeley (see UNIX Lic. table)	UNIX Courses	
	Plum Hall 303 Forest Drive Edison NJ 08817 201-572-1017	Courses nationwide: UNIX Pascal C, Advanced C	
	Santa Cruz Operation (see Vendor Info table)	UNIX Tutorials self-study tapes	
	Human Computing Res. (see Canadian User Grp)	UNIX Seminars nationwide USA	
	Computer Technology Grp Telemedia, Inc. 310 S. Michigan Ave Chicago IL 60604 800-621-3155	UNIX Training 8 courses, hands-on nationwide USA	
	User Training Corp P.O. Box 970 Soquel CA 95073 408-462-6527	UNIX System Training audiodigital courses novice, advanced, C	
	International Tech. Sem 2000 Center St. Suite 1036 Berkeley CA 94704	UNIX Courses James Joyce, UCB Prof at ITS or on-site	
	RLG Corporation (see Timesharing)	UNIX Courses at Reston facility C, Shell, System Admin	
~~	~~~~	~~~~	~~~~
Typesetting	UNICOMP Phototypesett'g 1580 Camino Redondo Los Alamos NM 87544 505-662-edit	from troff output	
	Image Network 1633 Bayshore Highway Suite 239 Burlingame CA 94010	from troff output	

~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~

VENDOR CONTACT INFORMATION

This table gives vendors' address/phone/telex, products offered, international distributors.

COMPANY	PRODUCTS	DISTRIBUTORS (COUNTRIES SERVED)
Alcyon 8474 Commerce Ave San Diego CA 92121 ph: 619-578-0860	C Regulus (Op Sys)	Europel Systems (England)
Boston Systems Off 469 Moody St. Waltham MA 02154 ph: 617-894-7800 tx: 710 324 0760	Assemblers Simulators Pascal	ASR Intl (Japan) China Computer (Rep of China) Contahal Ltd (Israel) Imdata A/S (Denmark) Interautomation AG (Switz) Interautomation GmbH (W Germ) Mimarobe OY (Finland) Mini Computer Sys (Australia) Nordqvist and Berg (Swe/Nor) Rohde & Schwarz (W Germ/Aust) Software Sciences (Engl/Ire) Software Sciences (Belgium) Southern Dynamics (India) Spetelec (France) Systems Tech Intl (China) Systime S.A. (S. Africa) Yezerski Roper Asso (Australia) Zeltron Automazione (Italy) Olivetti Africa (S. Africa)
Caine, Farber, Gordon 750 East Green Street Pasadena CA 91101 ph: 213-449-3070	PL/M PDL (SW design)	
Cymric Computer Syst PO Box 253 Concord MA 01742 ph: 617-369-9106	Assemblers Pascal Simulators	Camelot SW & Systems (England) Ing. Buero Ridderbusch (W Germ)
Enertec, Inc. 19 Jenkins Ave Lansdale PA 19446 ph: 215-362-0966	Pascal Assemblers	
First Systems Corp. 1112 Ocean Dr Suite 201 Manhattan Beach CA 90266 ph: ?	Pascal FORTRAN	
Green Hills Software 55 North St.John Pasadena, CA 91103 ph: 213-796-6543	C PLM	

COMPANY	PRODUCTS	DISTRIBUTORS (COUNTRIES SERVED)
Hunter & Ready 445 Sherman Ave Palo Alto CA 94306 ph: 415-326-2950 tx: 69-6191	OS Kernel	
Industrial Programming 100 Jericho Quad Jericho NY 11753 ph: 516-936-6600	OS Kernel	Betea (Belgium) Celdis (France) Alfred Neyeenatechnik (W Germ) Contahl Ltd (Israel) C.N. Rood (Netherlands) Frontec Microsatorcenrum (Scand) Hawker Siddeley (England) Saras Electronics (India) Tokyo Electron Ltd (Japan) Xmit AG (Switz)
Intelligent Devices P.O. Box 163 Dillon CO 80435 ph: 303-468-0112	Assembler Simulators	
Intelligent Ind. Syst. One Harmon Plaza Secaucus, N.J. 07094 ph: 201-866-3332	RTL/2 microMAGIC	
Interactive Systems 1212 Seventh St Santa Monica CA 90401 ph: 213-450-8363 tx: 910 343 6255	C, FORTRAN IS/1 (Op Sys)	
Intermetrics, Inc Software Products Div. 733 Concord Ave Cambridge MA 02138 ph: 617-661-1840 tx: 710 320 7523	Pascal	Micro General (Italy)
International Data 453-D Ravendale Dr Mountain View CA 94043 ph: 415-969-7222	Assemblers Simulators	
Introl Corp. 67 W. Virginia St Milwaukee WI 53204 ph: 414-276-2937	C	
JMI SW Consultants 1422 Easton Road Roslyn, PA 19001 ph: 215-657-5660	C Executive	Real Time Systems Advanced Data Controls

COMPANY	PRODUCTS	DISTRIBUTORS (COUNTRIES SERVED)
Language Resources 4885 Riverbend Road Boulder CO 80301 ph: 303-449-8087	Pascal PL/M	
Lantech Systems Inc 9861 Chartwell Drive Dallas, TX 75243 ph: 214-340-3900	C Assemblers Simulators	
Mark Williams Co 1430 West Wrightwood Chicago IL 60614 ph: 312-472-6659 tx: 910 221 1182	C	Toyko Electron Ltd (Japan)
Microsystems Inc 2500 East Foothill Blvd Suite 102 Pasadena CA 91107 ph: 213-577-1471	proFORTH RTOS-80	
Microtec PO Box 60337 Sunnyvale CA 94088 ph: 408-733-2919 tx: 4990808	Assemblers Simulators Pascal	Albetros Ltd (England) ASAHI Bus. Consult. (Japan) ASR Corp Intl (Japan) Contahl Ltd (Israel) Creative Daten Systeme (Austria, Beneiux, Scand, Switz, W Germ) Micro General (Italy)
NUVATEC/INC 261 Eisenhower Lane S Lombard IL 60148 ph: 312-620-4830	Assemblers	
Oasys (Office Auto Sys) 60 Aberdeen Ave. Cambridge MA 02138 ph: 617-491-4180	Assembler	
Oregon Software 2340 SW Canyon Rd Portland OR 97201 ph: 503-226-7760 tx: 910 464 4779	Pascal	
Pascal Development Co 1381 S De Anza Blvd Suite 205 Cupertino CA 95014 ph: 408-253-4280	Pascal	

COMPANY	PRODUCTS	DISTRIBUTORS (COUNTRIES SERVED)
Santa Cruz Operation 500 Chestnut Street Santa Cruz CA 95060 ph: 408-425-7222	C, Assemblers UNIX Applicat'n	
Software Components 97 La Quinta San Jose CA 95127 ph: 408-923-2741	OS Kernel	
Syscon Corp 4015 Hancock St San Diego CA 92110 ph: 619-222-6381 tx: 910 335 1660	PLMX	Micro Scope (England)
Systems and Software 1315 Butterfield, #230 Downers Grove, IL 60515 ph: 312-960-1181	OS Kernel Pascal Debugger	Electrodesign (Canada) Itech Information (England)
Telecon Systems 90 E Gish Rd Suite 25 San JOse CA 95112 ph: 408-275-1659	C	
US Software 5470 NW Innisbrook Pl Portland OR 97229 ph: 503-645-5043 tx: US 425133 COGI PTL	Pascal MICRO Libraries	
Unisoft of Berkeley 2405 4th St Berkeley CA 94710 ph: 415-644-1230	C	
Van Data Suite 107 17544 Midvale Ave N Seattle, WA 98133 ph: 206-542-7611 800:426-5248	C	
VenturCom, Inc 139 Main St Cambridge MA 02142 ph: 617-661-1230	C UNIX Applicat'n	
Virtual Systems 1500 Newell, Suite 406 Walnut Creek CA 94596 ph: 415-935-4944	Assemblers Pascal FORTRAN	Metrologie (France) Simac (Netherlands)

COMPANY	PRODUCTS	DISTRIBUTORS (COUNTRIES SERVED)
Whitesmiths, Ltd. Millbrook Tarry 97 Lowell Rd Concord MA 01742 ph: 617-369-8499	C, Pascal Idris (Op Sys)	Advance Industries (Japan) Fawnray Pty Ltd (Australia) Real Time Systems (England)
Wintek Corp 1801 South St Lafayette IN 47904 ph: 317-742-8428	Assemblers PL/W Simulator	
W S Ataras 40 Laughton St Upper Marlboro MD 20772 ph: 301-249-1141	CAD/CAM	

INTERNATIONAL DISTRIBUTORS

This table gives contact information for software distributors referenced in the Vendor Information list. It lists the software vendors whose products they distribute. It lists distributors by the country in which they are located. Other countries served by the distributor are listed in the Vendor Contact Information table.

COUNTRY	COMPANY	VENDORS
Australia	Fawnray Pty Ltd. P.O.Box 224 Hurstville NSW, 2220 ph: (02) 570-6100	Whitesmith
	Mini Computer Systems 368 Hawthorne Road S. Caulfield 3162 ph: 528-2711 tx: 34175	Boston Systems Office
	Yezerski Roper & Assoc 375 Pacific Hwy, Suite 3 Artarmon NSW 2064 ph: 439-7272 tx: 25468	Boston Systems Office
Belgium	Betea S.A. Chausse de Louvain 775 B-1140 Bruxelles ph: (02) 736 80 50 tx:846-23188	Industrial Programming

COUNTRY	COMPANY	VENDORS
~~	~~~~	~~~~
	Software Sciences Nederland BV Rue De Genevestraat 10 1140 Bruxelles (Evere) ph: (02) 216 6500 tx: 24015 Correspondence: P.O. Box 71881 1008 EB Amsterdam Netherlands	Boston Systems Office
~~	~~~~	~~~~
Canada	Electrodesign 1925 52nd Ave, Suite 1 Lachine Quebec H8T 3C3 ph: (514) 636-4838 tx: 05-821784	Systems & Software
~~	~~~~	~~~~
Rep. of China	China Computer Corp. Room A, 5th Floor 126 Nanking E. Road Section 4, Taipei Taiwan ph: (02) 731-0155 tx: 26834	Boston Systems Office
~~	~~~~	~~~~
People's Rep of China	Systems Technology Intl 156 Milk St. Boston MA 02109 USA ph: (617) 482-9430	Boston Systems Office
~~	~~~~	~~~~
Denmark	Imdata A/S Smedeland 8 DK-2600 Glostrup ph: 02 63 22 33 tx: 33285	Boston Systems Office
England	Albetros (Engrs) Ltd. Frances Road Basingstoke Hampshire RG21 3DA ph: (0256) 57551 tx: 858893	Microtec
	Camelot SW & Systems 79 London Road Knebworth, Herts. SG3 6HG England ph: (0438) 812215	Cymric
	Europel Systems Ltd. 15 Westmead Dr. Newbury, Berkshire ph: 635 31074	Alcyon

COUNTRY	COMPANY	VENDORS
~ ~ ~ ~	~ ~ ~ ~	~ ~ ~ ~
	Itech Information Technology Services 115 Grove Road Hitchin Hertfordshire SG4 0AA ph: (04380) 66561	Systems & Software Industrial Programming
	Micro Scope Ltd. Mill Lane, Taplow Maiden-Head Brookshire, SL6 0AA	Syscon
	Microsystem Services P O Box 37 Lincoln Road Cressex Indus Estate High Wycombe, Bucks HP12 3XJ England	Virtual Systems
	Real Time Systems Elliott Terrace Wkshops Newcastle upon Tyne NE4 6UP ph: 0632 732531,732639 tx: 53429 PACE G	Whitesmith JMI SW Consultants
	Software Sciences Ltd. 40 Invincible Road Farnborough Hampshire GU14 7QU ph: (02) 52 544321 tx: 858228	Boston Systems Office
	Unit-C Dominion Way West Broadwater, Worthing West Sussex BN14 8N7 ph: (903) 212114 tx: 877296	Cymric
~ ~ ~ ~	~ ~ ~ ~	~ ~ ~ ~
Finland	Mimarobe OY P.O. Box 361 SF-33101 Tampere 10 ph: (931) 36 333 tx: 22400	Boston Systems Office
~ ~ ~ ~	~ ~ ~ ~	~ ~ ~ ~
France	Celdis S.A. 53 Rue Charles Frerot 94250 Gentilly ph: 546.13.13 tx: 842-200485	Industrial Programming
	Metrologie ph: 1-791-4444 tx: 611448	Virtual Systems

COUNTRY	COMPANY	VENDORS
	Mondial Com, S.A. 3, Rue Cavalotti 75018 Paris ph: 294.12.21	Microtec
	Spetelec Tour Europa 111 94532 Rungis Cedex ph: (1) 686.56.65 tx: 250801	Boston Systems Office
India	Macro International Suite 121-123 Howland Pl 8256 East Market St. Warren OH 44484 USA ph: (216) 856-1866 tx: 241319	Microtec
	Saras Electronics N-47, Greater Kailash-I New Delhi-110048 ph: 698074 tx: 011-3532	Industrial Programming
	Southern Dynaics 93, Ramaswamy St Mannady, Madras 600 001 ph: 456799 tx: 417443	Boston Systems Office
Israel	Contahl, Ltd 54, IBN Gvirol St. Tel Aviv 64364 ph: (03) 269 379 tx: 922-33654	Boston Systems Office Industrial Programming Microtec
Italy	Micro General Corso Galileo Ferraris 75 10128 Torino ph: (011) 594 612 tx: 220644	Intermetrics Microtec
	Zeltron Automazione SPA Viale Bianca Maria 45 20122 Milano ph: 02-795802 tx: 312099	Boston Systems Office
Japan	Advance Data Controls Kyoritsu Kojimachi Bldg No. 5-5, Kojimachi Chiyoda-Ku Tokyo ph: (03) 263-0383	JMI SW Consultants

COUNTRY	COMPANY	VENDORS
~~	~~~~	~~~~
	Advance Industries Chiyoda-Ku Tokyo ph: (03) 258-0839	Whitesmith
	ASAHI Business Consult. 13-10 1-Chome, Tsukiji Chuo-ku Tokyo ph: (03) 543-3161 tx: 252-4215	Microtec
	ASR Corp International 3-23-8, Nishi-Shimbashi Minato-ku Tokyo 105 ph: (03) 437-5471 tx: 242-2723	Boston Systems Office Microtec
	Japan System Sci Co 305 Kojimachi 3 Chome, Chiyoda-Ku Tokyo	Virtual Systems
	Rikei Corporation Shinjuru Nomura Bldg 1-26-2 Nishi-Shinjuku Shinjuku-Ku Tokyo 160	Virtual Systems Oregon Software
	Tokyo Electron Ltd. Shinjuku Nomura Bldg 1-26-2 Nishi-Shinjuku Shinjuku-Ku Tokyo 160 ph: (03) 344-5893 tx: 781-232-2240	Industrial Programming Mark Williams Co.
~~	~~~~	~~~~
Netherlands	C.N. Rood B.V. Cort vd Lindenstraat 11-13 Postbus 42 2280 AA Rijswijk ph: 070-996360 tx: 844-31238	Industrial Programming
	Simac ph: 040-533725 tx: 51037	Virtual Systems
~~	~~~~	~~~~
Scandanavia	Frontec Microdatorcentrum Box 204 Malmvagen 28 Sollentuna, Sweden ph: 08-359360 tx:854-15130	Industrial Programming
~~	~~~~	~~~~

COUNTRY	COMPANY	VENDORS
South Africa	Olivetti Africa Ltd. 15 Steimens St. Johannesburg	Boston Systems Office
	Systime S.A. Pty Ltd 16th Fl, Nedbank E City 12 End St. Doornfontein 2094	Boston Systems Office
Sweden	AB Nordqvist & Berg Box 9145 S-10272 Stockholm ph: 08-69 04 00 tx: 10407	Boston Systems Office
Switzerland	Interautomation AG Neumarkt CH-5200 Brugg ph: (056) 41 94 00 tx: 52352	Boston Systems Office
	Xmit AG Bellikonerstrasse 218 CH-8967 Widen ph: 057-54656 tx: 845-59955	Industrial Programming
West Germany	Alfred Neyeenatechnik Schillerstrasse 14 2085 Quickborn -Hamburg tx: 841-213590	Industrial Programming
	Computer Beratung & SW Herdweg 1 D-7903 Laichingen ph: (07333) 3515	Mark Williams Co
	Ing. Buero Ridderbusch Steinstrasse 50 71 Heilbronn ph: 7131 80849	Cymric
	Interautomation GmbH Marburger Strasse 10 D-1000 Berlin 30 ph: (030) 211 50 57	Boston Systems Office
	Rohde & Schwarz GmbH Graf-Zeppelin Str. 18 5000 Koeln 90 ph: (02203) 49 (1)-347 tx: 8873288	Boston Systems Office Engineering and Sales

SOFTWARE PRODUCTS FOR 8550 & 8002

Vendors and their products specifically for the 8002A or 8550.

VENDOR	PRODUCT	MICROS	COMPATIBLE	PRICE
Microsystems	FORTH Package	8080/5,z80	8002,8550,load	$2250
Pascal Dev Co	Pascal	8085,z80,6800	8002,download	unk
Syscon	PLMX	8085,z80,9900 6800,6809,1802	8002,8550,Asm	$500
	Floating Pt	same as above	8002,8550,Asm	unk
	Library Source			$150
	Utilities		8002,8550	$95
US Software	MICRO	8085	8002,8550,Asm	$750
	Pascal	8085	8002,8550,Asm	$750
	Text Process'g		8002,8550	$235
	Floating Pt	8085,z80,6809	8002,8550,Asm	$500
WS Ataras Engr	Wirelist Proc'g	NA	8550	$950

CUSTOMER REFERENCES SOUGHT

Are you using your MDP system with a third party software product such as a compiler, spreadsheet, or even a database management system?

Tektronix would like to know your appraisal of that product. This information helps us determine which products deserve a stronger commitment from Tektronix. It may also be used by others to determine which product to buy. MDP would like to publish your testimonial letter or personal contact in our sales newsletter. If you have an appraisal, please contact

MDP Third Party Software
Tektronix, Inc.
P.O. Box 4600, M.S. 92-635
Beaverton OR 97075

SOFTWARE PRODUCT REFERENCES

Would you like a reference for a software product you are considering? MDP provides such references each month in its sales newsletter. Contact your local Tektronix salesperson for these references. Personal contacts or testimonial letters for these products have been published:

Product	Target uC	Host	Vendor	Issue
C	z80	8560	Interactive Systems	May 83
proFORTH	8085/z80	8002	Microsystems Inc.	May 83
proFORTH	8085/z80	8550	Microsystems Inc.	May 83

RUNNING UNIX SOFTWARE ON THE 8560

PDP11-UNIX software with the following characteristics will, without any change, run on the 8560:

1. Able to run on UNIX version 7
2. Able to run in 64K bytes without overlays
3. Able to run in common instruction & data (I & D) space

_Note: PDP-11 has either separate I & D (eg 11/70) or common I & D (eg 11/23); Much UNIX Software can be generated to run in either.

8560 MEDIA. To install software on the 8560, note these media specs:

Diskettes

- IBM-Compatible diskettes, soft sectored
- Single or double sided, single or double density
- track 0, side 0 is 128 bytes/sector, FM-encoded, always single density
- track 0, side 1 is 256 bytes/sector, MFM-encoded
- tracks 1-76 both sides are 256 bytes each sector
- In Practice:
  - Any IBM-compatible diskette with 26 sectors/track, single or double sided, either FM- or MFM-encoded, can be read.
  - 8560 is sensitive to drive alignment/timing. Use 8560 alignment disc (119-1354-00, Customer Service); align on track 38.
  - Formating the disks first on an 8560 increase probability of success

Sectors

- TNIX treats 2S-2D diskettes as 1995 512-byte blocks;
  1S-1D diskettes as 500 512-byte blocks
- There is no interleaving and no skewing
- In Practice:
  - Some floppy controllers can be programmed for no interleave/skew.
  - Otherwise read interleaved, skewed sectors and reorder the data on the 8560.

File Formats

tar: ____same as UNIX v7, found in Auxiliary Utilities Package
fbr: ____TNIX only, format documented in 8560 System Ref Manual;
dsc50: __TNIX only, for 8550 transfers, format is internally documented
others: _unreleased utilities to read diskettes from RT-11, ISIS, CP/M and Motorola systems; contact your local Tektronix salesperson.

Rodney Bell, MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~

PRODUCT PERFORMANCE SECTION

PASCAL FLOATING POINT SECTIONS, WRONG CLASS

Two sections in the PASCAL library pertaining to floating point computations, have been assigned to the class name "DATAQQ" which are RAM variable sections. They should have been assigned to the class name "CONSTQQ" which are ROM constants sections. This is a problem in all of the 16 bit PASCAL compilers. The section names are "FPP01QQ" and "FPP02QQ". If these sections are being used by your program, you will need to reassign them to the class "CONSTQQ". This can be done at link time by entering the following line in your link command line or command file.

```
-C CONSTQQ=FPP01QQ CONSTQQ=FPP02QQ
```

These sections contain tables of constants used by the floating point operations of **all 16bit PASCAL compilers.**

Gordon Glathar MDP Customer Support

~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~

PASCAL DEBUG USER NOTE

Pascal Debug issues an undocumented error message when program execution is stopped during a library routine or a user supplied assembly language routine.

"pdb:Cannot determine current location"

With the Z8000 and 68000 PDBs, this does not mean PDB is lost, it means there is no Pascal Statement Number or Symbol related to that location. To determine program state, you can use the "TB" command. The first level or perhaps more may give the message "Unrecognizable Activation", however, the total list will indicate the Pascal statement that issued the call to the assembly routine. While execution is stopped, you can access your data with fully qualified variable names, (i.e., MOD0.VAR1, instead of just VAR1). Also a "GO" command will continue program execution.

The 8086 PDB does not have this recovery capability at this time.

Marilyn Hanson MDP Product Marketing

~~~~~~~~~~~~~~~~~~~~

68000 ASSEMBLER SYNTAX ERRORS NOT FLAGGED

68000 Assembler and failure to trap syntax errors

The current evaluation of the 68000 assembler has uncovered problem. The assembler does not flag the following address register indirect addressing modes syntax as incorrect:

+(An)
(An)-

where An is an address register

NOTE: (An)+ and -(An) are valid addressing forms.

To fix this problem, a new error message has been added to the assembler,

Error 241 Invalid address register indirect addressing mode syntax

Explanation: The syntax +(An) or (An)- has been specified. These are not valid addressing mode forms.

Since the 68000 assembler manual will be updated in June, this error message will be included in this update. Until the manual update is released, it will be possible to get this error message and not have an explanation in the manual.

~~~~~~~~~~~~~~~~~~~~

PROBLEM WITH MULTIPLE ASSEMBLIES

All B-series assemblers create, use and ultimately remove a file called xref.tmp. If multiple assemblers are running in the same 8560 directory at the same time and need to use the xref.tmp file during overlapping times, the assembly will fail with an xref.tmp missing message.

The error is being corrected in version 2 of the assembler.

Again the error only occurs when:

- more then one assembler process is active
- AND they are in the same directory
- AND they both are using the xref.tmp file at the same time.

John Owens MDP Customer Support

~~~~~~~~~~~~~~~~~~~~

MDP BUG BASE

The following product performance reports are contained in our data base. If you have encountered additional problems not listed here, 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 to the problem. We will also try to provide a "work-around" immediately.

John Owens MDP Customer Support

PRODUCT 8301 and 8540 power supplies

CONFIG. 8301 and 8540

PROBLEM The 8301 and 8540 power supplies may occasionally fail due to voltage spikes on the AC power lines. Symptoms seen by customers are 1) no lights on the 8301 or 8540, 2) no response on the keyboard or 3) meaningless data displayed on the terminal.

SOLUTION If this is a problem at your installation, use a constant voltage transformer that provides "spike suppression."

ACE EDITOR AND EMPTY FILES

PRODUCT ACE Editor and doing a replace in an empty file

CONFIGURATION 8560 TNIX version 1.3 and ACE Editor version 2.08

PROBLEM Doing a replace on an end-of-line marker, when the text file is empty, will cause the end-of-line marker to disappear until a zero page is done.

COMMENT The next release of ACE (in June) will correct this problem.

NROFF -T FILES

PRODUCT 8560 Class C software with nroff and term ID

CONFIGURATION 8560 TNIX version 1.3 and text processing software
~~~~~~~~~~~~~~~~~~~~

PROBLEM The 8560 Class C software for text processing expects "nroff -T" to reference the files of the form /usr/lib/term/taba(term id). However, the files provided are of the form /usr/lib/term/tab(term id).

COMMENTS Field fix is to change the names.

68000 ASM AND THE MOVQ INSTUCTION

PRODUCT 68000 assembler and MOVQ instruction

CONFIGURATION 8560 TNIX version 1.2 and 68000 B-series assembler version 01.15-66

PROBLEM For the MOVQ instruction, the assembler issues error message #250 when the value is greater than 7F (hexadecimal). For example:

```
MOVQ 86H, D1
```

Issues an error message, but code generated will be correct unless the value supplied is greater then 7F.

B SERIES ASM AND EQU DIRECTIVE

PRODUCT B series assembler and redefined EQU statements

CONFIGURATION 8560 TNIX version 1.2 and 68000 B-series assembler version 01.15-66

PROBLEM The user needs to be aware that the B series assembler specifications do not allow the EQU to redefine a label, even with the same value and that an error message is issued when this restriction is violated.

Z80 EMULATOR ESC BUG

PRODUCT Z80B and Z80A Emulator

CONFIGURATION 8550 DOS-50 version 2.1A or 8540 OS-40 version 1 with the Z80AorB emulator

PROBLEM If an ESC key (control c) is detected during a double fetch instruction, upon continuing (GO) the program may start on the second fetch of the double fetch instruction instead of the next instruction.

COMMENTS The user can determine that the program stopped on a double fetch, then restart at the address of next instruction.

LINKER INTERNAL ERROR IN ROUTINE 35

PRODUCT LAS Linker and error message "internal error in routine 35"

CONFIGURATION 8560 TNIX version 1.3 and LAS Linker version 2.08-00

PROBLEM When any symbol is used as both a memory name (MEM=name) and as an entry point name (label), the error message "internal error in routine 35" is generated.

COMMENTS User must rename the memory name.

NATIVE C COMPILER INCLUDE FILES

PRODUCT 8560 system and include files in the Native Programming Pkg.

CONFIGURATION 8560 TNIX version 1.3 and Native Programming Package

PROBLEM Several items in the TNIX System Reference Manual have been updated. In section 2 for the IOCTL (2) command #include <sgtty.h> should be #include <tiop.h>. In section 4 for the TTY (4) command all occurrences of #include <sgtty.h> should be #include <tiop.h>. The last sentence "Full documentation not available at this printing" should be changed to refer the user to files "iopcmd.h" and "trm.def.h". Some of the include files are outdated. A new release is expected this summer and new versions are available from your Tektronix Seles or Applications Engineer.

8503 SINGLE FILE SYSTEM PROBLEM

PRODUCT 8560 and root file system on multiple drives

CONFIGURATION 8560 with 8503 and TNIX 1.3

PROBLEM TNIX 1.3 will not allow the root filesystem to span more than one disk drive. The system may or may not return an error message saying that it is out of swap space.

COMMENTS TNIX version 1.4 does not have this problem.

6809 DISM OF SUBD INSTRUCTION

PRODUCT 6809 disassembly of the SUBD instruction

CONFIGURATION 8540 OS-40 version 1 and the 6809 emulator

PROBLEM When using the 6809 disassembly command, the extended SUBD instruction will be disassembled as the SUBA instruction.

COMMENTS A mod will be available this summer.

8086 PASCAL WITH STATEMENT

PRODUCT 8086 Pascal compiler and "with" statement

CONFIGURATION 8560 TNIX version 1.3 or 8550 DOS-50 version 2.1A and 8086 Pascal compiler version 1.02-04(8560) or version 1.01-07(8550)

PROBLEM The 8086 Pascal compiler generates incorrect code if there is 1) multiple indexed array in a "with" statement, 2) optimization is "on" and 3) the array is used as an address of a common sub-expression (CSE).

EXTERNAL FUNCTION IN PASCAL

PRODUCT 8086 Pascal with optimizer on

CONFIGURATION 8560 TNIX version 1.2 and 8086 Pascal version 01.02-040

PROBLEM The compiler may loop forever with optimizer on with certain conditions, particularly if code is removed due to optimization.

COMMENTS This will be corrected with a mod this summer.

ACE EDIT AND START OF FILE TEXT INSERTION

PRODUCT 8560 ACE Editor inserting text at the start of a file

CONFIGURATION 8560 TNIX version 1.3 and ACE Editor version 2.08

PROBLEM Inserting pages of new text at the beginning of an existing file can cause loss of the new text. The data appears on the screen while inserting, but the data is not placed in the file when the editing session has been completed.

COMMENT Occasionally exit insert mode to prevent the problem. A new release this July will correct the problem.

LINKER AND CONCATENATED SECTIONS

PRODUCT Concatenating sections in the Linker with -r

CONFIGURATION 8560 TNIX 1.2 and Linker V 2.00

PROBLEM Sections of the same name are concatenated at link time. If additionally the -r (relinkable) option is specified; at relink time the addresses for local symbols within all but the first section (of the repeated section name) are incorrect.

COMMENTS Concatenating sections by using the same section name is useful in reducing symbols table space. This is corrected with TNIX 1.4 and Linker version 2.08.

ATOBASM AND LARGE DATA BLOCKS

PRODUCT A to B object converter with a large Text Block

CONFIGURATION 8560 TNIX version 1.4 and ATOBOBJ version 01.07-00

PROBLEM If the object code in a Text line equals or exceeds 246 bytes(a single line), ATOBOBJ will fail and stop execution.

COMMENTS Text lines of over 246 bytes long are not created by Tek assemblers. We have not been able to find an instance of a third party s/w product that does create text blocks longer then 256 chr. There are no plans to change this limitation.

PASCAL ERROR 19383

PRODUCT 8086 PASCAL compiler and error message 19838

CONFIGURATION 8560 TNIX version 1.2 and 8086 PASCAL compiler version 1.02-04

PROBLEM If the user declares a constant of type real and compiles the module with -d (debug) "on", the compiler generates error message "19838". No code is generated but the listing states "no errors."

COMMENTS Work around, use a literal in the code instead of a constant declaration for floating point constants.

LDE AND ALTERNATE FILE WRITES

PRODUCT LDE Editor writing to alternate files

CONFIGURATION 8560 TNIX version 1.3, LDE Editor version 1.3 and CT8500 terminal

PROBLEM During an edit session, if the user writes to another existing file, without the "more" modifier, the editor issues an incomplete and misleading message. If the user then follows the directions from the incomplete message, the original file (and not the alternate file) will be overwritten.

PDB AND INTERRUPT INITIALIZATION

PRODUCT 8086 Pascal Debug and interrupt-initialization code

CONFIGURATION 8560 TNIX V01.3 , 8086 Pascal compiler version 1.02.04 and 8086 Pascal debug version 1.07-00A

PROBLEM If ICS is used with 8086 Pascal to generate interrupt initialization code for a large number of interrupts, the address of "Pascal-Begin" may be in the wrong code segment for PDB. Thus PDB cannot find "Pascal-Begin." The program does execute without Pascal debug.

COMMENTS This will be corrected this summer.

ULOAD PROMPT SEQUENCE

PRODUCT 8560 uload and prompt sequence

CONFIGURATION 8560 TNIX version 1.3 and 8550 DOS-50 version 2.1 or 8540 OS-40 version 1.0

PROBLEM Users must be careful with the uload command and specifying a prompt sequence such as '**$$$$**'. Notice the single quotes enclosing the prompt sequence. If users specify a prompt sequence of **$$$$** (without single enclosing quotes), com will receive the process ID instead.

COMMENTS A caution about using **$$$$** or any other character(s) with a special meaning for the shell is described in the 8560 Reference Manual.

8048 ASSEMBLER AND IMPROPER LOADFILE

PRODUCT 8048 assembler with incorrectly loaded object code

CONFIGURATION 8560 TNIX 1.3 and 8048 V 01.04.18 with LINKER V02.05

PROBLEM The linker does not properly locate the 8048 object code if there is a jump within the first few instructions where the lister specified (also endrel does not always match the end of the loaded object code).

COMMENTS Version 2.00-06 is now available and fixes the bug.

ACE CRASH ON TEXT DELETION

PRODUCT ACE Editor deleting a portion of a file

CONFIGURATION 8550 DOS-50 version 2.1A and ACE Editor version 2.08

PROBLEM After deleting from the middle of a file (by setting a mark) to the last character in the file, the 8550 will stop functioning and 8301 program light will stay "on".

COMMENTS Limit text to be deleted by use of the mark to a few pages at most. Ace will crash if the mark is swapped to a temp file. A new release will fix this by this summer.

ACE AND CONTROL S AND CONTROL Q

PRODUCT 8550 ACE Editor and control S or Q

CONFIGURATION 8550 DOS-50 version 2.1A, ACE Editor version 2.060A and CT8500 terminal.

PROBLEM The "control S" and "control Q" fail to halt ACE Editor scrolling operations when using the CT8500 terminal.

COMMENTS The scroll and page features adequately provide display control, thus no change is anticipated.

8048 ASSEMBLER AND DATA TRUNCATED MESSAGE

PRODUCT 8048 B series assembler and data truncation warning

CONFIGURATION 8560 TNIX version 1.3 and 8048 assembler version 1.04-18

PROBLEM Even though the value of less than 256 is valid as an immediate operand for HI, LO, or BITS, a misleading warning message 240 is issued (immediate data is truncated).

COMMENTS This problem is fixed in Version 2.00-06 which is now available.

8086 ICS AND PROGRAMMER ORIGINED VARIABLES

PRODUCT 8086 Pascal compiler and RAM memory space

CONFIGURATION 8560 TNIX version 1.3, 8086 Pascal version 1.02-04 and Linker version 2.05

PROBLEM User must direct ICS to not use memory space that has been absolutely defined in a var origin or port declaration. If this step is not followed, then ICS and the Linker may place the customer origined arrays over the same address range as RAM VAR and generate link time errors.

COMMENTS This is not a problem with linker version 2.08.

EHEX AND MOTOROLA TERMINATION RECORDS

PRODUCT EHEX utility and Motorola termination record

CONFIGURATION 8550 DOS-50 version 2.1A or 8560 TNIX version 1.3 and EHEX version 1.0

PROBLEM The ehex utility doesn't generate the correct Motorola termination records (S7 and S8 hex). For the S8 record, the current ehex puts out a 2 byte long transfer address instead of 3 bytes. For the S7 record, the current ehex puts out a 2 bytes long transfer address instead of 4 bytes long. The termination record contains the execution start address.

COMMENTS The user must type go with an address after file loading. All other data is correctly transferred.

SHELL ESCAPE AND ERASED DISPLAYS IN ACE

PRODUCT 8560 ACE editor and escaping to the shell

CONFIGURATION 8560 TNIX version 1.3 and ACE version 2.08

PROBLEM While in the ACE editor escaping to shell (to list the directory) will display the directory listing for too short of a time before the screen is erased (about 1/2 second).

COMMENTS The user can issue a "sh" command while in ace and the screen won't be erased until control "D" is entered.

ERRONEOUS LINK ERROR 115

PRODUCT Linker error 115 and 8086 Pascal compiler (8560)

CONFIGURATION 8560 TNIX version 1.3, 8086 Pascal compiler version 1.02-04 and linker version 2.05

PROBLEM The linker error message 115 (truncated error at address), generated from linking compiled code, could be misleading since the resulting code may be valid.

COMMENTS Since the truncation could be valid, the generated code should be tested to verify any possible errors.

RESTARTING THE 8086 EMULATOR

PRODUCT 8086 emulator and RAM validity check

CONFIGURATION 8550 DOS-50 version 2.1A or 8540 OS-40 version 1.0 and 8086/ 88/87 Version 1.15 control software

PROBLEM Emulation control code is lost if the probe is deselected which can occur without the user being aware of it. Subsequent attempt to use emulation control commands yield unexpected results.

COMMENTS The use of commands that use local resources will cause the emulator to be deselected. Thus the emulator will need to be reselected and set up for emulation after such commands. For example, PROM programmer commands and 8550 edit, asm, link, etc., will cause the emulator to be deselected.

68000 PASCAL DEBUG AND SOME TRACE OPTIONS

PRODUCT 68000 Emulator and invoking Pascal debug or tracing

CONFIGURATION 8540 OS-40 V1.0 or 8550 DOS-50 V2.1A with 68000 emulator control software V1.10

PROBLEM In Pascal Debug, the use of "step" will miss one sequence of step if it is a call instruction; i.e., step or trace may miss some function or procedure calls.

Tracing doesn't work if the -s and jmp options or an address range within a memory space is specified.

COMMENTS A ROM patch for the 8540 and a syspatch for the 8550 is available.

8086 PASCAL - COMPILER HANG

PRODUCT 8086 Pascal compiler and looping forever with debug on

CONFIGURATION 8560 TNIX version 1.3, 8086 Pascal Compiler version 1.02-04

PROBLEM The 8086 Pascal compiler may loop forever during compilation if the debug option is set; i.e. pas -d.

COMMENTS This is sometimes caused by declaring constants inside a function or procedure. If problem persists, try breaking up the code into separate modules. A new version is expected soon which will correct the problem.

Z80B AND BUS CONTENTION

PRODUCT Z80 emulator and Z80 refresh causing bus contention

CONFIGURATION 8540/8550 with Z80B probe and emulator

PROBLEM During Z80 refresh we leave the data bus drivers turned outward. Hence, if a user's prototype wants to use the bus during the refresh, bus contention will result.

COMMENTS This bug will not affect normal refresh operation. It will affect the prototype operation where the prototype is using the refresh cycle to perform a DMA operation. A mod is available.

INTERNAL PDB ERROR ON LOADING

PRODUCT 8086 PDB (probably exists in Z8000 PDB)

CONFIGURATION 8560D02 VERSION 1.06-00 (8086 AND POSSIBLY Z8000)

PROBLEM There is a bug in PDB loader. Loadfiles with large numbers of symbols may cause an internal PDB error.

COMMENTS The work-around is to use the -d option in a limited number of modules at a time. A bug fix release is planned for later this summer.

PDB STATEMENT RANGE ERROR

PRODUCT 8560D02 PDB 8086

CONFIGURATION PDB Version 1.06-00

PROBLEM If an empty procedure or function is included in the PASCAL program, PDB's load command may fail with STATEMENT RANGING ERROR.

COMMENTS Always put at least one statement in any procedure stub before using PDB.

PROBLEM REPORT

Customer Name ______________________________ Date ______________

Company Name ______________________________ Title ______________

Company Address ____

Internal Address/Dept ____

City ______________ State ______________ Zip Code ______________

Area code ______________ Tel. No. ______________ Ext. ______________

Subscription Service No. ______________________________________

HARDWARE CONFIGURATION. Include serial number and firmware version numbers.

SOFTWARE CONFIGURATION. Include version numbers for all involved products and operating system.

PROBLEM. 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.

COMMENTS.

Send to:
MDP Technical Support Manager
Tektronix Inc
Del. Station 92-635
P.O. Box 4600
Beaverton, Oregon 97075

INDEX

The material contained in this document is for reference only, no assurance of suitability for a specific purpose is implied.